MODO

USER GUIDE

VERSION 17.1v1

FOUNDRY.

Modo® Online Help. Copyright © 2023 The Foundry Visionmongers Ltd. All Rights Reserved. Use of this Online Help and the Modo software is subject to an End User License Agreement (the "EULA"), the terms of which are incorporated herein by reference. This Online Help and the Modo software may be used or copied only in accordance with the terms of the EULA. This document, the Modo software and all intellectual property rights relating thereto are and shall remain the sole property of The Foundry Group LLC, ("Foundry") and/or Foundry's licensors.

The EULA can be found on our website at https://www.foundry.com/eula.

Foundry assumes no responsibility or liability for any errors or inaccuracies that may appear in this document and this document is subject to change without notice. The content of this document is furnished for informational use only.

Except as permitted by the EULA, no part of this document may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, recording or otherwise, without the prior written permission of Foundry. To the extent that the EULA authorizes the making of copies of this Reference Guide, such copies shall be reproduced with all copyright, trademark and other proprietary rights notices included herein. The EULA expressly prohibits any action that could adversely affect the property rights of Foundry and/or Foundry's licensors, including, but not limited to, the removal of the following (or any other copyright, trademark or other proprietary rights notice included herein):

Modo® and Nexus™ © Copyright 2023 The Foundry Visionmongers Ltd. All Rights Reserved. Modo® and Nexus™ are registered trademarks of The Foundry Visionmongers Ltd, in the UK and/or other countries. All other products or brand names are trademarks or registered trademarks of their respective holders. Modo is a licensed user of Pixar Subdivision Surfaces™. 'Full Body IK' based on the IKinema run time engine.

Mari® and Nuke® © Copyright 2023 The Foundry Visionmongers Ltd. All Rights Reserved. Mari® and Nuke® are registered trademarks of The Foundry Visionmongers Ltd, in the UK and/or other countries. All other products or brand names are trademarks or registered trademarks of their respective holders. Modo is a licensed user of Pixar Subdivision Surfaces™. 'Full Body IK' based on the IKinema run time engine.

In addition to those names set forth on this page, the names of other actual companies and products mentioned in this Online Help (including, but not limited to, those set forth below) may be the trademarks or service marks, or registered trademarks or service marks, of their respective owners in the United States and/or other countries. No association with any company or product is intended or inferred by the mention of its name in this Online Help.

Linux ® is a registered trademark of Linus Torvalds.

Windows ® is the registered trademark of Microsoft Corporation.

Mac, Mac OS X, and QuickTime are trademarks of Apple, Inc., registered in the U.S. and other countries.

The Qt GUI Toolkit is Copyright ©2011 Nokia Corporation and/or its subsidiary(-ies).

FFMpeg is Copyright ©2012 the FFmpeg developers.

Avahi is Copyright the Avahi developers. Parts of Avahi are Copyright 2003, 2004 Porchdog Software. All rights reserved.

This software contains Autodesk® FBX® code developed by Autodesk, Inc. Copyright ©2012 Autodesk, Inc.

All rights, reserved. Such code is provided “as is” and Autodesk, Inc. disclaims any and all warranties, whether express or implied, including without limitation the implied warranties of merchantability, fitness for a particular purpose or non-infringement of third party rights. In no event shall Autodesk, Inc. be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of such code.

Adobe® and Photoshop® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Maya® is a registered trademark of Autodesk, Inc., in the USA and other countries.

Mudbox™ is a trademark of Autodesk, Inc.

ZBrush™ © 2015 Pixologic, Inc. All rights reserved, Pixologic and the Pixologic logo, ZBrush, and the ZBrush logo are registered trademarks of Pixologic, Inc.

Patents - Granted and Pending

Protected by U.S. Patents:

- 8,994,737
- 8,581,926
- 7,808,501
- 7,688,323
- 7,538,779
- 7,496,416

Additional patents pending.

Licensed Patents

Additionally Foundry has been granted license to the following Pixar patents:

- 08/905,434 (US Patent No. 6,037,949)
- 08/905,435 (US Patent No. 6,222,553)
- 08/905,436 (US Patent No. 6,300,960)
- 09/756,112 (US Patent No. 6,489,960)
- 09/619,064 (US Patent No. 6,760,024)
- 10/686,551 (US Patent No. 7,023,438)
- 11/274,883 (US Patent No. 7,233,332)
- 10/810,064 (Publication No. 20040263511)

And all continuations, divisionals, continuations-in-part, reexaminations, and reissues of such patents, together with all foreign counterparts or equivalents of any of the foregoing.

Some images on Home landing page courtesy of Emilio J. Dominguez Calvo (www.abishai.artstation.com), Sergio Paulo Madeira (Madeira3D - www.behance.net/paulomadeira), and Theodoru (www.theodoru.com). Some images on User Guide landing page courtesy of Ken Barthelmey (www.theartofken.com), Maciej Frolow (www.maciejfrolow.myportfolio.com), Kion3D (www.kion3d.it), Tomas Sciskala (mytrixx.myportfolio.com), Emilio J. Dominguez Calvo (www.abishai.artstation.com), Sergio Paulo Madeira (Madeira3D - www.behance.net/paulomadeira), and Theodoru (www.theodoru.com). Some images on Resources landing page courtesy of Tomas Sciskala (mytrixx.myportfolio.com).

Thank you to William Vaughan for providing a number of his Python tool scripts (Pushing Point Hatchet Collection).

The Foundry Visionmongers Ltd.
5 Golden Square
London
W1F 9HT
UK

Rev: Tuesday, November 26, 2024

Contents

Installation and Licensing 32

Getting a Foundry User Account 32

Installing on Windows 33

Installing on Mac 50

Installing on Linux 67

Getting Help 84

What Is Modo? 88

Understanding the Interface 90

Using the Modo Introduction Screen 90

Exploring the Layout 91

User Interface Conventions 117

Customizing Your Layout 132

Using Layout Controls 136

Using Standard Tool Controls 140

Using the Work Plane 143

Understanding Viewports 155

3D (OpenGL) Viewport 156

3D Viewport Properties 189

UV Viewport 219

List and Info Viewports 233

Channels Viewport 234

Clip Browser 241

Images/Clips List 243

Command History 253

Info and Statistics 260

Items (Scenes) List 261

Properties Panel 285

Using Tool State Presets 289

Shader Tree 290

Vertex Map List 296

Utility Viewports 297

Assembly Viewport 298

Color Picker Viewport 301

Color Theme Editor 306

Display Viewport 310

Gradient Editor 323

Graph Editor 333

Groups Viewport 354

Preset Browser 366

Preset Browser Advanced Filtering 381

Creating Your Own Presets 384

Preview Viewport 402

Render Display 423

Tool Pipe 437

Form Workbenches 441

Using the Input Editor 446

Modeling 453

Modeling Overview and Techniques 453

Modeling Techniques 456
Procedural Modeling with MeshOps 474
Creating Interactive Custom Profile Presets 491
Deferring and Pausing Evaluation 511
Procedural Preset Browser 513
Procedural Selection 517
Using the Mesh Ops Tab 564

Creating Geometry 572

Adding Primitives 575
Cube 577
Sphere 580
Cone 585
Cylinder 588
Capsule 592
Torus 595
Ellipse 599
Tube 600
Teapot 602

Using Drawing Tools 604
Pen 604
Curves 608
Bezier 609
B-Spline 613
Bezier 618
Sketch 626
Arc 628

Adding Curves 630
Circle 634
Star 635
N-Sided 636
Rectangle 638
Ellipsoid 639
Diamond 642
Curve Convert 644
Curve to Bezier Mesh Operation 647

Using Other Geometry Tools 650
Modeling with Images 650

Solid Sketch 654
Text 656
Vertex 661
Patch Curves 662
Stitching 664
Lace Item 671
Create Polygons 676

Selecting Items 680

Using Vertex Component Tools 692
Vertex Bevel 693
Vertex Extrude 695
Vertex Merge Tool 696
Vertex Tools 698

Using Edge Component Tools 710
Edge Bevel 711
Edge Extrude 715
Edge Extend 716
Slide / Edge Slide 719
Add Point 735
Add Loop 736
Edge Relax 739
Edge Chamfer 744
Edge Tools 763
Split Normals 770
Edges to Curves 771
Unbevel 778

Using Polygon Component Tools 780
Detriangulate 782
Triangulate 783
Polygon Bevel 785
Polygon Extrude 799
Flip Polygon 800
PolyScale Offset 807
PolyScale Offset Detach 808
Flatten Ngons 809
Flip It 812
Triple Fan 814
Reduce Collapse Tools 816
Smooth Shift 819
Shift 820
Thicken 821
Spikey 826
Inset 828
Polygon Reduce 829
Set Polygon Type 831

    - Polygon Tools 833
  - Using Selection Modifiers 841
  - Using Selection Sets 866
  - Make Selections Fast Using Falloffs 873
- Applying Precision 877
  - Applying Snapping 880
    - Snaps and Precision Palette 888
  - Mesh Constraints 890
  - Using Utilities 906
- Transforming Items 914
  - Axis Rotate 918
  - Center Commands 922
  - Move 923
  - Rotate 926
  - Scale 929
  - Tack 933
  - Transform 934
  - Uniform Scale 938
  - Absolute Scaling 941
- Duplicating Items 943
  - Array 946
  - B-Spline Extrude 956
  - Bezier Clone 963
  - Bezier Extrude 974
  - Bridge 979
  - Clone 991
  - Content Preset 999

Curve Clone 1002
Curve Extrude 1012
Curve Fill 1016
Curve Offset by Mesh 1021
Curve Rebuild 1025
Rebuilding Curve Spans 1028
Curve Sweep 1034
Open Curve Offset 1050
Closed Curve Offset 1053
Duplicate Commands 1055
Lace Geometry 1056
Mesh Paint 1062
Mirror 1075
Paste Tool 1085
Path Segment Generator 1086
Pen Clone 1088
Pen Extrude 1096
Pen Generator 1099
Procedural Clone Effector 1103
Radial Array 1104
Radial Sweep 1113
Scatter Clone 1119
Sketch Extrude 1127
Sweep Effector 1129
Symmetrize 1131

Deforming Items 1137

Deform Tools 1140

Element Move 1140
Element Push 1144
Flex 1146
Soft Move 1150
Soft Drag 1153
Shear 1154
Smooth 1159
Jitter 1165
Quantize 1184
Symmetry Tool 1186
Soft Selection Move 1187
Soft Selection Rotate 1191
Soft Selection Scale 1194
Soft Selection Transform 1198
Linear Align 1202
Radial Align 1210
Quick Align 1212
Ground Align 1215
UV Transform 1218
Path Generator 1228

Scale Deform Tools 1230
Push 1230
Sculpt 1234
Taper 1238
Bulge 1242
Flare 1247

Rotate Deform Tools 1248
Twist 1248
Bend 1252
Vortex 1255
Swirl 1259

Editing Items 1262

Slice Tools 1264
Slice 1264
Curve Slice 1267
Pen Slice 1269
Slice Effector 1270
Edge Slice 1273
Loop Slice 1275
Axis Slice 1283
Julienne 1286
Topology Pen 1288
Contour 1299
Shatter 1302

- Edit Tools 1313
  - Edit Commands - Cut/Copy/Paste 1313
  - Edit Commands - Delete/Remove 1315
  - Automatic Retopology Tool 1321
  - Merge Meshes 1330
  - Freezing Mesh Operations 1346
  - Booleans 1347
  - Solid Drill 1351
  - Axis Drill 1353
  - Subdivide 1355
  - UnSubdivide 1358
  - Fix Black Dots 1360
- Other Geometry Editing Tools 1362
  - Mesh Cleanup 1362
  - Scene Cleanup 1364

Specifying Action Centers and Falloffs 1365

- Using Falloffs 1368
  - Airbrush Falloff 1370
  - Curvature Falloff 1371
  - Cylinder Falloff 1372
  - Element Falloff 1374
  - Image Falloff 1376
  - Lasso Falloff 1377
  - Linear Falloff 1379
  - Noise Falloff 1381
  - Path Falloff 1382
  - Radial Falloff 1383
  - Screen Falloff 1384
  - Selection Falloff 1385
  - Soft Selection Falloff 1386
  - Coplanar Falloff 1388
  - Vertex Map Falloff 1391

MeshFusion 1392

- Fusion Example 1401
- Mesh Roles and Relationships 1436
- Cutting, Copying, Pasting, and Splitting in MeshFusion 1443
- Drag and Drop Workflow 1446
- Using MeshFusion with Zero Subdivision 1449
- Fusing by Selection Set 1459

    Fusion Qbic Presets 1465
    Working with MeshFusion Assemblies 1468
    Using Curves with MeshFusion 1474
    Creating a New Fusion Item 1477
    Modeling in the Schematic Viewport 1478
    Fusion Surface Strips 1486
    Embossing Surface Strips with MeshFusion 1505
    Fusion Item 1526
    Duplicating Fusion Items 1543
    Fusion Mesh Conversion 1547

Sculpting 1548

  Sculpting Techniques 1548

  Using Sculpting Brush Tools 1551
    Push 1552
    Inflate 1556
    Fold 1559
    Clay 1561
    Smooth 1564
    Carve 1566
    Scar 1569
    Flatten 1570
    Smudge 1573
    Move 1574
    Tangent Pinch 1575
    Center Pinch 1577
    Spin 1578

    - Emboss 1580
    - Attenuate 1581
  - Adding Images 1582
  - Using Sculpting Masks 1587
  - Locking Borders and Corners 1592
  - Creating Sculpting Layers 1594
  - Using Sculpting Utility Commands 1597
  - Using the Multi Resolution Palette 1601
- UVing 1603
  - Working with Vertex Maps 1603
    - Vertex Mapping 1617
    - Vertex Normal Tools 1619
    - Create New Vertex Map 1647
    - Set Vertex Map Name 1648
    - Vertex Map Commands 1649
    - Working With Vertex Maps Procedurally 1654
    - Topological Vertex Map Transfer 1675
    - Transfer Vertex Map 1676
    - Smooth Vertex Map 1678
    - Apply Morph 1679
    - Morph Tool 1680
    - Background to Morph 1680
    - Topological Morph Creation 1681
    - Vertex Map Math 1684

Vertex Map Math Tool 1685

Vertex Shader Baking 1687

Vertex Curvature Baking 1689

Vertex Color Tool 1690

Vertex Map Weight Tool 1691

Edge Weight Tool 1692

UV to Weight 1692

Weight Map Slicer 1695

Morph Slicer 1697

Falloff Weight 1699

Convert Falloff Cache 1701

Set Vertex Normals 1704

Transform Vertex Normals 1709

Show Vertex Normals 1712

Vertex Map Tools 1713

Vertex Map Paint 1715

Working with UV Maps 1718

UV Mapping 1724
- Create UV Tool 1726
- UV Box Transform Tool 1735
- View Projection 1741
- UV Peeler 1742
- UV Unwrap 1743
- Select Error UVs 1754
- Select Unmapped Polygons 1756
- UV Relax 1759
- Symmetrize UVs 1766
- Align UV 1767
- Rectangle 1772
- Fit UVs 1779
- Select Border 1785
- Mirror UVs 1786
- Rotate UV 1786
- Merge UVs 1787

Flip UV 1787
Sew UVs 1792
Split, Move, and Sew UVs 1792
Orient Pieces 1802
Texel Density 1802
Pack UVs 1809
Cut, Copy, and Paste UVs 1828
Delete UVs 1830
Export UVs to EPS 1830
Export UVs to SVG 1831
Convert UVs to Mesh 1833
Transform UV Map 1838

UDIM Workflow 1839

UDIM Indicator 1852

Painting 1855

Painting Overview 1855

Using Painting Tools 1857

Airbrush 1859

Blur 1861

Clone 1863

Eraser 1866

Fill 1868

Lasso 1869

Line 1871

Linear Gradient 1873

Paintbrush 1874

Radial Gradient 1877

Sharpen 1879

Smudge 1881

Customizing Paint Tools 1911

Customizing Inks 1919
Tablet Nozzles 1927

Adding Images 1929
Image Adjustment Commands 1933

Using Hair Tools 1934
Blow 1936
Comb 1937
Curl 1939
Move 1940
Root Puff 1941
Separate 1942
Smooth 1943
Stretch 1944
Hair Guides 1945

Using Particle Tools 1954
Particle Paint Dissolve 1955
Particle Paint Flatten 1957
Particle Paint Move 1959
Particle Paint Pinch 1960
Particle Paint Size 1962
Particle Paint Smooth 1964
Particle Paint Smudge 1966
Particle Paint Spin 1967
Particle Paint Tangent Pinch 1969
Particle Paint Tool 1971

Creating a Layout 1974

Working with Items 1974

Understanding Items vs. Components 1977

Adding Layer Items 1979

- Scene Item 1980
- Dynamic Replicator 1983
- Fractal Gasket 1989
- Gear Item 1992
- Ground Plane 1997
- Group Locator Item 2000
- Instance Item 2001
- Locator Item 2004
- Mesh Item 2025
- Replicator Item 2053
- Rock Item 2060
- Static Mesh Item 2067

Transforming Items 2070

- Centers and Pivots 2070
- Transform Item Tools 2071
  - Transform 2073
  - Move 2076
  - Rotate 2078
  - Axis Rotate 2080
  - Scale 2082
- Transform Modifiers 2085

Duplicating Items 2085

- Instances 2086
- Replicators 2089

Freezing Geometry 2094
Freeze for Procedural Modeling 2098
Using Dynamics and Particles 2101
Dynamics 2101
Dynamics Solver 2113
Dynamic Item 2118
Constraints 2134
Pin Constraint 2136
Point Constraint 2139
Hinge Constraint 2142
Slide Hinge Constraint 2145
Spring 2149
Constraint Modifier 2151
Dynamics Forces 2154
Linear Force 2155
Radial Force 2157
Drag Force 2159
Turbulence Force 2161
Vortex Force 2163
Wind Force 2165
Curve Force 2167
Anchor 2170
Procedural Shatter 2172
Baking Dynamics 2176
Particles and Simulations 2178
Particle Simulation 2181
Particle Emitters 2184
Radial Emitter 2184
Surface Emitter 2189
Curve Emitter 2191
Source Emitter 2194
Collision Emitter 2197
Particle Operators 2200
Particle Operator 2200

Collector/Emitter 2202
Duplicate Geometry Using Particle Clone 2204
Flocking 2209
Dynamic Collider 2211
Dynamic Fluid 2212
Terminator 2215

Particle Modifiers 2216
Particle Basic Modifier 2217
Particle Random Modifier 2220
Particle Step Modifier 2224
Particle Sieve Modifier 2227
Particle Expression Modifier 2229
Particle Look At Modifier 2237
Particle Audio Modifier 2239

Point Clouds 2243
CSV Point Cache 2243
Particle Generator 2246
Particle Cloud 2250
Surface Particle Generator 2254
Curve Particle Generator 2259
RealFlow Particles 2271

Animating 2274

Overview, Timeline, & Hierarchy 2274

Working with the Timeline 2280

Hierarchy 2296

Animation Layout and Viewports 2300

Track View 2318

Schematic Viewport 2323

Channels Viewport 2349

Deformers Viewport 2358

Creating Actors, Actions, and Poses 2363

Animation Layers 2385

Using Channels 2391

Add User Channels 2393

Modifying Gradient Channels 2398

Using Modifiers & Constraints 2406

Constraints 2409
- Transform Constraints 2412
- Geometry Constraints 2414
- Particle Constraints 2415
- Path Constraints 2417
- Intersection Constraints 2419
- Camera Constraints 2420
- UV Constraint 2422
- Constraint Up Vectors 2424

Channel Modifiers 2424
- Array Modifiers 2432
- Channel Modifiers 2452
- Channel Waveform 2457
- Conditional Modifiers 2458
- Mesh Info Modifier 2459
- Gradient Modifiers 2462
- Integer Range Modifiers 2475
- Math Modifiers 2479
- Matrix Modifiers 2482
- Measurement Modifiers 2491
- Particle Modifiers 2493
- Probe Modifiers 2495
- Quaternion Modifiers 2498
- Simulation Modifiers 2506
- String Modifiers 2511
- Time Modifiers 2515
- Vector Modifiers 2518
- Clamp 2523
- Planar IK 2525
- Dynamic Parenting 2535
- Expression Modifier 2536
- Generate ID 2541
- Linear Blend 2541
- Numeric Switch Modifier 2542
- Randomize ID Modifier 2544
- Revolve 2545
- Simple Kinematics 2546
- Switch Modifier 2550

Using Setup Tools 2552

Set 2554

Hierarchy Tools 2555

Match and Align 2557

Playblast 2559

Skeleton Tool 2564

Binding and Setup Mode 2568

Pose Tool 2571

Spacing Chart 2573

Retargeting 2585

Tools and Commands 2604

Channel Handles 2608

Using Audio 2612

Command Regions 2616

Applying Deformers 2628

Applying Deformers 2632
- Bezier Effector 2634
- Bend Effector 2641
- Curve Constraint Effector 2644
- Lag Effector 2647
- Lattice Effector 2650
- Magnet Effector 2654
- MDD Deformer 2657
- Add Morph 2660
- Rotation Effector 2661
- Slack Effector 2663
- Spline Effector 2665
- Transform Effector 2670
- Twist Effector 2673
- Wrap Influence 2676
- Weight Map Deformer 2678

Using Deformer Falloffs 2680
- Area Falloff 2687
- Bezier Falloff 2690
- Capsule Falloff 2694
- Curve Falloff 2698
- Falloff Operator 2703

Incidence Falloff 2705
Linear Falloff 2708
Radial Falloff 2713
Spline Falloff 2717
Texture Falloff 2722
Weight Map Falloff 2728

Using Influences 2736
General Influence 2737
Item Influence 2739
MDD Influence 2741
Morph Influence 2743
Push Influence 2746

Creating a Full Body IK Setup 2748

Setting up a Full Body IK 2748

Positioning and Rotating the Full Body IK 2755

Working with Weight Maps 2761

Weighting Tools 2763

Working with Morph Containers 2767

Utilizing the Auto Character System (ACS) 2771

Shading and Lighting 2774

Defining Shaders 2774

Polygon Set Material/Part/Smoothing 2782

Shader Tree 2788

Shader Tree Item Layers 2804
Folder 2809
Material Group 2810
Material 2815
Custom Materials 2845
AxF Material 2846
Cel Edges 2854
Cel Material 2858
glTF Shader 2860
Hair Material 2868

- Halftone Material 2875
- Iridescence Material 2879
- Skin Material 2881
- Thin Film Material 2889
- Substance Material 2891
- Unity Material 2895
- Unreal Material 2899

Enhance:modo Textures 2903

- Agate 2912
- Art Deco 2916
- Basket 2920
- Bath Tile 2926
- Blast 2930
- Blister 2934
- Box 2938
- Bozo 2944
- Branches 2948
- Bricks 2953
- Camo 2958
- Caustic 2962
- Cellular 2967
- Checks 2971
- Cheesy 2976
- Circular 2980
- Concrete 2985
- Coriolis 2989
- Cornerless 2993
- Corners 2999
- Counter 1 3004
- Counter 2 3009
- Crackle 3015
- Cruddy 3019
- Crumpled 3023
- Cubes 3028
- Cubic 3032
- Dash Line 3038
- Dented 3043
- Diamond Deck 3047
- Dimples 3052
- Dino Skin 3057
- Dirt 3062
- Disease 3066
- Disturbed 3070
- Drip Drop 3074
- Easy Wood 3078
- Electric 3082
- Etched 3086
- Easy Gradient 3090
- FBM 3093

Fire 3097
Fire Wall 3101
Fish Scales 3106
Flare 3111
Flow Bozo 3115
Frog Skin 3119
Gas Giant 3124
Glint 3128
Grainy Wood 3132
Granite 3137
Grid 3141
Hard Wood 3146
Hexagonal Tiles 3150
Hurricane 3155
Hybrid 3159
Iris 3164
Lattice 1 3169
Lattice 2 3173
Lattice 3 3177
Leather 3181
Linear 3186
Lump 3191
Marble Noise 3195
Marble Vein 3199
Membrane 3203
Minky 3207
Monster 3211
Mosaic 3216
Multi-Fractal 3221
Nurnies 3225
Octagonal Tile 3229
Parquet 3234
Pastella 3239
Paving 3243
Pebbles 3249
Peel 3253
Peened 3258
Plaid 3263
Planet 3267
Planet Clouds 3271
Planks 3276
Plates 3281
Polygon 3286
Puffy Clouds 3291
Radial 3295
Rain 3301
Regional HSV 3305
Ribs 3308
Ridged 3313

- Ring 3318
- Rings 3323
- Ripples 3327
- Rivets 3331
- Rivet Rust 3337
- RndLinear (Random Linear) 3343
- Rounded Tile 3348
- Rust 3353
- Scar 3358
- Scatter 3362
- Scratches 3366
- Scruffed 3371
- Shingles 3375
- Sin Blob 3380
- Smear 3383
- Spiral 3389
- Spots 3394
- Stamped 3398
- Star 3404
- Star Field 3409
- Strata 3413
- Stucco 3417
- Surf 3422
- Swirl 3426
- Tacos 3430
- Tartan 3436
- Terra 3440
- Tiler 3444
- Tri Checks 3449
- Tri Hexes 3454
- Tri Tile 3458
- UVLEDs 3462
- Vector Bozo 3468
- Veins 3472
- Wall 3476
- Waveforms 3482
- Waves 3501
- Windows 3505
- Windy Waves 3509
- Wires 3514
- Worm Vein 3518
- Wrapped fBm 3522

Image Map 3526
- Gradient Fill Image 3534

Projection Shader 3537

Projection Texture 3541

Processing 3548
- Constant 3548
- Gradient 3550

      - Occlusion 3576
      - Tension Texture 3585
      - Variation Texture 3589
      - Vertex Map Texture 3592
    - Textures 3595
      - Cellular 3599
      - Checker 3607
      - Dots Procedural 3612
      - Gabor Noise 3617
      - Noise Procedural 3626
      - Poisson Cells 3634
      - Ripples Procedural 3643
      - RT Curvature 3647
      - Weave Procedural 3651
      - Wireframe Texture 3655
      - Wood Procedural 3660
    - Texture Locator 3665
      - Projection Type Samples 3671
    - Texture Switch Node 3677
    - Fur Material 3680
    - Principled Shader 3701
    - MatCap Shader 3707
    - Material Presets 3711
    - Refractive Index Table 3713
    - Shader 3716
    - Surface Generator 3725
    - Working with Fur 3730
  - Layer Effects 3736
    - Effect - Render Item 3737
    - Effect - Texture Item 3737
      - Anisotropic Direction 3771
    - Effect - Shader Item 3775
    - Effect - Environment Item 3775
    - Effect - Light Item 3776
    - FX Item 3779
  - Layer Blend Modes 3783
  - Advancing the Shader Tree 3792
  - PBR Textures 3796
- Nodal Shading 3818
  - Shading Nodes 3821
    - Illuminate Node 3822
    - Raycast Node 3824
    - Ray Type Node 3827

    - Shader Inputs 3829
    - Color Blend 3832
    - Color Correct 3834
    - Color Gamma 3837
    - Color HSV 3838
    - Color Invert 3840
    - Color Kelvin 3841

Environments 3842
  - Environment Item 3842
  - Environment Material 3845
  - Image Based Lighting 3850

Lights 3852
  - Area Light 3853
  - Cylinder Light 3859
  - Directional Light 3865
  - Dome Light 3873
  - Photometric Light 3877
  - Point Light 3884
  - Portal 3889
  - Spot Light 3893
  - Mesh Light 3900
  - Light Material 3909

Volumes 3915
  - Blobs 3915
  - Sprites 3921
  - Volumes 3927
  - Render Boolean 3939

Baking 3945
  - Working with Bake Items and Baking Tools 3955

Channel Swizzling 3984

Rendering 3986

Rendering with OctaneRender 3986

Rendering with mPath 3990

Render Items 4001

Render Item: Frame 4001

Render Item: Settings 4010

Render Item: Default Renderer Settings 4010

Render Item: mPath Renderer Settings 4025

Render Item: Global Illumination 4035

Camera Items 4049

Camera Matching 4087

Camera Images 4104

Render Outputs 4117

Render Output Items 4135

Basic Render Outputs 4136

Cryptomatte Render Output 4139

Geometry Render Outputs 4142

Lighting Render Outputs 4148

Material Render Outputs 4153

Particle Render Outputs 4160

Shading Render Outputs 4163

Volume Render Outputs 4169

Render Commands 4172

Render Passes 4178

Render Proxy 4184

Deferred Meshes 4190

VDB Voxel 4200

Driving Render Outputs Nodally 4209

Network Rendering 4210

Network Rendering Troubleshooting 4216

Render Display 4217

Tone Mapping 4231

Color Management 4236

Saving Images 4244

Modo VR 4250

Options, Scenes, and Performance 4253

Using Modo VR 4255

Menus in Modo VR 4262

Indicators in Modo VR 4268

Modo VR Additional Features 4270

Quick Start: Coffee Mug 4274

Model the Mug 4274

Render the Mug 4306

Resources 4324

Preferences 4324

Data Preferences 4324

Defaults Preferences 4326

Display Preferences 4346

File I/O Preferences 4358

Input Preferences 4379

Rendering Preferences 4392

File Commands 4399

Project Directories 4403

Edit Commands 4404

Import Reference 4406

Importing and Exporting File Formats 4412

Bitmap Image Formats 4431

Export MDD 4434

Game Asset Exporter 4435

Keyboard Shortcuts 4441

Mouse Standards 4458

Modo Markup for Forms 4463

Frequently Asked Questions 4464

Plug-Ins 4472

Modo Material Importer for Unity 4473

Modo Material Importer for Unreal Engine 4478

Transferring assets with GoZ Kit 4483

Modo Bridge 4484

Modo Bridge for Unreal Engine 4487

Modo Bridge for Unity 4501

ProRender Beta 4511

PView Viewport 4513

ProRender Settings and Limitations 4517

Third-Party Software

4530

Installation and Licensing

We know the installation and licensing of a new application can be a boring task that you just want to be done with as soon as possible. To help you with that, this chapter guides you to the point where you have the application in front of you and are ready to start work.

Operating Systems

To see the installation and licensing instructions for your operating system, go to:

- Installing on Windows,
- Installing on Mac, or
- Installing on Linux.

Getting a Foundry User Account

Before you can install Modo, you first need to download it. Modo program downloads and licensing are facilitated through the Foundry product download page.

First and foremost, you'll need an account for the Foundry website. If you purchased Modo directly from Foundry, your account was created automatically during the purchase process. Use the same username and password that you created when purchasing Modo to log in and gain access to the various installer downloads.

If you've purchased Modo from a reseller you'll need to create an account manually by visiting our website. To do this:

1. Click the **Create Account** link in the upper-right corner of the page.
2. Once the account is created, you should receive your 25-digit serial number from your reseller.

Note: Expect the serial number e-mail once the order process is completed. The timing of this varies by reseller.

To link your Modo to the Foundry account you've created:

1. Ensure that you are signed in, then navigate to **Support** > **Activate MODO Serial Code** on the Foundry website.
2. Enter the 25-digit serial code and click **Attach to my account**.

Tip: From time to time, Foundry makes available service packs and other updates that you can access from either your user account or from the Modo menu bar **Help** > **Check for Updates**. Once downloaded, you can follow the same procedure outlined here to install. It is recommended you un-install any previous version before initiating a new install (though backing up before un-installing isn't a bad idea, should something go wrong).

Installing on Windows

After installation, Modo is run from either desktop icons, the **Start** menu, or from the command line using command arguments.

Modo System Requirements

Qualified Operating Systems

- Windows 10 and Windows 11 (64-bit)

Note: Other operating systems may work with Modo, but have not been fully tested. If you have any problems with a particular operating system, please contact our support team.

See the Modo Release Notes for the full list of system requirements.

Installing Modo on Windows

To install Modo on Windows, see either:

- Installing Modo with the User Interface (UI), or
- Installing Modo from the Command Line.

Installing Modo with the User Interface (UI)

To install Modo using the default UI, do the following:

1. Download the correct **.msi** installation file from our website at www.foundry.com.

   The file is saved to your preferred download location.
2. Double-click on the file in the installed folder to install Modo.
3. Accept the EULA and the application is copied into the installation folder automatically and is ready to launch when the copy is finished.

   By default, Modo is installed to C:\Program Files\Modo<version>\.

   By default, **Run** Modo 17.1v1 is enabled.
4. Proceed to Launching Modo on Windows.

Installing Modo from the Command Line

To install Modo from the command line, do the following:

1. Download the correct **.msi** installation file from our website at www.foundry.com.

   The file is saved to your preferred download location.
2. Open a command prompt window, by clicking **Start** and typing **cmd**.
3. Click **Command Prompt** to launch the application.
4. Use the **cd** (change directory) command to move to the directory where you saved the installation file. For example, if you saved the installation file in **C:\Temp**, use the following command and press **Enter**:

   ```
   cd \Temp
   ```
5. To install Modo, do one of the following:

   - To install Modo to the current directory and display the installation dialog, use **msiexec** and point to the installer:

     ```
     msiexec /i "PathToMsi.msi"
     ```
   - To install Modo to a specified directory and display the installation dialog, use the **INSTALL_ROOT** argument:

     ```
     msiexec /i "PathToMsi.msi" INSTALL_ROOT=D:\Modo\MyPath
     ```
   - To install Modo silently so that the installer does not prompt you for anything but displays a progress bar, enter **/quiet** after the installation command:

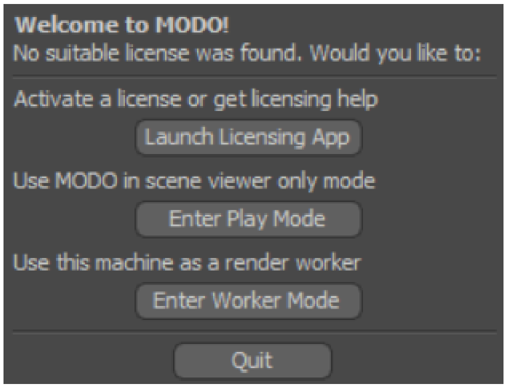
     ```
     msiexec /i "PathToMsi.msi" INSTALL_ROOT=D:\Modo\MyPath /quiet
     ```

Launching Modo on Windows

To launch Modo on Windows, do one of the following:

- Double-click the **Modo 17.1v1** icon on the Desktop.
- Click **Start** and type **Modo 17.1v1**, and then click **Modo 17.1v1** to launch the application.
- Using a command prompt, navigate to the Modo application directory (by default, **C:\Program Files\Modo17.1v1\modo**) and enter **modo.exe** to launch Modo.

If you already have a valid license, the graphical interface appears. If you don't have a license, or haven't installed one yet, the **Licensing** dialog appears on start up.



Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

Launch Licensing App

Clicking this button opens the Licensing app where you can manage your licenses and run diagnostic tools.

For more information on how to license Modo, refer to Licensing Modo on Windows.

Running in Play Mode

Play mode places Modo in a special state where it acts as a scene viewer. In this state the interfaces are mostly disabled, but it allows you to load scenes and render and save them, both in Preview and using **F9**; opening the **Render Display** window.

One unique function of Play mode is its ability to detect scene changes and auto-reload the scene. What this means is any scene opened from over a network and previewed, when updated on another system, saves files to the scene file, reloads the scene on the networked system and updates the preview viewport. This allows the secondary system to act as a dedicated preview system without the need for a second license.

Running in Worker Mode

Worker mode is related to network rendering and places Modo in a state where other Modo installations can control its renderer. For information on how to set up Network Rendering, please reference the Network Rendering page.

Updates and Statistics

When Modo launches for the first time with a valid license, you are presented with a dialog, which enables you to set preferences to check for updates and to submit usage statistics.

Updates

Get the latest reminders to update Modo. You can have Modo automatically check for updates on every launch, daily or weekly. By default, this preference is set to **Every Launch**.

Usage Statistics

To help improve the quality of Modo, you can agree to submit usage data to the Modo Development team. These statistics are submitted anonymously and include details on the build number run, the system platform, the application active and idle times, and whether sessions exit successfully or not. In the event of a crash, the most recently executed commands are also submitted.

Note: Graphics card information is not submitted.

Tip: To reset your settings, open **System** > **Preferences**, expand **Defaults** > **Application** and reset the options for the following:
Automatically Check for Updates
Automatically Submit Usage Statistics

The following settings are available: **Never**, **Every Launch**, **Daily**, or **Weekly**.

Article: You can disable the Alert dialog in the Form Editor. For details on how to do this, see Q100345: How to disable all new Alert dialog boxes for the Modo 11 series onward.

Running in Safe Mode

In the unlikely event of a crash, you can run Modo in Safe Mode to help us determine what is causing the problem. When Modo runs in Safe Mode, it doesn't load any third-party plug-ins, kits, or user customizations and only runs using the contents of the Modo install directory. Safe Mode is primarily used for debugging purposes. When running in Safe Mode, a pristine configuration file is used.

> **Note:** When exiting Safe Mode, your existing configuration directory (**MODO17.1.CFG**) is not overwritten.
>
> A backup of configuration directory is created at shutdown, when changes are saved. This backup directory can be renamed to restore the previous state if configuration changes cause issues. When a corrupt configuration is detected, it is renamed, instead of deleted, which allows you to recover data from corrupted configuration directory.
>
> Save your configuration directory and submit it to Support for debugging. Your configuration directory is stored in the **C:\Users\**<user name> **\AppData\Roaming\Luxology** directory.

To run in Safe Mode:

- Click **Start** and type **Modo 17.1v1**, and then click **Modo 17.1v1(Safe Mode)** to launch the application.

If Modo successfully launches in Safe Mode, click **File** > **Config Save** to overwrite your existing configuration file that was causing Modo to crash.

Licensing Modo on Windows

The following licensing methods are available:

- **Log-in Licenses** - A license tied to your Foundry website account, available for either subscriptions or perpetual maintenance entitlements. They can be activated on up to 2 machines at once and deactivated if you need to use the entitlement elsewhere. Activating Modo installs a rolling temporary license which can be used offline for up to 37 days. The license is refreshed whenever Modo is connected to the internet.

The following video walks you through using Modo's log-in licensing tool.

Note: Log-in licenses for Modo plug-ins Power Translators and Power Sub-D to NURBS are added alongside your Modo log-in license entitlements. If you install one of these plug-ins when the device is already authorized, please go to **Help** > **Modo Licensing** > **Advanced** > **Deauthorise this device**, and then log-in again in order to display plug-in licenses. All of your entitlements are added together.

- **Activation Keys** - A code which can be used to generate a license after purchase. Activation Keys can be provided for node-locked, floating and log-in licenses. They can be activated from the application's Licensing dialog or using the Activate a Product Page: https://www.foundry.com/licensing/activate-product.
- **Node Locked Licenses** - A license locked to a specific machine. Node-locked licenses, sometimes called uncounted licenses, do not require a license server or other resources to run. The license is for a single machine, if you need to move the license you'll have to submit a transfer request.
- **Floating Licenses** - A license locked to a machine that acts as a license server to enable any machine on the network to run an application. The license is installed on the server machine along with license server software. The license server software is installed with the Foundry Licensing Utility.

The number of machines that can run the application at the same time is limited by the number of floats in the license. Any machine can request a license, the server will issue a license as long as one is available. When the application exits it releases the license.

You can find more information about licensing Foundry products at learn.foundry.com/licensing

Note: If you are simply installing render nodes on a networked system, no license is required for Modo. Modo can also be run in Play mode to load scenes, navigate the viewport to look at them, and to render them (stills or animations). This allows you to render scenes on one system while continuing to work on another using your license. Play mode is available in the licensing dialog by clicking **Enter Play Mode**. For more information, see Running in Play Mode.

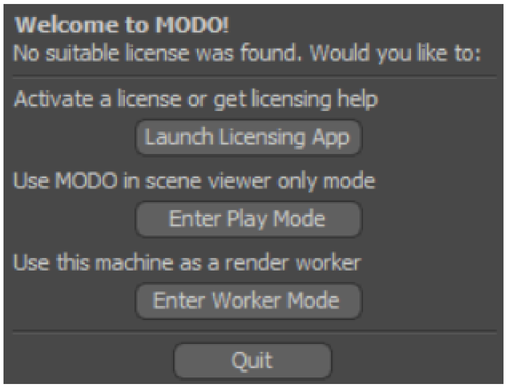
In all other instances, you need to have a license or a trial version of Modo.

To Install a Log-in License

To activate your license, follow these steps:

1. Launch Modo using any of the methods described in Launching Modo on Windows.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

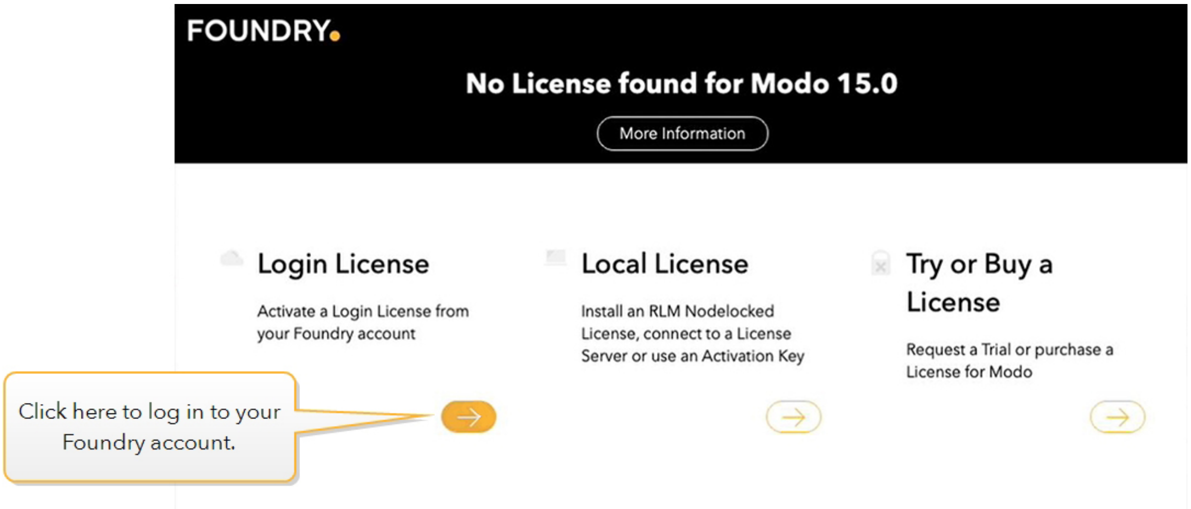
Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

2. Click **Launch Licensing App**.

   The following window appears.



No License found for Modo 15.0

Login License

Activate a Login License from your Foundry account

Local License

Install an RLM Nodelocked License, connect to a License Server or use an Activation Key

Try or Buy a License

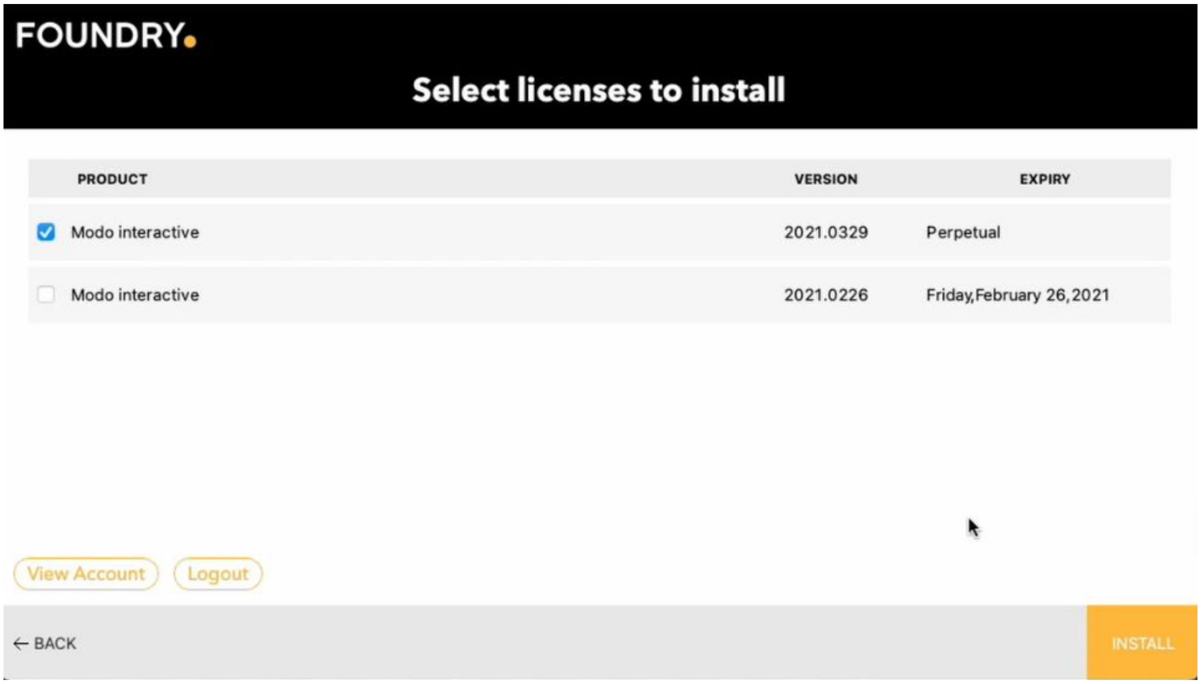
Request a Trial or purchase a License for Modo

Click here to log in to your Foundry account.

3. Click the button for **Login License** and **Sign In** with your Foundry account email address and password.

   A license is created in the following directory: **C:\Users\{username}\AppData\Local\Foundry\Tokens** or **%LOCALAPPDATA%\Foundry\Tokens** if the variable is set to a different location.

   The license key looks something like this: **c58edf7e-17ab-435b-8d8a-b3a9b347ab11.lic**.
4. Select the license you want to install from the list and click **Install**.



PRODUCT	VERSION	EXPIRY
☑ Modo interactive	2021.0329	Perpetual
☐ Modo interactive	2021.0226	Friday,February 26,2021

5. Once the license is installed, click **Done** to close the Licensing App.

   You can now use Modo.

Note: A license is activated for a period of 38 days. Make sure that you connect to the internet once during the activation period to extend your access. You can check the activation period of your license in the **Help** > **Modo Licensing** menu.

6. If you need to deactivate your license on the current machine, navigate to **Help** > **Modo Licensing**, click **Manage Login Licenses**.
   - Click on **Logout** to return all licenses and logout from the machine or;
   - Select the license(s) you need to deactivate and click **Return**.

Obtaining Legacy Licenses

To obtain a legacy license, you'll need your machine's System ID (sometimes called Host ID or rlmhostid). Just so you know what a System ID number looks like, here’s an example: 000ea641d7a1.

Bear in mind that, for floating licenses, you'll need the System ID of the license server, not the machines on which you intend to run the application. There are a number of ways you can find out your machine's System ID:

- Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing and run it. Click **System ID** to display it on screen.

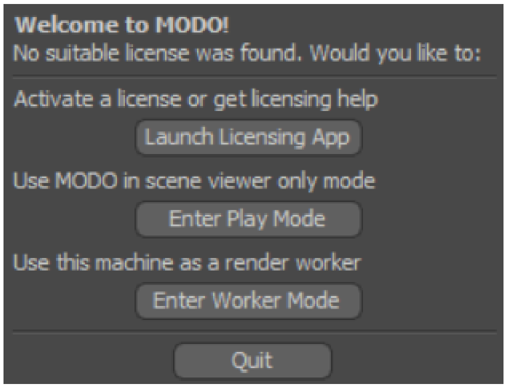
When you know your System ID, you can request a license for Foundry products from Foundry's Sales Department at sales@foundry.com.

To Install a License from Disk

If you've copied a license to a known location on disk:

1. Launch Modo using any of the methods described in Launching Modo on Windows.

   The **Licensing** dialog displays.

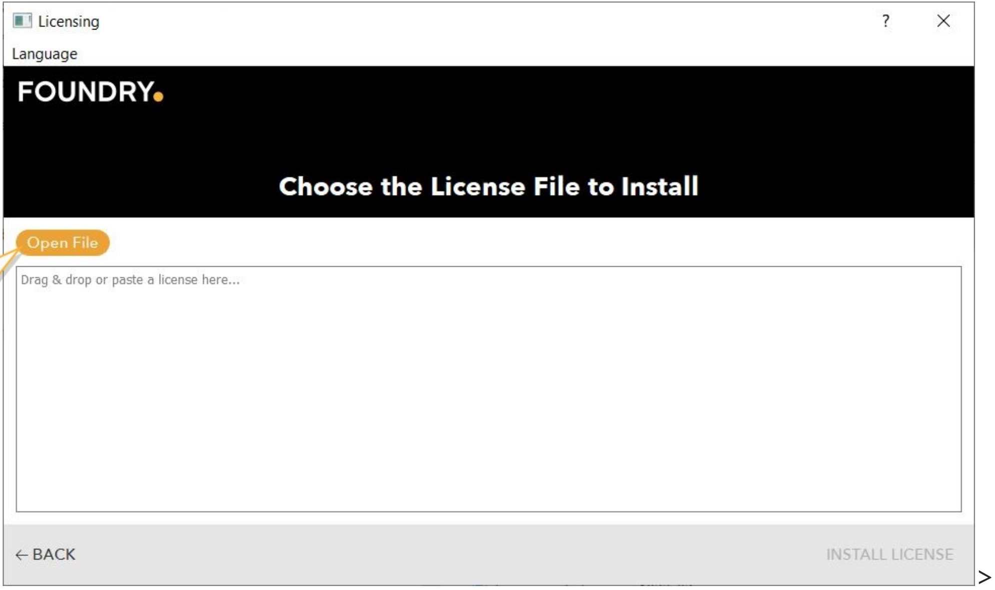


Welcome to MODO!
No suitable license was found. Would you like to:
Activate a license or get licensing help
Use MODO in scene viewer only mode
Use this machine as a render worker

Launch Licensing App
Enter Play Mode
Enter Worker Mode
Quit

2. Click **Launch Licensing App** and then click the button for **Local License**.
3. Click **License File.**

   The following window opens.



Choose the License File to Install

Licensing
?
×
Language
FOUNDRY
Open File
Drag & drop or paste a license here...
← BACK
INSTALL LICENSE

Browse to the location of the license file.

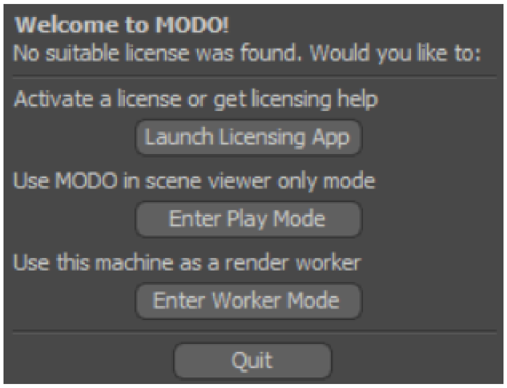
4. Click **Open File** to locate the license file.
5. Click **Open** to install the license automatically in the correct directory.

   Once the license is validated, Modo displays a notification dialog.

To Install a License from a Text File

1. Launch Modo using any of the methods described in Launching Modo on Windows.

   The **Licensing** dialog displays.

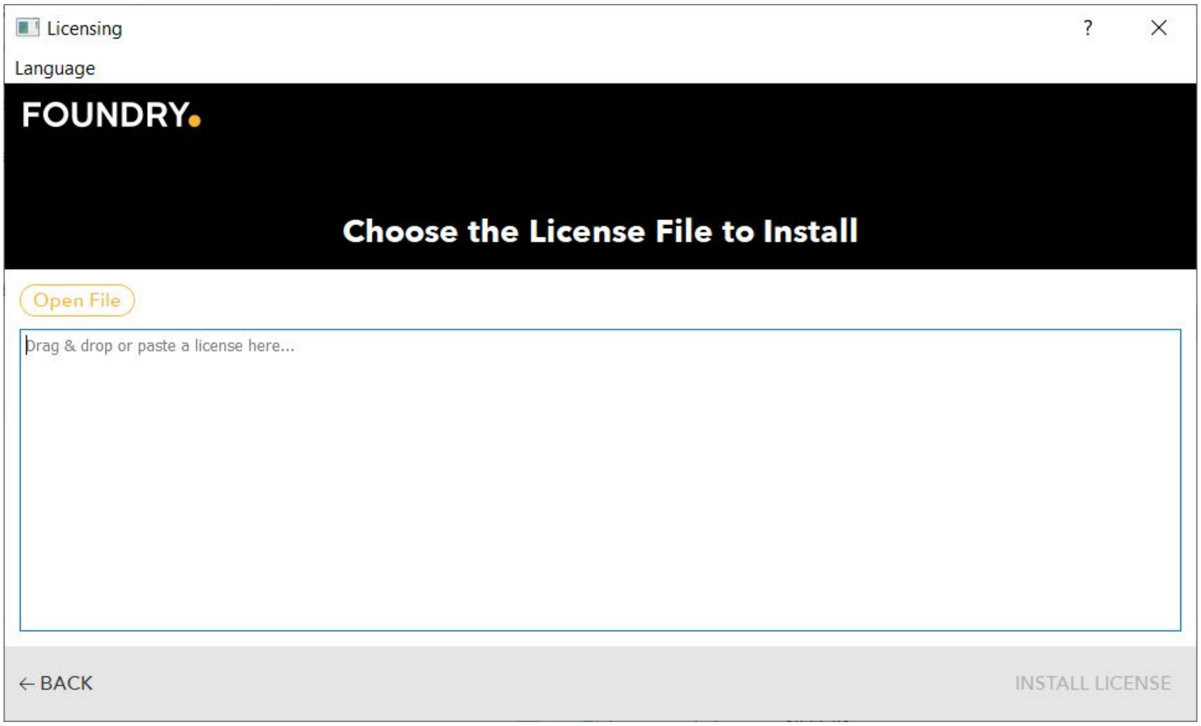


Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help
Launch Licensing App
Use MODO in scene viewer only mode
Enter Play Mode
Use this machine as a render worker
Enter Worker Mode
Quit

2. Click **Launch Licensing App** and then click the button for **Local License**.
3. Click **License File.**

   The following window opens.



Licensing
?
Language
FOUNDRY.
Choose the License File to Install
Open File
Drag & drop or paste a license here...
← BACK
INSTALL LICENSE

4. Copy and paste the license text into the window and click **Install license**.

   The license is automatically installed on your machine in the correct directory.

To Install a Floating License

If you requested a floating license from Foundry, you receive your license key (**foundry.lic**) in an e-mail or internet download. You should also receive the Foundry License Utility (FLU) application to help you install the license key on the license server machine. The server manages licenses for the client machines on your network.

> **Note:** The FLU is also available to download from https://www.foundry.com/licensing.

1. Make sure you have saved both the license key (**foundry.lic**) and the FLU application in the same directory.
2. Run the FLU application.

   The license key automatically appears in the FLU window if the FLU and **foundry.lic** are in the same directory.

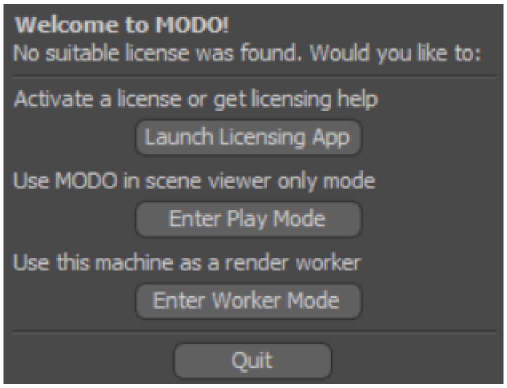
> **Tip:** If they are not in the same directory, you can either copy and paste the contents of the license key or drag-and-drop the file into the FLU window.

3. Click **Install**.

   This checks the license file and, provided that the license is valid, installs it into the correct directory.
4. In order for the floating license to work, you will need to install the Foundry Licensing Tools (FLT) on the license server machine.

   For more information on how to install floating licenses, refer to the *FLT User Guide*, which you can download from our website https://www.foundry.com/licensing
5. Once your license server is up and running, launch Modo on the client machine.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

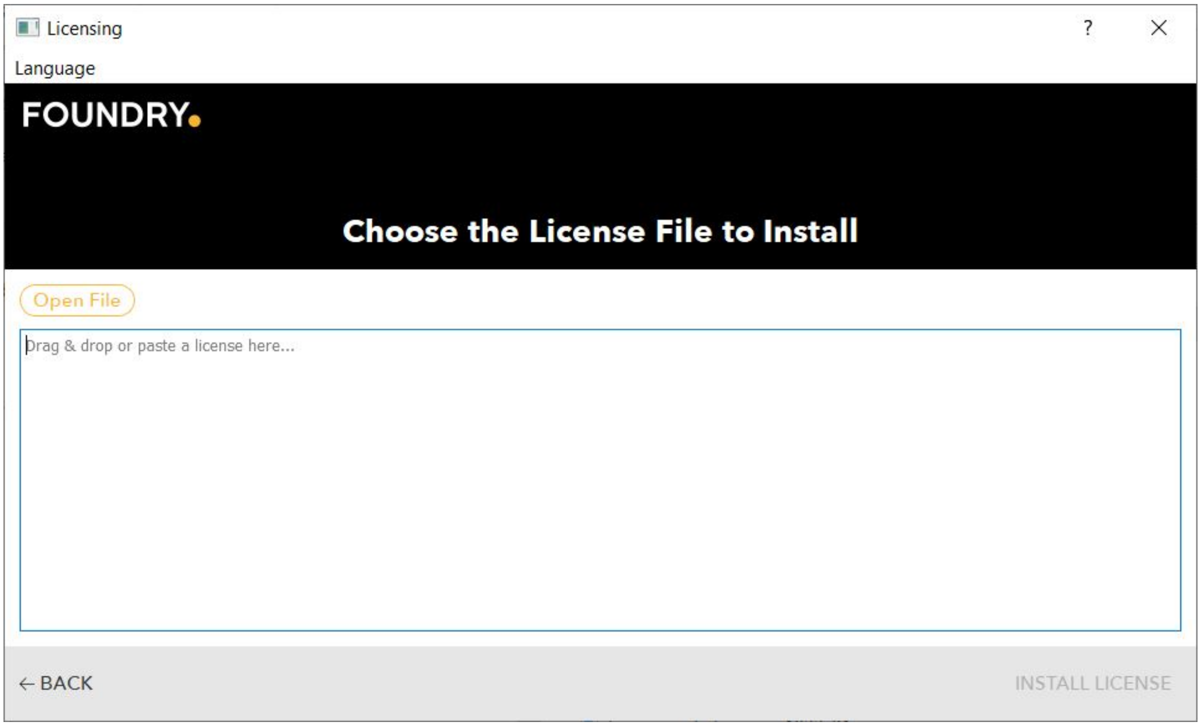
Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

6. Click **Launch Licensing App** and then click the button for **Local License**.

7. Click **Server** and enter the server address in the field provided. The format for the server name is: <port>@<servername>, for example, 30001@red.



Choose the License File to Install

Drag & drop or paste a license here...

Note: You must perform these steps on each client machine that requires a license from the server.

Roaming for Floating Licenses

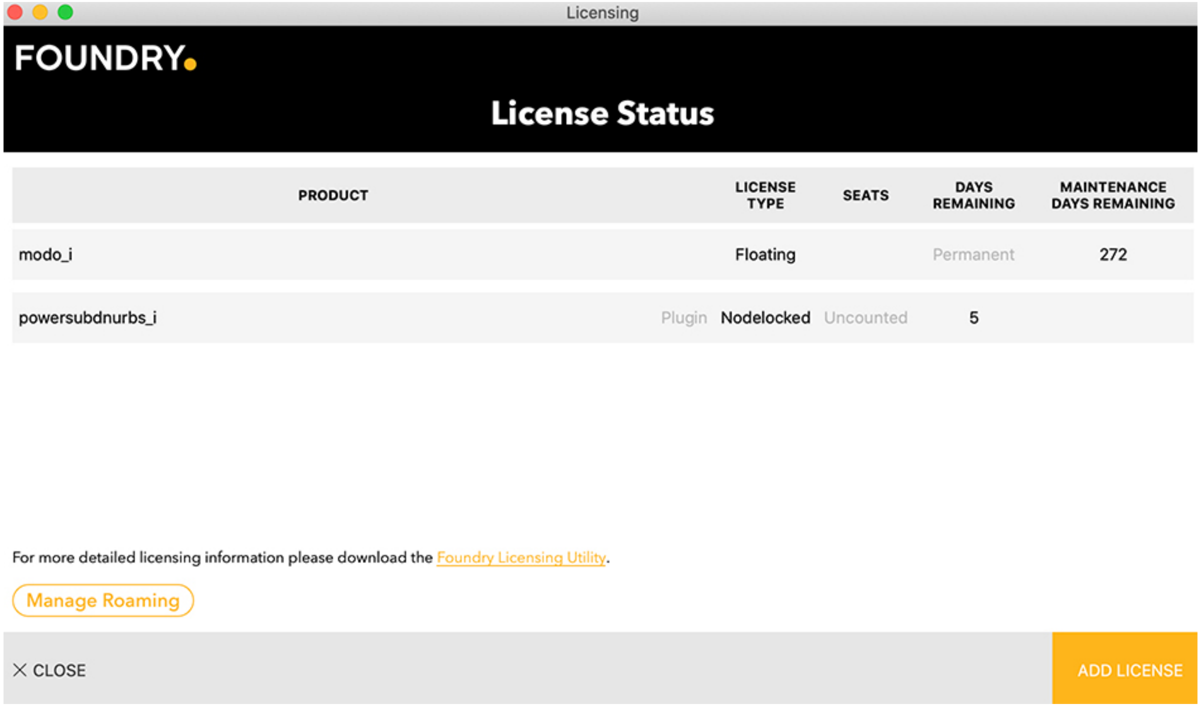
You can check out a roaming license from the pool of floating licenses available on a server and use it offline, on the system used to check it out, for a specified number of days, up to a maximum of 30 days. After the specified time the license expires and you have to connect to the server to renew or return it.

When you claim a license to use offline, the number of licenses available on the server decreases by one until you return the license.

To start roaming:

1. Make sure you're connected to your license server.
2. From the application menu bar, click **Help** > **Modo Licensing**.

   This displays the **Licensing App**.


Licensing
FOUNDRY.

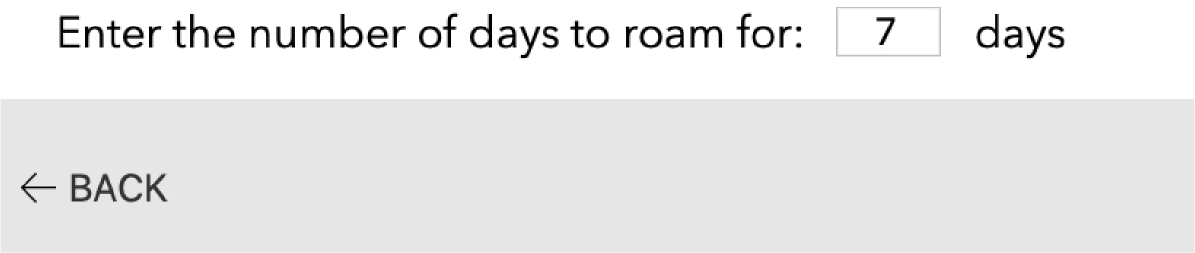
License Status

PRODUCT		LICENSE TYPE	SEATS	DAYS REMAINING	MAINTENANCE DAYS REMAINING
modo_i		Floating		Permanent	272
powersubdnurbs_i	Plugin	Nodelocked	Uncounted	5	

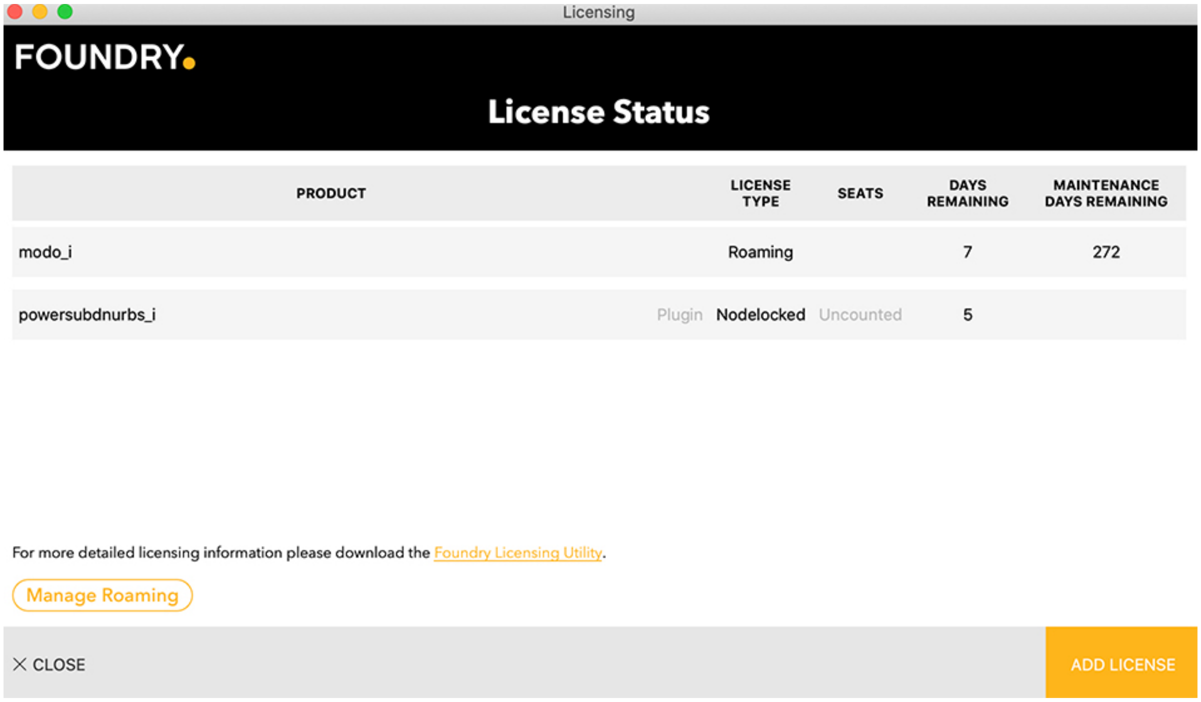
For more detailed licensing information please download the Foundry Licensing Utility.

Manage Roaming
CLOSE
ADD LICENSE

3. Click **Manage Roaming**.
4. At the bottom of the dialog, enter the number of days for which you want to check the license out, then click **Roam Licenses**.


Enter the number of days to roam for: 7 days

Note: The value must be between **1** and **30**.



License Status

PRODUCT		LICENSE TYPE	SEATS	DAYS REMAINING	MAINTENANCE DAYS REMAINING
modo_i		Roaming		7	272
powersubdnurbs_i	Plugin	Nodelocked	Uncounted	5	

For more detailed licensing information please download the Foundry Licensing Utility.

Manage Roaming

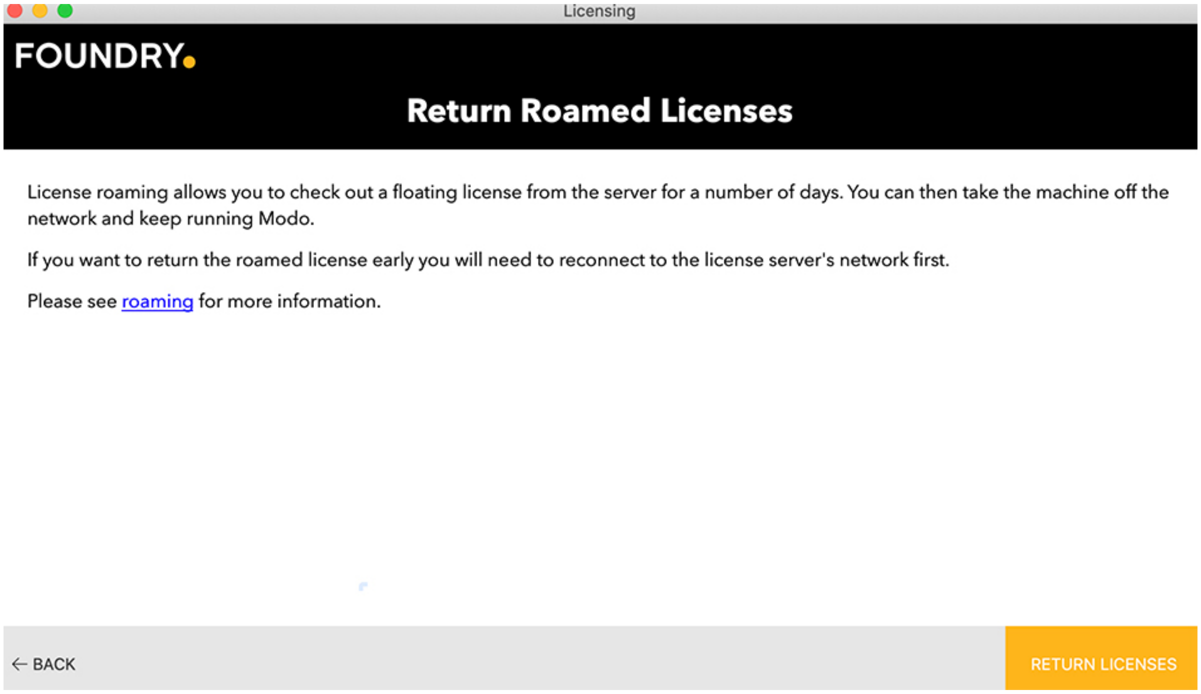
On the license status page, your license is now listed as Roaming.

You can now run Modo offline for the specified number of days.

To return a license:

1. Make sure you're connected to the license server.
2. From the application menu bar, click **Help** > **Modo Licensing**.

   This displays the **Licensing App**.
3. Click **Manage Roaming**.
4. Click **Return Licenses**.



License roaming allows you to check out a floating license from the server for a number of days. You can then take the machine off the network and keep running Modo.

If you want to return the roamed license early you will need to reconnect to the license server's network first.

Please see roaming for more information.

5. Click **Close**.
6. Close and re-open Modo for the changes to take effect.

Deactivating a Modo License

Login-based licensing allows you to deactivate your license from one machine and then activate it on another. This gives you the flexibility to easily use your Modo Subscription or Modo with Maintenance license on different machines when or if you need to.

You can deactivate your login-based license directly in Modo or through the Foundry website.

Deactivating a License in Modo

1. Launch Modo and click **Help** > **Modo Licensing** in the main menu bar.

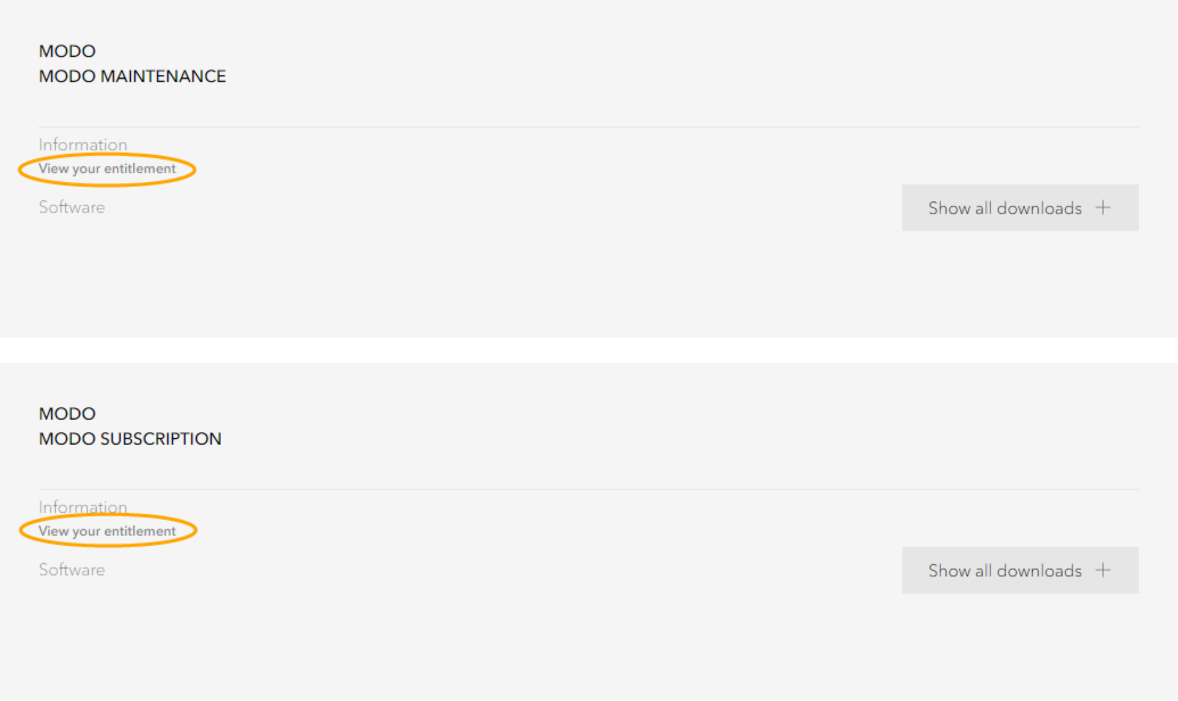
   The **Licensing** dialog displays the active license installed on the machine.
2. Click **Advanced** and then **Deauthorise this device** to deactivate the license and remove it from the machine.
3. Quit Modo.

Deactivating a License on Our Website

You can also deactivate the license from your activated machine through the **My Account** pages on our website. This is useful if you need to use your license at home but forgot to log out of Modo at work.

1. Go to the Foundry My Account - Products page and login if prompted.

   For more information about Foundry user accounts, see Getting a Foundry User Account.
2. Navigate to your **Subscription** or **with Maintenance** product and click **View your entitlement**.



MODO
MODO MAINTENANCE

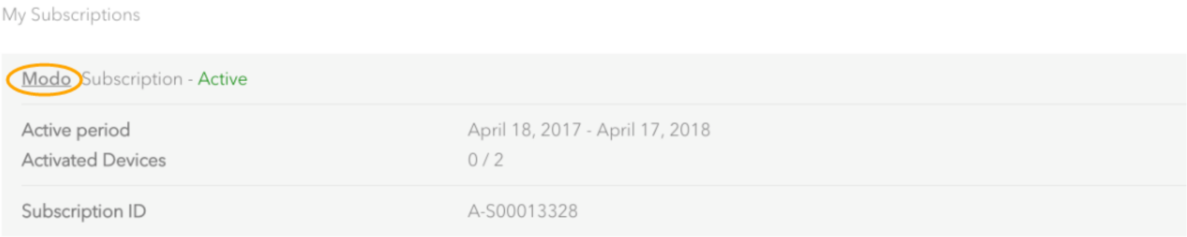
Information
View your entitlement
Software

MODO
MODO SUBSCRIPTION

Information
View your entitlement
Software

Tip: If you're having difficulty viewing the entitlement page log out of the website using this link, https://www.foundry.com/user/logout, and then log back in again.

3. Click **Modo** to see the details of your subscription.

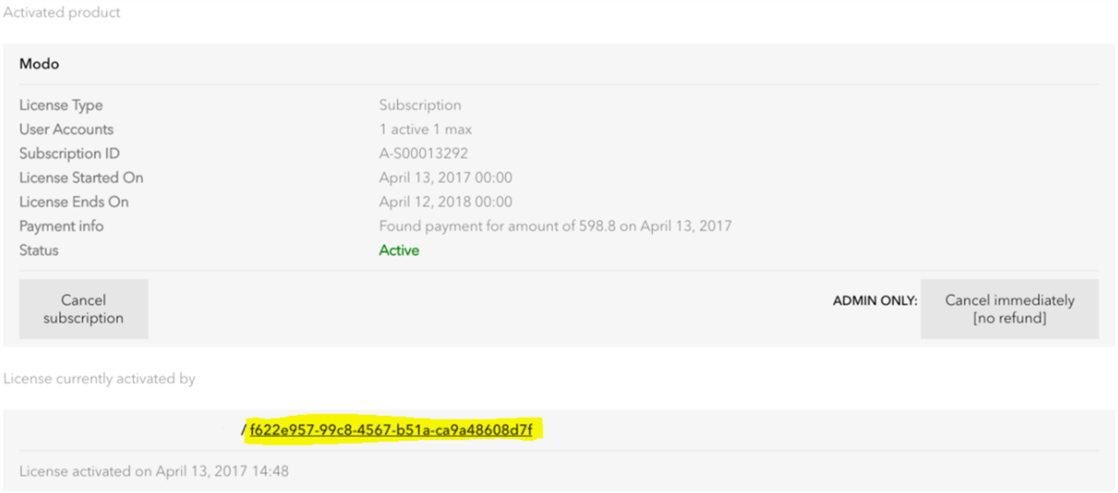


My Subscriptions

Modo Subscription - Active

Active period	April 18, 2017 - April 17, 2018
Activated Devices	0 / 2
Subscription ID	A-S00013328

4. Click on the license key under the **License currently activate by** section.



Activated product

Modo

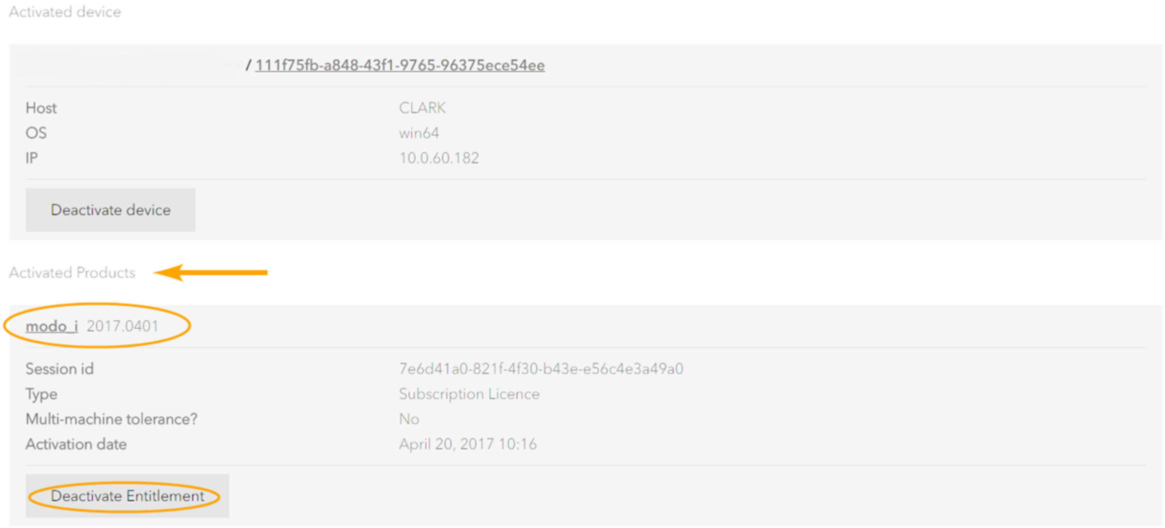
License Type	Subscription
User Accounts	1 active 1 max
Subscription ID	A-S00013292
License Started On	April 13, 2017 00:00
License Ends On	April 12, 2018 00:00
Payment info	Found payment for amount of 598.8 on April 13, 2017
Status	Active

License currently activated by

/ f622e957-99c8-4567-b51a-ca9a48608d7f

License activated on April 13, 2017 14:48

5. In the **Activated Products** section, under **modo_i**, click **Deactivate Entitlement** to remove the Modo license from that machine.



Activated device

/ 111f75fb-a848-43f1-9765-96375ece54ee

Host	CLARK
OS	win64
IP	10.0.60.182

Activated Products

modo_i 2017.0401

Session id	7e6d41a0-821f-4f30-b43e-e56c4e3a49a0
Type	Subscription Licence
Multi-machine tolerance?	No
Activation date	April 20, 2017 10:16

Warning: If you click on **Deactivate device**, all login-based licenses (including any Nuke Non-Commercial or Mari Non-Commercial) will be removed from the machine.

Tip: For an overview of the documentation and quick access to the main topics, go to Docs Home.

Installing on Mac

After installation, Modo is run from either desktop icons, the **Finder**, or from the **Terminal** using arguments.

Modo System Requirements

Qualified Operating Systems

- macOS 11.x (Big Sur), 12.x (Monterey), and 13.x (Ventura)

Note: Modo has been tested and validated on Apple Silicon hardware, running under Rosetta 2 emulation.

Other operating systems may work with Modo, but have not been fully tested. If you have any problems with a particular operating system, please contact our support team.

See the Modo Release Notes for the full list of system requirements.

Installing Modo on Mac

To install Modo on Mac:

- Installing Modo with the User Interface (UI), or
- Installing Modo from the Terminal.

Installing Modo with the User Interface (UI)

To install Modo using the default UI, do the following:

1. Download the correct **.dmg** installation file from our website at www.foundry.com.

   The file is saved to your preferred download location.
2. Double-click on the **.dmg** file to start the installation.
3. Accept the EULA and then drag the **Modo** icon into the **Applications** folder.

The application is copied into the folder automatically and is now ready to launch when the copy is finished.

4. Proceed to Launching Modo on Mac.

Installing Modo from the Terminal

To install Modo from the Terminal, do the following:

1. Download the correct **.dmg** installation file from our website at www.foundry.com.

   The **.dmg** is installed to your preferred download location.

2. Launch a Terminal window.
3. Use the **cd** (change directory) command to move to the directory where you saved the installation file. For example, if you saved the installation file in **/Users/<username>/Downloads/**, use the following command and press **Enter**:

   ```
   cd /Users/<username>/Downloads/
   ```

4. Enter:

   ```
   open Modo17.1v1.app
   ```

5. Accept the EULA and then drag the **Modo** icon into the **Applications** folder.

   The application is copied into the folder automatically and is now ready to launch when the copy is finished.

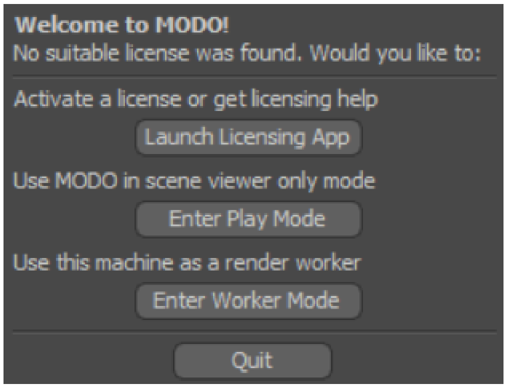
6. Close the Terminal and proceed to Launching Modo on Mac.

Launching Modo on Mac

To launch Modo on macOS, do one of the following:

- Click the Modo icon in the **Applications** folder.
- Double-click the Modo dock icon.
- Using the Terminal, navigate to the **/Applications** directory and enter **Modo17.1v1.app** to launch Modo.

If you already have a valid license, the graphical interface appears. If you don't have a license, or haven't installed one yet, the **Licensing** dialog appears on start up.


Welcome to MODO!
No suitable license was found. Would you like to:
Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

Launch Licensing App

Clicking this button opens the Licensing app where you can manage your licenses and run diagnostic tools.

For more information on how to license Modo, refer to Licensing Modo on Mac .

Running in Play Mode

Play mode places Modo in a special state where it acts as a scene viewer. In this state the interfaces are mostly disabled, but it allows you to load scenes and render and save them, both in Preview and using **F9**; opening the **Render Display** window.

One unique function of Play mode is its ability to detect scene changes and auto-reload the scene. What this means is any scene opened from over a network and previewed, when updated on another system, saves files to the scene file, reloads the scene on the networked system and updates the preview viewport. This allows the secondary system to act as a dedicated preview system without the need for a second license.

Running in Worker Mode

Worker mode is related to network rendering and places Modo in a state where other Modo installations can control its renderer. For information on how to set up Network Rendering, please reference the Network Rendering page.

Updates and Statistics

When Modo launches for the first time with a valid license, you are presented with a dialog, which enables you to set preferences to check for updates and to submit usage statistics.

Updates

Get the latest reminders to update Modo. You can have Modo automatically check for updates on every launch, daily or weekly. By default, this preference is set to **Every Launch**.

Usage Statistics

To help improve the quality of Modo, you can agree to submit usage data to the Modo Development team. These statistics are submitted anonymously and include details on the build number run, the system platform, the application active and idle times, and whether sessions exit successfully or not. In the event of a crash, the most recently executed commands are also submitted.

Note: Graphics card information is not submitted.

Tip: To reset your settings, open **System** > **Preferences**, expand **Defaults** > **Application** and reset the options for the following:
Automatically Check for Updates
Automatically Submit Usage Statistics

The following settings are available: **Never**, **Every Launch**, **Daily**, or **Weekly**.

Article: You can disable the Alert dialog in the Form Editor. For details on how to do this, see Q100345: How to disable all new Alert dialog boxes for the Modo 11 series onward.

Run in Safe Mode

In the unlikely event of a crash, you can run Modo in Safe Mode to help us determine what is causing the problem. When Modo runs in Safe Mode, it doesn't load any third-party plug-ins, kits, or user customizations and only runs using the contents of the Modo install directory. Safe Mode is primarily used for debugging purposes. When running in Safe Mode, a pristine configuration file is used.

Note: When exiting Safe Mode, your existing configuration (**com.luxology.modo17.1**) is not overwritten.

A backup of configuration directory is created at shutdown, when changes are saved. This backup directory can be renamed to restore the previous state if configuration changes cause issues. When a corrupt configuration is detected, it is renamed, instead of deleted, which allows you to recover data from corrupted configuration directory.

Save your configuration directory and submit it to Support for debugging. Your configuration directory is stored in the **/Users/qa/Library/Preferences/** folder.

To run in Safe Mode:

- Open **Applications** > **Modo**, and double-click **Modo17.1v1 (Safe Mode)**.

Note: If you have installed the Modo application in another folder, launch the **Modo17.1v1 (Safe Mode)** shortcut from that folder.

If Modo successfully launches in Safe Mode, click **File** > **Config Save** to overwrite your existing configuration file that was causing Modo to crash.

Licensing Modo on Mac

The following licensing methods are available:

- **Log-in Licenses** - A license tied to your Foundry website account, available for either subscriptions or perpetual maintenance entitlements. They can be activated on up to 2 machines at once and deactivated if you need to use the entitlement elsewhere. Activating Modo installs a rolling temporary license which can be used offline for up to 37 days. The license is refreshed whenever Modo is connected to the internet.

The following video walks you through using Modo's log-in licensing tool.

Note: Log-in licenses for Modo plug-ins Power Translators and Power Sub-D to NURBS are added alongside your Modo log-in license entitlements. If you install one of these plug-ins when the device is already authorized, please go to **Help** > **Modo Licensing** > **Advanced** > **Deauthorise this device**, and then log-in again in order to display plug-in licenses. All of your entitlements are added together.

- **Activation Keys** - A code which can be used to generate a license after purchase. Activation Keys can be provided for node-locked, floating and log-in licenses. They can be activated from the application's Licensing dialog or using the Activate a Product Page: https://www.foundry.com/licensing/activate-product.
- **Node Locked Licenses** - A license locked to a specific machine. Node-locked licenses, sometimes called uncounted licenses, do not require a license server or other resources to run. The license is for a single machine, if you need to move the license you'll have to submit a transfer request.
- **Floating Licenses** - A license locked to a machine that acts as a license server to enable any machine on the network to run an application. The license is installed on the server machine along with license server software. The license server software is installed with the Foundry Licensing Utility.

The number of machines that can run the application at the same time is limited by the number of floats in the license. Any machine can request a license, the server will issue a license as long as one is available. When the application exits it releases the license.

You can find more information about licensing Foundry products at learn.foundry.com/licensing

Note: If you are simply installing render nodes on a networked system, no license is required for Modo. Modo can also be run in Play mode to load scenes, navigate the viewport to look at them, and to render them (stills or animations). This allows you to render scenes on one system while continuing to work on another using your license. Play mode is available in the licensing dialog by clicking **Enter Play Mode**. For more information, see Running in Play Mode.

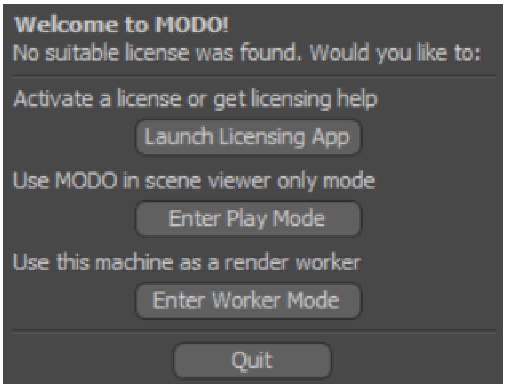
In all other instances, you need to have a license or a trial version of Modo.

To Install a Log-in License

To activate your license, follow these steps:

1. Launch Modo using any of the methods described in Launching Modo on Mac.

   The **Licensing** dialog displays.



Welcome to MODO!

No suitable license was found. Would you like to:

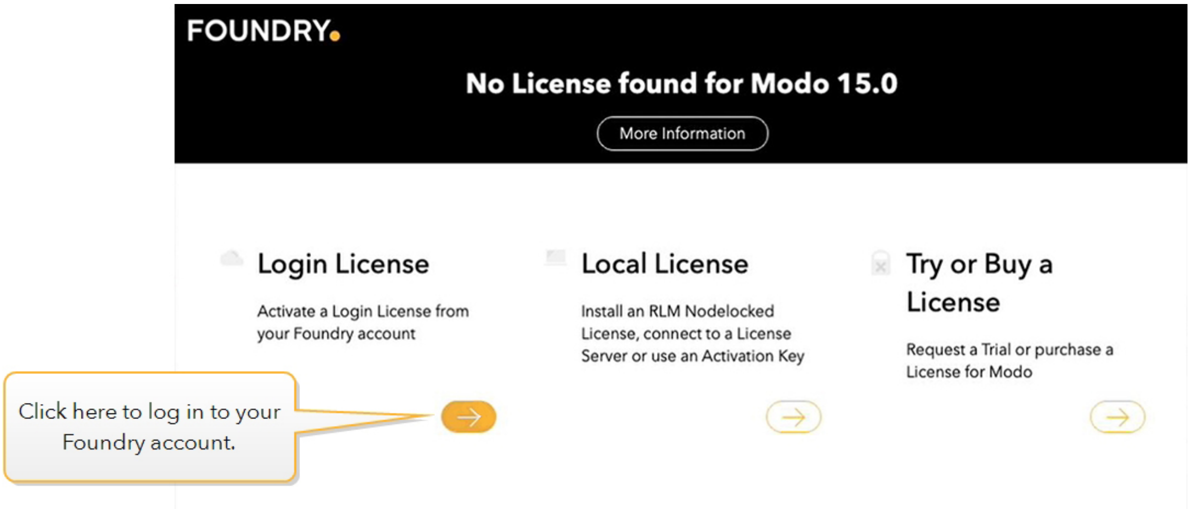
Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

2. Click **Launch Licensing App**.

   The following window appears.



No License found for Modo 15.0

Login License

Activate a Login License from your Foundry account

Local License

Install an RLM Nodelocked License, connect to a License Server or use an Activation Key

Try or Buy a License

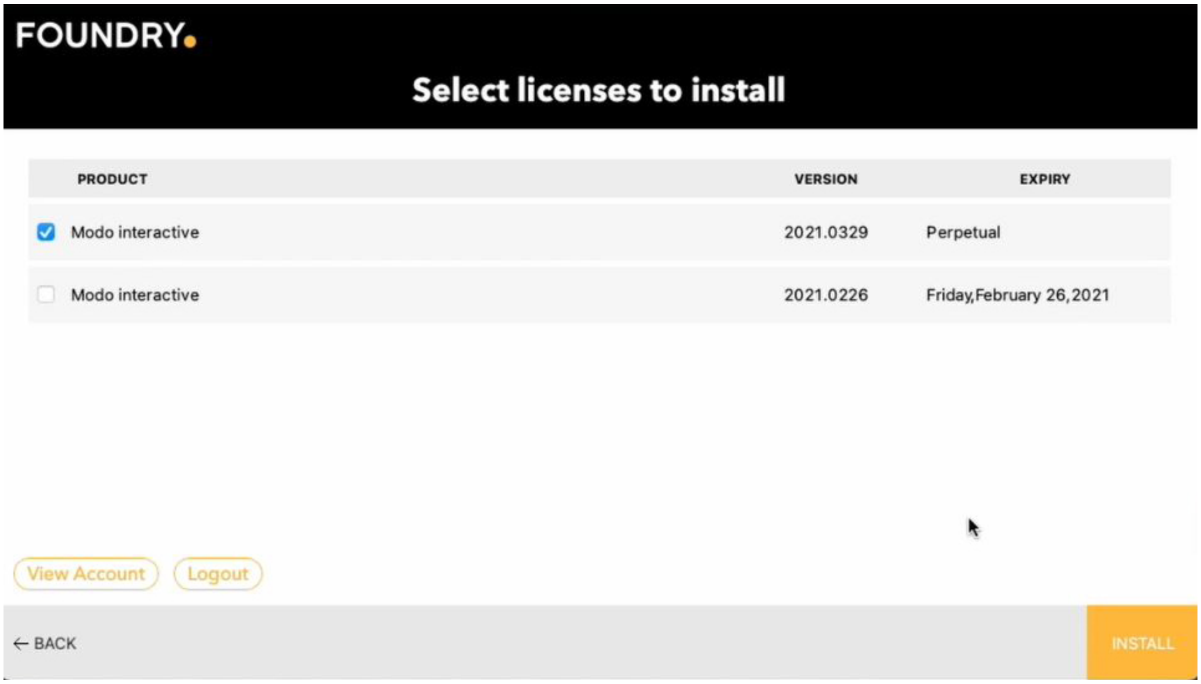
Request a Trial or purchase a License for Modo

Click here to log in to your Foundry account.

3. Click the button for **Login License** and **Sign In** with your Foundry account email address and password.

   A license is created in the following directory: **/Users/{username}/Library/Application Support/Foundry/Tokens**

   The license key looks something like this: **c58edf7e-17ab-435b-8d8a-b3a9b347ab11.lic**
4. Select the license you want to install from the list and click **Install**.



PRODUCT	VERSION	EXPIRY
☑ Modo interactive	2021.0329	Perpetual
☐ Modo interactive	2021.0226	Friday,February 26,2021

5. Once the license is installed, click **Done** to close the Licensing App.

   You can now use Modo.

Note: A license is activated for a period of 38 days. Make sure that you connect to the internet once during the activation period to extend your access. You can check the activation period of your license in the **Help** > **Modo Licensing** menu.

6. If you need to deactivate your license on the current machine, navigate to **Help** > **Modo Licensing**, click **Manage Login Licenses**.

   • Click on **Logout** to return all licenses and logout from the machine or;

   • Select the license(s) you need to deactivate and click **Return**.

Obtaining Legacy Licenses

To obtain a legacy license, you'll need your machine's System ID (sometimes called Host ID or rlmhostid). Just so you know what a System ID number looks like, here’s an example: 000ea641d7a1.

Bear in mind that, for floating licenses, you'll need the System ID of the license server, not the machines on which you intend to run the application. There are a number of ways you can find out your machine's System ID:

- Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing/tools and run it. Click **System ID** to display it on screen.

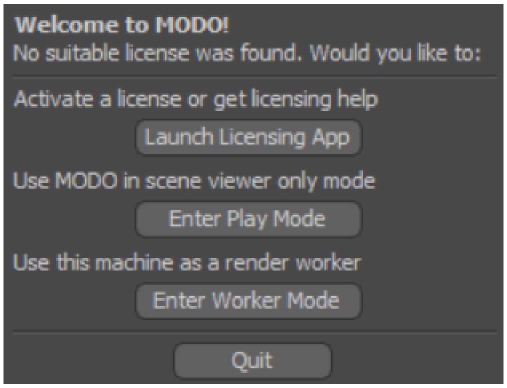
When you know your System ID, you can request a license for Foundry products from Foundry's Sales Department at sales@foundry.com.

To Install a License from Disk

If you've copied a license to a known location on disk:

1. Launch Modo using any of the methods described in Launching Modo on Mac.

   The **Licensing** dialog displays.

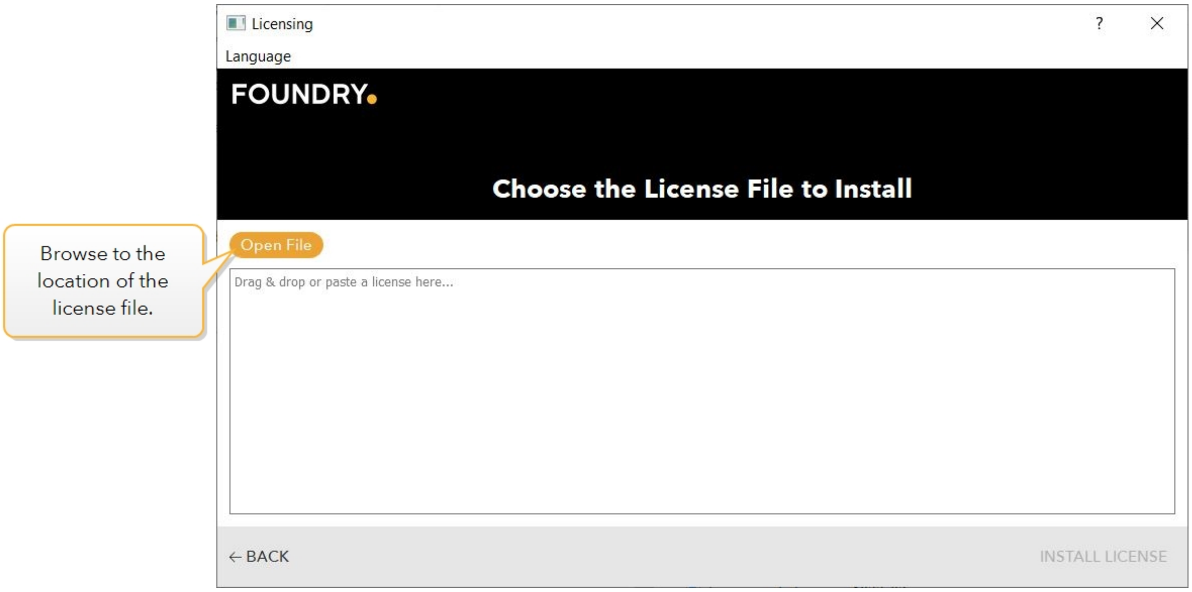


Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help
Launch Licensing App
Use MODO in scene viewer only mode
Enter Play Mode
Use this machine as a render worker
Enter Worker Mode
Quit

2. Click **Launch Licensing App** and then click the button for **Local License**.
3. Click **License File.**

   The following window opens.



Licensing
?
Language
FOUNDRY.
Choose the License File to Install
Open File
Drag & drop or paste a license here...
← BACK
INSTALL LICENSE

Browse to the location of the license file.

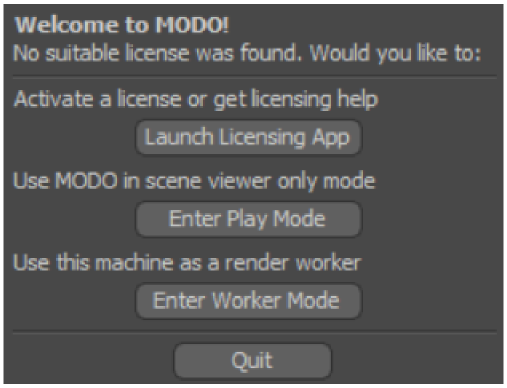
4. Click **Open File** to locate the license file.
5. Click **Open** to install the license automatically in the correct directory.

   Once the license is validated, Modo displays a notification dialog.

To Install a License from a Text File

1. Launch Modo using any of the methods described in Launching Modo on Mac.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help

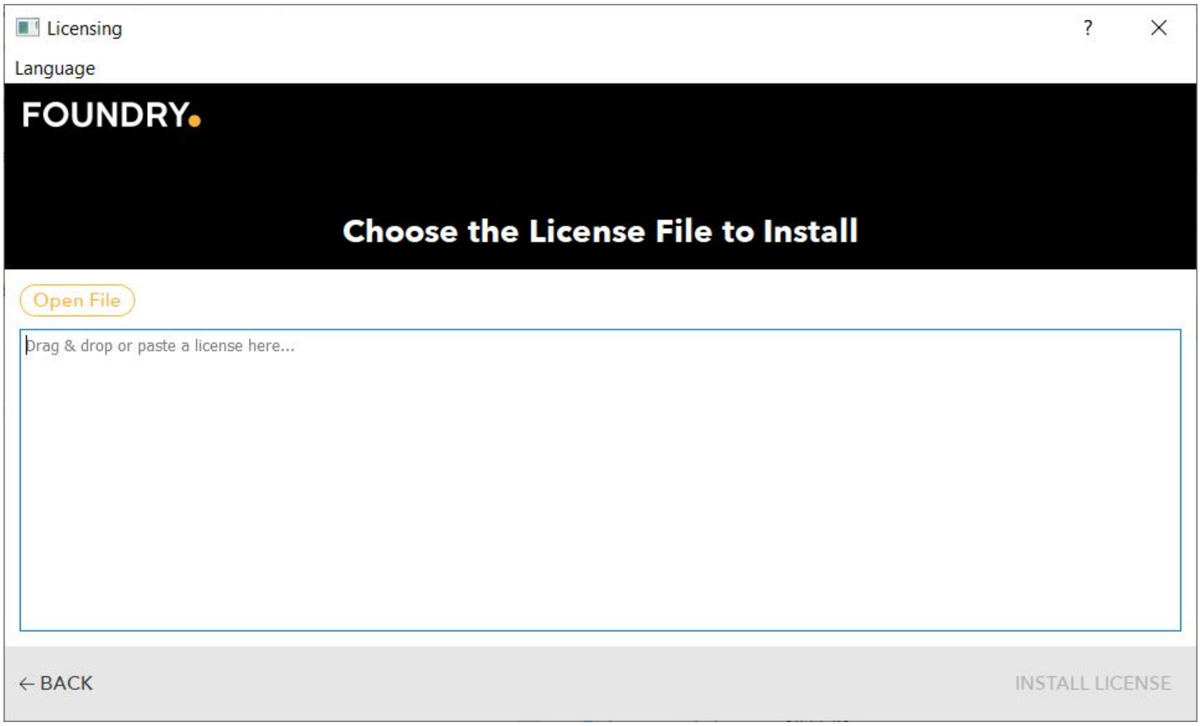
Use MODO in scene viewer only mode

Use this machine as a render worker

Launch Licensing App
Enter Play Mode
Enter Worker Mode
Quit

2. Click **Launch Licensing App** and then click the button for **Local License**.
3. Click **License File.**

   The following window opens.



Choose the License File to Install

Drag & drop or paste a license here...

Licensing
?
Language
FOUNDRY.
Open File
← BACK
INSTALL LICENSE

4. Copy and paste the license text into the window and click **Install license**.

   The license is automatically installed on your machine in the correct directory.

To Install a Floating License

If you requested a floating license from Foundry, you receive your license key (**foundry.lic**) in an e-mail or internet download. You should also receive the Foundry License Utility (FLU) application to help you install the license key on the license server machine. The server manages licenses for the client machines on your network.

Note: The FLU is also available to download from https://www.foundry.com/licensing/tools.

1. Make sure you have saved both the license key (**foundry.lic**) and the FLU application in the same directory.
2. Run the FLU application.

   The license key automatically appears in the FLU window if the FLU and **foundry.lic** are in the same directory.

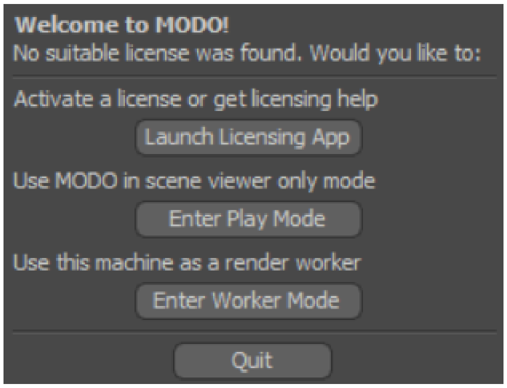
Tip: If they are not in the same directory, you can either copy and paste the contents of the license key or drag-and-drop the file into the FLU window.

3. Click **Install**.

   This checks the license file and, provided that the license is valid, installs it into the correct directory.
4. In order for the floating license to work, you will need to install the Foundry Licensing Tools (FLT) on the license server machine.

   For more information on how to install floating licenses, refer to the *FLT User Guide*, which you can download from our website https://www.foundry.com/licensing
5. Once your license server is up and running, launch Modo on the client machine.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

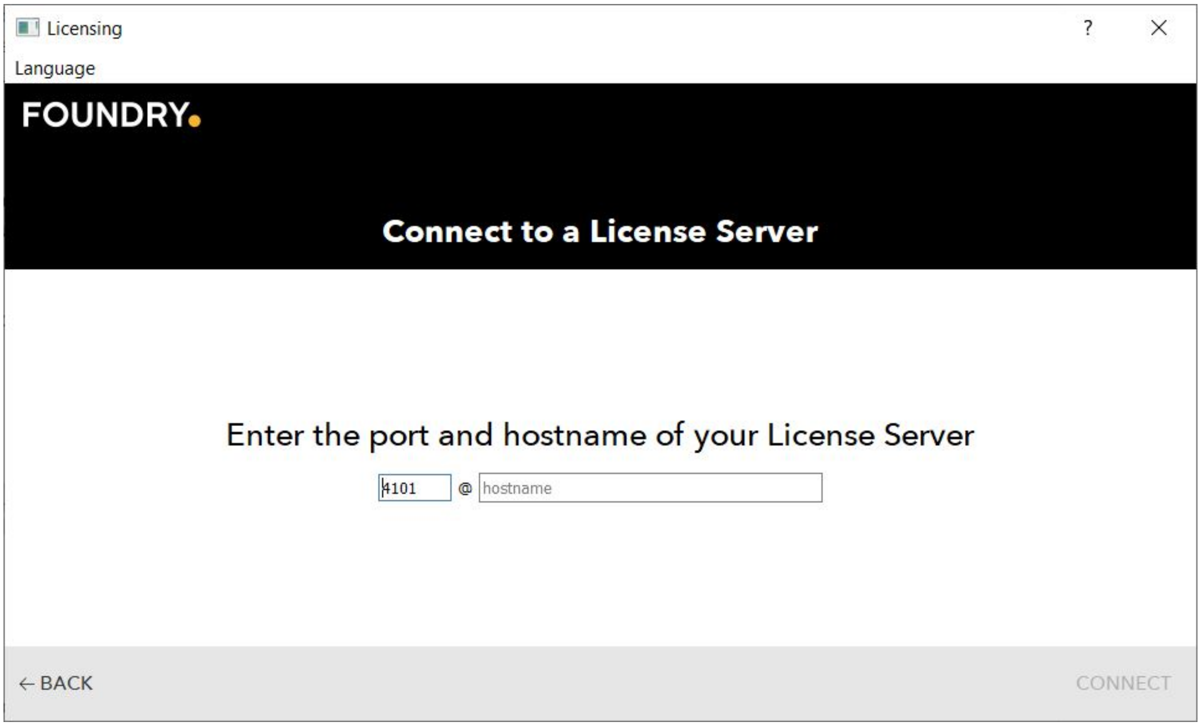
Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

6. Click **Launch Licensing App** and then click the button for **Local License**.

7. Click **Server** and enter the server address in the field provided. The format for the server name is: <port>@<servername>, for example, 30001@red.


Licensing
?
Language
FOUNDRY.
Connect to a License Server
Enter the port and hostname of your License Server
4101
@
hostname
← BACK
CONNECT

Note: You must perform these steps on each client machine that requires a license from the server.

Roaming for Floating Licenses

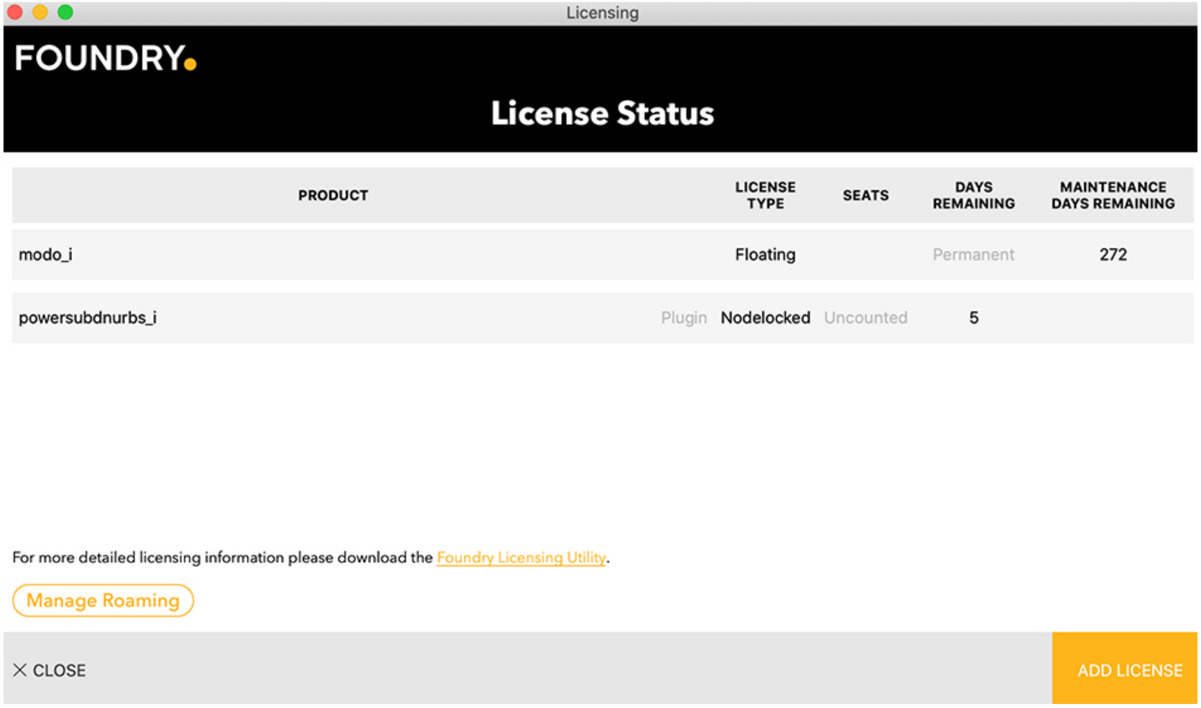
You can check out a roaming license from the pool of floating licenses available on a server and use it offline, on the system used to check it out, for a specified number of days, up to a maximum of 30 days. After the specified time the license expires and you have to connect to the server to renew or return it.

When you claim a license to use offline, the number of licenses available on the server decreases by one until you return the license.

To start roaming:

1. Make sure you're connected to your license server.
2. From the application menu bar, click **Help** > **Modo Licensing**.

   This displays the **Licensing App**.

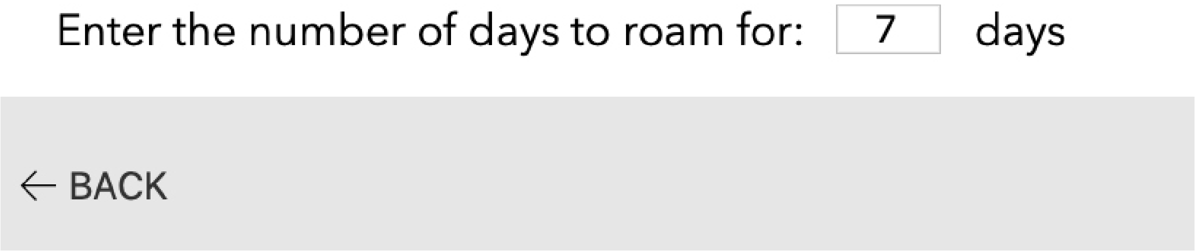

Licensing
FOUNDRY
License Status

PRODUCT		LICENSE TYPE	SEATS	DAYS REMAINING	MAINTENANCE DAYS REMAINING
modo_i		Floating		Permanent	272
powersubdnurbs_i	Plugin	Nodelocked	Uncounted	5	

For more detailed licensing information please download the Foundry Licensing Utility.

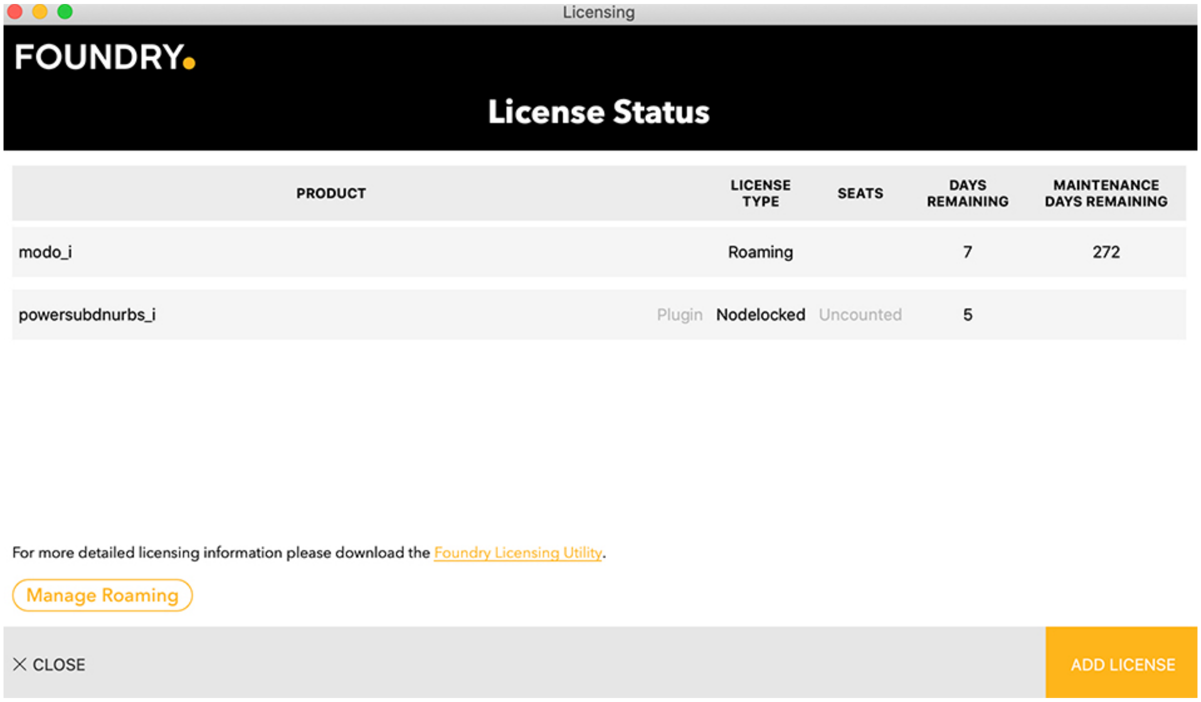
Manage Roaming
CLOSE
ADD LICENSE

3. Click **Manage Roaming**.
4. At the bottom of the dialog, enter the number of days for which you want to check the license out, then click **Roam Licenses**.



Enter the number of days to roam for: 7 days

Note: The value must be between **1** and **30**.


Licensing
FOUNDRY
License Status
Manage Roaming
CLOSE
ADD LICENSE

PRODUCT		LICENSE TYPE	SEATS	DAYS REMAINING	MAINTENANCE DAYS REMAINING
modo_i		Roaming		7	272
powersubdnurbs_i	Plugin	Nodelocked	Uncounted	5	

For more detailed licensing information please download the Foundry Licensing Utility.

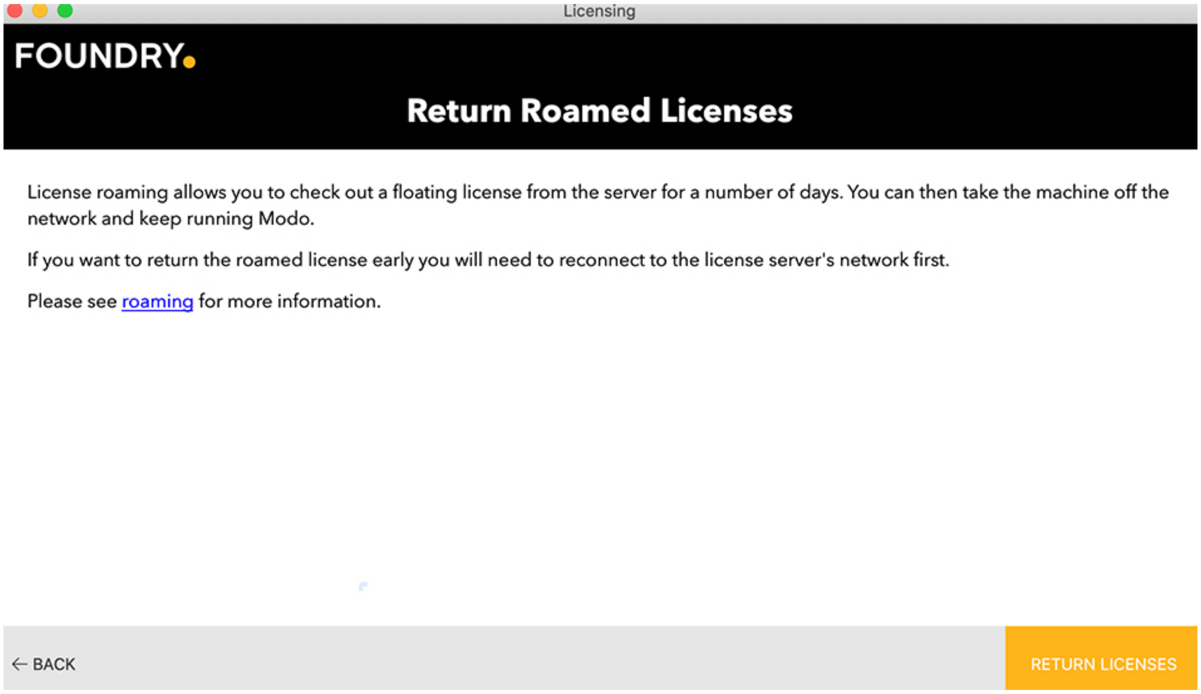
On the license status page, your license is now listed as Roaming.

You can now run Modo offline for the specified number of days.

To return a license:

1. Make sure you're connected to the license server.
2. From the application menu bar, click **Help** > **Modo Licensing**.

   This displays the **Licensing App**.
3. Click **Manage Roaming**.
4. Click **Return Licenses**.



Return Roamed Licenses

License roaming allows you to check out a floating license from the server for a number of days. You can then take the machine off the network and keep running Modo.

If you want to return the roamed license early you will need to reconnect to the license server's network first.

Please see roaming for more information.

5. Click **Close**.
6. Close and re-open Modo for the changes to take effect.

Deactivating a Modo License

Login-based licensing allows you to deactivate your license from one machine and then activate it on another. This gives you the flexibility to easily use your Modo Subscription or Modo with Maintenance license on different machines when or if you need to.

You can deactivate your login-based license directly in Modo or through the Foundry website.

Deactivating a License in Modo

1. Launch Modo and click **Help** > **Modo Licensing** in the main menu bar.

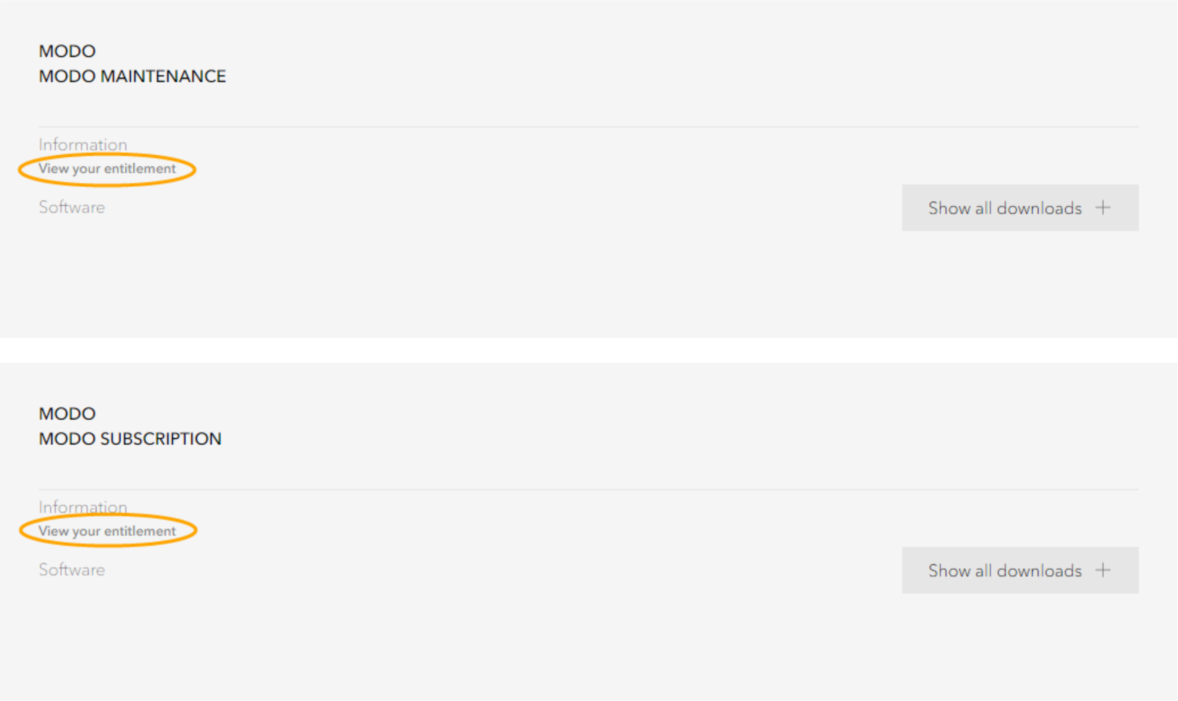
   The **Licensing** dialog displays the active license installed on the machine.
2. Click **Advanced** and then **Deauthorise this device** to deactivate the license and remove it from the machine.
3. Quit Modo.

Deactivating a License on Our Website

You can also deactivate the license from your activated machine through the **My Account** pages on our website. This is useful if you need to use your license at home but forgot to log out of Modo at work.

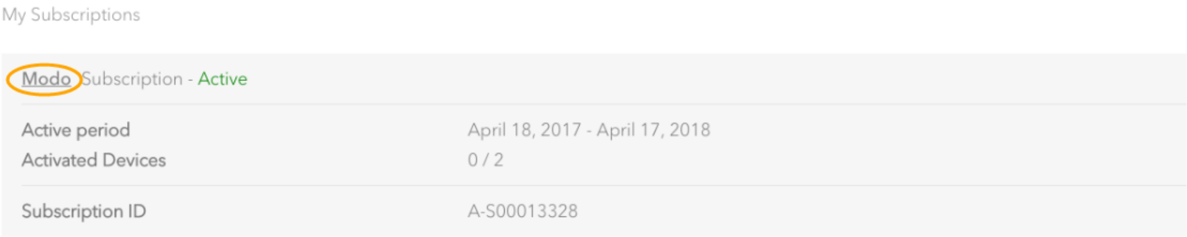
1. Go to the Foundry My Account - Products page and login if prompted.

   For more information about Foundry user accounts, see Getting a Foundry User Account.
2. Navigate to your **Subscription** or **with Maintenance** product and click **View your entitlement**.


MODO
MODO MAINTENANCE
Information
View your entitlement
Software
Show all downloads +
MODO
MODO SUBSCRIPTION
Information
View your entitlement
Software
Show all downloads +

Tip: If you're having difficulty viewing the entitlement page log out of the website using this link, https://www.foundry.com/user/logout, and then log back in again.

3. Click **Modo** to see the details of your subscription.


My Subscriptions

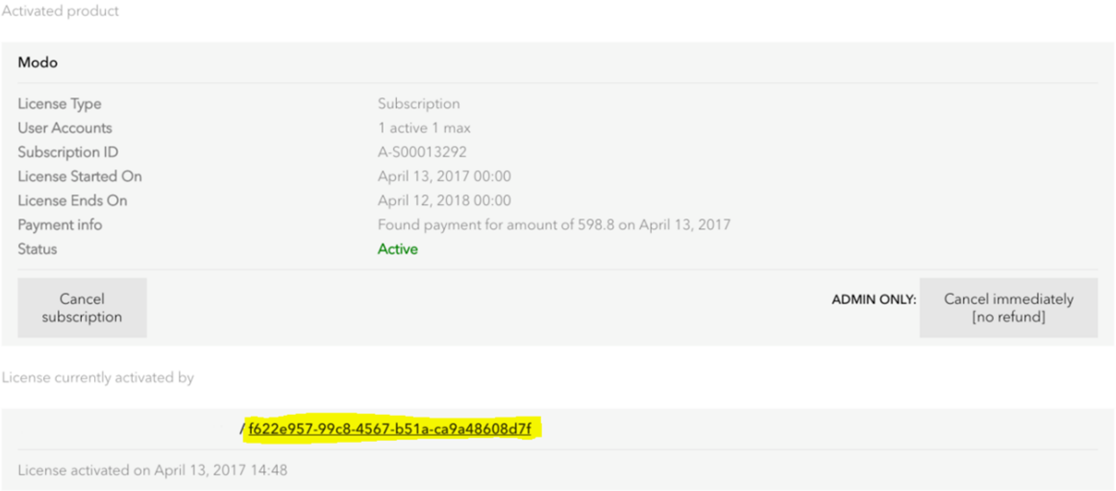
Modo Subscription - Active

Active period April 18, 2017 - April 17, 2018

Activated Devices 0 / 2

Subscription ID A-S00013328

4. Click on the license key under the **License currently activate by** section.



Activated product

Modo

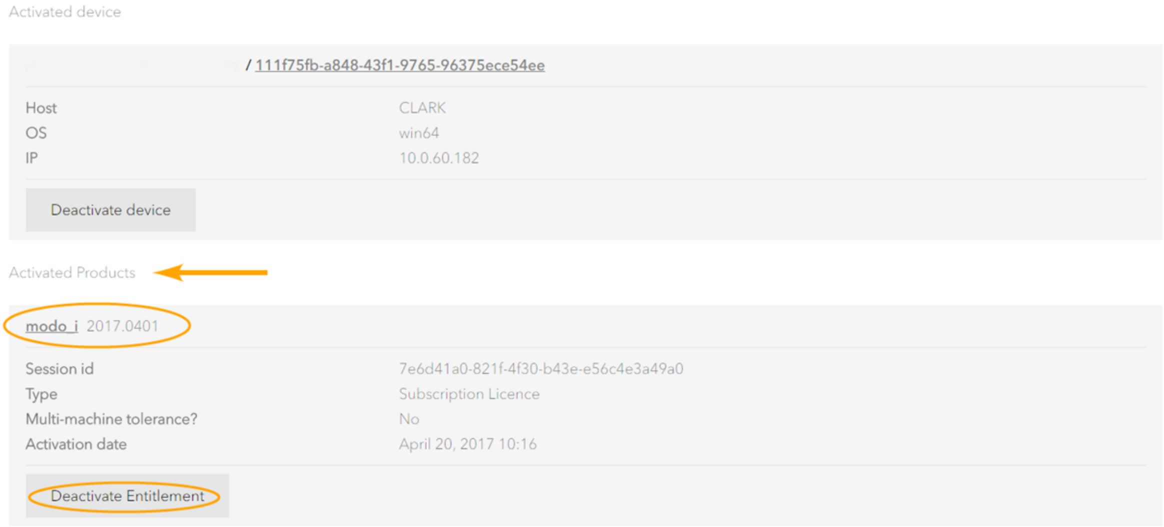
License Type	Subscription
User Accounts	1 active 1 max
Subscription ID	A-S00013292
License Started On	April 13, 2017 00:00
License Ends On	April 12, 2018 00:00
Payment info	Found payment for amount of 598.8 on April 13, 2017
Status	Active

License currently activated by

/ f622e957-99c8-4567-b51a-ca9a48608d7f

License activated on April 13, 2017 14:48

5. In the **Activated Products** section, under **modo_i**, click **Deactivate Entitlement** to remove the Modo license from that machine.



Activated device

/ 111f75fb-a848-43f1-9765-96375ece54ee

Host	CLARK
OS	win64
IP	10.0.60.182

Activated Products

modo_i 2017.0401

Session id	7e6d41a0-821f-4f30-b43e-e56c4e3a49a0
Type	Subscription Licence
Multi-machine tolerance?	No
Activation date	April 20, 2017 10:16

Warning: If you click on **Deactivate device**, all login-based licenses (including any Nuke Non-Commercial or Mari Non-Commercial) will be removed from the machine.

Tip: For an overview of the documentation and quick access to the main topics, go to Modo Home.

Installing on Linux

After installation, Modo is run from either desktop icons, the browser, or from the **Terminal** using arguments.

Modo System Requirements

Qualified Operating Systems

- Linux Rocky 9 (64-bit)

Note: For Ubuntu 22.04 installs (which have not been validated by formal QA testing), additional libraries need to be installed for Modo to work. From the terminal issue the following two commands, to prevent a crash on launch:

```
sudo apt-get install libxcb-xinerama0
sudo apt-get install libxcb-xinput0
```

Other operating systems may work with Modo, but have not been fully tested. If you have any problems with a particular system, please contact our support team.

See the Modo Release Notes for the full list of system requirements.

Installing Modo on Linux

To install Modo on Linux, see either:

- Installing Modo from the Terminal, or
- Installing Modo Remotely from the Terminal.

Installing Modo from the Terminal

To install Modo from the Terminal, do the following:

1. Download the correct **.sh** installation file from our website at www.foundry.com.

The **.sh** file is saved to your preferred download location.

2. Open the Terminal.
3. Use the **cd** (change directory) command to move to the directory where you saved the installation file. For example, if you saved the installation file in **/home/user1/Downloads**, use the following command and press **Return**:
   ```
   cd /home/user1/Downloads
   ```
4. Run the installer.
   ```
   ./Modo17.1v1.<build_num>.sh
   ```
5. Follow the on-screen instructions. By default, Modo is installed to the directory to which you saved the installer.
6. If you didn't add a license key during the installation, do that now. Proceed to Launching Modo on Linux.

Note: By installing Modo silently, you agree to the terms of the End User Licensing Agreement. To see this agreement, please refer to the **End User License Agreement** link listed under the **Resources** tab in the online help or run the installer in standard, non-silent mode.

Installing Modo Remotely from the Terminal

If you need to install Modo on render machines using the Terminal, do the following:

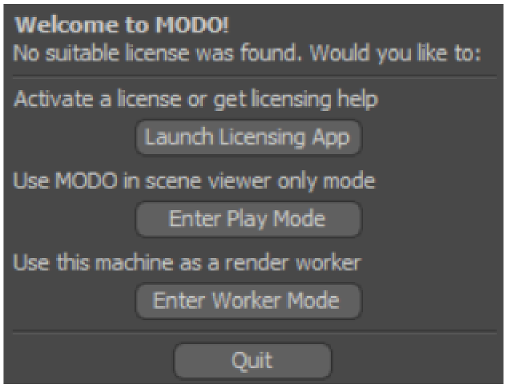
1. Download the correct **.sh** installation file from our website at www.foundry.com.
2. Use the following Terminal command to log in to your render machine as root:
   ```
   ssh root@render_machine
   ```
   Replace render_machine with the name of your render node.
3. Make a directory to install Modo to:
   ```
   mkdir /usr/local/Modo17.1v1
   ```
4. Copy the installer file from the machine that you downloaded it on to your render machine with a command like:
   ```
   scp root@download_machine:/tmp/Modo17.1v1.<build_num>.sh root@render_
   machine:/usr/local/Modo17.1v1/
   ```
   Replace download_machine with the name of the machine you downloaded the installer file to, and render_machine with the name of your render node.
5. Run the installer.
   ```
   ./Modo17.1v1.<build_num>.sh
   ```
6. Repeat steps 3-6 for each render machine.

Launching Modo on Linux

To launch Modo on Linux, do one of the following:

- Open the Modo application directory where you installed Modo, and double-click the Modo icon.
- Using the Terminal, navigate to the Modo application directory where you installed Modo, and enter:
    - **./modo** to launch Modo.

If you already have a valid license, the graphical interface appears. If you don't have a license, or haven't installed one yet, the **Licensing** dialog appears on start up.



Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

Launch Licensing App

Clicking this button opens the Licensing app where you can manage your licenses and run diagnostic tools.

For more information on how to license Modo, refer to Licensing Modo on Linux.

Running in Play Mode

Play mode places Modo in a special state where it acts as a scene viewer. In this state the interfaces are mostly disabled, but it allows you to load scenes and render and save them, both in Preview and using **F9**; opening the **Render Display** window.

One unique function of Play mode is its ability to detect scene changes and auto-reload the scene. What this means is any scene opened from over a network and previewed, when updated on another system, saves files to the scene file, reloads the scene on the networked system and updates the preview viewport. This allows the secondary system to act as a dedicated preview system without the need for a second license.

Running in Worker Mode

Worker mode is related to network rendering and places Modo in a state where other Modo installations can control its renderer. For information on how to set up Network Rendering, please reference the Network Rendering page.

Updates and Statistics

When Modo launches for the first time with a valid license, you are presented with a dialog, which enables you to set preferences to check for updates and to submit usage statistics.

Updates

Get the latest reminders to update Modo. You can have Modo automatically check for updates on every launch, daily or weekly. By default, this preference is set to **Every Launch**.

Usage Statistics

To help improve the quality of Modo, you can agree to submit usage data to the Modo Development team. These statistics are submitted anonymously and include details on the build number run, the system platform, the application active and idle times, and whether sessions exit successfully or not. In the event of a crash, the most recently executed commands are also submitted.

Note: Graphics card information is not submitted.

Tip: To reset your settings, open **System** > **Preferences**, expand **Defaults** > **Application** and reset the options for the following:
Automatically Check for Updates
Automatically Submit Usage Statistics

The following settings are available: **Never**, **Every Launch**, **Daily**, or **Weekly**.

Article: You can disable the Alert dialog in the Form Editor. For details on how to do this, see Q100345: How to disable all new Alert dialog boxes for the Modo 11 series onward.

Running in Safe Mode

In the unlikely event of a crash, you can run Modo in Safe Mode to help us determine what is causing the problem. When Modo runs in Safe Mode, it doesn't load any third-party plug-ins, kits, or user customizations and only runs using the contents of the Modo install directory. Safe Mode is primarily used for debugging purposes. When running in Safe Mode, a pristine configuration file is used.

Note: When exiting Safe Mode, your existing configuration directory (**modo_17.1.rc**) is not overwritten.

A backup of configuration directory is created at shutdown, when changes are saved. This backup directory can be renamed to restore the previous state if configuration changes cause issues. When a corrupt configuration is detected, it is renamed, instead of deleted, which allows you to recover data from corrupted configuration directory.

Save your configuration directory and submit it to Support for debugging. Your configuration directory is stored in the **/home/**<username>**/.luxology** directory.

To run in Safe Mode:

- Using the Terminal, navigate to the Modo application directory where you installed Modo, and enter:

```
./Modo17.1v1 -safemode -dbon:noconfig
```

If Modo successfully launches in Safe Mode, click **File** > **Config Save** to overwrite your existing configuration file that was causing Modo to crash.

Licensing Modo on Linux

The following licensing methods are available:

- **Log-in Licenses** - A license tied to your Foundry website account, available for either subscriptions or perpetual maintenance entitlements. They can be activated on up to 2 machines at once and deactivated if you need to use the entitlement elsewhere. Activating Modo installs a rolling temporary license which can be used offline for up to 37 days. The license is refreshed whenever Modo is connected to the internet.

The following video walks you through using Modo's log-in licensing tool.

Note: Log-in licenses for Modo plug-ins Power Translators and Power Sub-D to NURBS are added alongside your Modo log-in license entitlements. If you install one of these plug-ins when the device is already authorized, please go to **Help** > **Modo Licensing** > **Advanced** > **Deauthorise this device**, and then log-in again in order to display plug-in licenses. All of your entitlements are added together.

- **Activation Keys** - A code which can be used to generate a license after purchase. Activation Keys can be provided for node-locked, floating and log-in licenses. They can be activated from the application's Licensing dialog or using the Activate a Product Page: https://www.foundry.com/licensing/activate-product.
- **Node Locked Licenses** - A license locked to a specific machine. Node-locked licenses, sometimes called uncounted licenses, do not require a license server or other resources to run. The license is for a single machine, if you need to move the license you’ll have to submit a transfer request.
- **Floating Licenses** - A license locked to a machine that acts as a license server to enable any machine on the network to run an application. The license is installed on the server machine along with license server software. The license server software is installed with the Foundry Licensing Utility.

The number of machines that can run the application at the same time is limited by the number of floats in the license. Any machine can request a license, the server will issue a license as long as one is available. When the application exits it releases the license.

You can find more information about licensing Foundry products at learn.foundry.com/licensing

Note: If you are simply installing render nodes on a networked system, no license is required for Modo. Modo can also be run in Play mode to load scenes, navigate the viewport to look at them, and to render them (stills or animations). This allows you to render scenes on one system while continuing to work on another using your license. Play mode is available in the licensing dialog by clicking **Enter Play Mode**. For more information, see Running in Play Mode.

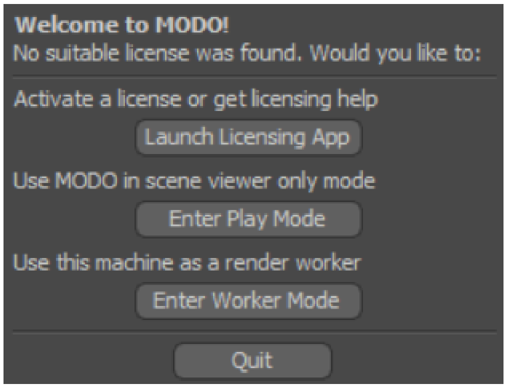
In all other instances, you need to have a license or a trial version of Modo.

To Install a Log-in License

To activate your license, follow these steps:

1. Launch Modo using any of the methods described in Launching Modo on Linux.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help

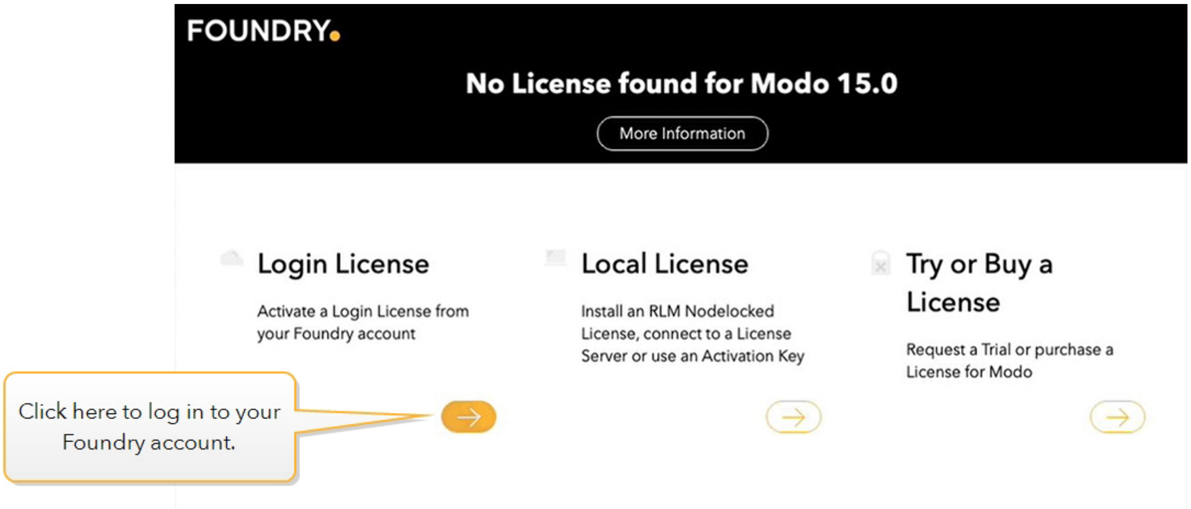
Use MODO in scene viewer only mode

Use this machine as a render worker

Launch Licensing App
Enter Play Mode
Enter Worker Mode
Quit

2. Click **Launch Licensing App**.

   The following window appears.



FOUNDRY
More Information
Login License
Local License
Try or Buy a License

No License found for Modo 15.0

Activate a Login License from your Foundry account

Install an RLM Nodelocked License, connect to a License Server or use an Activation Key

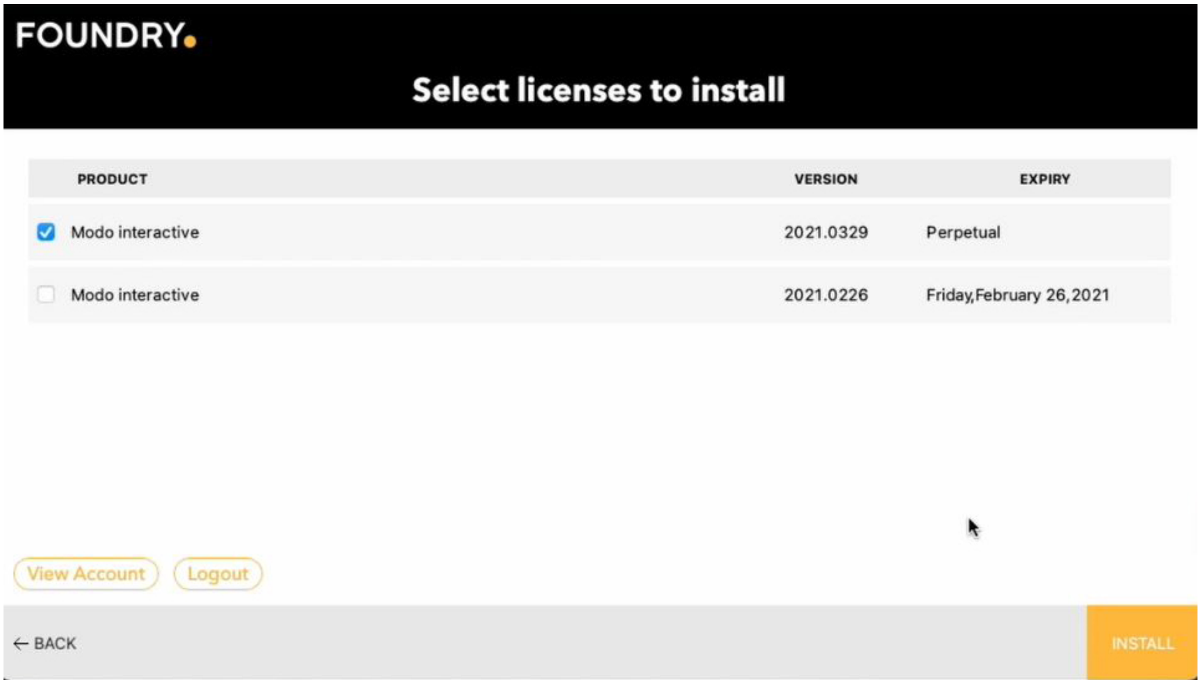
Request a Trial or purchase a License for Modo

Click here to log in to your Foundry account.

3. Click the button for **Login License** and **Sign In** with your Foundry account email address and password.

   A license is created in the following directory: **$HOME/.local/share/Foundry/Tokens** or **$XDG_DATA_HOME/Foundry/Tokens** if the $XDG_DATA_HOME variable is set.

   The license key looks something like this: **c58edf7e-17ab-435b-8d8a-b3a9b347ab11.lic**
4. Select the license you want to install from the list and click **Install**.



	PRODUCT	VERSION	EXPIRY
☑	Modo interactive	2021.0329	Perpetual
☐	Modo interactive	2021.0226	Friday,February 26,2021

5. Once the license is installed, click **Done** to close the Licensing App.

   You can now use Modo.

Note: A license is activated for a period of 38 days. Make sure that you connect to the internet once during the activation period to extend your access. You can check the activation period of your license in the **Help** > **Modo Licensing** menu.

6. If you need to deactivate your license on the current machine, navigate to **Help** > **Modo Licensing**, click **Manage Login Licenses**.
   - Click on **Logout** to return all licenses and logout from the machine or;
   - Select the license(s) you need to deactivate and click **Return**.

Obtaining Legacy Licenses

To obtain a legacy license, you'll need your machine's System ID (sometimes called Host ID or rlmhostid). Just so you know what a System ID number looks like, here’s an example: 000ea641d7a1.

Bear in mind that, for floating licenses, you'll need the System ID of the license server, not the machines on which you intend to run the application. There are a number of ways you can find out your machine's System ID:

- Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing and run it. Click **System ID** to display it on screen.

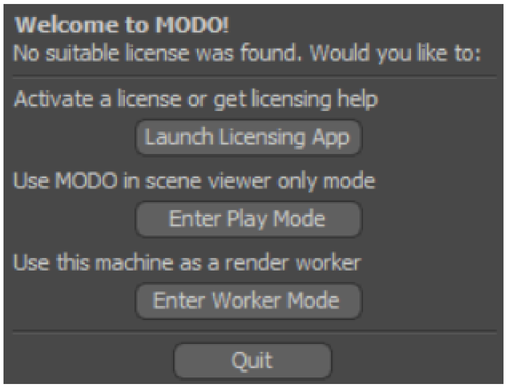
When you know your System ID, you can request a license for Foundry products from Foundry's Sales Department at sales@foundry.com.

To Install a License from Disk

If you've copied a license to a known location on disk:

1. Launch Modo using any of the methods described in Launching Modo on Linux.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help

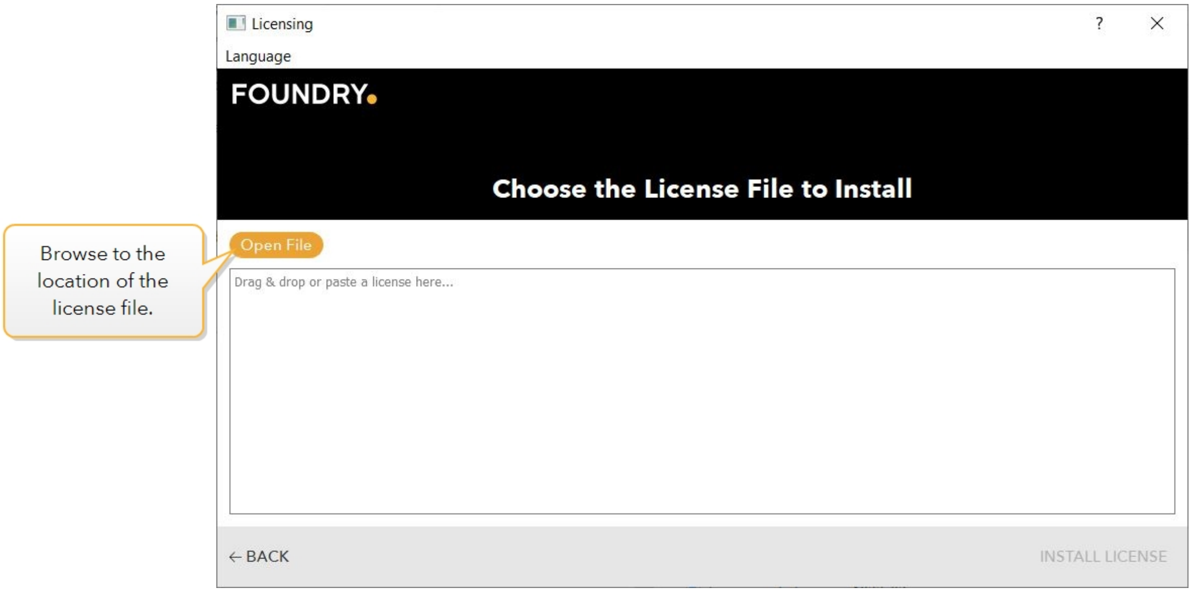
Use MODO in scene viewer only mode

Use this machine as a render worker

Launch Licensing App
Enter Play Mode
Enter Worker Mode
Quit

2. Click **Launch Licensing App** and then click the button for **Local License**.
3. Click **License File.**

   The following window opens.



Licensing
?
×
Language
FOUNDRY
Choose the License File to Install
Open File
Drag & drop or paste a license here...
← BACK
INSTALL LICENSE

Browse to the location of the license file.

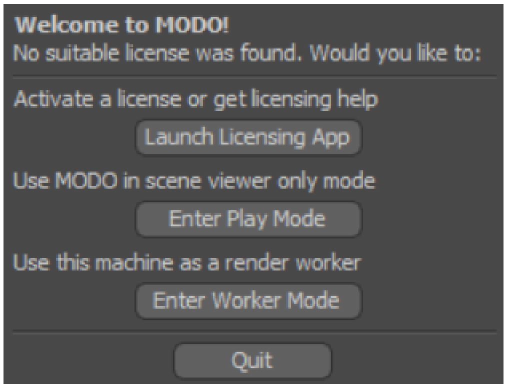
4. Click **Open File** to locate the license file.
5. Click **Open** to install the license automatically in the correct directory.

   Once the license is validated, Modo displays a notification dialog.

To Install a License from a Text File

1. Launch Modo using any of the methods described in Launching Modo on Linux.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

Activate a license or get licensing help

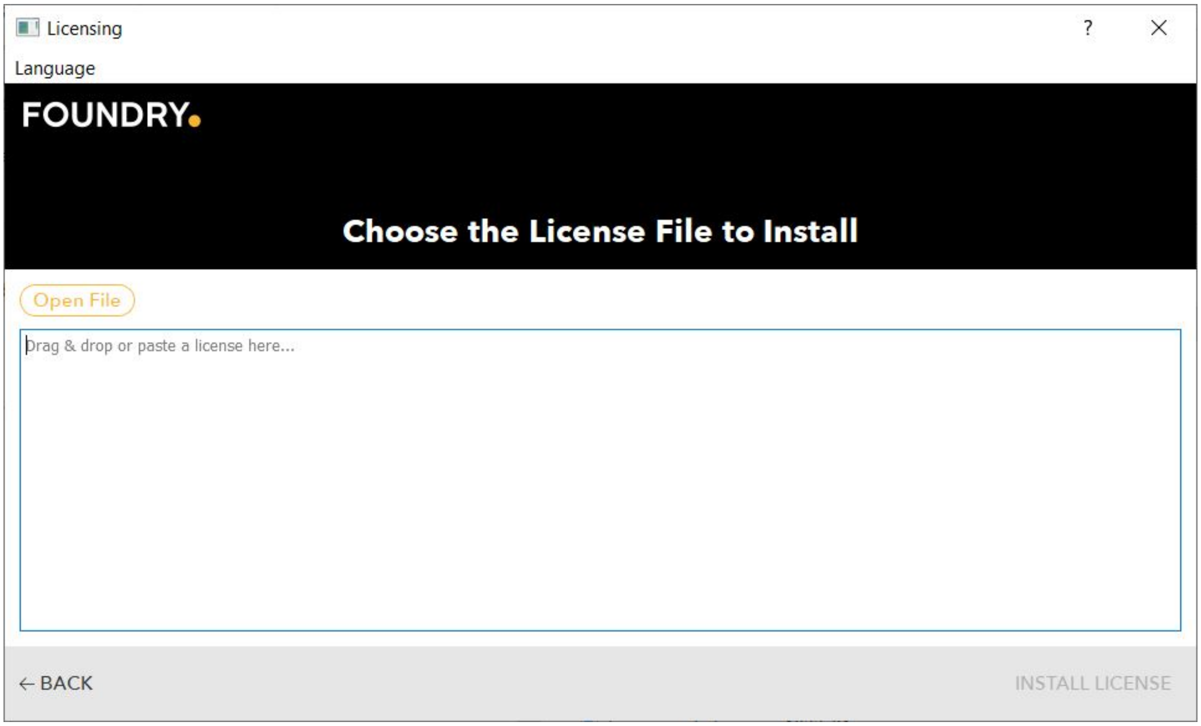
Use MODO in scene viewer only mode

Use this machine as a render worker

Launch Licensing App
Enter Play Mode
Enter Worker Mode
Quit

2. Click **Launch Licensing App** and then click the button for **Local License**.
3. Click **License File.**

   The following window opens.



Choose the License File to Install

Drag & drop or paste a license here...

Licensing
?
Language
FOUNDRY
Open File
← BACK
INSTALL LICENSE

4. Copy and paste the license text into the window and click **Install license**.

   The license is automatically installed on your machine in the correct directory.

To Install a Floating License

If you requested a floating license from Foundry, you receive your license key (**foundry.lic**) in an e-mail or internet download. You should also receive the Foundry License Utility (FLU) application to help you install the license key on the license server machine. The server manages licenses for the client machines on your network.

Note: The FLU is also available to download from https://www.foundry.com/licensing.

1. Make sure you have saved both the license key (**foundry.lic**) and the FLU application in the same directory.
2. Run the FLU application.

   The license key automatically appears in the FLU window if the FLU and **foundry.lic** are in the same directory.

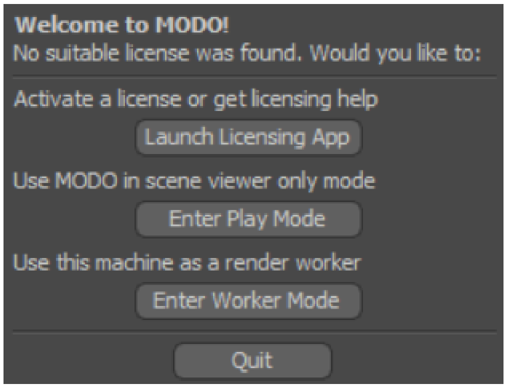
Tip: If they are not in the same directory, you can either copy and paste the contents of the license key or drag-and-drop the file into the FLU window.

3. Click **Install**.

   This checks the license file and, provided that the license is valid, installs it into the correct directory.
4. In order for the floating license to work, you will need to install the Foundry Licensing Tools (FLT) on the license server machine.

   For more information on how to install floating licenses, refer to the *FLT User Guide*, which you can download from our website https://www.foundry.com/licensing
5. Once your license server is up and running, launch Modo on the client machine.

   The **Licensing** dialog displays.



Welcome to MODO!
No suitable license was found. Would you like to:

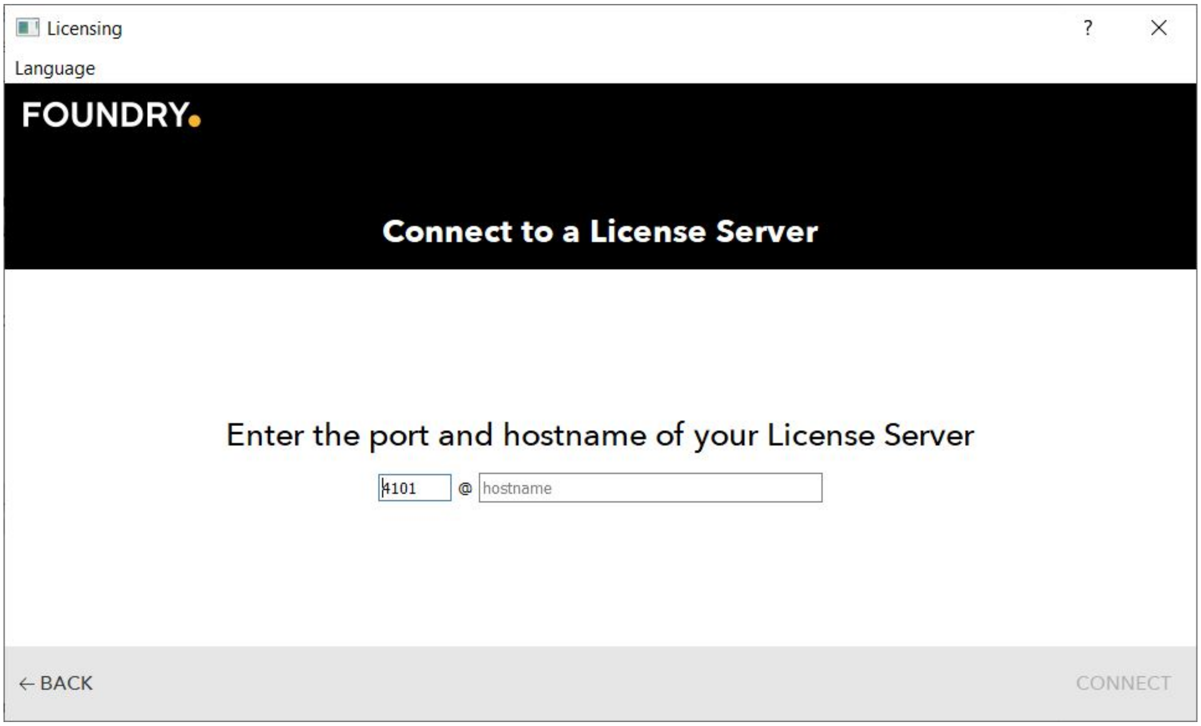
Activate a license or get licensing help

Use MODO in scene viewer only mode

Use this machine as a render worker

6. Click **Launch Licensing App** and then click the button for **Local License**.

7. Click **Server** and enter the server address in the field provided. The format for the server name is: <port>@<servername>, for example, 30001@red.



Connect to a License Server

Enter the port and hostname of your License Server

4101 @ hostname

Note: You must perform these steps on each client machine that requires a license from the server.

Roaming for Floating Licenses

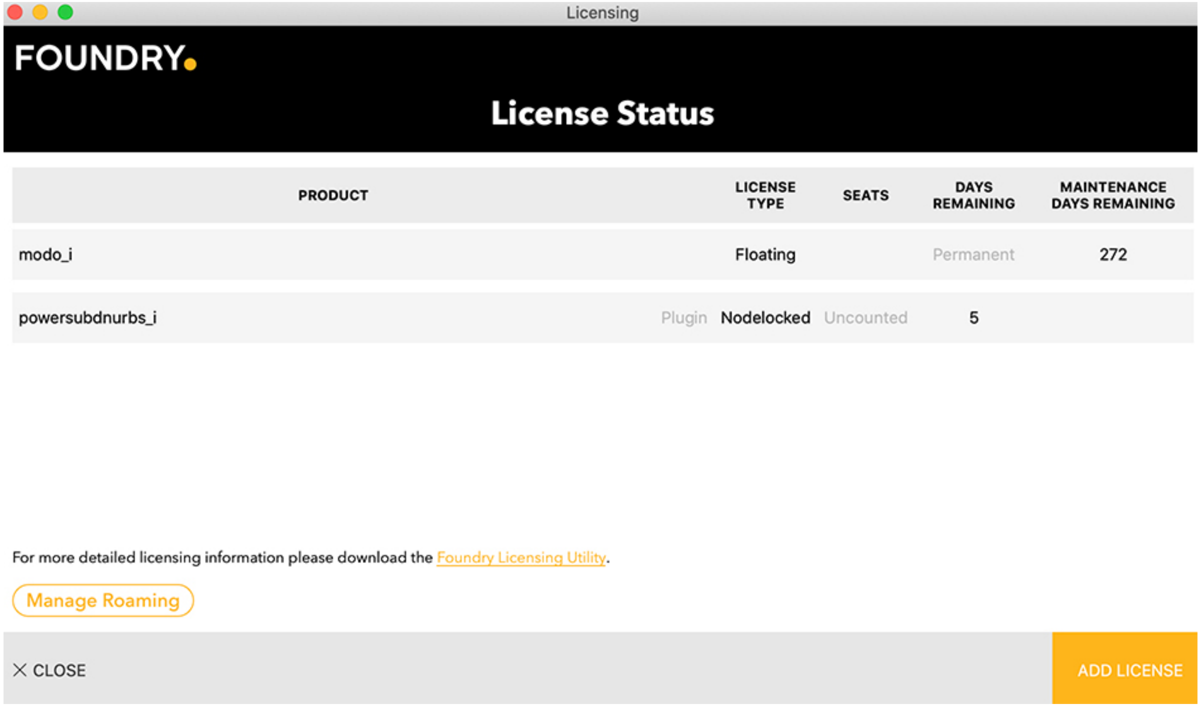
You can check out a roaming license from the pool of floating licenses available on a server and use it offline, on the system used to check it out, for a specified number of days, up to a maximum of 30 days. After the specified time the license expires and you have to connect to the server to renew or return it.

When you claim a license to use offline, the number of licenses available on the server decreases by one until you return the license.

To start roaming:

1. Make sure you're connected to your license server.
2. From the application menu bar, click **Help** > **Modo Licensing**.

   This displays the **Licensing App**.

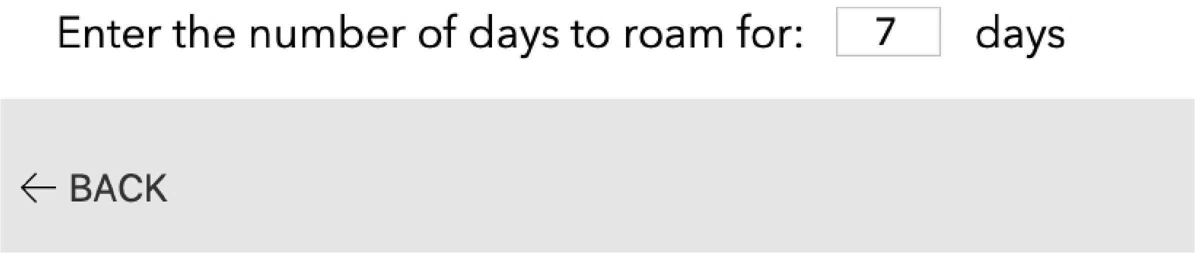

Licensing
FOUNDRY
License Status

PRODUCT		LICENSE TYPE	SEATS	DAYS REMAINING	MAINTENANCE DAYS REMAINING
modo_i		Floating		Permanent	272
powersubdnurbs_i	Plugin	Nodelocked	Uncounted	5	

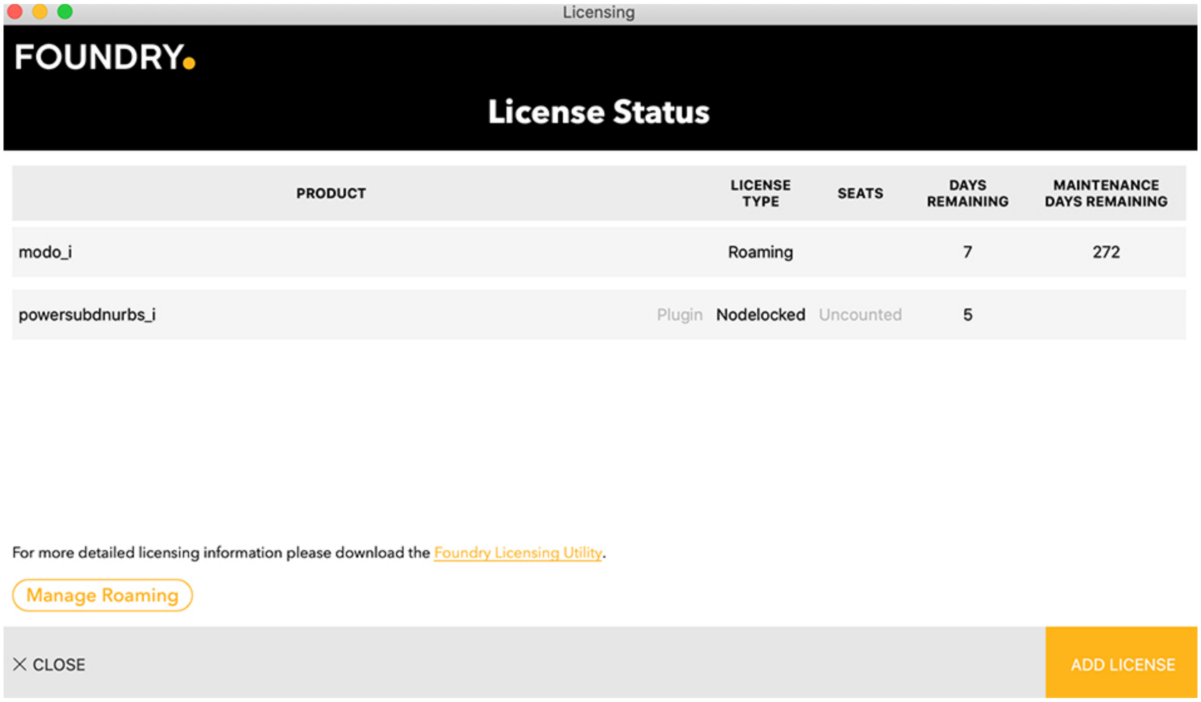
For more detailed licensing information please download the Foundry Licensing Utility.

Manage Roaming
CLOSE
ADD LICENSE

3. Click **Manage Roaming**.
4. At the bottom of the dialog, enter the number of days for which you want to check the license out, then click **Roam Licenses**.


Enter the number of days to roam for: 7 days

Note: The value must be between **1** and **30**.


Licensing
FOUNDRY
License Status

PRODUCT		LICENSE TYPE	SEATS	DAYS REMAINING	MAINTENANCE DAYS REMAINING
modo_i		Roaming		7	272
powersubdnurbs_i	Plugin	Nodelocked	Uncounted	5	

For more detailed licensing information please download the Foundry Licensing Utility.

Manage Roaming
CLOSE
ADD LICENSE

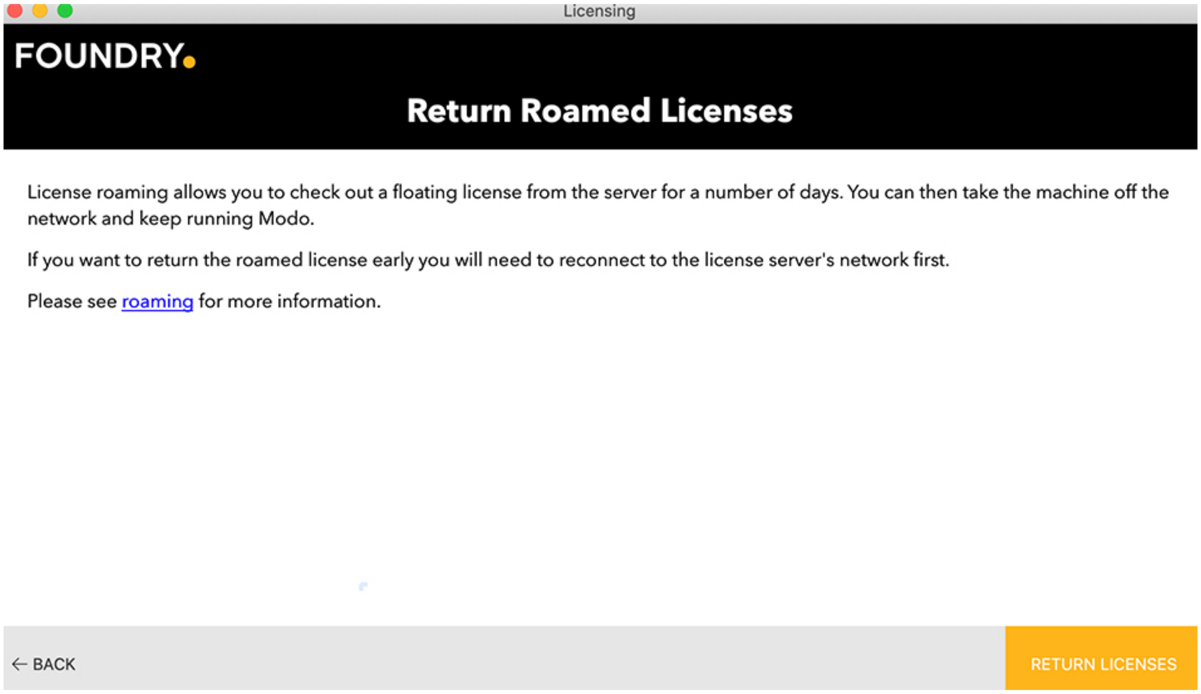
On the license status page, your license is now listed as Roaming.

You can now run Modo offline for the specified number of days.

To return a license:

1. Make sure you're connected to the license server.
2. From the application menu bar, click **Help** > **Modo Licensing**.

   This displays the **Licensing App**.
3. Click **Manage Roaming**.
4. Click **Return Licenses**.



License roaming allows you to check out a floating license from the server for a number of days. You can then take the machine off the network and keep running Modo.

If you want to return the roamed license early you will need to reconnect to the license server's network first.

Please see roaming for more information.

5. Click **Close**.
6. Close and re-open Modo for the changes to take effect.

Deactivating a Modo License

Login-based licensing allows you to deactivate your license from one machine and then activate it on another. This gives you the flexibility to easily use your Modo Subscription or Modo with Maintenance license on different machines when or if you need to.

You can deactivate your login-based license directly in Modo or through the Foundry website.

Deactivating a License in Modo

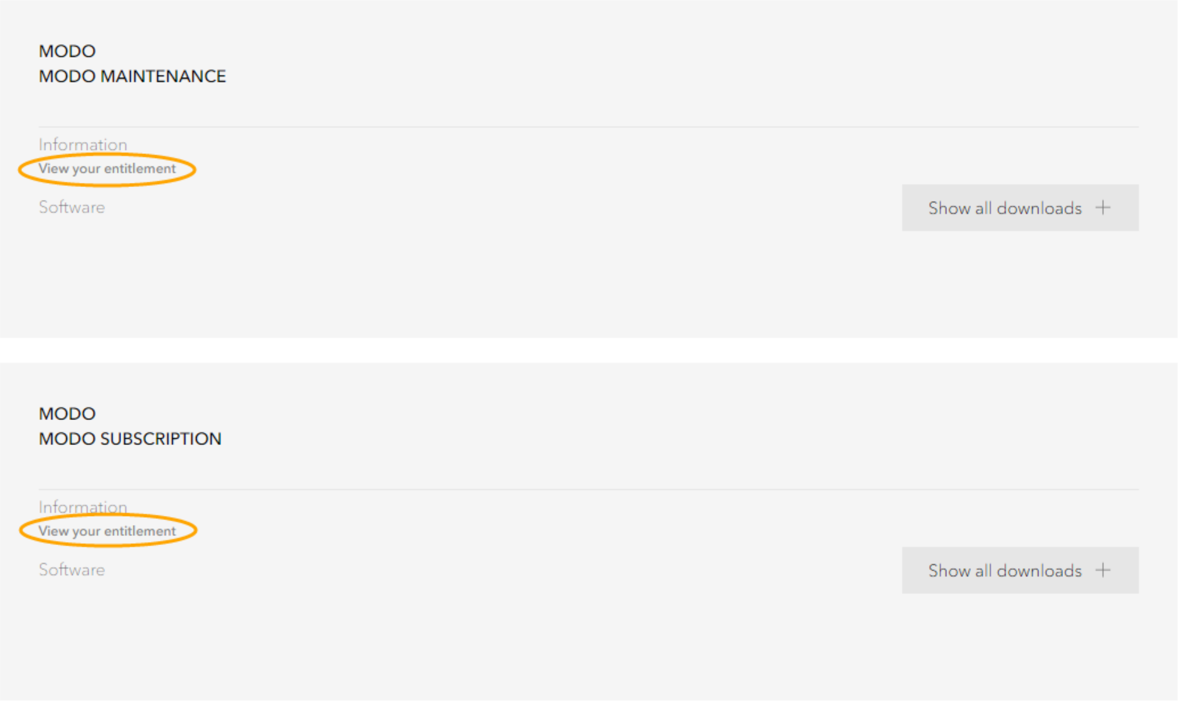
1. Launch Modo and click **Help** > **Modo Licensing** in the main menu bar.
   The **Licensing** dialog displays the active license installed on the machine.
2. Click **Advanced** and then **Deauthorise this device** to deactivate the license and remove it from the machine.
3. Quit Modo.

Deactivating a License on Our Website

You can also deactivate the license from your activated machine through the **My Account** pages on our website. This is useful if you need to use your license at home but forgot to log out of Modo at work.

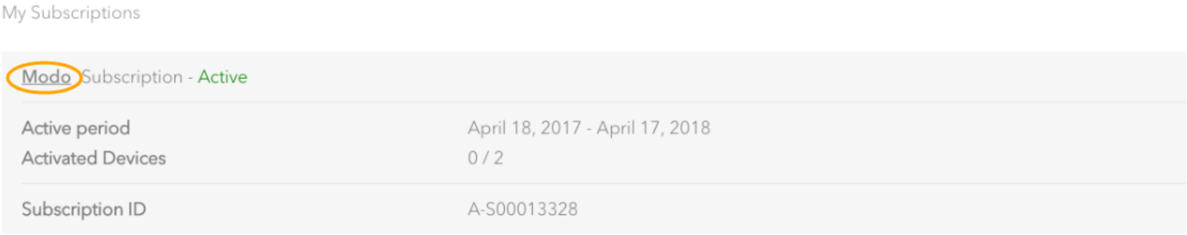
1. Go to the Foundry My Account - Products page and login if prompted.

   For more information about Foundry user accounts, see Getting a Foundry User Account.
2. Navigate to your **Subscription** or **with Maintenance** product and click **View your entitlement**.


MODO
MODO MAINTENANCE
Information
View your entitlement
Software
Show all downloads +
MODO
MODO SUBSCRIPTION
Information
View your entitlement
Software
Show all downloads +

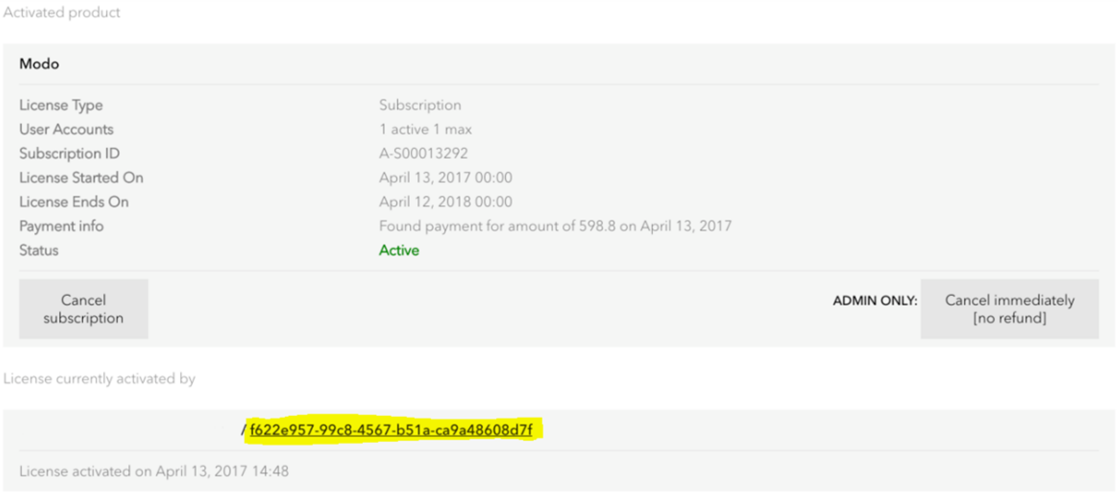
Tip: If you're having difficulty viewing the entitlement page log out of the website using this link, https://www.foundry.com/user/logout, and then log back in again.

3. Click **Modo** to see the details of your subscription.


My Subscriptions
Modo Subscription - Active

Active period	April 18, 2017 - April 17, 2018
Activated Devices	0 / 2
Subscription ID	A-S00013328

4. Click on the license key under the **License currently activate by** section.



Activated product

Modo

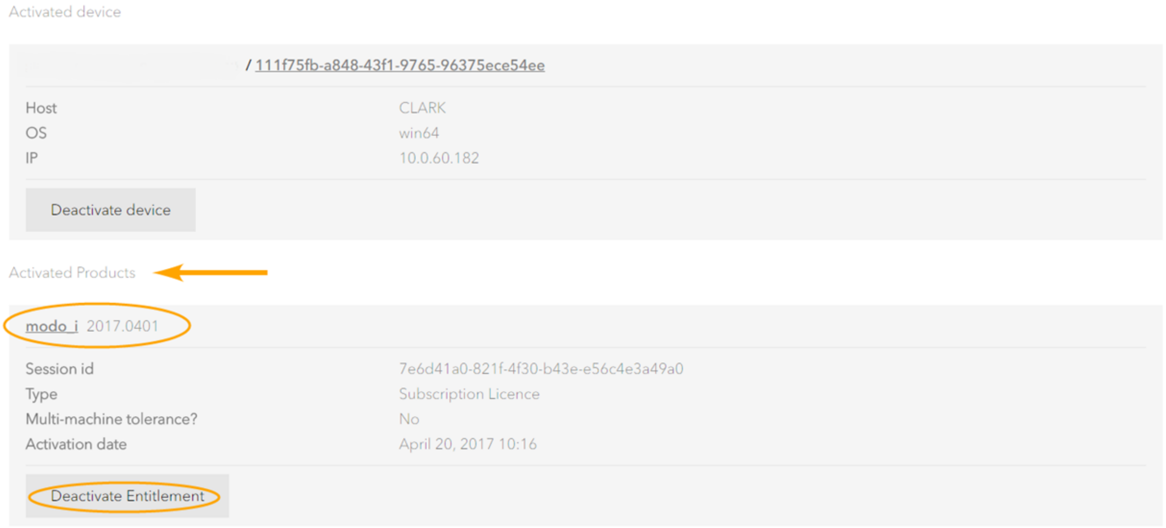
License Type	Subscription
User Accounts	1 active 1 max
Subscription ID	A-S00013292
License Started On	April 13, 2017 00:00
License Ends On	April 12, 2018 00:00
Payment info	Found payment for amount of 598.8 on April 13, 2017
Status	Active

License currently activated by

[redacted] / f622e957-99c8-4567-b51a-ca9a48608d7f

License activated on April 13, 2017 14:48

5. In the **Activated Products** section, under **modo_i**, click **Deactivate Entitlement** to remove the Modo license from that machine.



Activated device

[redacted] / 111f75fb-a848-43f1-9765-96375ece54ee

Host	CLARK
OS	win64
IP	10.0.60.182

Activated Products

modo_i 2017.0401

Session id	7e6d41a0-821f-4f30-b43e-e56c4e3a49a0
Type	Subscription Licence
Multi-machine tolerance?	No
Activation date	April 20, 2017 10:16

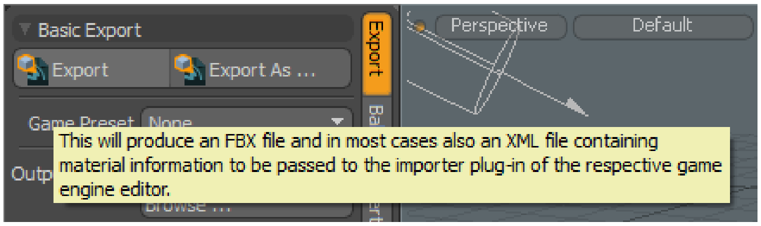
Warning: If you click on **Deactivate device**, all login-based licenses (including any Nuke Non-Commercial or Mari Non-Commercial) will be removed from the machine.

Getting Help

Modo features several forms of help, in a variety of locations.

Viewing Tool Tips

Most controls offer concise instructions in the form of tool tips. To display the tool tips, move your mouse pointer over an interface control or parameter.



This will produce an FBX file and in most cases also an XML file containing material information to be passed to the importer plug-in of the respective game engine editor.

Viewing the Context-Sensitive Help

The context-sensitive help allows you to open the documentation for a particular tool or UI element by clicking on it in the interface.

To display these descriptions in your default web browser, press **F1** on the keyboard. You can also activate the context-sensitive help in the menu bar, under **Help** > **Activate Help**.

The cursor turns into a **?** (question mark), and when you click on a UI element, the relevant Help topic is displayed in a new browser window.

Using the Help Menu in Modo

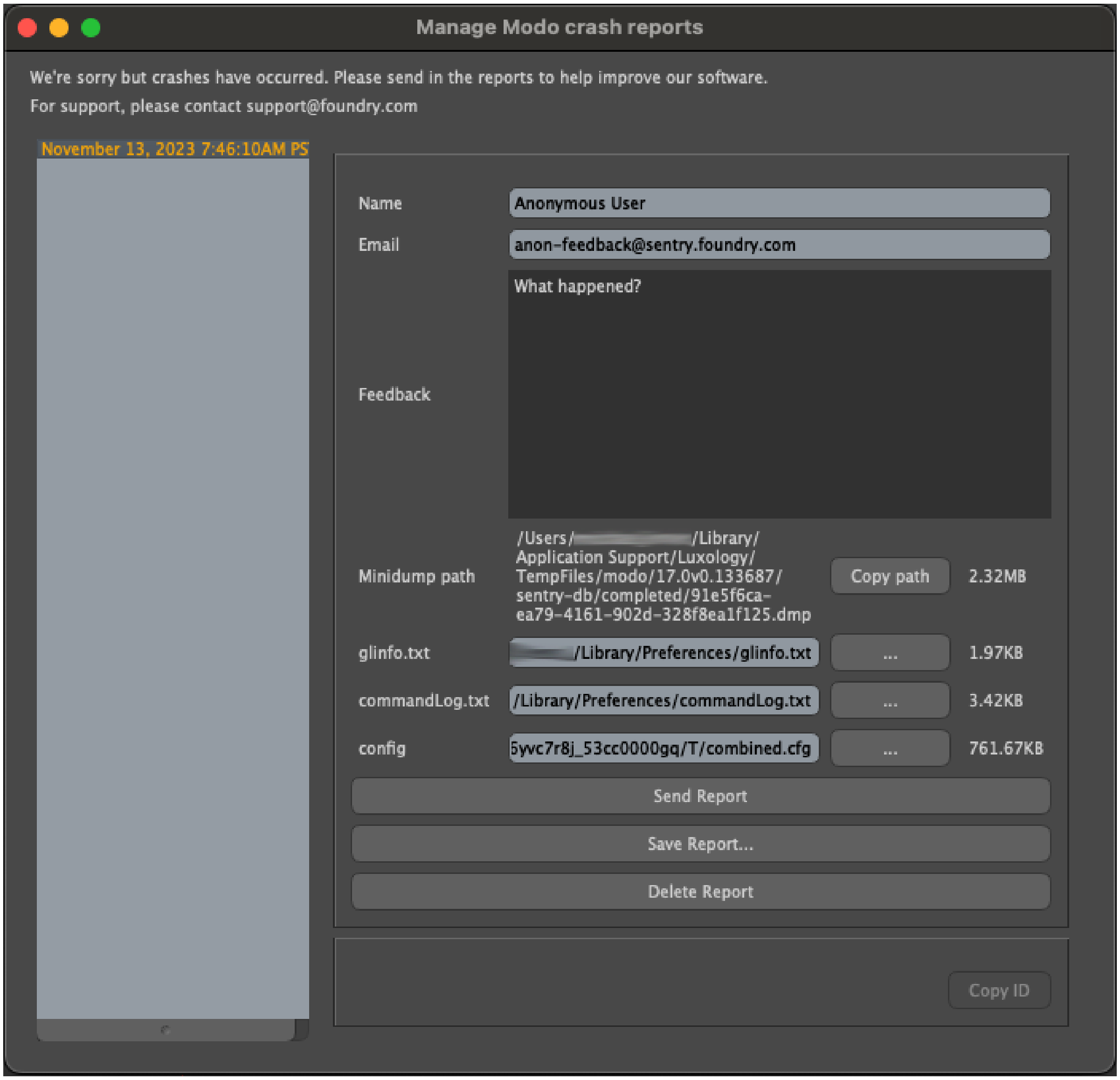
You can click the main Help menu to access the following:

- **Activate Help** - activates the context-sensitive help.
- **Learn Modo** - opens the Foundry Learn page featuring video tutorials and links to other resources.
- **Online Documentation...** - opens the Modo Online Help.

- **Documentation Download** - opens a page featuring links to online and offline help for all versions of Modo.
- **SDK Developers Wiki** - opens Modo’s SDK documentation designed for developers.
- **TD SDK Documentation** - opens Modo’s TD SDK documentation designed for developers interested in Modo's Python API.
- **Kit Help** - opens a submenu where you can find options for various Modo kits, such as installation and help options.
- **Kit Version** - opens a submenu featuring documentation for various Modo kits, including Fusion and ProRender.
- **What's New in Modo...** - opens the *What's New in Modo* topic of the documentation, containing the list of new features.
- **Foundry Homepage...** - opens the Foundry website.
- **Foundry Asset Portal...** - opens a source for sharing and downloading assets, scenes, and scripts.
- **Foundry User Community...** - opens the Foundry forums.
- **Foundry TV...** - access to training videos, tutorials, feature demos, and user videos.
- **Foundry Blog...** - opens the Foundry blog.
- **Foundry User Gallery...** - opens the Foundry User Gallery, where you can find images and videos created by other users and submit your own work.

Improved Crash Reporting

Modo has a crash reporting system which you can use to report any crashes to Foundry.



Manage Modo crash reports

We're sorry but crashes have occurred. Please send in the reports to help improve our software.
For support, please contact support@foundry.com

November 13, 2023 7:46:10AM PS

Name: Anonymous User
Email: anon-feedback@sentry.foundry.com
Feedback: What happened?
Minidump path: /Users/[redacted]/Library/Application Support/Luxology/TempFiles/modo/17.0v0.133687/sentry-db/completed/91e5f6ca-ea79-4161-902d-328f8ea1f125.dmp 2.32MB
glinfo.txt: [redacted]/Library/Preferences/glinfo.txt 1.97KB
commandLog.txt: /Library/Preferences/commandLog.txt 3.42KB
config: 5ywc7r8j_53cc0000gq/T/combined.cfg 761.67KB

Please enter your name and account **Email**, so that we can follow up with you for more information. Please include any information in the **Feedback** section, which might help our QA and Development teams to track down and fix your crash.

By default the reporter will submit your **minidump** (containing the crash information), as well as the **glinfo** (system GPU information), **commandLog** (list of commands run during your Modo session) and **config** (preferences and customizations) files. If you have information in any of those files that you do not want to submit, you can remove them by deleting the file and path info in the text field next to them.

After clicking **Send Report**, the report will be transmitted to Foundry and will return a **crash report ID**, which will be shown (and can be copied from) the bottom of the report dialog, before closing it.

You can alternatively **Save Report...**, which will store the report locally for manual submission, or you can **Delete Report**, if you do not want to submit it.

If you don’t submit, save or delete the report and close the **Crash Reporter** window, you can access the crash reports saved on your system at a later time using the **Help** > **Manage Crash Reports...** menu option.

Windows/Linux

The crash report dialog should appear immediately after a crash on these platforms. If unsent/unsaved crashes are present on your system, you can also access the dialog from the **Help** > **Manage Crash Reports...** menu.

macOS

Due to security limitations on macOS, we are unable to submit a crash report immediately after a Modo session has crashed. However, on the next launch of Modo the Manage Modo crash reports dialog should appear and the dialog is also accessible from **Help** > **Manage Crash Reports...**

Warning: The crash reporter for the OSX/Intel builds will not appear on macOS ARM hardware (limitation), but should on macOS Intel hardware running OSX/Intel builds and on macOS ARM hardware running ARM/Apple Silicon builds.

Contacting Customer Support

Should questions arise that the online help system fails to address, you can visit Foundry's Support Portal.

What Is Modo?

Modo is a 3D modeling and animation software that includes powerful and flexible sculpting, texturing, lighting, and rendering toolsets. Modo is used by artists around the world to create assets and shots for various industries including animation, VFX, games, architecture and design.

Courtesy of Madeira3D

Courtesy of Kion3D

Courtesy of Emilio J. Dominguez Calvo

Modo can be run on Windows, Mac, and Linux, and is designed as a tool for both artists and designers. Modo was developed with artists' workflows in mind to create a carefully considered interface and toolsets, allowing you to work intuitively and with more creative freedom.

Modo was created with the understanding that its users would come from a broad spectrum of industries and disciplines, meaning it would need to be flexible. Because of this, Modo can be easily implemented into existing pipelines, providing individual functionality, like modeling or texture painting, or it can be used for the entire pipeline itself, allowing you to design, model, animate, and render entirely in the same environment.

Courtesy of Tomas Sciskala

For users who employ Modo as a component in a multi-application pipeline, there are various file format options that allow Modo to inter-operate with most pipeline systems. For more information on these file formats, see Importing and Exporting File Formats.

The Modo user interface also allows you to remap controls, so that you can set viewport controls and hotkeys to be the same as another pipeline application. This means that artists can bounce between software without their workflow being impaired by having to keep tracking of different controls.

To get started in Modo, take a looking at the following documentation:

Understanding the Interface

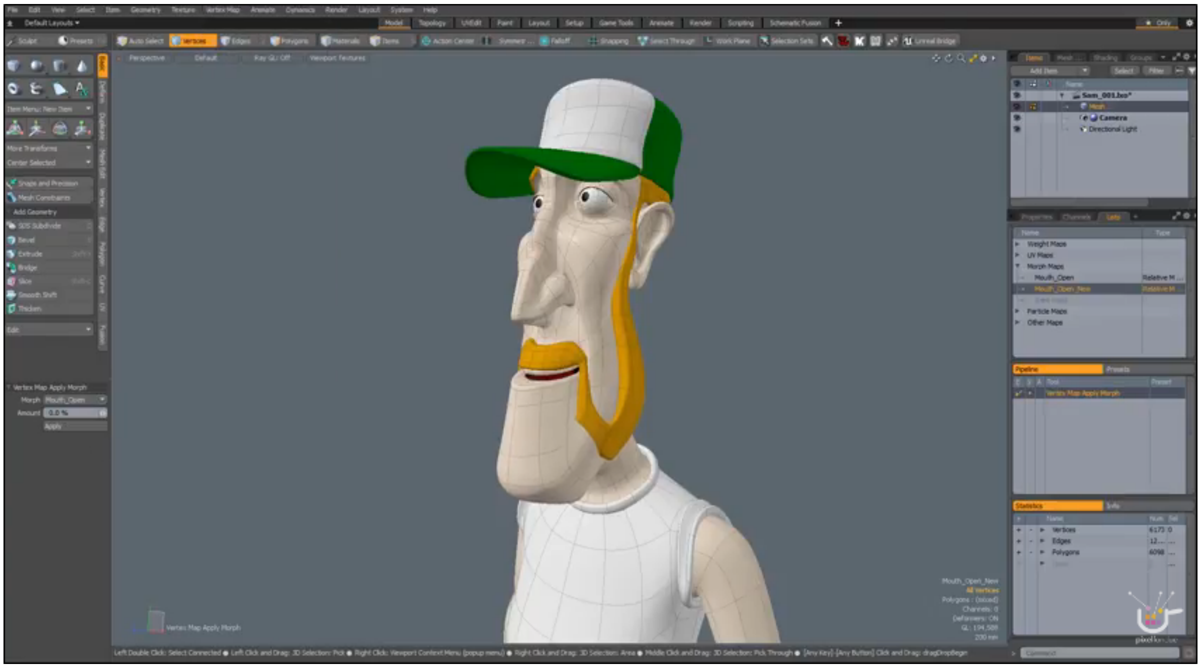
This is a great place to start if you are new to Modo. Learn how to navigate the interface, use common toolsets, and customize the layout to meet your needs.

Quick Start: Coffee Mug

A written tutorial and accompanying video to help you get started with Modo. Discover how to model, texture and render a mug, while learning some key tips and tricks along the way.

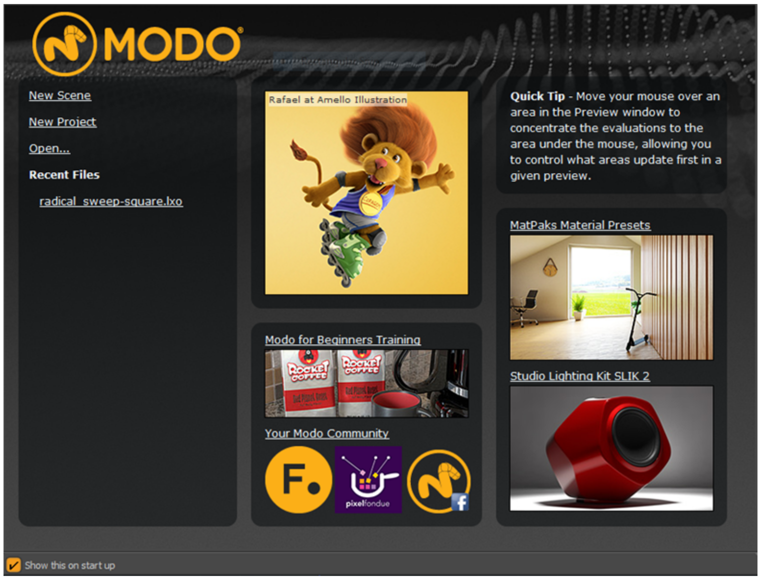
Understanding the Interface

To get the most out of Modo, it is helpful to first understand its unique interface designed for maximum flexibility and fluid navigation. Modo provides you with the most powerful tools, and this is exemplified by every viewport, panel, and menu. The User Interface Conventions and Using Standard Tool Controls topics provide an overview of interacting with Modo's interface using the mouse, and subsequent topics cover the various windows within the application.



Using the Modo Introduction Screen

To enhance your first time experience, you are presented the Modo Introduction Screen when you launch Modo for the first time.


MODO
New Scene
New Project
Open...
Recent Files
radical_sweep-square.lxo
Rafael at Amello Illustration
MatPaks Material Presets
Modo for Beginners Training
Your Modo Community
Studio Lighting Kit SLIK 2
Show this on start up

Quick Tip - Move your mouse over an area in the Preview window to concentrate the evaluations to the area under the mouse, allowing you to control what areas update first in a given preview.

The Modo Introduction Screen provides the ability to quickly create a new scene, create a new project, open an existing scene, and review recently opened files. It also provides the **Image of the Week** to get you inspired!

From here you can quickly access learning material on **Modo Foundations**, **Basics**, **Modo Tutorials**, **Quick Tips**, and download the **MatPaks Material Presets** and the **Studio Lighting Kit SLIK 2**.

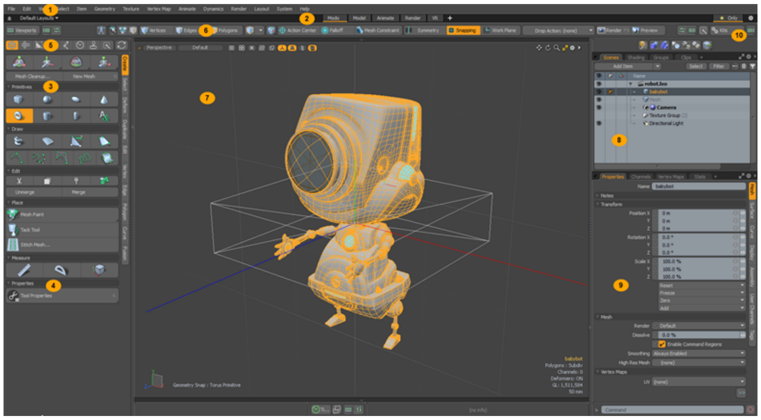
Tip: The Modo Introduction Screen is also accessible from the **Help** > **MODO Introduction Scene** menu.

Exploring the Layout

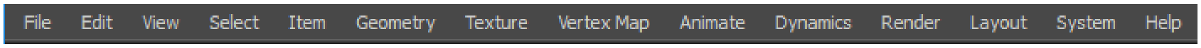
When you first open Modo after installing it, this is the interface you're presented with. It was designed and refined over multiple versions for the smoothest user experience. Users of any previous version of Modo are very familiar with this tabbed version of the interface, where each focused workspace is selected directly using the tabs across the viewport.

Each of the workspaces are set up as three main sections. The left-side column is for tools and commands, the center viewport is the most frequently used in Modo, and lastly, the right-hand side houses information regarding the scene: its many items and their respective attributes. Understanding all the various areas and their roles in working in Modo is essential to getting the most out of it.

The image below highlights the main interface sections. See the corresponding numbers below for details.

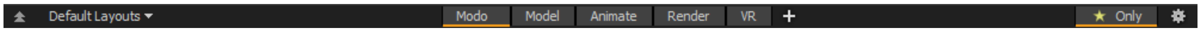

1
2
3
4
5
6
7
8
9
10

1 — Application Menu Bar


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help

Any function in Modo is available from the menu bar, organized by their respective application. Click any of the menu bar items to reveal a list of commands. Most commands in the menu bar can also be found elsewhere in the application as buttons, or are available through keyboard shortcuts, listed to the right of the menu item (for a list of Modo keyboard shortcuts, see Keyboard Shortcuts). This functional duplication enhances overall workflow.

2 — Switcher Bar

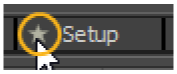

Default Layouts
Modo
Model
Animate
Render
VR
+
Only

Directly below the menu bar is the switcher bar. Each tab in the switcher bar represents a focused layout, optimized for their specific task. Click on a tab to open the layout. To minimize the switcher bar, click the two upward-facing arrows (⏫) on the far left, temporarily hiding it. To reopen it, click the remaining thin line.

You can add your own layouts using the plus button (⊞) located to the right of the last tab. When you create a new layout, an additional tab is added to the rightmost position of the bar.

You can reorder tabs by clicking and holding a tab and dragging it to a new location. The drop position is indicated by a small orange line.

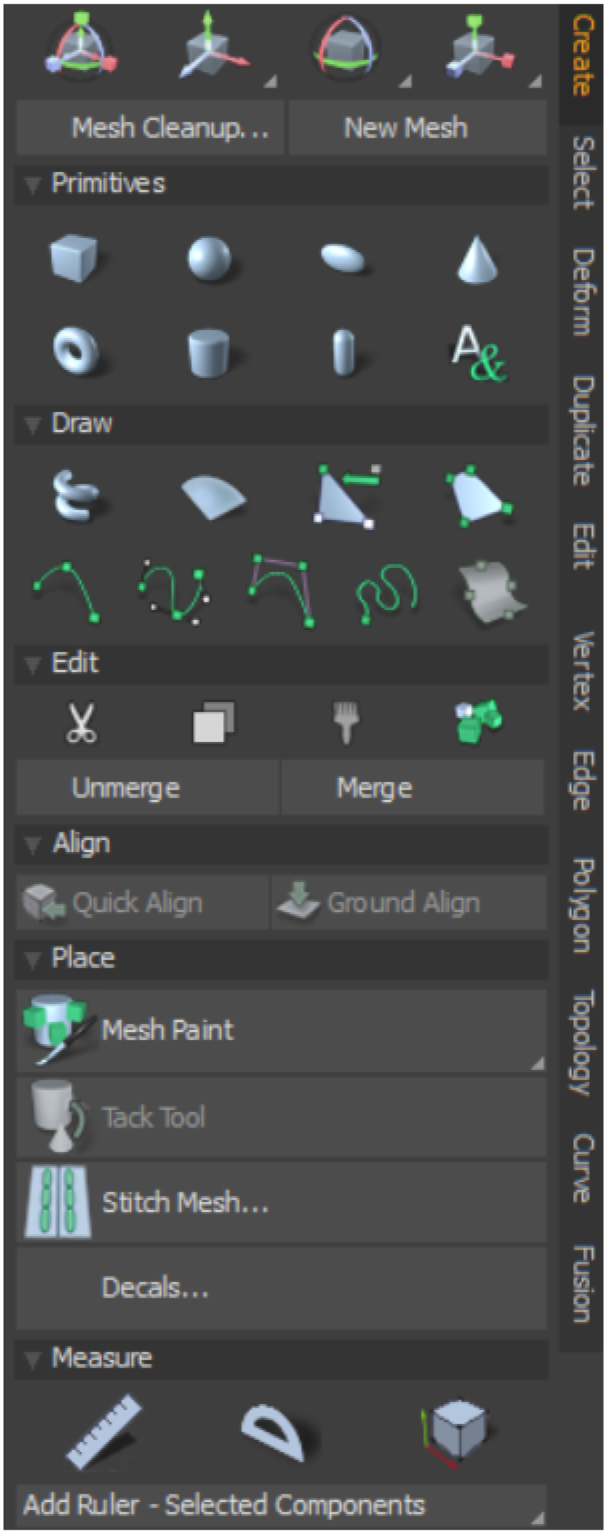
By the default, you can see the **Modo**, Model, Animate, Render, and VR layouts. To reveal additional layouts, click the Starred Only ★ Only button on the right. You can also click the star button in front of a layout name to highlight specific, often used tabs.



To hide unstarred tabs, click the Starred Only ★ Only button again.

You can reset a layout to its default state by double-clicking its tab. A dialog opens where you need to confirm that you want to reset the layout.

3 — Toolbox


Create
Select
Deform
Duplicate
Edit
Vertex
Edge
Polygon
Topology
Curve
Fusion
Mesh Cleanup...
New Mesh
Primitives
Draw
Edit
Unmerge
Merge
Align
Quick Align
Ground Align
Place
Mesh Paint
Tack Tool
Stitch Mesh...
Decals...
Measure
Add Ruler - Selected Components

The Toolbox is in the upper area of the left side panel. Nearly every layout has its own dedicated toolbox, illustrated here with the **Model** layout's Model Toolbox. The Toolbox itself houses the various tools and commands. On the right side of the Toolbox, there are a series of rotated tabs or sub-tabs that organize the various tools by their function. Click on any of them to open the related tools.

To activate a tool, click its button. The active tool is highlighted in orange. You can then manipulate the active tool in the 3D viewport. You can modify some tools using the **Ctrl**/**Cmd**, **Alt**, and **Shift** keys, and certain tools support additional functions for the middle and right mouse buttons. Depending on the tool's function, its properties appear in the section directly below, where you can modify and apply settings. You can drop (deactivate) a tool by pressing the **Space** key, or by selecting another tool. Buttons to quickly access alternate toolboxes are available just above the Toolbox.

Recently used tools are also available in the Tool HUD. You can display the HUD by pressing **Ctrl**/**Cmd**+**Tab**. For more details, see Tool HUD.

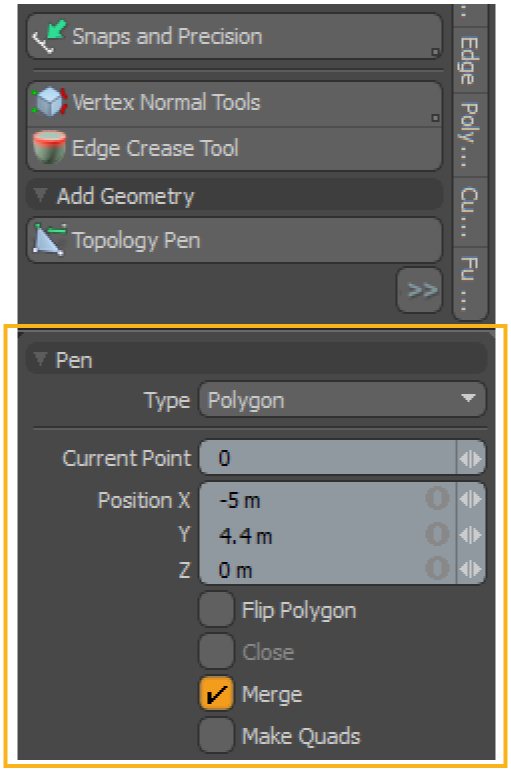
4 — Tool Properties

Depending on whether you are using Direct Modeling tools, Procedural Modeling , or Schematic Modeling tools, properties are displayed in different areas of the interface.

Direct modeling

For direct modeling tools, the properties dedicated to the currently-active tool are displayed on the left panel, directly below the Toolbox.

Note: In the Modo layout, a **Tool Properties** button is displayed. Click it to reveal the properties.



Snaps and Precision
Vertex Normal Tools
Edge Crease Tool
Add Geometry
Topology Pen
>>
Edge
Poly...
Cu...
Fu...
Pen

Type Polygon

Current Point 0

Position X -5 m

Y 4.4 m

Z 0 m

☐ Flip Polygon

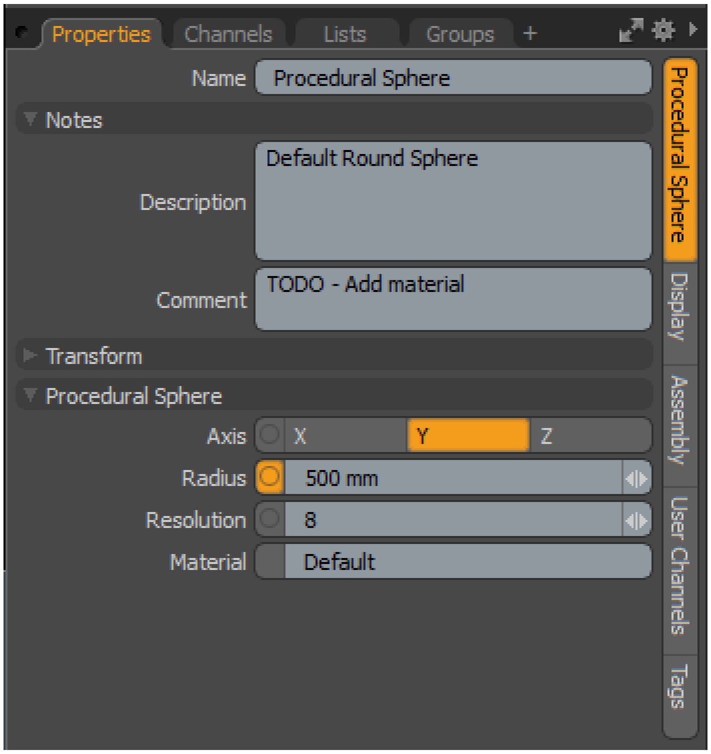
☐ Close

☑ Merge

☐ Make Quads

Procedural and Schematic Modeling

For Procedural and Schematic modeling tools, the properties dedicated to the currently-active tool are displayed on the right panel.


Properties
Channels
Lists
Groups
+
Procedural Sphere
Display
Assembly
User Channels
Tags

Name: Procedural Sphere

Notes

Description: Default Round Sphere

Comment: TODO - Add material

Transform

Procedural Sphere

Axis: X Y Z

Radius: 500 mm

Resolution: 8

Material: Default

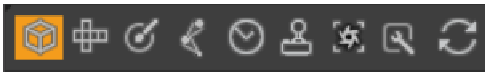
Tool properties only appear when a tool is activate, and they change depending on which tool is selected. For most tools, you can click and drag in the viewport, and see the properties update in real time. Conversely, you can directly edit the properties by clicking over a value and typing in a new value. Press **Enter** to see the update in the viewport.

Tip: You can press the **Tab** key to jump between value input fields.

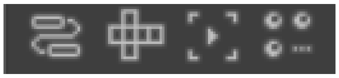
Once you are satisfied with your settings, drop the tool by pressing the **Space** key.

For information about item properties, see Items (Scenes) List.

5 — Toolbars and Palettes

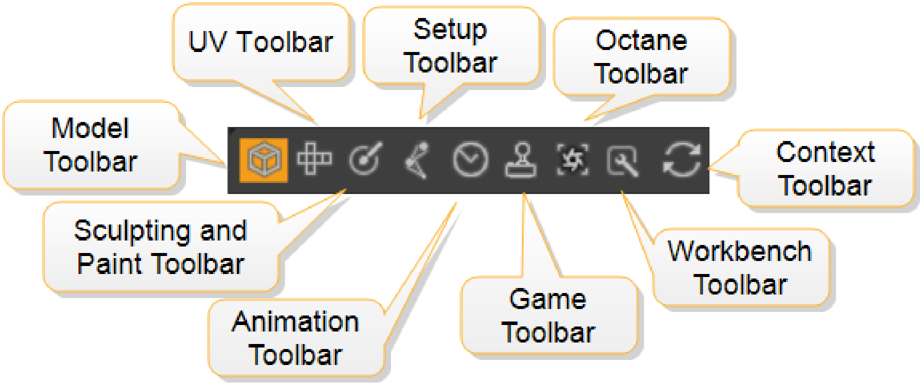


Toolbars

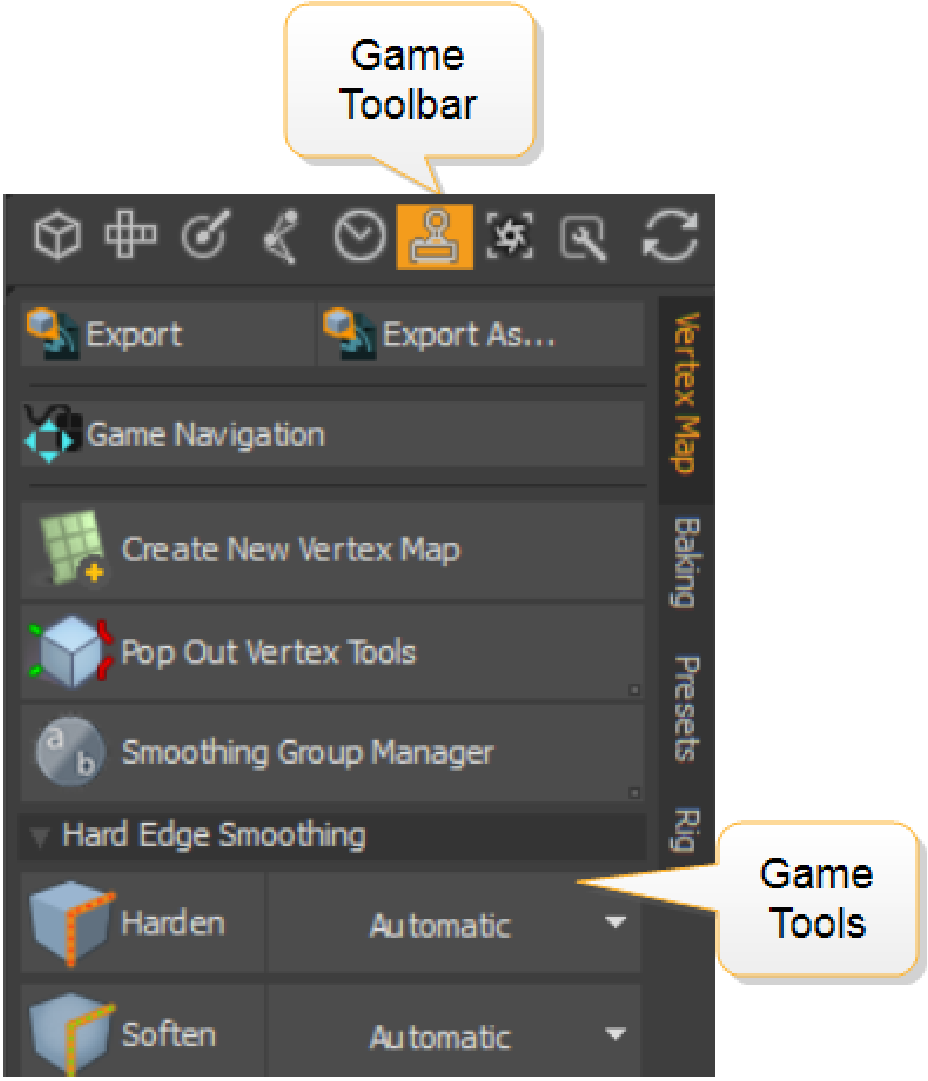


Palettes

Accessing Tools


UV Toolbar
Setup Toolbar
Octane Toolbar
Model Toolbar
Context Toolbar
Sculpting and Paint Toolbar
Workbench Toolbar
Game Toolbar
Animation Toolbar

To gain faster access to a wide variety of Modo tools, click any of the toolbar buttons in the top-left corner of the interface. Once activated, the related tools are displayed in the Toolbox on the left panel. For example, click on the Game Toolbar and the related tools appear in the Toolbox.

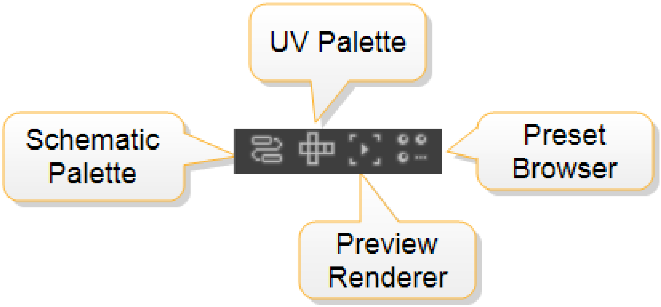

Game Toolbar
Export
Export As...
Game Navigation
Create New Vertex Map
Pop Out Vertex Tools
Smoothing Group Manager
Hard Edge Smoothing
Harden
Automatic
Soften
Automatic
Vertex Map
Baking
Presets
Rig
Game Tools

In addition, when you select a different layout, the associated toolbar is displayed automatically. For example, open the **UV** layout and the **UV** toolbar is displayed on the left panel. You can click on any toolbar button to change the tools displayed in the left panel while working in the selected layout.

Toolbar Shortcuts:

- Hold **Alt**+click a toolbar button to open as a popover. Moving the popover pins it open.
- Hold **Ctrl**/**Cmd**+click to open it as a palette.

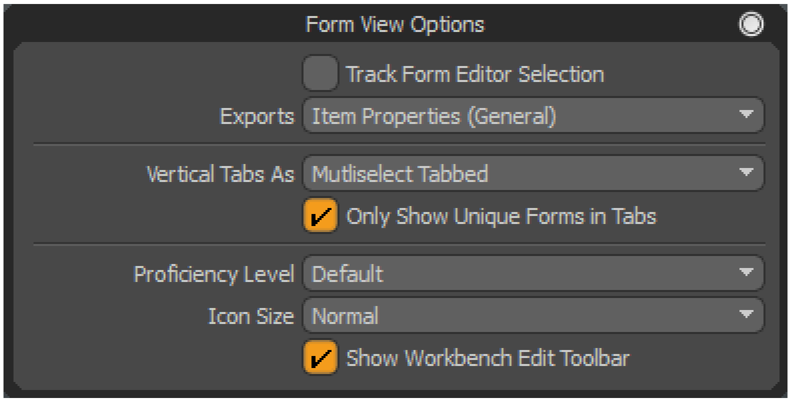
Accessing Common Palettes



To open a separate window for the Schematic palette, UV palette, Preview renderer, or Preset Browser while working in a layout, click on the palette's button.

Form View Options

Pressing the **O** key in a form opens the **Form View Options**, where you can customize the appearance and content of the form.



Form View Options

☐ Track Form Editor Selection

Exports: Item Properties (General)

Vertical Tabs As: Mutliselect Tabbed

☑ Only Show Unique Forms in Tabs

Proficiency Level: Default

Icon Size: Normal

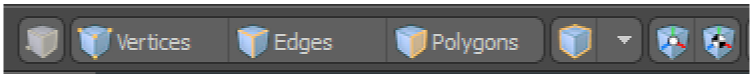
☑ Show Workbench Edit Toolbar

>

Track Form Editor Selection	When enabled, the form view displays what is currently selected in the Form Editor.
Exports	When a form's **Exported** option is enabled in the Form Editor, the form can be selected in a form view. This is also indicated by the X column in the tree. This dropdown lists the forms with **Exported** enabled. Selecting one displays it in the form view.

Vertical Tabs As	Select how vertical tabs are displayed. The following options are available: • **Tabbed** - Tabs are on the right, with one tab selectable at a time. • **Multiselect Tabbed** - Tabs are on the right, and you can select multiple tabs at the same time by **Ctrl**+clicking them. • **Merged Tabbed** - All tabs are merged into a single scrolling form. The tabs remain visible on the right and can be used to jump to different parts of the form. Tabs that are visible are drawn as selected, and change as you scroll the form. • **Merged** - All tabs are merged into a single scrolling from and the tab bar is hidden.
Only Show Unique Forms in Tabs	When enabled (the default), this hides subsequent references to the same sub-form when in a **Merged** or **Multiselect** mode. This helps to avoid redundant controls, such as the **Name** field at the top of many Item **Properties** forms.
Proficiency Level	This option is used to change the level of detail displayed in the form. • **Default** - Uses the **Base Form Proficiency Level** preference in the form. For more information, see Defaults Preferences. • **Basic** - A short list of the common core properties. • **Standard** - A list of all of the core properties. • **Advanced** - A list of all of the core properties plus custom functions for special behaviors.
Icon Size	Override's the form's default icon size with a custom size. Normal uses the icon size set in the form itself in the **Form Editor**. The following other options are available: • **Extra small Icons** • **Small Icons** • **Large Icons**
Show Workbench Edit Toolbar	Enabled by default, when there are any Workbenches in the form, this shows/hides their Edit toolbar.

6 — Modo Modes



The **Modes** section can be found under the menu bar and above the 3D viewport. The first four buttons from the left are the Component editing modes, an important distinction in Modo. These modes set how you select and work directly with geometry for manipulation: pushing, pulling, and modifying polygons. You can toggle between any of these modes by pressing the **Space** key, continuously looping through them, or by pressing the numbers **1**, **2**, or **3** on the keyboard.

To enable **Items** mode for editing items, deselect the component editing modes. To understand the difference between Component modes and Item mode, think of each item layer as a container: the Component modes edit what's inside the container, and Item mode edits the container itself. A separate Item mode is necessary for some functions in Modo. For example, when animating, you are transforming items (the containers), and individual vertices are not directly animateable. Also, Replicators and Instances work by duplicating the container.

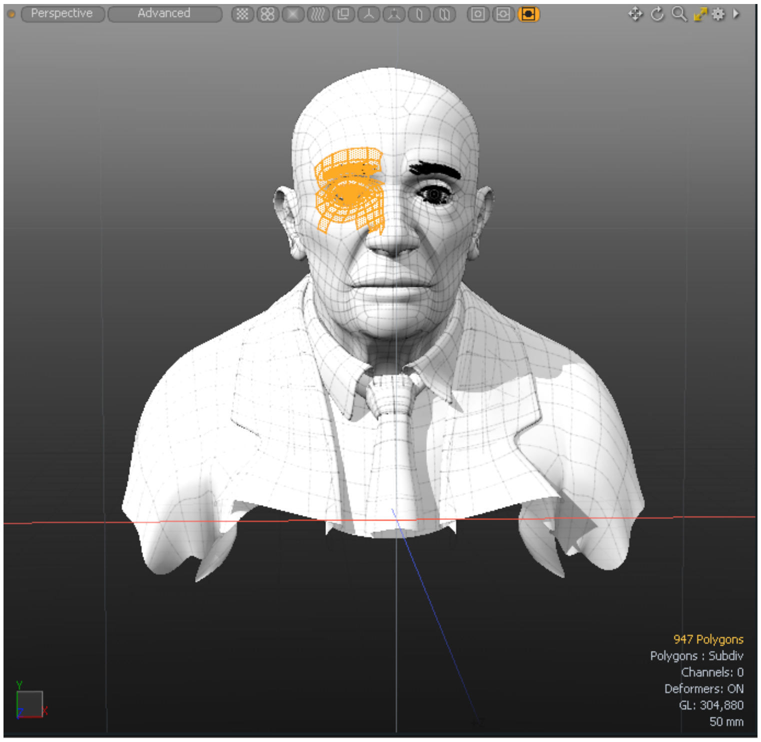
You can set an item's **Center Selection** and **Pivot Selection** points. These are the defined surfaces within the Shader Tree, which is the heart of Modo's shading system.

If you click the Set Polygon Tags button , a pop-up reveals additional material selection modes. For more information, see Polygon Set Material/Part/Smoothing.

The rest of the buttons are modifiers for tools, changing their basic behaviors:

- **Action Centers** - Automatically adjusts a tool's axis and orientation. For more information, see Specifying Action Centers and Falloffs .
- **Symmetry** - Duplicates manipulations across an axis. For more information, see Symmetry Tool.
- **Falloff** - Controls the effect a tool has on a selection based on an interactively applied distance. For more information, see Using Falloffs.
- **Snapping** - Eases certain workflows that require levels of precision. For more information, see Applying Snapping.
- **Work Plane** - An interactive and modifiable virtual workbench that simplifies many modeling tasks among other things. For more information, see Using the Work Plane.

7 — 3D Viewport



The 3D viewport is the most frequently used interface in Modo. It contains a dark grid going off into the distance. This is the ground plane, denoting the zero position of the Y axis. As items move up, they increase their Y position value.

Note: The Y axis is also referred to as the Up axis.

The X values move from left to right, and the Z axis (depth) moves from front to back. (If you are used to an alternate Up axis, you can modify the setting in **System** > **Preferences**) The darkest lines represent zero for the X and Z planes, the origin is where all three axes intersect, '0,0,0' on all three axes. Item level transforms all take place from this position.

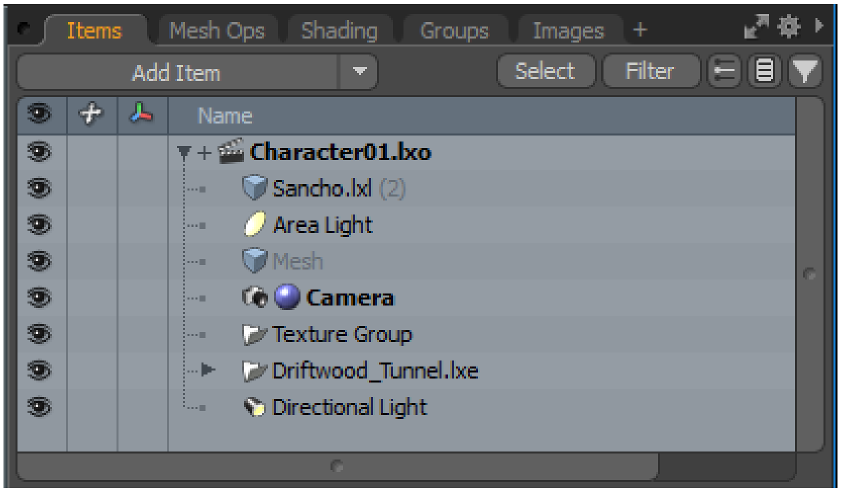
The lighter grid is the **Work Plane**, the virtual desktop within the viewport. It interactively changes position based on the current viewpoint's position. For example, if you select the **Cube** tool and click and drag over the viewport, the first plane of the cube originates on the work plane itself. Drop the tool and rotate your viewpoint until the Work Plane changes position and then activate the **Cube** tool again and draw another square. This time it originates from the new **Work Plane** position. Based on your viewpoint, the **Work Plane** always tries to face the user, and appears within the viewport as a reference to where new items are created.

If you're used to working in a four-viewport system where you can easily place items in 3D space by moving them around in each viewport, the **Work Plane** allows you to work in a single large perspective viewport, and easily position items within the scene without surprises as to where items are created. There are several functions for modifying the Work Plane's position and how it adapts to the viewport, covered more fully in the Work Plane section.

You can navigate in the viewport using the three icons in the upper right-hand corner, or keyboard shortcuts.

- To move the viewport position, drag over the Four Arrows icon, or hold **Shift**+**Alt** while clicking and dragging in the viewport. Middle-click to move the view forwards and backwards, and right-click to restrict the view change to up and down.
- To rotate the view, drag on the Rotation icon, or hold **Alt** and click and drag over the viewport. Middle-click and dragging restricts the rotation around the Y axis, and right-click and dragging flicks the view: it slowly stops when the button is released instead of directly stopping.
- To zoom in, drag on the Magnifying Glass icon, scroll the mouse wheel, or hold **Ctrl**/**Cmd**+**Alt** and click and drag in the viewport. Right-clicking and dragging defines a rectangular area that, when the button is released, Modo instantly zooms to.

8 — Items List, Mesh Ops, Shading, Groups, and Images



The top right area of the interface lists all the items that make up a Modo scene. These lists are important to the overall Modo workflow, so they appear in nearly all the various workspaces with some variation. The tabs across the top of this section filter which items are listed, grouped by how Modo sees them.

The **Items** list shows all the items that make up the scene, such as geometry layers, cameras, and lights. Within the list, items can be grouped together by parenting for organization or for animation Hierarchy. Selecting the various items displays their properties in the **Properties** viewport along with numerous other functions. For more information, see Items (Scenes) List.

The **Shader Tree** organizes all the surfaces within the scene and their respective items, such as materials, shaders, textures, and groups. When rendering a scene, it is the **Shader Tree** that controls how the final rendered image appears. For more information on these workflows, see Shading and Lighting and Rendering.

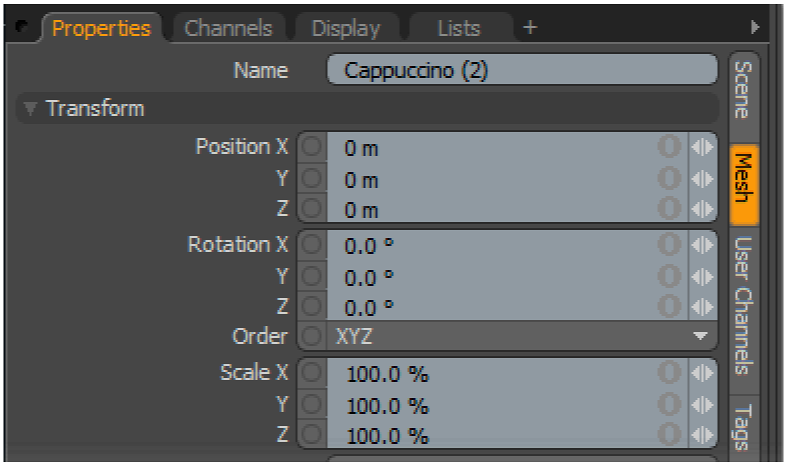
The Groups Viewport is useful for creating groupings of items in a scene, making it easier to select and manipulate multiple items, as well as for controlling lighting (light linking), making multi-item Replicators and for working with animated scenes when setting keyframes.

The **Images** tab lists all the external images currently loaded into Modo and offers functionality for working with them.

Note: Images aren't saved in the Modo file itself, so they need to be available each time a scene is loaded.

Lastly, the **Quick Tips** viewport offers some shortcuts to videos and such for quickly getting up to speed with Modo.

9 — Properties, Channels, Display, and Lists



Name Cappuccino (2)

Transform

Position X 0 m

Y 0 m

Z 0 m

Rotation X 0.0 °

Y 0.0 °

Z 0.0 °

Order XYZ

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

Directly below the **Items** list and the **Shader Tree** is another panel of lists. As scenes fill up with items, textures, UV maps, and so on, a serious amount of information is created along with it, and Modo offers many ways of viewing and manipulating all that info. In this group of lists, **Properties** is the most important. It is useful to keep this open most of the time.

The first tab across the top is the item **Properties** tab. Every item in Modo has a series of attributes associated with it, and this is the viewport where you can edit them. Selecting any item displays the item's properties, dynamically changing as different types of items are selected.

The next tab is **Channels**. Every animateable attribute of an item is listed here for easy view, selection and manipulation. For more information on the **Channels** tab, see Channels Viewport.

The **Display** tab controls the visibility and display of items in the 3D viewport. For more information on this tab, see Display Viewport.

The next tab, **Lists**, consists of three parts. The top section lists all the associated Vertex Maps. A Vertex Map is an instruction associated with a vertex telling it to do a certain thing, such as move a specific offset distance (morph map), specify a 2D position on an image (UV) or simply a value that is fed into another function such as Weight Maps do. To see the Vertex Map associated with an item, the item must be selected (highlighted in the item list). In this window, you can create, rename and delete any of the various maps.

Under the **Lists** tab is the **Pipeline** viewport or Tool Pipe. This is the heart of the Modo toolbox. A tool can be made up from a single function like the Cube tool, or it can be a combination of functions rolled together, such as a transform tool with a certain falloff and action center, like the Flex tool. The **Tool Pipe** is where you can make custom tool combinations. **Presets** contains the various pre-made tool combinations included in the various tool boxes that install with Modo.

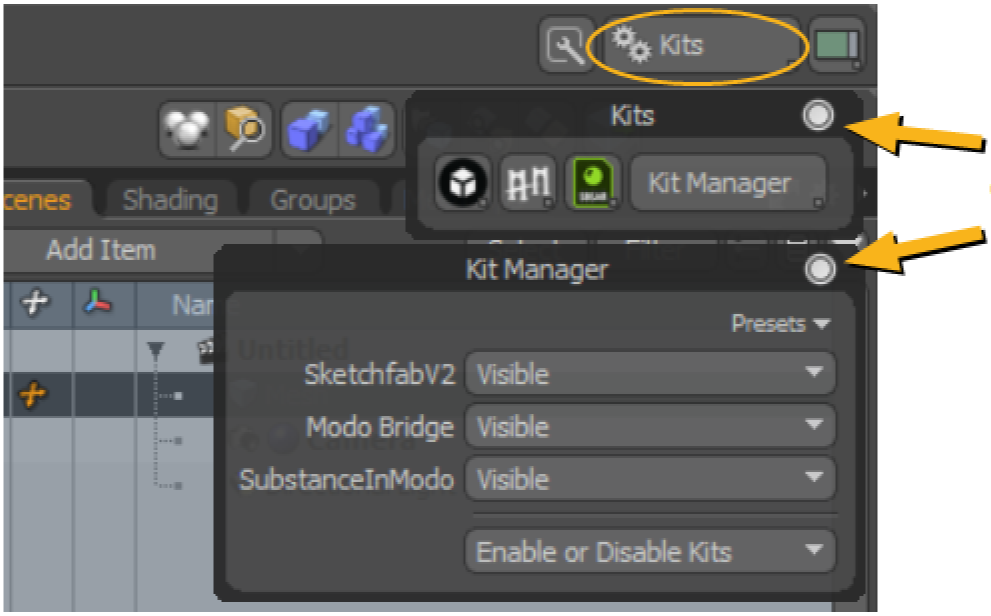
Below the Pipeline viewport, you can find **Statistics**. This is a listing of scene statistics: the total number of polygons, how many of them are triangles or have more than 5 sides. You can use the +/- icons preceding each line for selecting and deselecting geometry in the scene based on the specific criteria.

Lastly, the **Info** tab gives you all the information associated with a selection. Selecting an item provides information on that item. Selecting some geometry provides information on all the associated vertices and map values as well as applied surfaces and selection sets associated with it. For more information on the contents of these tabs, see Info and Statistics.

10 - Kits Menu

The **Kits** menu is used to manage the visibility of kit icons and buttons displayed in the toolbar.

Tip: Modo Kits can be installed by downloading them from the Preset Browser's **Cloud Assets** > **AddOns** > **Foundry** > **Kits** directory. For more information, see Preset Browser.


Kits
Kits
Kit Manager
Scenes
Shading
Groups
Add Item
Kit Manager
Presets
Enable or Disable Kits

SketchfabV2 Visible
Modo Bridge Visible
SubstanceInModo Visible

Click to Pin

To hide or view kit icons and buttons:

- Click the dropdown menu beside one of the kits listed and select one of the following options:
  - **Visible** - Shows the kit button on the toolbar.
  - **Hidden** - Hides the kit button from the toolbar.

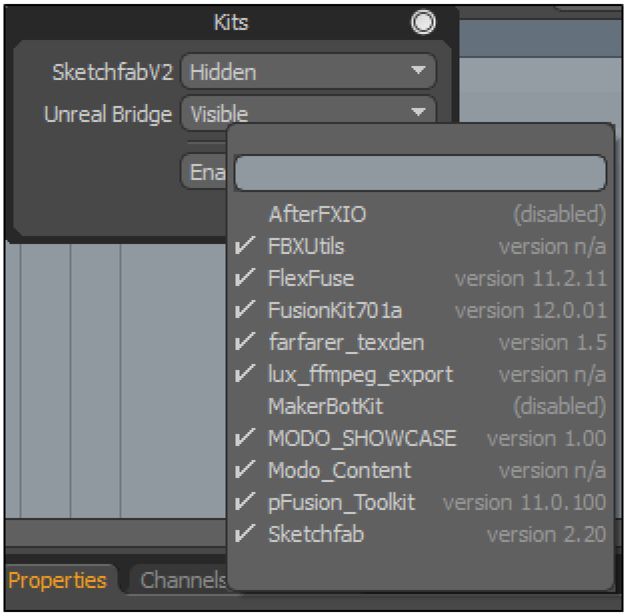
To enable or disable kits:

1. At the bottom of the **Kits** popover, click **Enable or Disable Kits**.

Note: Modo Kits are not removed from your system.

2. From the dropdown menu, click on the appropriate kit to enable or disable.

   A check mark is displayed for the kits currently enabled. A **(disabled)** tag is displayed for the kits disabled.

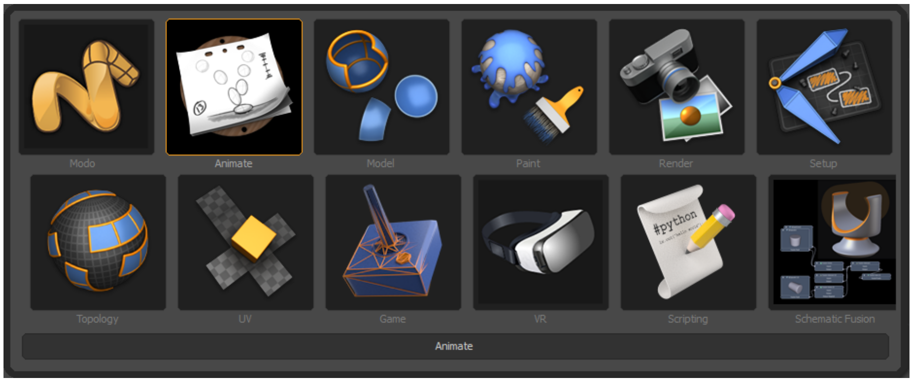


SketchfabV2 Hidden
Unreal Bridge Visible

	AfterFXIO	(disabled)
✓	FBXUtils	version n/a
✓	FlexFuse	version 11.2.11
✓	FusionKit701a	version 12.0.01
✓	farfarer_texden	version 1.5
✓	lux_ffmpeg_export	version n/a
	MakerBotKit	(disabled)
✓	MODO_SHOWCASE	version 1.00
✓	Modo_Content	version n/a
✓	pFusion_Toolkit	version 11.0.100
✓	Sketchfab	version 2.20

Note: You must restart Modo after enabling or disabling any kit. Once done, the Kits menu is updated.

Layout Switching

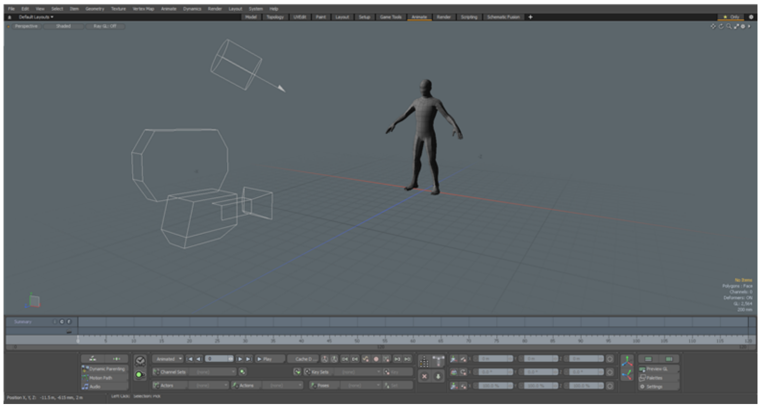


Modo employs an interface switching option using **Shift**+**Ctrl**+**Tab** to quickly switch workspace layouts. Press **Shift**+**Ctrl** and then **Tab** to open the selection pop-up. Continue to hold **Ctrl** and then release and press **Tab** again to cycle to the next workspace highlighted by the orange outline. Subsequent **Tab** presses continue to cycle through the selections organized by the layout's most recent use. Releasing the keys opens the currently selected interface. This method eliminates the space used by the tabs in the interface.

You can also open the Layout Switcher from the lower toolbar, by clicking **Layouts**. This option is also available from the menu bar, in **Layout** > **Recent Layouts...**. Specific layouts are accessible from the menu bar under **Layout** > **Layouts**. Each of the various workspaces is covered in more details below. Additionally, you can access the pie menu version of the Layout Switcher by pressing **Ctrl**+**~** (tilde).

Layouts

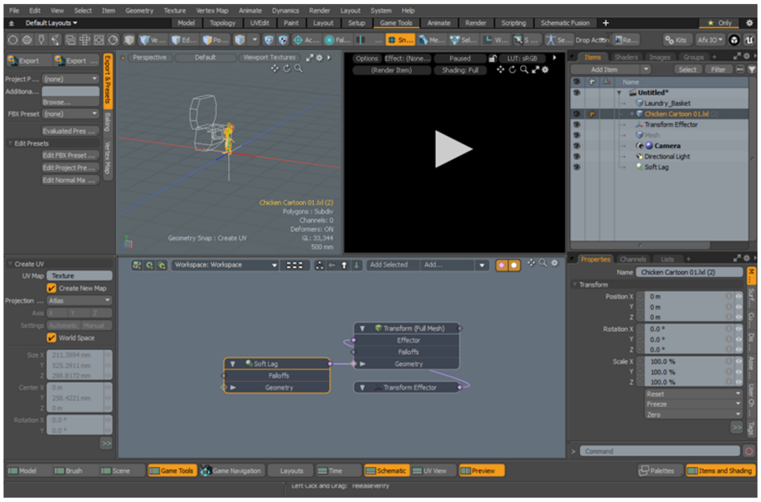
Animate



The **Animate** tab focuses specifically on animation, especially refined for character work. This viewport is unique in that the standardized layout is gone, in order to maximize the 3D viewport. Toolboxes have been reduced to pop-up palettes with the essential keying tools laid out as a special toolbox below the **Timeline**. Hidden just above the **Timeline** is the Track View that allows you to easily re-time animated elements. To open it, drag the top frame edge of the **Timeline** upward revealing the hidden viewport. **Timeline** playback controls run across the bottom of the interface with additional controls for setting keyframes and controlling the auto keying feature with additional easy access to the Channel Sets and Actor/Pose functions. For more information on the Animate viewport, see Animation Layout and Viewports.

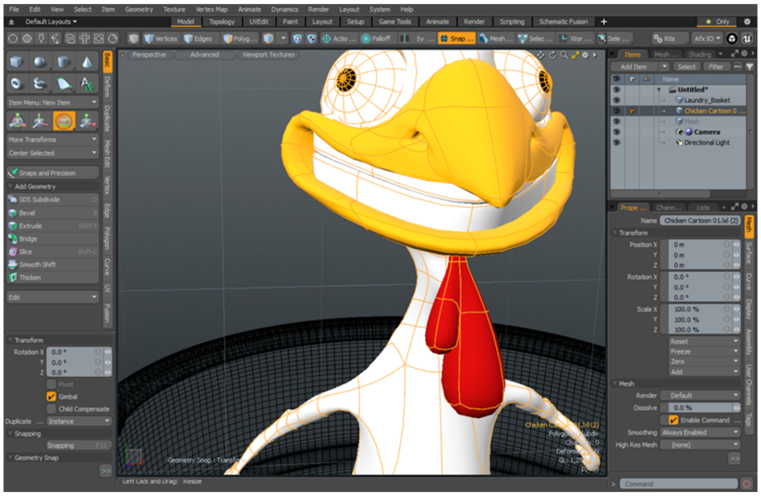
Tip: If you are used to animating in previous versions of Modo, you can use the **Layout** and **Setup** interfaces with the more familiar controls.

Game



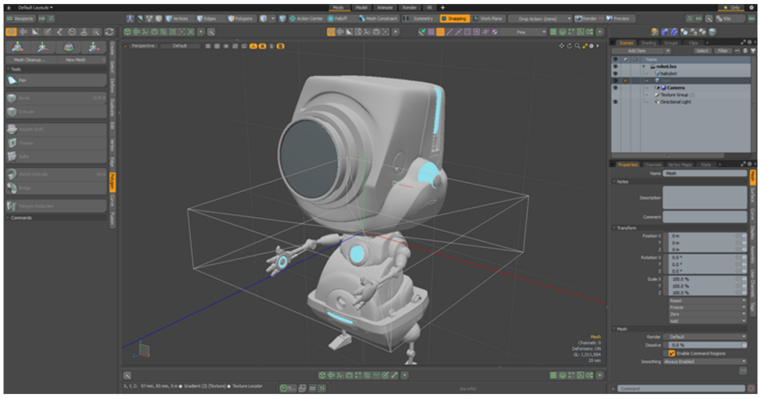
The **Game** layout provides you with a flexible structure, which makes it suitable to be used as your primary layout. The center of the layout is divided into a 3D viewport, a preview viewport, and a schematic viewport. The panels on the left and right are easily collapsible, and the toolbar on the bottom gives you quick access to a wide range of tools. For more information on this layout, see .

Model



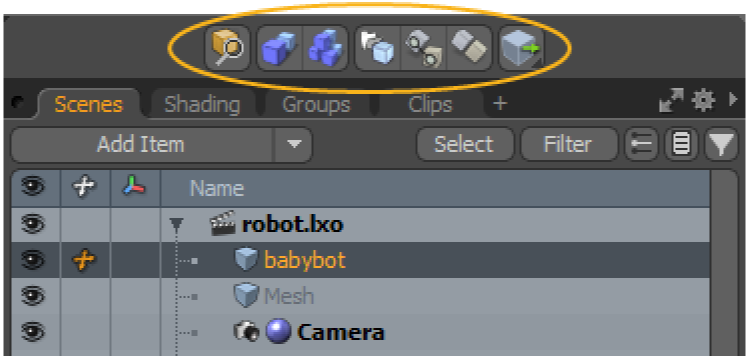
The **Model** interface is the dedicated model creation space, with all the creation and deforming-specific tools close at hand in the Toolbox on the left side. Below the Toolbox is the standard tool **Properties** panel. The middle section is the maximized 3D (OpenGL) viewport, where you can interact with the tool handles and selected geometry. The right side panel contains all the lists viewports: the **Item List** for selecting item layers in a scene, the **Lists** viewport with access to Vertex Maps, the **Pipeline** for customizing tools, and finally the **Statistics** viewport that allows statistical selections based on component features.

Modo

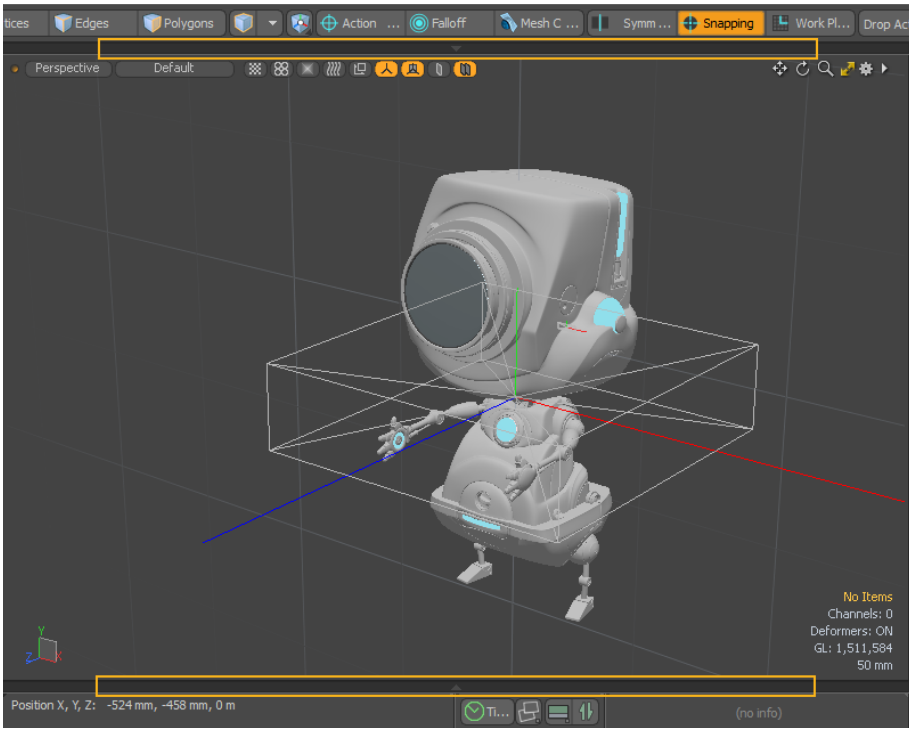


The **Modo** layout is an all-purpose interface that allows you to do virtually any task Modo is capable of. In the Toolbox on the left, you can find all tools arranged into sub-tabs. In addition to the tabs familiar from the Model layout, there is a **Select** sub-tab containing all selection tools.

The buttons above the right panel are context-sensitive, and change depending on what tab is active. For example, opening the **Scenes** tab displays controls for duplicating, instancing, parenting, and grouping items.

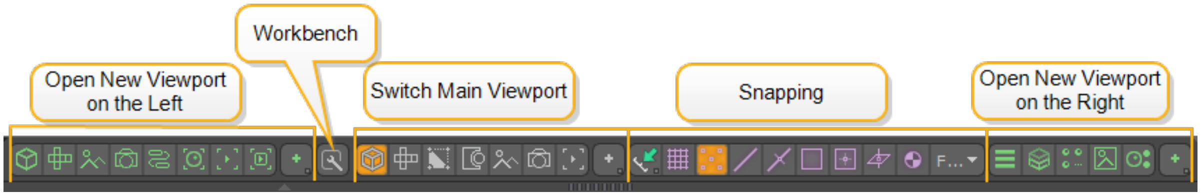


This layout has a collapsible area at the top and the bottom. To open them, click the thin line at the top or the bottom of the 3D viewport.

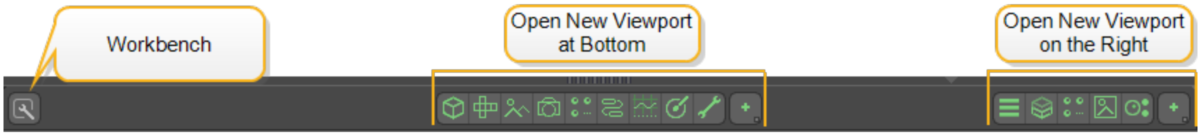

tices
Edges
Polygons
Action
Falloff
Mesh C ...
Symm ...
Snapping
Work Pl...
Drop Ac
Perspective
Default
No Items
Channels: 0
Deformers: ON
GL: 1,511,584
50 mm
Position X, Y, Z: -524 mm, -458 mm, 0 m
Ti...
(no info)

When expanded, additional controls appear that allow you to open more viewports. These areas are also equipped with workbenches, so you can add additional tool controls here. For more information, see Form Workbenches.

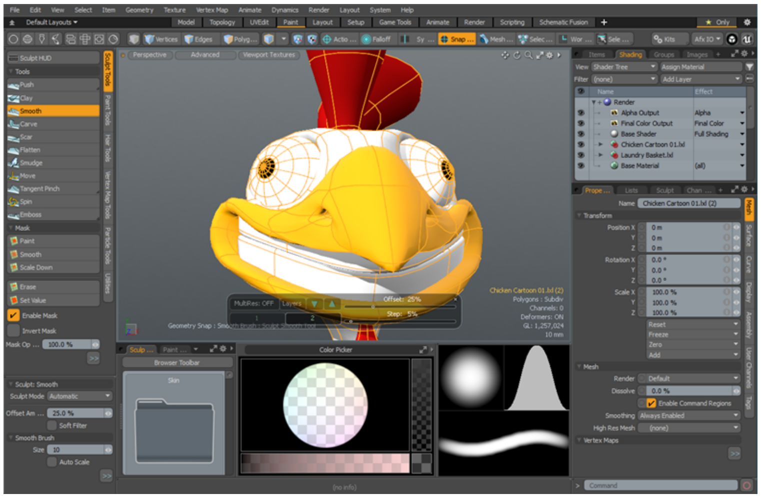
Top area:


Workbench
Open New Viewport on the Left
Switch Main Viewport
Snapping
Open New Viewport on the Right
F...

Bottom area:

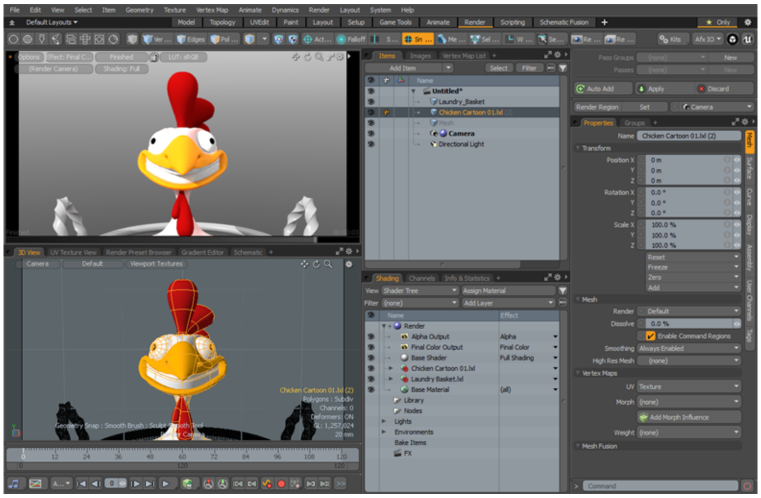

Workbench
Open New Viewport at Bottom
Open New Viewport on the Right

Paint



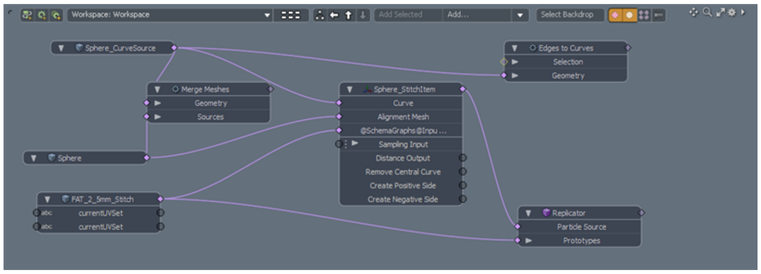
The **Paint** interface tab focuses on painting and sculpting. The left **Toolbox** is separated into the various tools and controls for painting textures and sculpting meshes. Directly below the 3D viewport is located a **Preset Browser** for viewing brushes and preset tool options. You can also directly access a **Color Picker** viewport and see the details of the current brush tip in the **Tip View** window.

Render



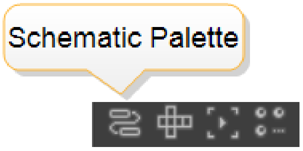
The **Render** layout provides streamlined access to the main options you need when focusing on the task of rendering a project. On the right-hand side is the **Item List** and **Shader Tree** panels, stacked with a tall **Render Properties** panel, so you can easily access and edit the various material and texture items and settings. The main 3D workspace is split into two views; a Preview Viewport that provides an extremely fast near final preview of the rendered scene, and a **Camera View** providing direct control over the positioning of the camera. To render in this view, press **F9** or select the **Render** command from the **Render** options of the menu bar.

Schematic

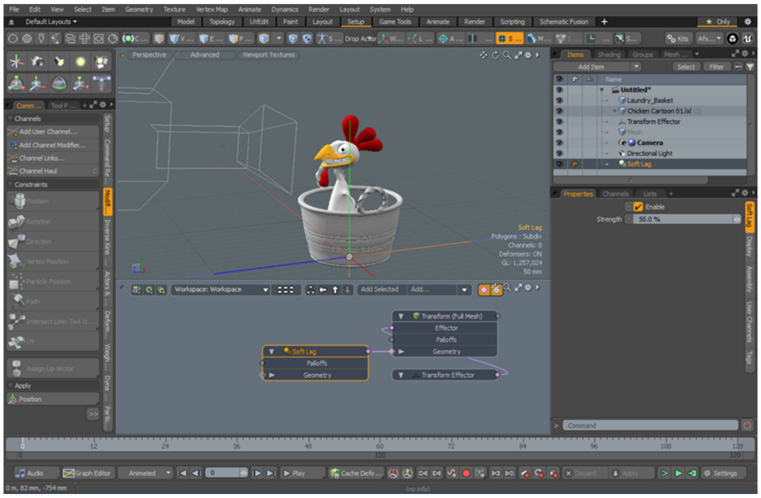

Workspace: Workspace
Add Selected
Add...
Select Backdrop
Sphere_CurveSource
Edges to Curves
Selection
Geometry
Merge Meshes
Geometry
Sources
Sphere_StitchItem
Curve
Alignment Mesh
@SchemaGraphs@Inpu ...
Sampling Input
Distance Output
Remove Central Curve
Create Positive Side
Create Negative Side
Sphere
FAT_2_5mm_Stitch
currentUVSet
currentUVSet
Replicator
Particle Source
Prototypes

The **Schematic** layout allows you to set up symbolic relationships between items and their channels. It is a 2D environment where elements are visually linked, providing a more intuitive and powerful way to define relationships between elements; building networks of operations that are combined to produce a specific result.

Tip: You can also open the **Schematic** layout as a separate viewport by clicking the **Schematic** palette icon on the top-left corner of the interface.

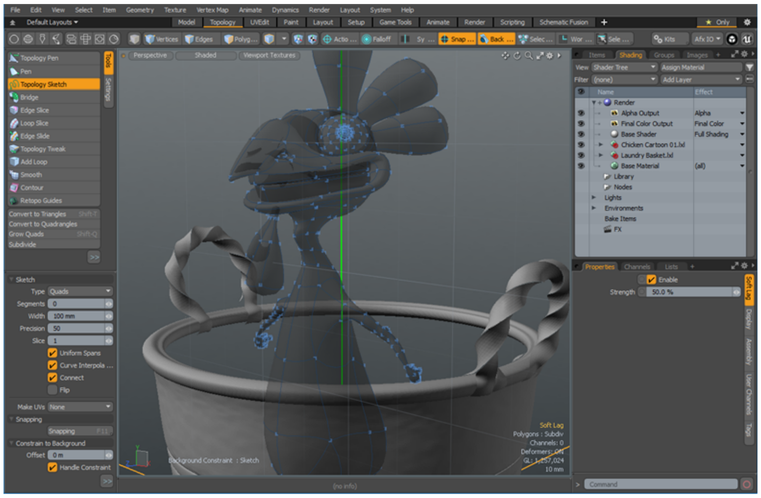

Schematic Palette

Setup



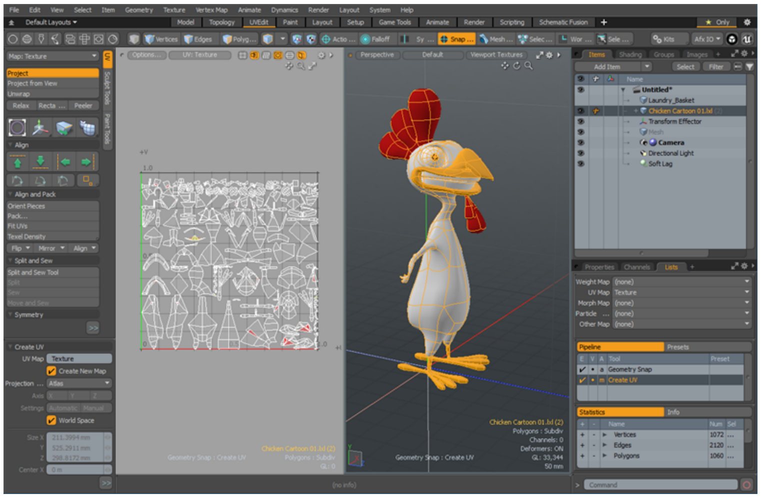
The **Setup** layout is mainly designed for rigging using the Schematic Viewport. You can select items in the **Items** list and **Shader Tree**, grab their associated channels in the **Channel** list and drag them into the **Schematic View** for node linking. The **Toolbox** also offers commands related to channel links and modifiers. Additionally, this is the place for setting up dynamics and particle simulations.

Topology



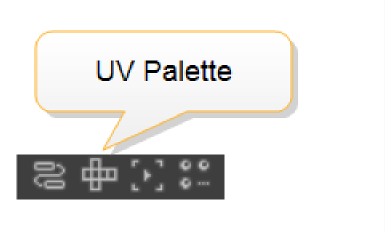
The **Topology** interface tab contains all the retopology related tools. Retopology is the act of taking a very dense (high resolution) mesh and using its surface in the background as a constraining surface to build a clean, lower resolution mesh. This is a common task when creating game assets where low polygon counts are a requirement, also when cleaning up scanned mesh data for animation, or when simply making it easier to change the topology and polygon flow of an existing model. The **Toolbox** contains tools that automatically conform to any item in the background, with a special OpenGL drawing mode to make it easier to see what is being drawn in the foreground over what is located in the background.

UV



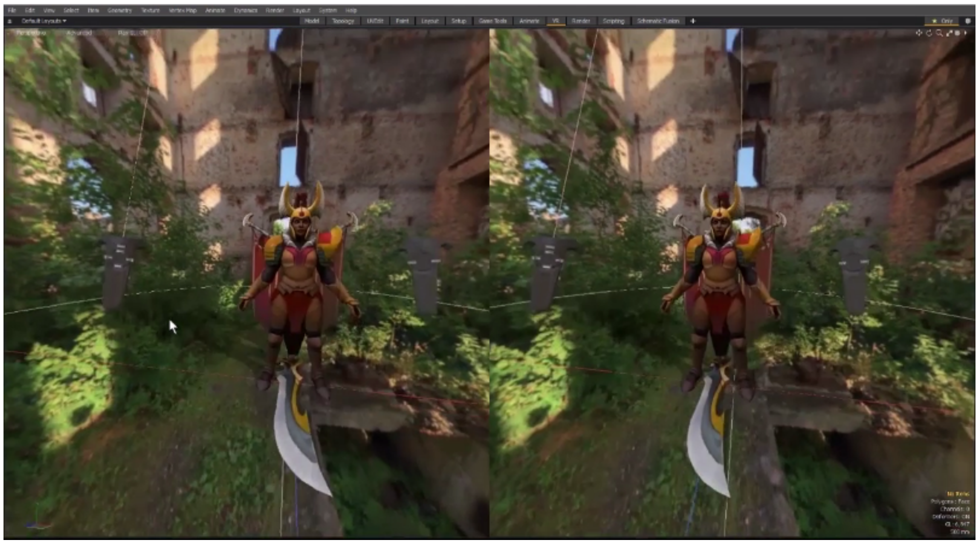
The **UV** interface splits the 3D viewport vertically adding a **UV View** on the left. You can select from the various maps in the **Vertex Map** list on the right-hand side. The **Toolbox** focuses on the various UV Mapping and editing tools.

Tip: You can also open the **UV** layout as a separate viewport by clicking the **UV** palette icon on the top-left corner of the interface.



For more information, see UV Viewport.

VR



The **VR** layout allows you to create content using virtual reality technology to interact with a scene in a real time 3D environment. By using Modo as your full 3D modeling and animation software with a Head Mounted Device (HMD), you can step inside the scene to move your models around, scale your scene, add annotations, save camera shots, teleport yourself from one position to the next and apply commands such as actions, using the HMD controls. Modo allows you to return to the desktop experience to fine tune your models and animations.

Opening the **VR** layout initializes a connected VR headset and allows you to immediately view and adjust your model within a virtual environment. In this layout you can complete item transformations, scene layouts, and design reviews.

For more information, see Modo VR.

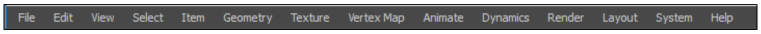
User Interface Conventions

Modo follows a series of user interface (UI) conventions across the application that can be different from what you are accustomed to, but provide a very intuitive and fluid platform for you to accomplish your creative vision. Understanding of these conventions is the first and most logical step in using Modo to its fullest potential.

Menus and Switcher Bar

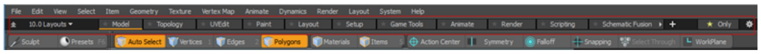
Menu Bar

The menu bar provides a quick access to all of Modo's core tools.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help

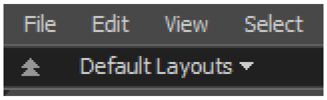
Clicking on any of the menu options activates a dropdown menu with many entries and sub-categories for you to select from. The menu bar is actually composed from a standard Modo Attribute Sheet and can be customized through **System** > **Form Editor** on the menu bar. Menu option names that display a dialog are followed by ellipsis (three dots).

Switcher Bar


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
10.0 Layouts
Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
Schematic Fusion
+
Only
Sculpt
Presets
Auto Select
Vertices
Edges
Polygons
Materials
Items
Action Center
Symmetry
Falloff
Snapping
Select Through
WorkPlane

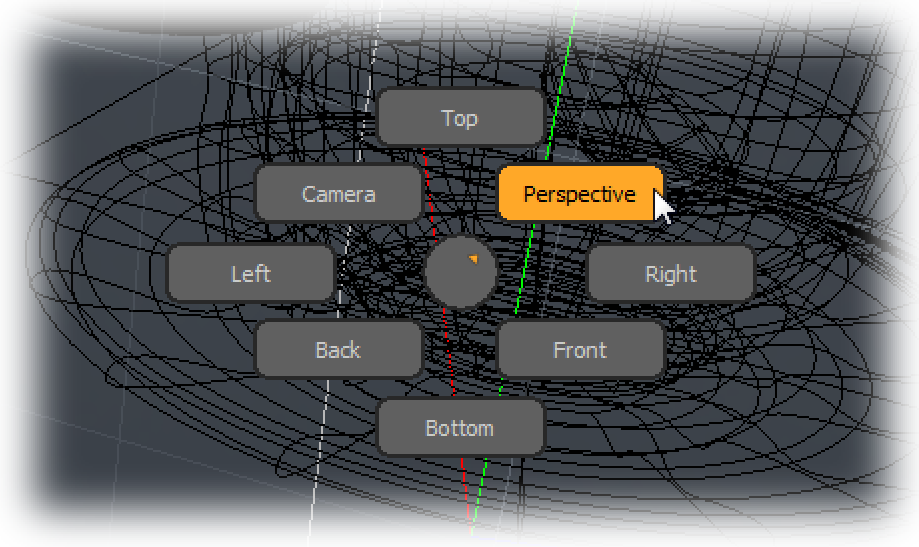
Directly below the menu bar is the switcher bar. Each tab in the switcher bar represents a focused workspace, optimized for their specific task. Left-clicking on a tab opens that particular layout. You can add your own layouts using the plus icon located to the right of the last tab. Once created, an additional tab is added to the right-most position of the bar. Tabs can be reordered by clicking and dragging a tab to a new location. The drop position is indicated by a small orange line. The star on each tab allows you to highlight specific, often used tabs. When you click the **Star Only** button, all the unstarred tabs are suppressed from display. Layouts can be reset to their initial default states by double-clicking on a specific tab, opening a requester asking if you are sure you want to reset the display. To minimize the display, click the two upward facing arrows on the far left, temporarily hiding it. To reopen it, click over the remaining small line.

The switcher bar also allows you to switch between new and old layouts. Click the **Default Layouts** field to reveal a list of available options:


File
Edit
View
Select
Default Layouts

Pie Menus

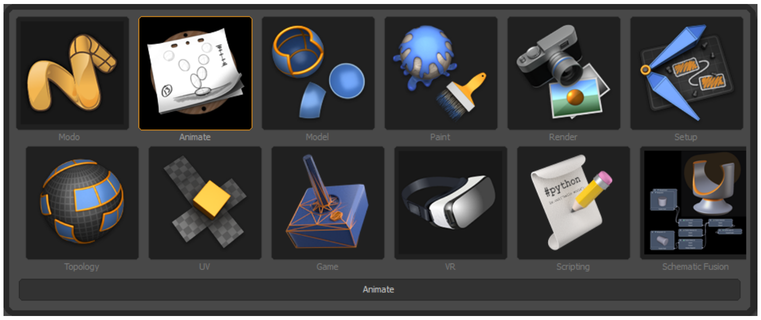
Pie menus are quick contextual pop-up menus that appear using keyboard shortcuts. The default keyboard shortcut is [**Ctrl/Cmd**] + [**Space Bar**}.


Top
Camera
Perspective
Left
Right
Back
Front
Bottom

Menu items are laid out in a wheel so you can access them gesturally for speedier workflow. By default, when a keyboard shortcut is invoked to recall a menu, the menu opens instantly. You can adjust this behavior with the preference setting **System** > **Preferences** > **Input** > **Remapping** with the **Open Pies Immediately** toggle. When disabled, opening pie menus requires you to use the keyboard shortcut and click over any viewport to open. Pie menus are created using the **Forms Editor**.

Note: Note that pie menus are limited to 8 items only. The first 8 items that appear in the Forms Editor are used for the pie menus, including disabled items. The Keyboard Shortcuts page references the various default menus available.

Layout Switching


Modo
Animate
Model
Paint
Render
Setup
Topology
UV
Game
VR
Scripting
Schematic Fusion
Animate

Modo also employs an interface switching option using **Shift** + **Ctrl**+**Tab** to quickly switch workspace layouts. Press **Shift** +**Ctrl** and then **Tab** to open the selection pop-up. Continuing to hold down **Shift** + **Ctrl** and then release and press **Tab** again to cycle to the next workspace highlighted by the orange outline.

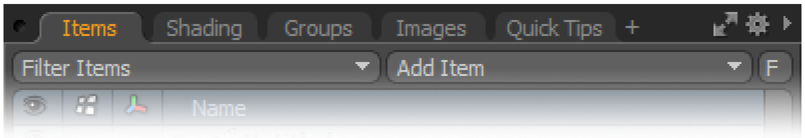
Subsequent **Tab** presses continue to cycle through the selections organized by the layout's most recent use. Releasing the keys opens the currently selected interface. This method eliminates the space used by the tabs in the interface.

You can also open the layout switcher from the lower toolbar, by clicking **Layouts**. This option is also available from the menu bar, in **Layout** > **Recent Layouts...**. Specific layouts are accessible from the menu bar under **Layout** > **Layouts**. Additionally, you can access the pie menu version of the layout switcher by pressing **Ctrl**+ **~** (tilde).

Tabs and Buttons

Tabs

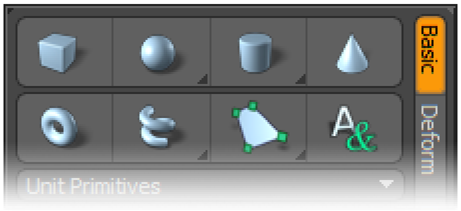
Modo organizes certain aspects of the interface into what are known as tabs.



In Modo, tabs allow multiple viewports to exist in the same layout (or to organize multiple layouts) and are a handy way to quickly get access to certain facets of Modo important only to a particular function. In some cases a viewport's display is dependent on information selected in another viewport, which is sometimes hidden under a different tab (such as item selection displaying their attributes in the **Properties** viewport). Tabs in Modo do not auto-select, so you have the power to control what information you want displayed at a particular moment without the surprise of viewports suddenly changing. You only need to click directly on a tab to open it for display. When a tab is active, the tab name changes from a light gray to an orange color. Background tabs are a darker gray color.

Buttons

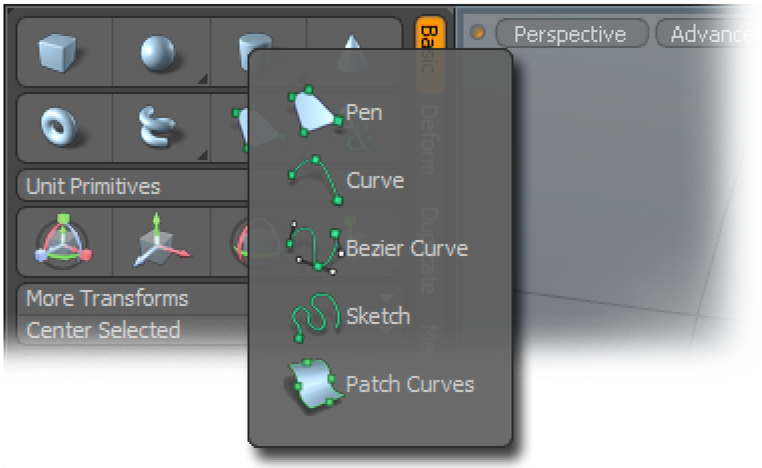
Modo displays commands and state for many functions as buttons.



Buttons exist as an icon, as an icon and text, or as just text (and any available keyboard shortcut, if applicable). These buttons appear throughout the interface, but mainly in the toolbox down the left side of the default interface. You need to click on a button to select the function. When active, the button highlights in orange, indicating its selected (active) state. For certain buttons, like tools, you can deactivate them by clicking the tool button again or pressing **Q** or the **Esc** key to drop the tool. For state buttons, such as the mode buttons, you can't deactivate the state, but you can choose an alternate button to change to a different state.

Button Pop-ups

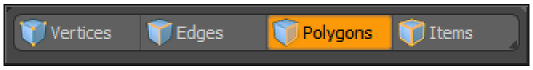
A button pop-up is a button that displays a secondary list or window.



In the toolbox, these buttons are separated by a small gray arrow pointing toward the lower right hand corner. To open, right-click on a pop-up button to access this list and make selections from it. You can also click and hold to open the pop-up as well. The GL Viewport pop-ups which allow for Viewport Type and Shading choices are special cases. The options list is displayed when you right-click the buttons in the 3D (OpenGL) Viewport. You can adjust pop-up transparency by navigating to **System** > **Preferences**... > **Default** > **UI** > **Popup Transparency** and changing the percentage there. For more information, see Defaults Preferences.

Alternate Buttons

Buttons can be set to perform alternate actions based on key modifiers. As the modifiers are pressed and released, the buttons change to show the alternate commands. Alternates are also listed on tooltips.



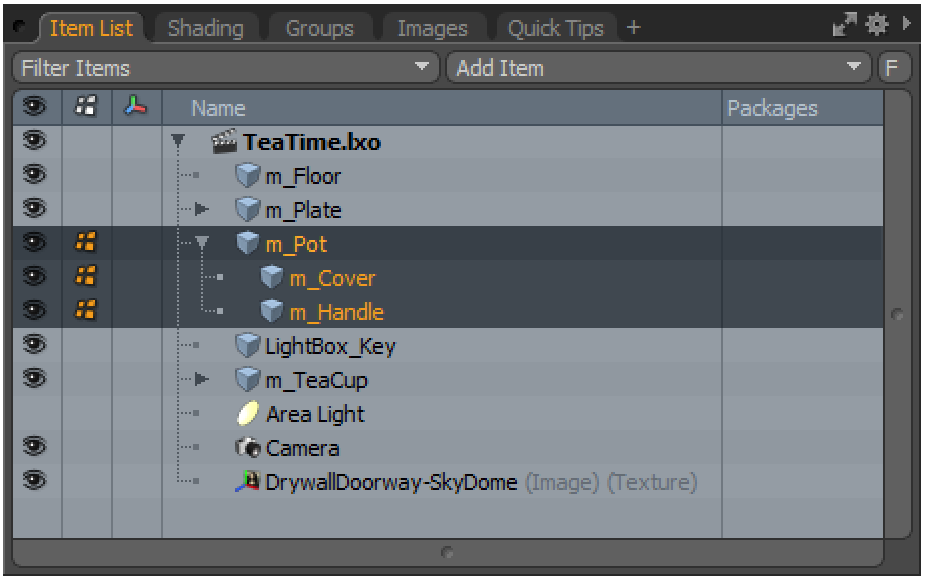
An example of an alternate action-based key modifier is the **Boundary** command. When you press the **Ctrl** key, the **Edge** button in the Modo modes toolbar displays **Boundary** instead of **Edges**. If you press the **Alt** key instead, the **Edges** button displays **Convert**. Another example is the use of the **Ctrl** key with the

Primitives. Pressing the **Ctrl** key changes the display of the Primitive buttons, and clicking them creates a Unit Primitive of whatever type is pressed.

For more information about selection modes, see Selecting Items.

Item Selections

When working with any item or layer in Modo, you need to tell Modo precisely which layer to work with by selecting the particular item.


Item List
Shading
Groups
Images
Quick Tips
+
Filter Items
Add Item
F
Name
Packages
TeaTime.lxo
m_Floor
m_Plate
m_Pot
m_Cover
m_Handle
LightBox_Key
m_TeaCup
Area Light
Camera
DrywallDoorway-SkyDome (Image) (Texture)

This can be done in the tree views of the **Item List** and **Shader Tree**, or in the list view of the **Vertex Maps** viewport. To select a layer, you only need to click directly on the layer name. Selected layers are differentiated by the dark highlighting of the layer, with the layer name changing to orange. You can select multiple layers in a variety of ways allowing gang edits of multiple items. You can select a series of items by clicking the top most item layer, and then pressing and holding the **Shift** key and then clicking on the bottom most layer, selecting all the layers in-between. You can select non-sequential layers by pressing and holding the **Ctrl** key and clicking to select the layers. Clicking on another layer without pressing the **Ctrl** key automatically clears the selection of other layers. You can remove individual layers from a selection by pressing the **Ctrl** key and clicking on the individual layers.

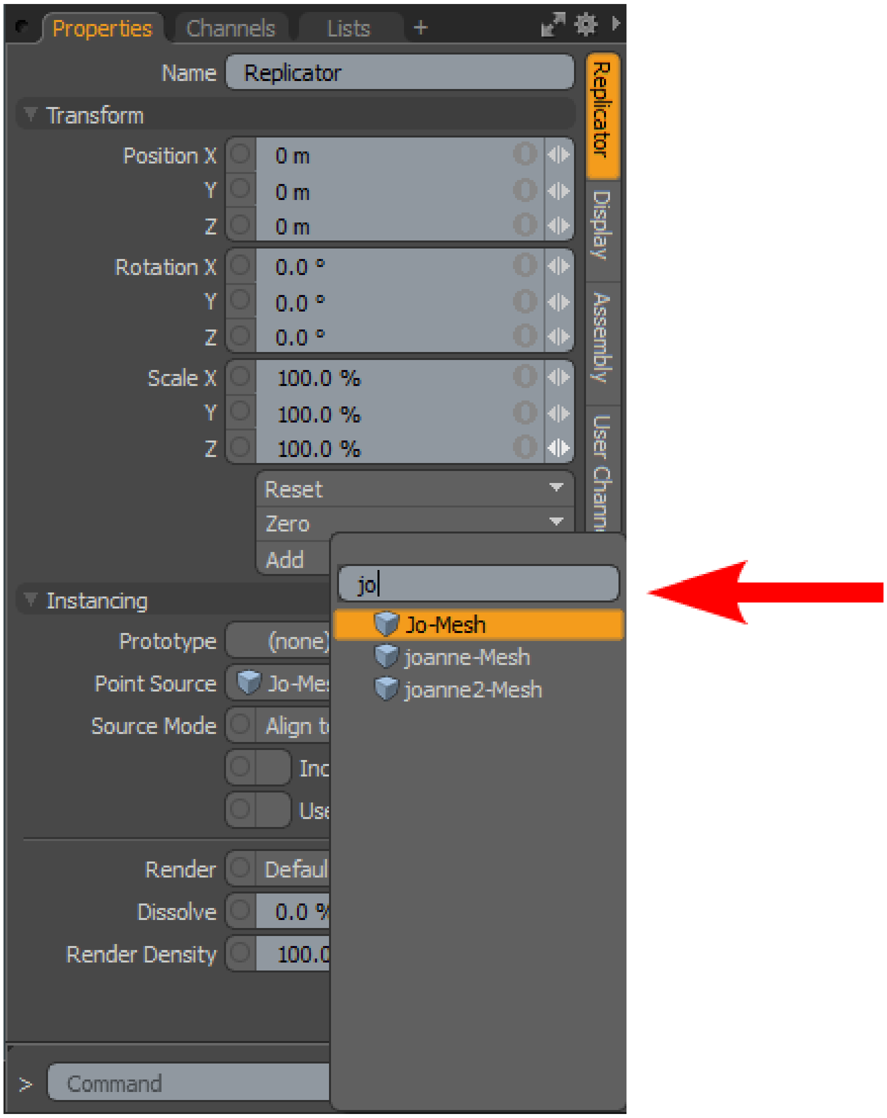
The first item selected always is the primary selection, and it is separated from subsequent item selections by a slightly darker color and a slightly lighter layer name. This is an important function for recognizing ordered selections necessary for certain animation and item creation commands, such as when setting up Replicators.

Filtering Menu Items

The Filter field is designed to assist you in locating menu options quickly. As you type into the field, a list of matching options is presented. Use the up and down arrow keys to scroll through the filtered list and then click **Enter**/**Return** to accept the menu item. The following is an example for the **Replicator** > **Instancing** > **Prototype** field.

Filter fields are found throughout the interface when a property contains a long list of possible options.

Note: The Filter is a case-insensitive.


Properties
Channels
Lists
Replicator
Display
Assembly
User Chann
Transform
Position X
Y
Z
Rotation X
Scale X
Reset
Zero
Add
Instancing
Prototype
Point Source
Source Mode
Render
Dissolve
Render Density
Command
Jo-Mesh
joanne-Mesh
joanne2-Mesh

Name: Replicator

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

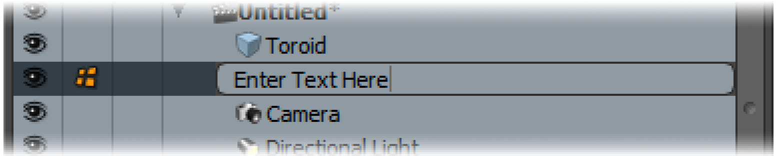
Filter: jo

Nesting and Hierarchies

Entries in a tree have sub-entries, which you can expand by clicking on an arrow icon next to the parent ▸. One column indicates the nesting, and sub-entries are indented relative to their parent. Entries also have attributes, which you can expand or contract by clicking on a plus/minus (**+**/**-**) icon next to the entry with attributes. Attributes are also nested under their parent, but for clarity attributes cannot themselves have sub-entries. Double-click on the parent item to select all the children. **Shift**+click the parent arrow ▸ to expand all subsequent nested/child item layers simultaneously.

Inline editing

Text fields can be edited inline wherever possible. In particular, text fields in trees that you can alter, can be edited directly in place.

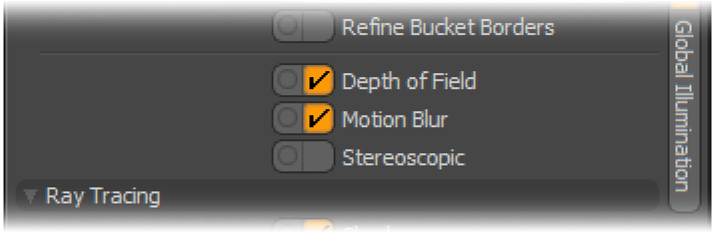


Enter Text Here

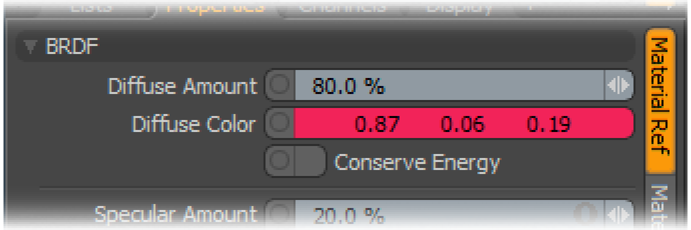
To edit text, click to select the text, pause, and then click again to convert the text to an inline edit field. Once converted, you can type in the new name and press **Enter** to save the new name.

Checkboxes

Checkboxes or boolean buttons act as simple toggles to allow you to set the enabled/disabled state of any option. They appear as small gray squares when the state is set to disabled. When you enable the button by clicking on the square area, it becomes highlighted in orange and a black checkmark appears in the square. Clicking the button again toggles to the previous state.


☐ Refine Bucket Borders
☑ Depth of Field
☑ Motion Blur
☐ Stereoscopic

Advanced Color Navigation


BRDF
Diffuse Amount
80.0 %
Diffuse Color
0.87
0.06
0.19
Conserve Energy
Specular Amount
20.0 %
Material Ref

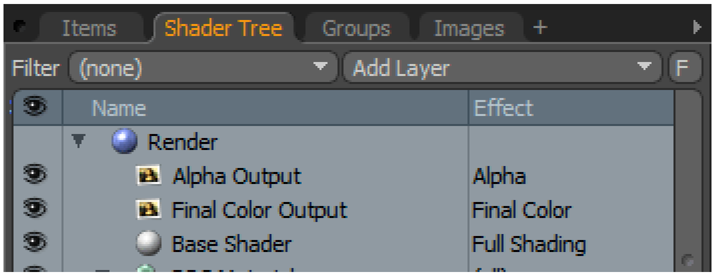
Modo has unique color selection behavior. In any color swatch, three numbers are displayed, by default, each a floating point value between **0** and **1**. Each value represent individual RGB color settings. You can click and drag on any of the numbers to directly modify the color, which is displayed interactively in the background of the swatch as the mouse moves. Additionally, holding the following keys modifies all three values at once:

- **Shift**+click+drag = Brightness
- **Ctrl**+click+drag = Hue
- **Alt**+click+drag = Saturation

Or, simply clicking anywhere on the color swatch itself opens the Color Picker Viewport.

Tip: You can change the default floating point color system by using the **Preferences** dialog. In the menu bar, select **System** > **Preferences...** and in the **Preferences** viewport, select **Input** > **Accuracy and Units**. Here you can modify the **Color Unit System** to **Floating Point** (**0** - **1**), **Percentage** (**0**% - **100**%), **Integer** (**0**-**255**), and **Hexadecimal** (**00 00 00** to **ff ff ff**).

Simple String Search Filter

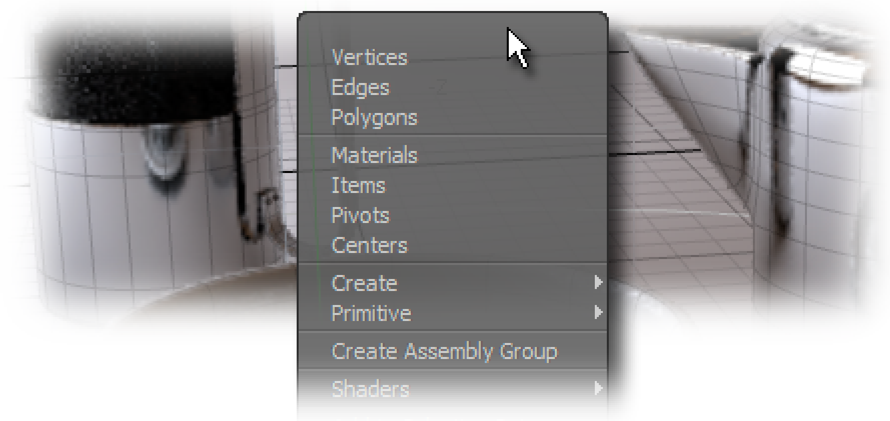

Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer
F
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading

Modo provides a function that allows you to use simple search strings to help in locating layers in a list viewport, such as the **Items** list or **Shader Tree**. It is located toward the top right of the viewport as a small **F** button, where you can click on the button to open the filter input field. Text is then input onto the value field same as elsewhere by simply clicking on the input field to set focus and then typing on the keyboard. When you type strings of text, the viewport updates to display the results of the search. You can then select a layer and click the **F** button again to hide the search options returning the list view to normal, with the layer remaining selected. You can select two modes for the search with the pop-up menu, either **Simple Search** or **Pattern Matching**.

When searching in the **Clips** list (images viewport) the original file name is considered, for other viewports, the actual layer name is considered. When the **Match Case** toggle is enabled, the search considers the letter-case of the text. If you prefer a flat display of the list results, you can toggle the **Flat** button to allow easy view and selections of multiple layers.

Contextual Pop-ups

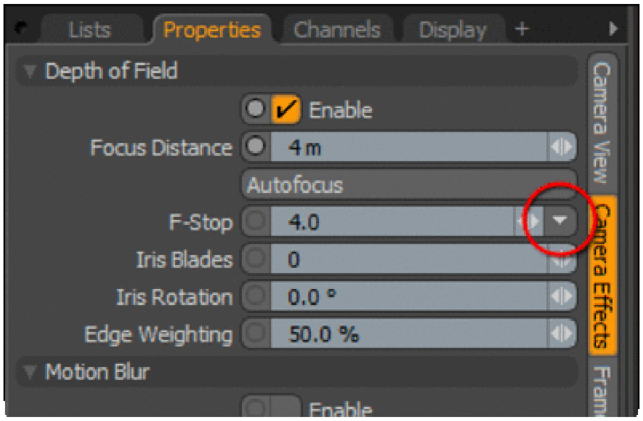
The various viewports (including the 3D view and UV edit views) display pop-up option lists when you right-click within the viewport as well as in the header.



Here you can apply functions to that particular viewport, activate certain tools, and affect selections within the viewport, all dependent on the mode, and over what element in the interface the particular right-click occurred. If there is no option available, then right-clicking does nothing, the menus only appear when the context is correct. For example, right-clicking in the viewport while in any component selection mode offers a number of choices to affect the geometry selections.

Value Presets

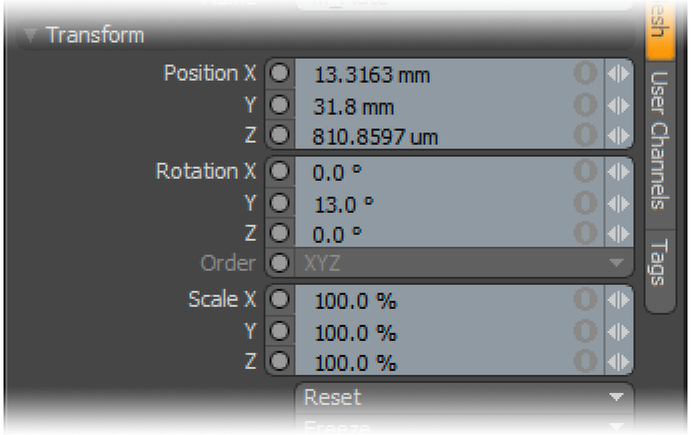
Value Presets are the small white downward-facing arrows that appear on the far right side of many numeric entry fields.



Depth of Field
☑ Enable
Focus Distance 4 m
Autofocus
F-Stop 4.0
Iris Blades 0
Iris Rotation 0.0 °
Edge Weighting 50.0 %
Motion Blur
Enable

By clicking on the arrow you can open a mini pop-up window for selecting and saving often used values. The values presented within each property are specific to that particular property, meaning you only see the F-Stop values in that particular field. Simply clicking on any of the values when highlighted inserts that particular value into the associated data field. You can use the **(new)** command to create a preset, and the **(delete)** command to remove any values from the pop-up. Once a value is created, it is saved to Modo's config file.

Numeric Entry Fields and Mini-sliders



Transform
Position X 13.3163 mm
Y 31.8 mm
Z 810.8597 um
Rotation X 0.0 °
Y 13.0 °
Z 0.0 °
Order XYZ
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Numeric attributes can be edited directly inline by simply clicking directly on the value, converting it to an inline edit field, a second click sets the insertion point, then with the keyboard, you can enter a new value, then pressing **Enter** to accept. Numeric properties can also be modified interactively through the mini-sliders, the small white arrows that appear on the far right hand side of most numeric entry fields. By clicking and dragging on the mini-sliders, you can quickly increase (dragging right) or decrease (dragging left) the current value in the field. You can step values by clicking on the right or left side of the slider, click on the right to increase the value, click on the left to decrease it. The mini-sliders also have a system of detents that cause tiny pauses to numeric values as you drag over numbers such as 100% or 180°. Depending on the input the values are:

- angles, major: 0, increments of +/- 90
- angles, minor: increments of +/- 30 and +/- 45
- percent, major: 0, increments of +/- 100%
- percent, minor: increments of +/- 25%
- distance, major: 0, powers of 10
- distance, minor: nice grid increments
- float, major: 0, powers of 10
- float, minor: multiples of 2, 2.5, 5

This is very useful for quickly and accurately dragging out a value. Pressing the **Shift** key while using a mini-slider increases the speed, a value is incremented by a factor of 10, while the **Ctrl** key reduces the speed by a factor of 10.

You can gang edit all three fields simultaneously as well by clicking the ⓿ to the left of the mini-slider enabling the feature:

- The first click changes to ⊖ Copy - identical values are applied to all like controls.
- The second click changes to ⊗ Proportional - value changes are applied proportionally to all like controls.
- The third click changes to ⊕ Relative - value changes are added to all like controls.
- The fourth click returns to ⓿ Independent - controls are edited independently.

When entering numeric values, you can use keyboard equivalents as well:

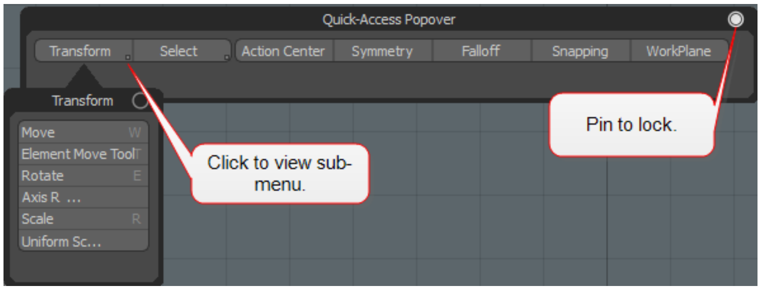
- **Ctrl**+**Alt**+**Enter** - Copy ⊖,
- **Alt**+**Enter** - Relative ⊕, and
- **Ctrl**+**Enter** - Proportional ⊗.

Tip: Numeric entry fields allow you to enter simple equations and the value is computed automatically. This allows for very simple calculations without using a calculator.
For example, when you use the Clone tool to create seven copies of the original mesh, each with a small amount of rotation so that by the seventh clone their total is 360 degrees, you can enter 360/7 in the **Rotation** numeric entry field. When pressing **Enter**, Modo handles the calculation and the resulting **51.4 degrees** is in the field. Accepted symbols are **/** = divide, ***** = multiply, **-** = subtract, and **+** = add.

Popover Window

Any type of viewport or content can be opened as a popover. Once activated, interaction can happen normally in the window. Moving the mouse outside the popover bounds dismisses it. To see an example of a popover window, press and hold **Alt**+**Spacebar** anywhere in the Modo interface. The **Quick-Access Popover** window displays, showing tools for Transform, Select, Action Center, Options, Symmetry, Falloff, Snapping, and WorkPlane.

Popovers can speed your workflow by putting the tools directly under your fingertips, such as pressing **Transform** > **Scale** in viewport to display the tool properties in the Toolbox in the left pane.

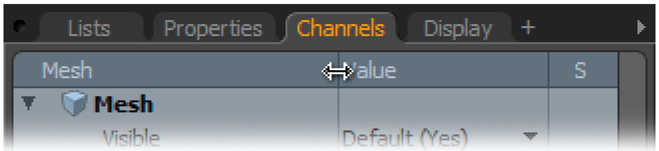

Quick-Access Popover
Transform
Select
Action Center
Symmetry
Falloff
Snapping
WorkPlane
Transform
Move W
Element Move Tool T
Rotate E
Axis R ...
Scale R
Uniform Sc...
Click to view sub-menu.
Pin to lock.

You can also pin popovers in the UI by clicking the small circle in the upper right corner of the window. To dismiss the window, click the pin again. Pinned popovers are persistent across Modo sessions.

Note: If a popover form becomes empty, it is automatically closed. If it gets populated, it becomes visible again.

List Columns

To facilitate the display of various types of information in a list viewport, data is organized into columns.


Lists
Properties
Channels
Display
+
Mesh
Value
S
Mesh
Visible
Default (Yes)

The default column sizes are not always a proper fit for the data they contain. You can resize column widths by hovering the mouse pointer on the divider line until the cursor turns into two opposing arrows, then clicking the divider and dragging left or right resizes the column.

Drag and Drop

Modo supports Drag and Drop behavior for loading object files and bitmap images. Since Modo is a multi-document system, each 3D object or project that you drag in appears as its own document layer. When you drag and drop images, if dropping them onto geometry they appear as a texture in a material. If you drag them into a 3D viewport, they appear as a Backdrop Item. If you want to load a bitmap, but don't want to apply it to geometry or as a Backdrop Item, drag it into a non-3D viewport. Items can be dragged from the 3D views directly onto groups (where they become members of the group) or into the Schematic Viewport. Channels can be dragged from the Channel List onto groups or into the Schematic.

Viewport Maximize and Restore

You can maximize or minimize a viewport within a group to temporarily expand the amount of space given to a specific viewport. This is very useful when working in a quad view modeler layout as it allows you to quickly maximize a particular viewport, then restore back to the previous quad view state. By default this is mapped to the **0** (zero) key on the numeric keypad. To use, hover the mouse pointer over the target window and press **0**. The target viewport maximizes to full size based on the group where it is embedded. Pressing **0** again while hovering over the same viewport return the viewport to its previous state. This can also be useful when working in long **Item Lists** and **Shader Trees** and searching for particular layers.

Floating a Window

Viewports can be torn off and converted into floating windows, or new floating windows can be created which can be manipulated like the main window. To float any window, simply click the thumb (the circular indentation at the upper left hand corner of the window) and drag the position of the window. The window is removed from its position in the interface and floats when you do this. To dock the floating window, just reverse the process. Click the thumb and then move the cursor back to the newly desired position in the interface, by hovering over a viewport frame edge. A yellow line displays indicated the new location of the window. Releasing the mouse button docks viewport. Floating windows are persistent between sessions.

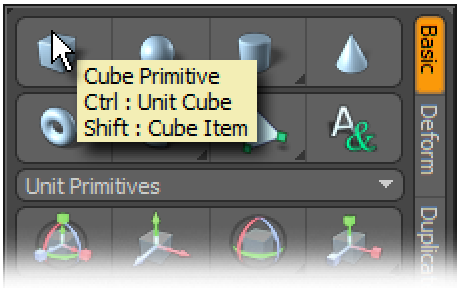
Hauling

Hauling is an off-handle click+drag action which controls one or two values of a tool directly in the viewport using the left/right, and possibly up/down motion of the mouse. Many tools have an advanced handle setting that displays hauling feedback in the form of lines drawn horizontally and vertically on the screen corresponding to the direction of mouse deltas. The lines are labeled, and persist as long as hauling can be resumed. Clicking directly on a line highlights it and starts hauling for that parameter only. For more information, go to OmniHaul.

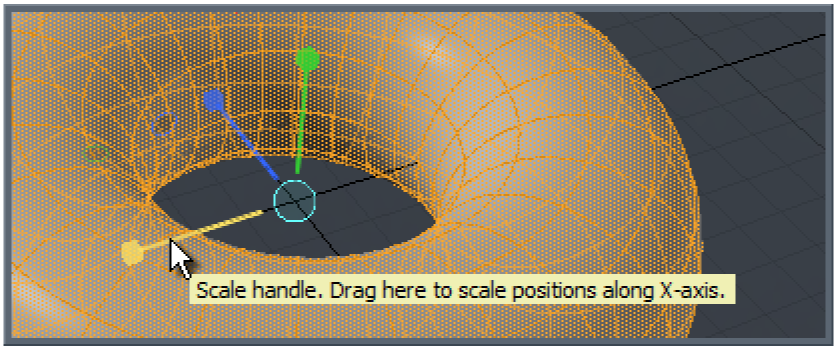
Help

Tooltips

Hovering the mouse cursor over a button displays a tooltip, a small text hint with the name of a function or tool, and keyboard equivalents if applicable. For certain commands, tooltips also contain useful hints for using the function.


Cube Primitive
Ctrl : Unit Cube
Shift : Cube Item

Hovering the mouse over any tool handle (when it is highlighted) pops open a handle specific tooltip, often with instructions related to the function of the specific handle.


Scale handle. Drag here to scale positions along X-axis.

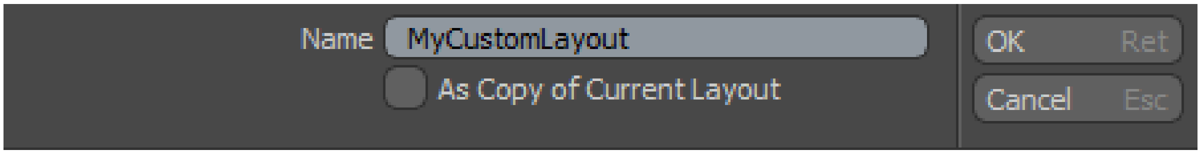
Context Help

You can access context sensitive help by pressing the **F1** key or by choosing **Help** > **Activate Help** from the menu bar. The cursor changes into a question mark. Then clicking on a tool or UI element opens a page of related information displays in an HTML browser window. For complete help information, go to the menu bar and choose **Help** > **Documentation** > **Inline Help System**. This displays the main index page of this documentation. If you still need help, you can visit the Foundry community forum where you can directly post questions to be answered by expert users and helpful Modo fans everywhere.

Customizing Your Layout

Modo provides one of the most flexible UIs of any 3D application, providing a variety of controls that allows you to customize the program to your liking. Understanding how to tweak an existing layout can be better explained with a run through of making a layout from scratch. When creating a completely custom layout, the first thing to do is define a new layout. This also ensures that the system defaults are not mistakenly changed. To do this, follow the steps below.

1. In the menu bar, click on **Layout** > **New,** and in the pop-up window, enter a unique name and make sure the **As Copy of Current Layout** option is disabled.



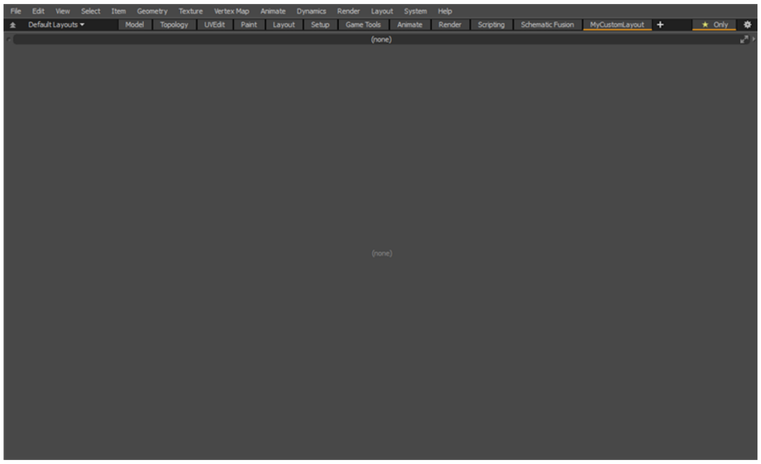
Name MyCustomLayout

☐ As Copy of Current Layout

OK Ret
Cancel Esc

2. Click **OK**.

   This creates the named layout in the menu bar under **Layouts** and automatically switches to it. You are now presented with a completely blank workspace.



File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Default Layouts
Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
Schematic Fusion
MyCustomLayout
+
Only
(none)
(none)

The controls to customize the layout are all that are left within the viewport. In the upper left corner below the menu bar, you can see the **Thumb** (●), a tiny shaded dot next to the title bar. All viewport frame specific controls reside here. On the opposite side of the title bar rests the **Widget (▸)**, a small right facing arrow. This controls the contents of a frame. You can right-click on each of them to open a pop-up menu. Every window has these controls making it easy for you to modify any layout to your liking.

To continue the exercise, recreate a single tab of the default tabbed layout. To do this, you need to divide the interface.

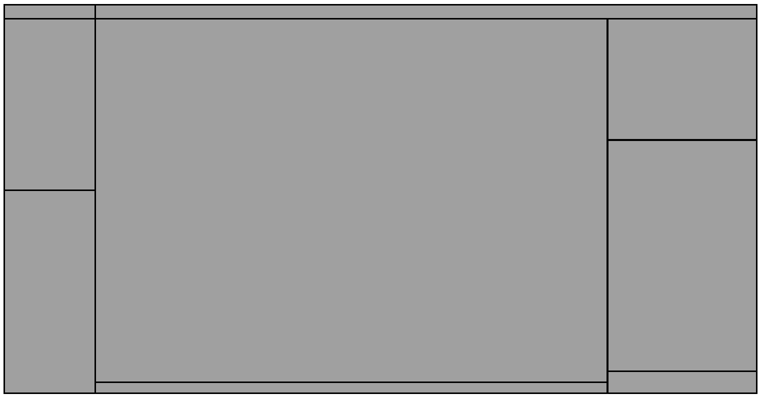
3. Select the **Widget**, and from the menu, choose **Viewport Group**.

   This nests two viewports inside one another. Now you have two widgets on the right.
4. From the upper viewport's **Widget** menu, select **Tabbed Viewport**.

   This creates the tabbed interface, where each tab can hold a completely different layout.
5. In the nested viewport below, right-click the **Thumb**, and choose the **Split Vertical** option.

   This divides the window into two equal sized viewports stacked on top of one another.
6. To resize the viewports, hover the mouse pointer directly over the border between the frames, then click and drag to position the edge in a new location.

   Dragging the border toward the top makes the first viewport.
7. To finish off, continue dividing the windows using the split commands found under the **Thumb**. Follow this diagram, splitting viewports horizontally or vertically, as necessary.



   Split the topmost bar horizontally once, and the lower window twice, resulting in three columns. Split each of those columns vertically once, with an additional split in the right column. Position the viewport frames, so the layout loosely mimics the above template.

Note: If you accidentally split a viewport, you can delete viewport panes in **Thumb** > **Delete**.

Next, the windows need to be populated with premade interface content. The most common viewports are easily accessible from the **Widget** on the right side of the title bar of each viewport. You can add content by right-clicking the **Widget** of the window you would like to populate and selecting the viewport type from the popup menu.

Let's traverse the layout in a top to bottom, and left to right fashion filling in the viewports.

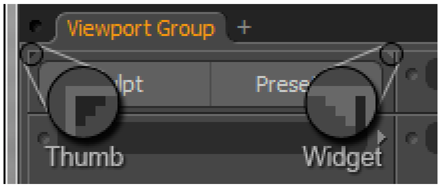
1. In the first viewport, select **Form View** in the **Widget** menu.

   This changes the window to a generic form but it still looks empty.
2. From the **Widget** again, choose **Viewport Settings** > **Sculpt and Presets Quick Access**.

This adds the quick access buttons to the viewport. but the title bar is a bit in the way.

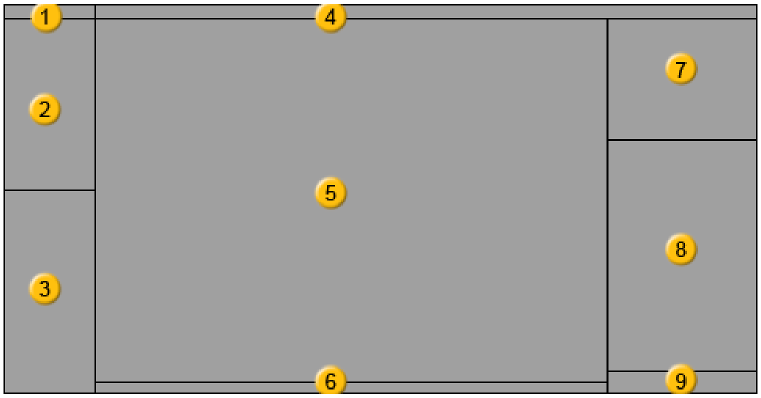
3. To remove the title bar, right-click the thumb and select the **Min Header** option to minimize it.

   The **Thumb** and the **Widget** become tiny triangles in the corners of the window, as illustrated below.



This window is ready, so move on to the next one.

4. Right-click the widget and select **Toolbars** > **modo Tools** and again, minimize the header.
5. Below that window in the next frame's **Widget**, select **Properties** > **Tool Properties** and then minimize the header.



For the remaining windows, the menus are as follows:

1. For Window 4, select **Widget** > **Form View** and **Widget** > **Viewport Settings** > **modo Modes**.
2. For Window 5, select **Widget** > **3D Edit** > **3D Model View.**
3. For Window 6, select **Widget** > **Status Feedback**.
4. For Window 7, add some tabs: select **Widget** > **Tabbed Viewport**, right-click on the tab itself and choose **Data Lists** > **Item List**.
5. Click the **+** icon on the right of the tab and then choose **Data Lists** > **Shader Tree**, that adds an additional tab.

   Window 8 serves multiple purposes, with nested viewports, similar to the main window.

6. **Select Widget** > **Application** > **Viewport Group**, set the upper **Widget** to **Tabbed Viewport** and split the lower window into three even sized viewports.
7. In the top viewport, set **Widget** > **Data Lists** > **Vertex Map List**.
8. In the next window, set **Widget** > **Data Lists** > **Tool Pipe** and in the third, set **Widget** > **Info** > **Info and Statistics**.

9. Next, across the top of the group, add an additional tab, as in step 6 above, and select **Widget** > **Properties** > **Item Properties**.
10. Minimize the header in all the viewports you just created, forgoing the rest of the tabs for now.
11. For Window 9, select **Widget** > **Utility** > **Command History**.

This re-creates the **Model** view of the default interface, from a blank slate. Make sure to save it again in the menu bar, selecting **Layout** > **Save** command.

Between the commands of the **Thumb** and the **Widget**, a great deal of customization power can be wielded. Any Modo viewport can be recalled from the **Widget**, although some are available only within the **Viewport Settings** sub-menu choices.

Note: Many viewports change when scaled, providing slightly different ways of interacting with the information contained in them. These options can be useful in setting up interfaces that mimic other popular applications. A thorough understanding of the common viewports helps in achieving your intended result.

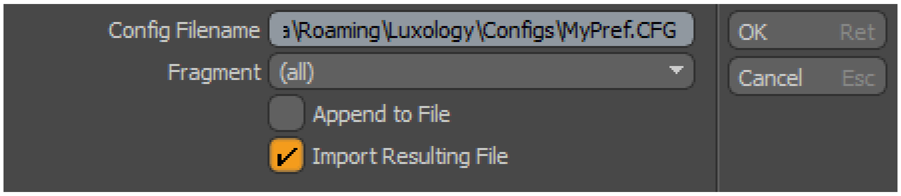
Saving Your Custom Preferences

Depending on the workflow you typically use, you may want to customize your Preferences. For example, set the default image format to use PNG 16-bit. Each time you upgrade your version of Modo, these preference changes can be used again without having to reconfigure your environment.

To change and save your preferences:

1. Click **Systems** > **Preferences**.
2. Make the appropriate changes and then close the **Preferences** dialog.
3. Click **File** > **Config Export**, name your file, and click **Save**.

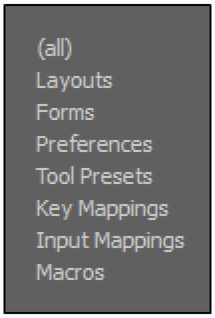
   The **Export Config Fragment** dialog displays.



Config Filename: a\Roaming\Luxology\Configs\MyPref.CFG
Fragment: (all)
☐ Append to File
☑ Import Resulting File

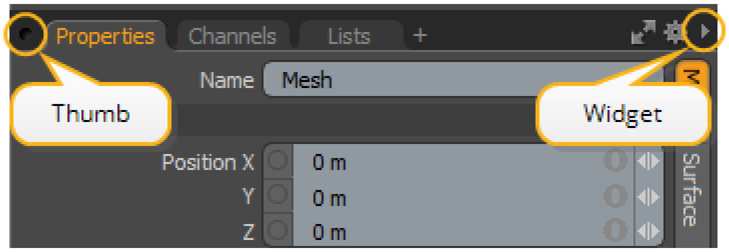
4. Specify the following options and then click **OK**.

Config Filename	Allows you to specify the location to save your configuration file to and to name your file.

	By default, this file is saved to the following directory: Windows: **C:\Users\'user name'\AppData\Roaming\Luxology\Configs** Mac OS: **/Users/qa/Library/Preferences/** **Tip:** Each time you make a change in your Preferences, save your changes as individual configuration files. Ensure that you have enabled the Append to File option to save your changes in your main configuration file. Then quit and restart your session of Modo to ensure there are not issues. Continue to save each preference change as individual files. Once all your changes are made and tested on restart, and these accumulated changes launch each time you start Modo.
Fragment	Allows you to specify where you have made changes in your configuration file. The following options area available:  (all) Layouts Forms Preferences Tool Presets Key Mappings Input Mappings Macros
Append to File	Enable this option if the same configuration file already exist to append these changes to that file without overriding it.
Import Resulting File	By default, this option is enabled to import the changes made to the configuration file automatically into your scene when you launch Modo.

Using Layout Controls

Modo's interface is made up of a flexible number of windows, called viewports, which can each have their own contents. Most of the interface elements can be found under **Layout** on the menu bar. However, a couple of interface elements also exist elsewhere to easily control the viewports. In the viewports menu bar are the Thumb (or Dimple) and Widget controls. They appear in the upper-left and -right corners, respectively.



Right-clicking on either icon opens a context menu with a variety of options for controlling viewports. The Thumb is used for manipulating the viewport frame, meaning the viewport itself can be split, deleted, or otherwise customized using its commands. The Widget controls the contents of the viewport, where you can choose from the entire array of possible options. You can open either menu by right-clicking on the icon.

Thumb

Controlling the Viewport

To control the viewports using the Thumb button:

- Click and drag the Thumb icon to detach the viewport into a floating window,
- Shift+click and drag the Thumb icon to clone the current viewport into a floating window,
- Ctrl+click and drag the Thumb icon to split the viewport in the direction of the drag (either horizontally or vertically), or
- Click and drag on a viewport divider to scale or resize the associated viewports.

Thumb Menu

Once you have opened the Thumb menu, a number of options are available.

- **Duplicate** - creates a copy of the selected viewport and deposits it as an embedded viewport on the right-hand side of the layout, leaving the original in place.
- **Copy** - creates a detached copy of the selected viewport, leaving the original in place.
- **Detach** - removes the embedded viewport and floats it in its own window.
- **Delete** - removes the current viewport from the interface.
- **Save Viewport** - saves any changes made to the current viewport preset.

- **Save Viewport As** - saves a new viewport preset, allowing you to set the name and category.
- **Save Viewport Group** - saves any changes made to the current viewport, as well as any child viewports, to the current preset. Child viewports are defined as those that are nested within an overall viewport frame, such as the **Model Quad** layout.
- **Save Viewport Group As** - saves a new viewport group preset, allowing you to set the name and category.
- **Delete Saved Viewport** - provides a means to remove custom-made viewport presets.
- **Revert** - restores the viewport to its default preset settings.
- **Maximize/Restore** - fills the viewport group frame with the current viewport. This option is highly-dependent on the viewport's layout. This is mostly useful in the case of maximizing a 3D viewport inside a collection of 3D views, as in the case of the **Model Quad** preset.
- **Min Header** - removes the viewport header, leaving mini-versions of the Thumb and Widget in their respective corners. This preserves the screen real-estate when necessary and is convenient when using viewports for simplified data displays, such as the coordinates viewport in the bottom-left of the default Modo layout.
- **Lock Height** - fixes the viewport height at the current setting. Subsequent resizing of the interface won't affect any particular window that has its size locked, allowing for fixed-size viewports.
- **Lock Width** - fixes the viewport width at the current setting. Subsequent resizing of the interface won't affect any particular window that has its size locked, allowing for fixed-size viewports.
- **Split Horizontal** - evenly splits the current viewport into two viewports of the same type. These are arranged next to each other.
- **Split Vertical** - evenly splits the current viewport into two viewports of the same type. These are arranged one on top of the other.

Widget

Widget Menu

The Widget menu is content-sensitive and the menu options change based on the category of the currently-loaded viewport preset. The sections are separated by thin shaded lines. The upper-most section changes to the most common types of viewports for any given category. The next section down provides all the defined categories and their related viewport types. For example, for the following menus:

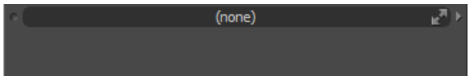
- **Viewport Settings** - contains all the viewport-specific options. For **Form View** viewports, this is a long list of the various forms available such as the mode options buttons. For the **Shader Tree**, this changes to **Filter** and **Preview Icon** options. The settings of many of the viewport options are covered in their respective sections under Understanding the Interface.

- **Viewport Controls** - duplicates the functionality (and therefore the menu) of the Thumb context menu as a convenience.
- **Layout Controls** - provides the same functionality of the options available in Modo's **Layout** menu option, in a sub-heading, as a convenience. The options available are as follows:
    - **Layout Presets** - choose from the various preset layouts or load a custom-saved layout.
    - **Save Layout** - save any interface customization into the currently-loaded preset. This action overwrites the original layout, replacing it fully.
    - **Save Layout As** - save any interface customization into a new Modo **Default Layout** that appears at the bottom of the **Layout Presets** list. When invoked, a menu opens, requesting a name for the new preset.
    - **Revert Layout** - provides a means for resetting any layout to "factory original", should one be mistakenly modified.
    - **Clear Layout** - removes all viewports from a layout, leaving only a single empty "(none)" viewport form. For more information on building a viewport from an empty frame, see Customizing Your Layout for a tutorial.
    - **Delete Layout** - opens a pop-up window, allowing you to remove any custom-created interface layouts.

Note: It's not possible to remove the default layouts.

Additional controls for layouts are available in the **Layout** menu on the Modo menu bar:

- **Lock Thumb Dragging** - interactively modify viewports, using several combination actions, when dragging the mouse pointer over the Thumb icon. You may find that, from time-to-time, you might mistakenly modify the layout and want to disable the Thumb dragging functionality.
- **Lock Divider Dragging** - interactively resize viewports by clicking and dragging the mouse pointer when over the frame dividers. You may find that, from time-to-time, you may mistakenly modify the layout and want to disable the divider dragging functionality.
- **Toggle Palettes** - toggle the visibility of floating palettes using this command or the keyboard shortcut ` (backtick).
- **Windows/Palettes** - while all the default layouts have viewports that are embedded, it's also possible to have floating viewports. When viewports are floating (or detached) they can be **Window** or **Palette** style. The only difference between them is that palettes have a thinner header, with a single control for closing the **Palette**, whereas Windows provide full OS-specific controls for minimizing or maximizing and closing the viewport. The menu bar command **Layout** > **Windows/Palettes** provides additional options to open custom pre-made viewports or varying types.

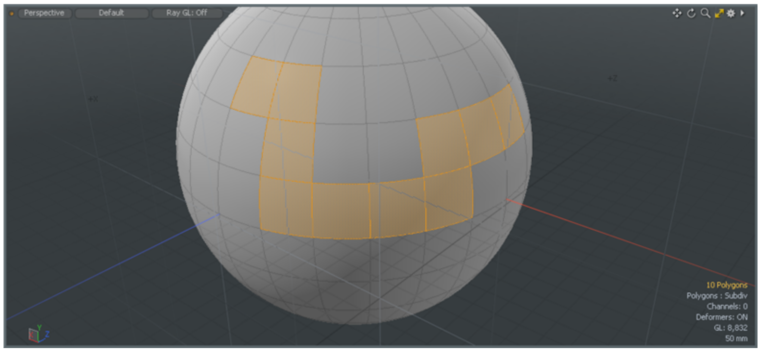


Using Standard Tool Controls

Because Modo is a 3D program, the way in which you interact with the interface and the elements within is important to understand. Beyond the standard ways you interact with any software program, such as selecting menu items or entering text, Modo uses a series of universal functions when modifying and manipulating elements within a scene.

Selections

Before manipulating any element, you need to specify what particular element you wish to manipulate by way of a selection.



Selections are made in several ways, but most often it is with the left click. Modo uses a technique called pre-highlighting to cue what element is selected when you click the mouse. By default this pre-highlighting is a light blue color. Once you click on an element, it changes to an orange color to signify its selected state. Clicking and dragging continues to select any elements the mouse is dragged over, this is known as a paint selection. Modo provides a number of function to aid in making selecting easier. For more information, see Using Selection Modifiers.

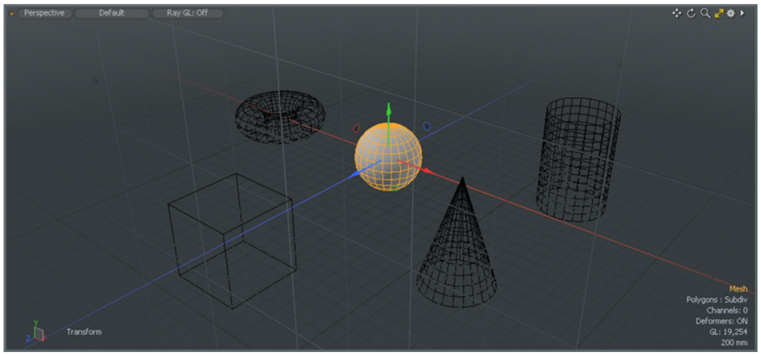
Note: When in a component mode (vertex, edge, or polygon) no elements are selected, Modo considers everything selected, this is an implied selection that has no highlighting. This is not the case for other modes, where elements need to be explicitly selected in order to modify them.

You can do multiple selections in a variety of ways. Click to select something, then press **Shift** and click on additional components to add them to the selection.

To remove an element (a vertex, edge, polygon, or item) from a selection, press **Ctrl** and click on the selected element to remove it from the selection. You can assign selection set to recall a selection at a later time. Use the menu bar command **Select** > **Assign Selection Set** to set the set. Use the command **Select** > **Use Selection Set** to load the saved selection. Even better, you can use the **Statistics** tab on the **Info and Statistics** toolbar. Click the plus button next to the named selection to activate the selection, or the minus button to subtract it from the viewport selection.

Tool Handles

In Modo, viewport manipulators or gizmos are referred to as Tool Handles. They are used to interactively control modeling operations in the viewport similar to the way an artist works with sculpting or drawing tools.



Handles have consistent color coding that acts as a quick indicator of the value that they edit and/or the context in which they are going to perform. In the default color scheme, tool handles use red, green, and blue when editing in 3D model space. When working in UV space, the handles are drawn as mustard yellow and teal (blue-green). The colors are used to represent the major axes when applicable (such as with the Move or Scale tools):

- In 3D space: Red = X, Green = Y, Blue = Z
- In UV space: Teal = U, Yellow = V

To access different tool handle styles, on the menu bar, navigate to **View** > **Tool Handles** > **Draw Style**. The following options are displayed:

- **Invisible** - Usefor hiding the Tool Handle entirely.
- **Basic** - Use for a minimum viewport display for the handles.
- **Advanced** - Use for more precision using the tool handles.
- **Default** - Use for returning to the factory preset.

You can also use the minus **-** or equals **=** keys to make the tool handles larger or smaller while working.

Key Toggles (aka Sticky Keys)

Keyboard shortcuts can be used as soft toggles to speed up workflow. For instance, if you need to move vertices in a mesh, invoking the menu bar command **Edit** > **Transform** > **Move**, you might find it cumbersome to constantly drop the active tool, select new vertices, and reselect the tool. Instead, you can use the Key Toggles functionality. For any tool that has a keyboard shortcut, such as the **W** key used to activate the **Move** tool, you can press the key, temporarily activating the tool, then use the mouse (or tablet Pen) in the 3D viewport to apply the adjustment (all while continuing to press the appropriate key). Once the adjustments are made, you can release the key, deactivating the tool without further mouse click or keypress necessary, making it very easy to make further selections.

Input Mapping

In Modo, the primary input device is the mouse. To optimize its use and to create a consistent workflow, there are a number of rules for using the mouse depending on the action that you are working with. Let's begin by looking at the mouse behaviors when using a primitive tool:

- Click and drag - adjust handles (continue)
- **Shift**+click and drag - adjust handles (reset)
- Right-click and drag - haul alternate parameters
- Middle-click+haul - adjust handles (clone previous)

Transform tools adjust existing mesh elements, and typically have parameters that are hauled (clicking and dragging away from the tool handles with the tool active), or optionally set with handles. This includes all the basic transforms, most point pushers, and many Vertex Map tools:

- Click and drag - haul parameters (reset)
- **Shift**+click and drag - haul parameters (continue)
- Right-click+haul - falloff setup parameters (if any)
- Middle-click+click drag - adjust parameters (clone previous)

Non-transform edit tools adjust existing mesh elements, but they typically don't want to reset as their default mouse-down behavior. These may still work with falloffs. An important exception is when a tool of this type is used with the screen falloff, in which case they act like transforms with reset as the default:

- Click+haul - parameters and handles (continue)
- **Shift**+click and drag - haul parameters (reset)
- Right-click+haul - falloff setup parameters (or alt parameters)
- Middle-click+click drag - adjust parameters (clone previous)

The airbrush falloff tool is a special case when used with any other tool, transform, or otherwise.

- Click and drag - paint weights (continue)
- **Shift**+click and drag - paint inverse weights
- **Ctrl**+click and drag - erase weights
- Right-click+haul - adjust airbrush size
- Middle-click+haul - parameters of action tool

Apply

When the Tool pipe is inactive, you can press the **Apply** button to perform the action of the Tool pipe with its current settings. This does not activate the pipe, and the **Apply** button is disabled when the pipe is active. The **Apply** Button, when available, appears on the tool's **Properties** panel. You can also right-click on the Tool Pipe entry and choose **Apply Tool**.

Reset Tool Attributes

When using a tool interactively or when using the numeric input fields for a tool, you can reset the values that have changed to defaults by choosing **Edit** > **Reset Tool Attributes**, or press **CTRL**+**D** with the tool active for quick keyboard access.

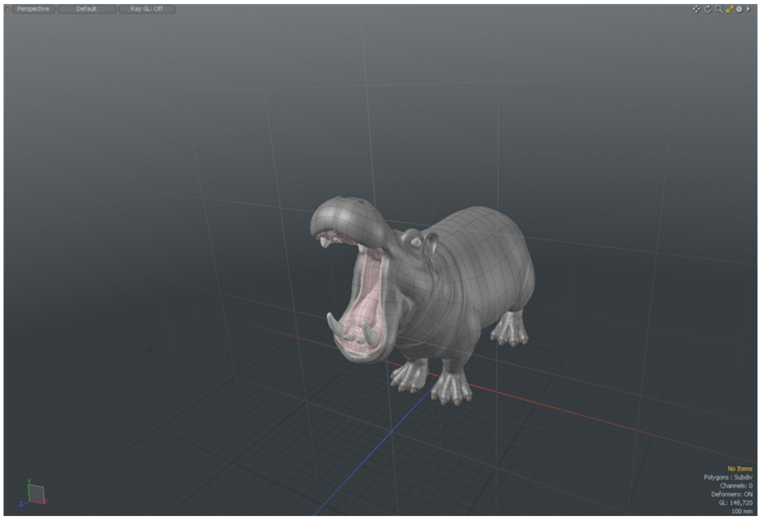
Keyboard Access

Press **K** to open the tool properties popover form.

Using the Work Plane

Using a flat screen to create and edit content in 3D space can sometimes feel awkward and unnatural. Software has tried to resolve the disconnect using different solutions over the years.

The most popular is the four viewport method. Each orthogonal viewport only provides two axes at any given time, eliminating surprises as you create and modify geometry in a given scene. Another solution is working with tool handles constraining operations performed in a perspective viewport to the same flat 2D axes of the four viewport method. Both of these solutions are reasonable and available in Modo. However, Modo also employs a third method called the Work Plane.



The Work Plane is a white grid that appears in the 3D viewports. It is an adaptive modeling aid that automatically adjusts itself to match the two major axes most closely aligned to your current screen axis, represented by a white grid in any of the 3D viewports. When you rotate the **Perspective** viewport, the Work Plane snaps to one of the dominant planes, XY, XZ, or XY. Action Centers often use the Work Plane position to determine the center and axis of tools when activated. When hauling, many tools also use the Work Plane to determine the axes of operation when this information cannot be derived from a specific tool handle.

While the default mode for the Work Plane is to automatically adjust to the screen's orientation, it can also be locked and used as a construction plane. When applied as such, the Work Plane effectively positions the entire Modo universe to that fixed angle and position. There are a number of methods for manually setting the Work Plane to a specific center and axis. The Work Plane can be easily locked to a specific major axis and position, or you can snap the Work Plane to selected geometry or even just to the polygons directly under the mouse. This level of flexibility and control provides you with a fluid and rich set of options for editing your workspace.

Mastering the Work Plane allows you to do most, if not all, of your modeling in a single perspective viewport. That means more screen space remains for viewing the geometry and performing the various modeling tasks, allowing you to work more fluidly and accurately. Experiment with the Work Plane when using tools and hauling, so you can get a better grasp of its behaviors and use it to your advantage.

To apply the Work Plane:

1. In the 3D viewport, make your selection.
2. Under the interface layout tabs, click **Work Plane**.

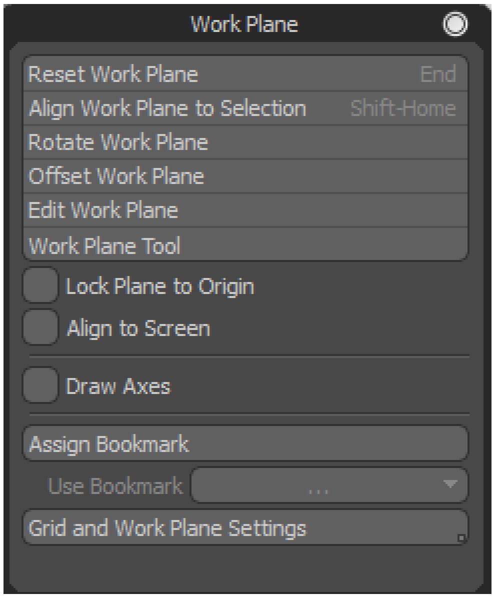
   The Work Plane is immediately aligned to what you had selected in the 3D viewport. The Work Plane button is now displayed in orange to indicate that it is active.

Tip: You can quickly offset the position of the Work Plane using a keyboard shortcut. Hover the mouse over some geometry in the scene where the Work Plane should move to, then press **Alt**+**O**. The Work Plane retains its orientation but moves to the location where the mouse pointer intersects with the geometric element.

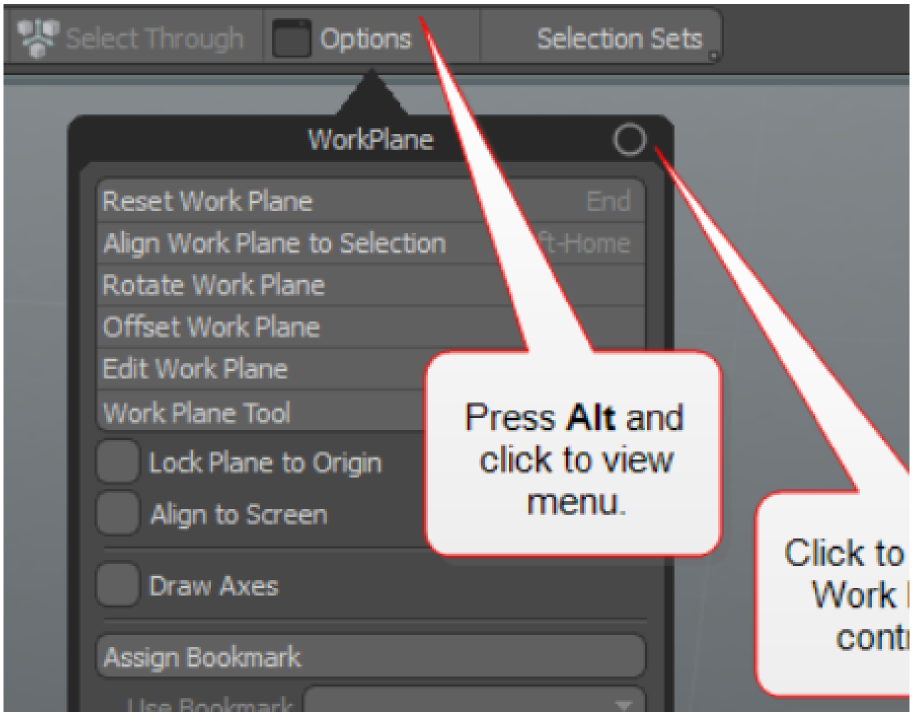
Work Plane Controls

The **Work Plane** is a persistent function of the 3D viewports, and is visible by default in the menu below the interface layout tabs. You can toggle its display under the **Visibility** tab of the **3D viewport Properties** panel. To open it, hover the mouse over a viewport and press the **O** key. Additionally, you can toggle the Work Plane visibility by pressing the * (asterisk) key on the number pad.

The controls for the Work Plane are displayed by pressing **Alt** + **Work Plane**, or by clicking **Edit** > **Work Plane**. You can also use the **Home** key to dynamically position the Work Plane to any geometry directly under the mouse pointer, centering under where the pointer tip intersects the polygon's surface and aligning it to the polygon's normal direction.


Work Plane
Reset Work Plane
End
Align Work Plane to Selection
Shift-Home
Rotate Work Plane
Offset Work Plane
Edit Work Plane
Work Plane Tool
Lock Plane to Origin
Align to Screen
Draw Axes
Assign Bookmark
Use Bookmark
...
Grid and Work Plane Settings

Tip: To pin the Work Plane control options dialog box, hold down the **Alt** key + click **WorkPlane** and then click the circle on the top right corner. To reposition the dialog box, click-and-drag it to a new position on the viewport.


Select Through
Options
Selection Sets
WorkPlane
Reset Work Plane End
Align Work Plane to Selection ft-Home
Rotate Work Plane
Offset Work Plane
Edit Work Plane
Work Plane Tool
Lock Plane to Origin
Align to Screen
Draw Axes
Assign Bookmark
Use Bookmark

Press **Alt** and click to view menu.

Click to Work cont

Reset Work Plane

This command returns the Work Plane to its default behavior. It only applies if the Work Plane has been aligned as a fixed construction plane.

To reset the Work Plane:

- Under the interface layout tabs, click **Work Plane**.

  The Work Plane is reset back to its original default position and its button is now displayed in gray to indicate that it is not active.

Align Work Plane to Selection

Aligns the Work Plane to the current component selection as described below. If nothing is selected, the Work Plane is aligned to the average normal of all foreground polygons.

To activate the command, click **Edit** > **Work Plane** > **Align to Selection**. Alternatively, **Alt**+**click** on the **Work Plane** button on the menu bar and select **Align Work Plane to Selection**.

When selecting vertices:

- If a single vertex is selected, only the center of the Work Plane is changed.
- If two vertices are selected, the first vertex sets the center and the Work Plane is altered to align the Z axis with the second vertex.

- If three vertices are selected, it uses the first vertex to set the center, the second to set the Z axis and the third defines the XZ plane.
- If more than three vertices are selected, the center is located at the average of their positions, and the plane is angled to put all vertices as close as possible to the XZ plane.

When selecting edges:

- If a single edge is selected, it is treated the same way as two vertices using the ends of the edge (above).
- If two edges are selected, they are treated the same as three vertices (above).
- If three or more edges are selected, this results in the same solution as vertices, using the end points of the edges as the vertices

When selecting polygons:

- If a single polygon is selected, the Work Plane is centered on that polygon with the Y axis aligned to the polygon normal. The longest edge of the polygon is along the Z axis.
- If multiple polygons are selected, the average normal is along the Y axis, and the system uses a best guess to determine correct orientation.

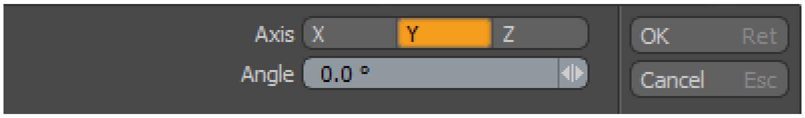
Tip: Right mouse click over any Item in the 3D Viewport and select **Set Work Plane** from the contextual menu to align the work plane to the item center.

Rotate Work Plane

The **Rotate Work Plane** function provides a way to accurately rotate the Work Plane a defined number of degrees away from its current position.

To activate the command, click **Edit** > **Work Plane** > **Rotate**. Alternatively, **Alt**+**click** on the **Work Plane** button on the menu bar and select **Rotate Work Plane**.

Invoking the command opens the following dialog.


Axis X Y Z
Angle 0.0 °

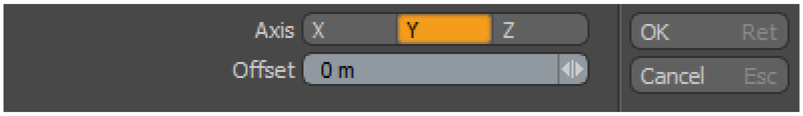
Axis	This option defines around which axis the Work Plane rotates, originating at the Work Plane origin (0,0,0) position.
Angle	Defines the number of degrees to rotate the Work Plane away from its home position.

Offset Work Plane

The **Offset Work Plane** function provides you with a means to accurately position the Work Plane by way of an offset from its current position.

To activate the command, click **Edit** > **Work Plane** > **Offset**. Alternatively, **Alt**+**click** on the **Work Plane** button on the menu bar and select **Offset Work Plane**.

Invoking the command opens this dialog.



Axis: Y

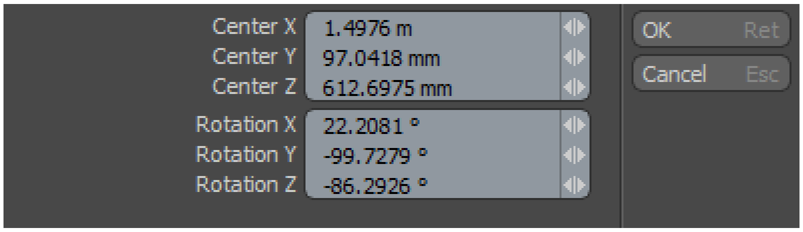
Offset: 0 m

Axis	This option defines around which axis the Work Plane offsets its position along.
Offset	Defines the distance to offset the Work Plane away from its current position.

Edit Work Plane

Displays the current values of the Work Plane as offset from the true World Origin position at 0,0,0. You can edit the values to precisely position and rotate the Work Plane.

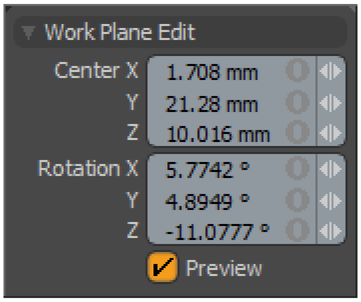
To activate the command, click **Edit** > **Work Plane** > **Edit**. Alternatively, **Alt**+**click** on the **Work Plane** button on the menu bar and select **Edit Work Plane**.



Center X: 1.4976 m
Center Y: 97.0418 mm
Center Z: 612.6975 mm
Rotation X: 22.2081 °
Rotation Y: -99.7279 °
Rotation Z: -86.2926 °

Work Plane Tool

The Work Plane Tool allows for interactive placement of the Work Plane itself. When combined with Snapping, the Work Plane can be placed very accurately. The Snapping tool is found in the menu bar below the interface layout tabs. Click in the 3D viewport to draw the tool handles that allow for interactive placement. The initial click places the handles at the intersection of the mouse click position and the existing Work Plane. You see the standard 3-point axes handles colored accordingly, with a small square at the center and two more at the ends of the axes pointers. The initial square at the center defines the Work Plane's center and the other two work together with the center to define a plane.



Work Plane Edit

Center X 1.708 mm
Y 21.28 mm
Z 10.016 mm
Rotation X 5.7742 °
Y 4.8949 °
Z -11.0777 °
☑ Preview

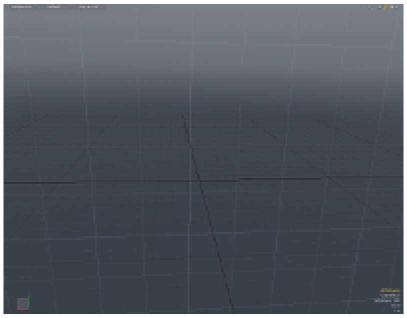
To use it, activate the tool, and then ensure that the **Preview** option is enabled in the tool's **Properties** panel. Then click in the 3D viewport, use the resulting tool handles to place the center and define the plane by clicking and dragging them in to the proper position. Numeric values on the tool's **Properties** panel can also be modifies to adjust the plane manually. Disabling the **Preview** option with the tool still active places the Work Plane, locking it to the defined location. Press the **spacebar** to drop the tool, allowing further editing of the scene with the newly defined Work Plane.

Lock Plane to Origin

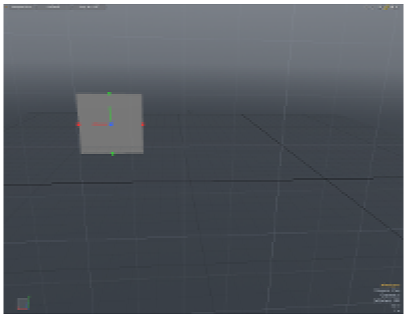
The **Lock Plane to Origin** toggle forces the Work Plane to intersect the World Origin position at 0,0,0. The Work Plane still aligns to the facing plane, and it may appear that it has shifted away from the origin. However, when you draw an object, it aligns to the Work Plane at zero distance from the World Origin along the axis that is perpendicular to the Work Plane.

To activate this command, **Alt**+**click** on the **Work Plane** button on the menu bar and enable the check box for **Lock Plane to Origin**.

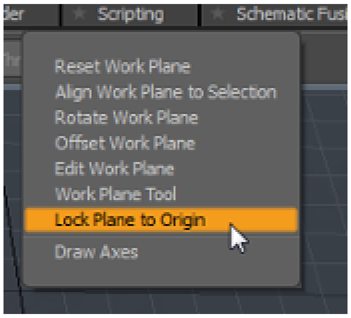
For example, if the Work Plane appears to be aligned to the XY plane and you draw a primitive, the X,Y coordinates are determined as normal, by mouse position. However, the primitive origin is placed at Z=0. In this case, the Z-axis is perpendicular to the Work Plane, and the Work Plane is positioned at Z=0 to intersect the World Origin. The images below illustrate this process.



1. The Work Plane is aligned to the



2. A flat cube primitive is drawn.

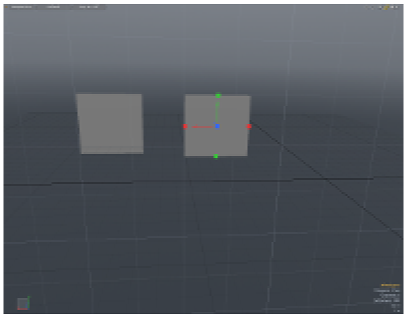


3. **Lock Plane to Origin** is

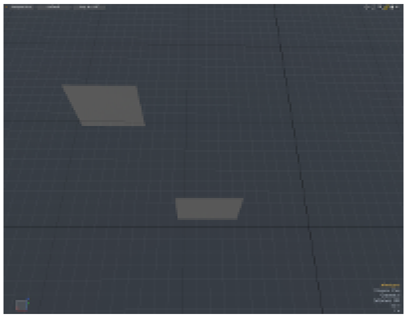
XY plane, a short distance behind Z=0.

(**Lock Plane to Origin** is off.)

activated.



4. A second flat cube primitive is drawn.



5. Looking overhead, notice that the second primitive is located at Z=0.

Tip: To help draw primitives on the ground (XZ) plane, toggle **Lock Plane to Origin** on, and rotate the viewport so that the Work Plane aligns to the ground plane. If the primitive has a flat base, such as a cube or cylinder, draw the flat object and then extrude the height. This locates the object on the ground plane. For objects that are initially drawn in 3D, or objects with a rounded base, you need to adjust the Y-position to sit the object on the ground plane.

Align to Screen

The **Align to Screen** option sets the default Work Plane to the screen direction instead of a major axis plane. Primitive tools, such as Pen and Sketch, stroke point positions on a major axis plane when there is no custom Work Plane set. **Align to Screen** unlocks the default Work Plane from a major axis to allow the stroke points on exact screen axis plane. This is useful when you want to draw curves using Sketch on the current screen space without setting custom Work Plane.

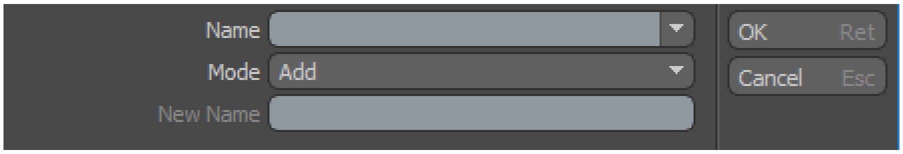
Draw Axes

This toggle enables and disables the drawing of an axis handle widget in the viewport at the origin position of the Work Plane, making it easier to recognize the center and axis directions.

Assign Bookmark

Work Plane bookmarks can be assigned to polygons and re-used later on. Different bookmarks can be saved for each of your scenes. This is useful to store construction planes in the geometry itself. To reset your selected polygons to the origin, select **Work Plane** > **Reset Work Plane**.

Tip: The 3D viewport item contextual menu also has an option to set the Work Plane on the item, making it easier to model in item space. Right-click on a mesh item and select **Set Workplane**. An item Work Plane aligns to meshes that have been moved away from the world origin in **Items** selection mode. It aligns the Work Plane to an item's center locator.



The following options are available:

- **Name** - Apply a name for a new bookmark or select an existing bookmark and then select a **Mode**.
- **Mode** - Select one of the following operations to apply.
  - **Add** - To add a new bookmark.
  - **Remove** - To remove an existing bookmark.
  - **Rename** - To rename an existing bookmark.
  - **Set** - To set the current view as a bookmark.
- **New Name** - Rename a bookmark.

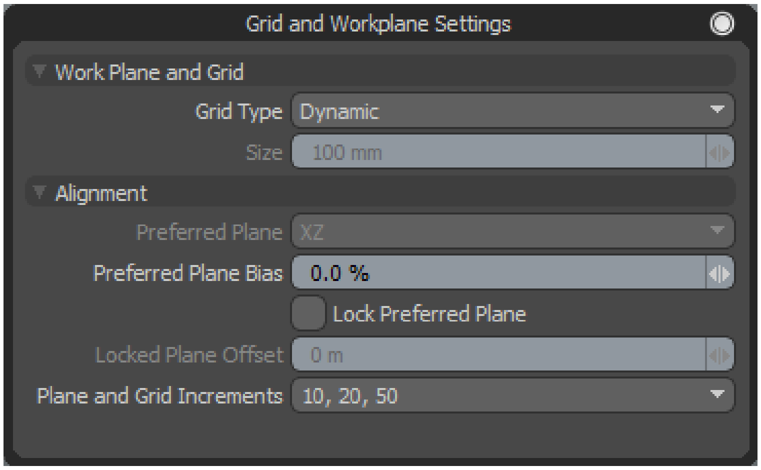
Use Bookmark

A list of previously created bookmarks, made for a particular scene, are listed. By selecting a bookmark, the polygon(s) assigned to that bookmark are fitted to the specified Work Plane.

Grid and Workplane Settings

This **Work Plane** option allows you to specify the grid and Work Plane settings. All the settings are applied globally to all viewports and are stored inside a configuration file found in the application menu **System** > **Open User Configs Folder**.

Note: The **Snapping** option no longer controls the grid display. The Snapping tool simply allows you to accurately create and align elements in relation to one another, or in relation to 3D space. For more information, see Applying Snapping.



Grid and Workplane Settings

Work Plane and Grid

Grid Type: Dynamic

Size: 100 mm

Alignment

Preferred Plane: XZ

Preferred Plane Bias: 0.0 %

☐ Lock Preferred Plane

Locked Plane Offset: 0 m

Plane and Grid Increments: 10, 20, 50

Work Plane and Grid	
Grid Type	An option for selecting which type of grid to display in the viewport. The following options are available: • **Fixed** - The grid is displayed in regular intervals. • **Dynamic** - The grid is dynamic depending on the zoom level, a small display in the lower right corner gives you the real world equivalent measurements for each square unit.
Size	This option is enabled when the **Grid Type** is set to **Fixed**, allowing you to specify the size of the grid squares displayed in the viewport. When zooming out, grid lines can merge together when they become too small. When this occurs, Modo converts fixed grids to dynamic grids.
Alignment	
Preferred Plane	When the **Preferred Plane Bias** option is set above 0%, you can choose a preferred Work Plane. Based on the bias amount, Modo attempts to keep the work plane positioned at the preferred plane determined by its two major axes.
Preferred Plane Bias	When working with the Work Plane, Modo tries to keep the plane perpendicular to the viewport window when navigating and rotating the view. The higher the value set, the more the **Preferred Plane** setting is displayed.
Lock Preferred Plan	You can lock the Work Plane to a specific plane-facing direction by

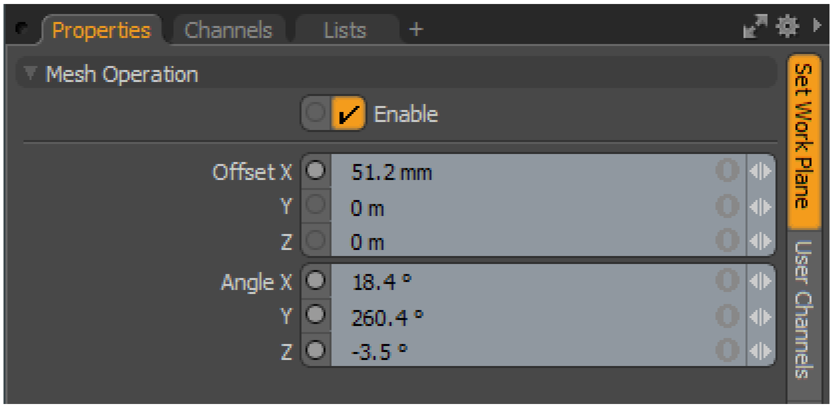
	enabling this option. Once fixed, rotating the viewport does not affect the position or angle of the Work Plane.
Locked Plane Offset	When the Work Plane is locked, you can additionally set an offset from the origin, providing accurate control over the position and placement of the Work Plane when used as a construction plane.
Plane and Grid Increments	As you zoom in and out of a scene, the Work Plane grid adjusts the density of the grid divisions, adding or removing subdivisions based on the plane increments. You can modify this behavior using this option.

Procedural Set Work Plane

Modo includes a procedural sub-tool to position and orient the Work Plane, called **Set Work Plane**. This can be added to the **Tool Pipe** input of any procedural tool that supports a Work Plane to transform its operation. Click the tool's **(Add Tool Pipe)** link to open the Procedural Preset Browser. **Set Work Plane** is in the **Procedural** > **Tool Pipe** > **Sub-tools** category.

If connected to a primitive tool, rotating and moving the Work Plane rotates and moves the primitive.

Procedural Set Work Plane Properties



☑ Enable

Offset X 51.2 mm
Y 0 m
Z 0 m
Angle X 18.4 °
Y 260.4 °
Z -3.5 °

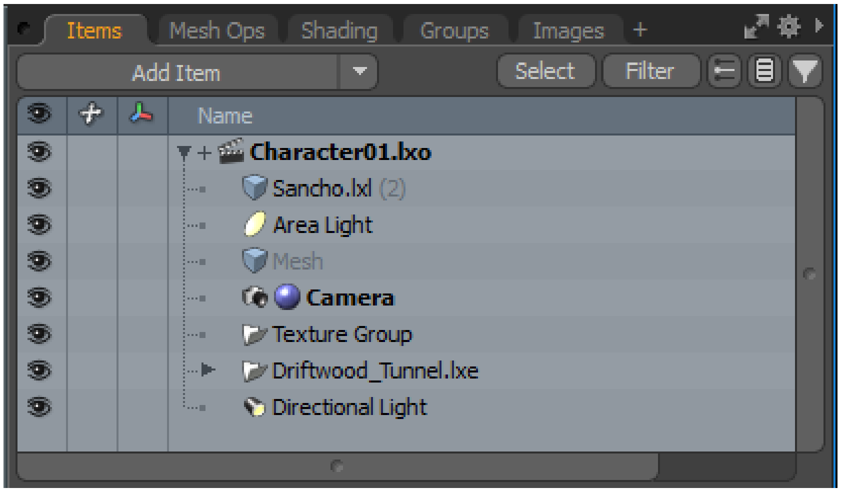
Set Work Plane	
Enable	Toggles the tool on or off.

Set Work Plane	
Offset X, Y, Z	Sets the Work Plane position.
Angle X, Y, Z	Sets the rotation of the Work Plane on each axis.

Understanding Viewports

The Modo interface is made up of a collection of frames called viewports.

A viewport is an empty container whose content is determined by the options of the viewport **Widget** (see Using Layout Controls). Selecting the **3D Model View** option changes a viewport to a 3D openGL view with its own functions and settings, but it is still just a viewport. From the Widget, you can subsequently select other options, such as **Shading**, and the 3D view now becomes a **Shading** Viewport, which is the identical size as the previous 3D view. This is because you control the sizes of viewports by clicking and dragging on their edges (the small dividers between the frames). Viewports are dynamic and react to the size of the frame they exist in.



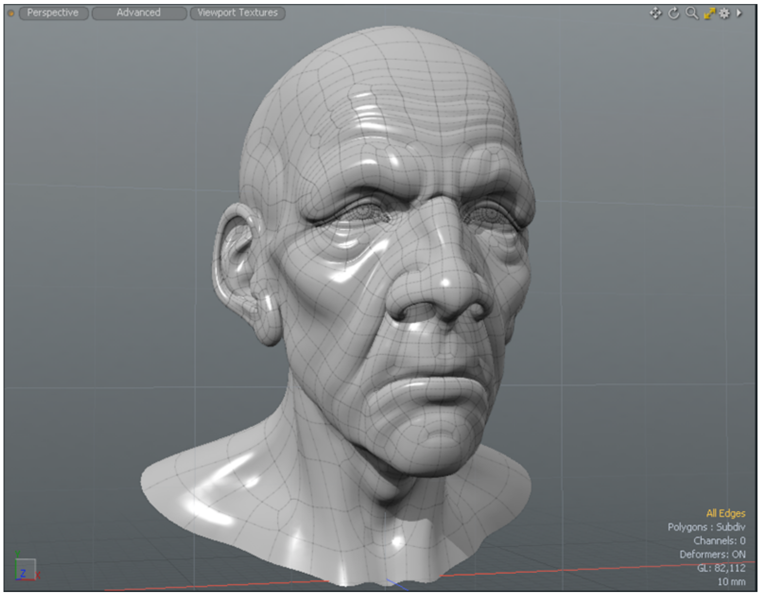
Modo uses several different viewport types, **Forms** are the most common. **Forms** are generalized containers that can be either a button, a toolbox, or a menu. The content of these viewports can be modified through the **Form Editor**. The next most common are the GL Viewports, which display 3D content (named as such because they use the prominent open source graphic library named OpenGL). These viewports cannot be modified outside of the customization option available within the viewport itself (and the **Preferences** settings). Other viewport categories include **Data Lists** which contain lists of information regarding various Modo items, these include the **Items List** and the **Shader Tree**. You interact with the various items in these views, but associated attributes for specific items are viewed in another frame called a **Property** viewport. Property views are available for various functions including Items, Render Settings and Tool Properties.

Viewports don't have to exist within the frame of the interface itself either, Modo provides both Window and Palette viewports that float above the standard interface. These windows work in the same fashion as normal embedded frames, but their visibility can be toggled on and off with the ` (backtick). The only difference between a Window and Palette is that the windows retain the operating systems controls to individually move and minimize the viewports.

Viewports can be visualized like children's blocks, they can be stacked and arranged any way you see fit, but better than blocks, they can also be scaled, divided, and removed providing the most adaptable and customizable interface of any 3D application.

3D (OpenGL) Viewport

The 3D viewport is the most frequently used interface in Modo. It is the window into Modo's virtual 3D world. It serves many purposes, allowing you to sculpt, paint, animate, deform, move elements about, and position cameras and lights. With so many jobs to do, it's important that it be flexible as well. There are numerous functions within the 3D viewport that allow you to customize what is seen and how it is seen, depending on the task at hand.



When Modo is opened for the first time, it opens to the default **Model** layout with a large 3D viewport set to **Perspective** view. When examining the viewport, you see a dark grid moving in perspective off into the distance some, with another coarser light colored grid perpendicular to the dark one.

Grid Displays

The dark grid represents the ground plane, the zero position of the Y (or Up) axis. Each square of that grid represents a fixed distance. Since this grid is dynamic depending on the zoom level, a small display in the lower right corner gives you the real world equivalent measurements for each square unit, in the case above it is 50mm.

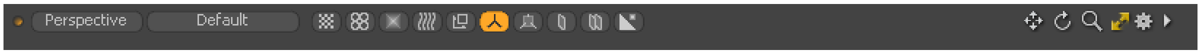
The lighter grid represents Using the Work Plane, which is also dynamic depending on your rotational orientation to the world. While the ground plane is purely for reference purposes, the Work Plane's purpose is to show you where objects get created in the 3D space. Think of it as a construction plane, dynamically adjusting to the most appropriate angle.

Tip: Right mouse click over any Item in the 3D Viewport and select **Set Work Plane** from the contextual menu to align the work plane to the item center.

Also within the 3D viewport, there are text and information overlays that give you pertinent information with regards to items visible within the viewport as well as guides to selections and the currently selected tool. All of these work together, providing important feedback to speed and enhance the overall workflow.

Additional Controls

To get the most of the 3D views, there are also many additional controls you can directly access when necessary. In the upper left corner of the window is a small circle, called the **Thumb**. Next to it, there are three capsule shaped buttons. The **Thumb** is dark gray until a viewport becomes active in which case it turns orange. To make any viewport active, simply click anywhere within the window frame. Right-clicking on the **Thumb** presents you with a series of options regarding the frame. See Using Layout Controls for further details regarding the options.



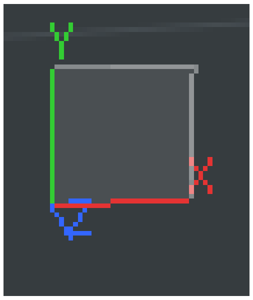
The left-most capsule button next to the **Thumb** is for choosing the way the viewport sees the scene (see Tool HUD). Click on it to open a pop-up menu that allows you to choose from the available options of Perspective and various isometric views, or viewing the scene from the Camera or Lights point of view or **Ray GL** options giving you a fast, fully rendered in-viewport representation of the scene. The **Viewport Textures** icon allows you to quickly select **Environments**, **Matcaps**, and **UV Textures** to display in the 3D Viewport without having to add them to the Shader Tree. For more information, see Viewport Textures.

The upper right corner has several small icons. The first three let you manipulate the view itself, move, rotate and adjust the zoom level of the active viewport. Clicking and dragging over any of the icons makes the appropriate view adjustment providing an intuitive way for manipulating the position of the viewport's virtual camera. For easier and faster viewport navigation, see Navigation. The next one is the **Minimize/Maximize** arrow. Click to expand the active viewport to fill the entire local frame, allowing for additional viewport space to manipulate the contents. When the arrow is yellow, click it to minimize the viewport and reveal the other associated viewports.

The gear contains all the available viewport customization control options. Here you can change how the geometry is represented in the viewport, control wireframe overlays and change how items in the background are displayed (items that are visible, but unselected are considered background elements), as well as set the visibility of many individual viewport elements. These options can also be accessed by hovering the mouse pointer over the target window and pressing **O** on the keyboard. For more information on viewport customization, see below.

The tiny arrow icon to the right, is called the **Widget**. For more information on this control, see Using Layout Controls. Right-click to open a pop-up menu that allows you to select one of the Modo viewports. Selecting any of these items converts that window into the chosen viewport.

Note: Don't select anything unless you mean to change it, as viewport changes cannot be undone. See Customizing Your Layout for more information regarding viewport customization (and how to get back the default interface when you need to).

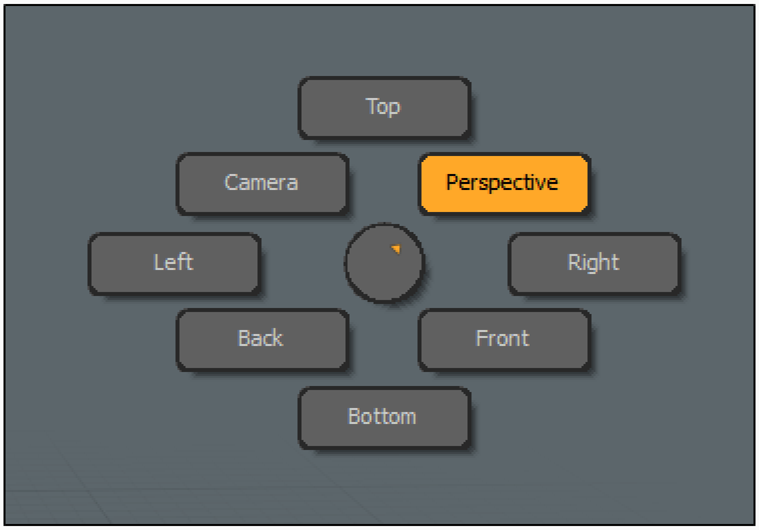


On the lower left of the viewport, there is an XYZ axis gizmo that provides a visual reference to the orientation of the window as well as the current work plane orientation. For orthographic views, the icon displays the axis plane the viewport windows faces toward. In perspective view, you can rotate the view and see the widget update in real time. The lower right corner has several informational text displays that dynamically update, showing the current number of selected items, channels and deformers. The GL display references the number of live polygons active in your viewport (including Sub-D surfaces) and the number at the bottom gives the real world size reference to the squares within the grid of the ground plane. The specific information displayed is controlled in the GL section of the Display Preferences.

Navigation

Modo provides a number of functions to aid in the navigation of GL viewport allowing you to navigate in 3D space in various ways. Most are mouse button and keyboard combinations. Shortcuts are based on the default interface values. For additional navigation options, see Remapping in Input Preferences.

Keystroke(s)	**Action**
Alt+left-click	Rotates view
Alt+middle-click	Rotates viewport Z axis (bank) view
Alt+right-click	Flick Rotate (also called Free Wheeling) view slows down when mouse is released.
Shift+**Alt**+left-click	Pans View
Shift+**Alt**+right-click	Up/Down Pan only
Ctrl+**Alt**+left-click	Zooms to mouse position
Ctrl+**Alt**+right-click	Box Zoom (drag box to zoom)
. (period)	Zooms In
, (comma)	Zooms Out
G	Centers view (Current mouse position to viewport center)
/	Turns (Spinning view of scene)
Shift+**/**	Turns upright
Mouse Wheel Scroll	Zooms view in & out to mouse pointer
A	Fits (Zoom to scene extents/all Items)
Shift+**A**	Fits selected (Zooms to selected elements)
Ctrl+**A**	Aligns selected (Aligns view to current selection)
Ctrl+**Shift**+**A**	Fits and aligns selected (Zooms and aligns view to current selection)
Ctrl+ **Spacebar**	Displays a pie menu. Hover your cursor over any of the options and release to select a view.

Keystroke(s)	Action
	 Top Camera Perspective Left Right Back Front Bottom

Visibility

Modo provides a way to modify the visibility of object in the viewport directly. While generally you can toggle visibility of items in the Items (Scenes) List that changes visibility of the entire layer. The following commands allow the temporary toggling of visibility of only the selected element, including components, depending on the selection mode. Once hidden, geometry becomes uneditable. To hide a piece of geometry, select the geometry component you wish to hide and in the menu bar, go to **View** > **Hide Selected** or press **H** on the keyboard. To make the geometry visible again, click **View** > **Unhide** or press the **U** key.

Note: When a file is saved and re-opened later, all hidden polygons are visible.

Keystroke(s)	Action
H	Hides selected geometry (hides all if nothing is selected).
Shift+**H**	Hides unselected geometry.
Ctrl+**H**	Hides invert (toggles visibility, inverting the present state).
U	Makes all hidden geometry visible for the currently selected layers.

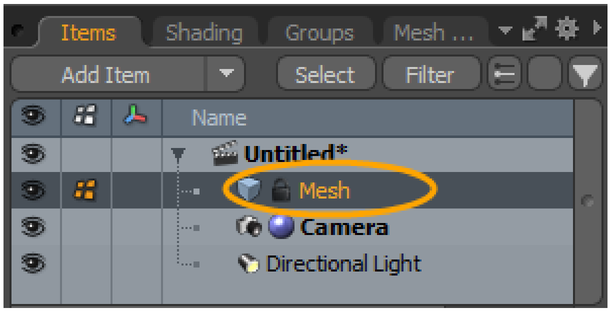
Locking

Providing a similar function to hiding, locking allows you to fix a selection of geometry so that it is not editable, however it is still visible.

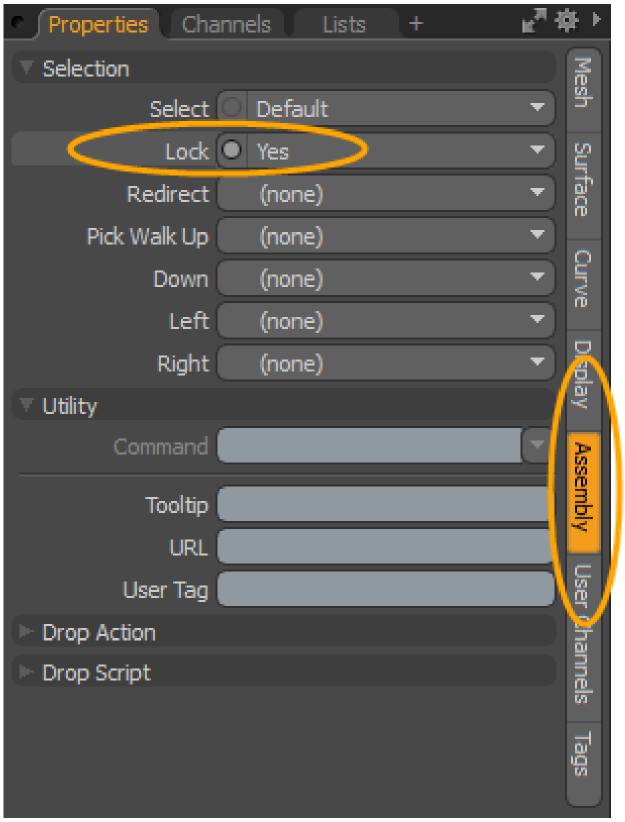
To lock item layers:

1. On the right panel, open the Items (Scenes) List, and right-click the item you want to lock.
2. Select **Lock/Unlock** from the dropdown menu.

   A lock icon appears on the mesh item in the **Items** list.


Items
Shading
Groups
Mesh ...
Add Item
Select
Filter
Name
Untitled*
Mesh
Camera
Directional Light

- Alternatively, in the **Properties** tab of the selected mesh item, open the **Assembly** tab, and set **Lock** to **Yes**. For more information, see Assembly Viewport.


Properties
Channels
Lists
+
Selection
Utility
Drop Action
Drop Script
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

Select: Default
Lock: Yes
Redirect: (none)
Pick Walk Up: (none)
Down: (none)
Left: (none)
Right: (none)
Command
Tooltip
URL
User Tag

You can unlock a locked element by right-clicking on the item and selecting **Lock/Unlock**. The lock icon is removed when unlocked.

While working in the 3D viewport, you can lock selected components to assist you in your workflow. Modo displays locked polygons, vertices, and edges with a specific color in the 3D viewport.

Note: In edge selection mode, Modo also displays locked vertices since the locked edge elements are associated with vertex or polygon elements.

To lock a component selection of geometry in the 3D viewport:

3. In the 3D viewport, select the geometry you wish to lock using one of the selection modes (Vertices, Edges, Polygons, or Items).

   For more information about selecting geometry, see Selecting Items.

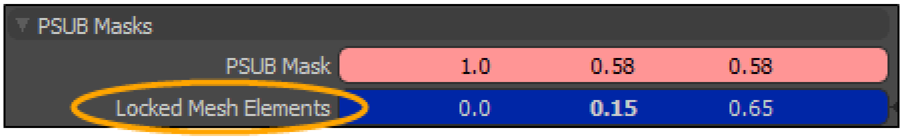
4. In the menu bar, click **Edit** > **Lock Selected** or press **J**.

Tip: You can unlock any locked elements in the 3D viewport by clicking **Edit** > **Unlock** or by pressing **I**.

Lock Key Shortcuts	Action
J	Locks selected geometry (locks all if nothing is selected).
Shift+**J**	Locks unselected geometry.
Ctrl+**J**	Locks invert (toggles locked state of all geometry).
I	Makes all locked geometry editable.

To change the color of locked elements:

1. In the menu bar, click **System** > **Preferences**, expand **Display**, and click **Colors**.
2. Under **PSUB Masks**, set the **Locked Mesh Elements** color by clicking-and-dragging the values in the field or click on the field to open the Color Picker to select your color. For more information, see Color Picker Viewport.



PSUB Masks

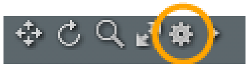
PSUB Mask 1.0 0.58 0.58

Locked Mesh Elements 0.0 0.15 0.65

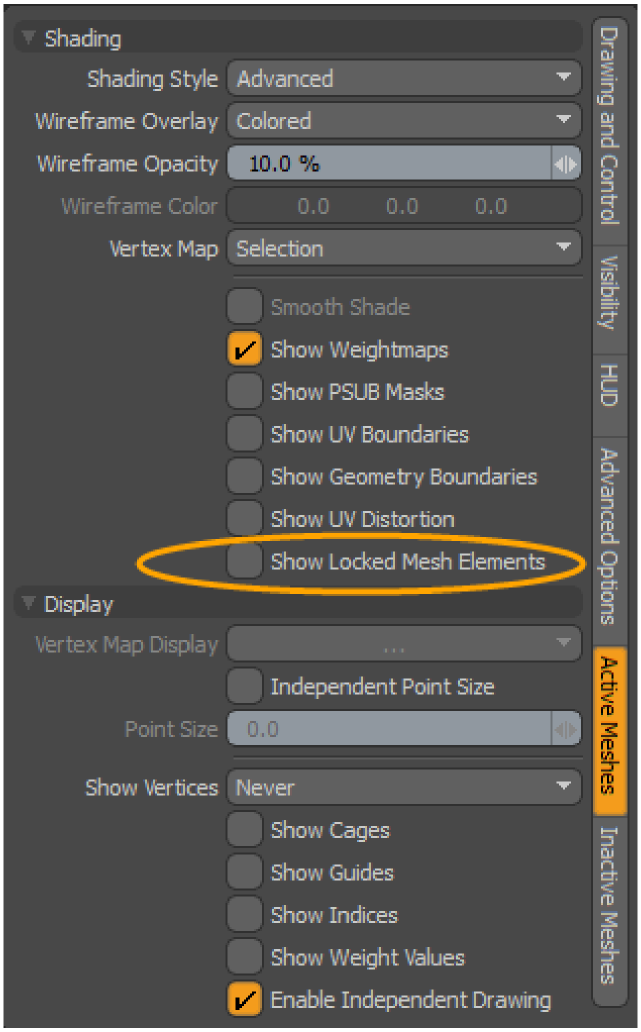
To disable **Show Locked Mesh Items**:

By default, Modo displays locked mesh items in the viewport. You can disable this in the viewport properties.

1. In the top-right corner of the 3D viewport, click the gear icon to display the properties.



2. Open the **Active Meshes** tab, expand the **Shading** section, and uncheck **Show Locked Mesh Items**.



Shading

Shading Style: Advanced

Wireframe Overlay: Colored

Wireframe Opacity: 10.0 %

Wireframe Color: 0.0 0.0 0.0

Vertex Map: Selection

☐ Smooth Shade

☑ Show Weightmaps

☐ Show PSUB Masks

☐ Show UV Boundaries

☐ Show Geometry Boundaries

☐ Show UV Distortion

☐ Show Locked Mesh Elements

Display

Vertex Map Display: ...

☐ Independent Point Size

Point Size: 0.0

Show Vertices: Never

☐ Show Cages

☐ Show Guides

☐ Show Indices

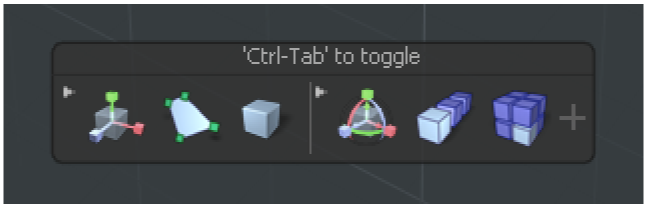
☐ Show Weight Values

☑ Enable Independent Drawing

Tool HUD

The Tool HUD is available in every layout of the 3D viewport. It is a context-sensitive form containing the most recent tools for each layout. It also displays different sets of tools depending on the active selection mode.

To enable the Tool HUD press **Ctrl**/**Cmd**+**Tab** on the keyboard.

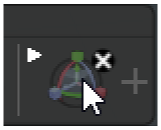


When you first open the Tool HUD , it displays a set of custom tools for the particular layout and selection mode. As you keep using various tools, they are added to the Tool HUD's most recent tools section on the left.

When you click a tool in the Tool HUD, the tool becomes active and its icon is highlighted in orange until you drop it.

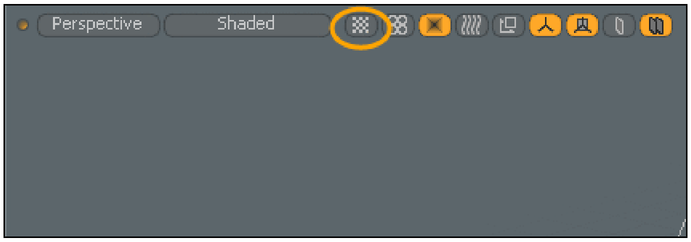
You can add new tools to the Tool HUD by activating the tool, then clicking the **+** button on the right.

To remove a tool, click **x** in the upper-right corner of the tool's icon in the Tool HUD.



You can customize the Tool HUD in the **3D Viewport Properties**, in the **HUD** sub-tab. For more information, see 3D Viewport Properties.

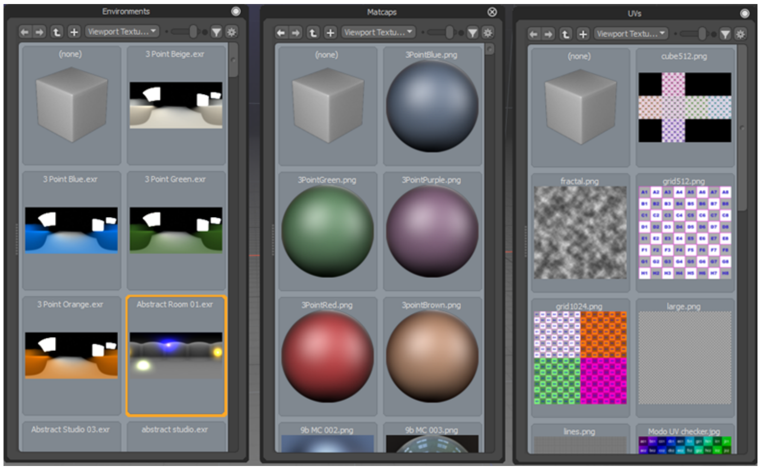
Viewport Textures



Predefined textures can be applied to your scene to quickly see a preview in the 3D Viewport using any of the 3D Viewport Styles. A variety of environments, MatCaps shaders, and UV textures are available. When you double-click on a texture, your scene is updated in the 3D Viewport without having to add any items in the Shader Tree. Using this feature enables you to experiment with different textures before actually creating Shader Tree layers. For more information, see Shader Tree Item Layers.

Note: Items that you have set in the **Shader Tree** are not overwritten. To view your set **Shader Tree** items, click **Render** > **Open Preview** and click the arrow to render your scene.

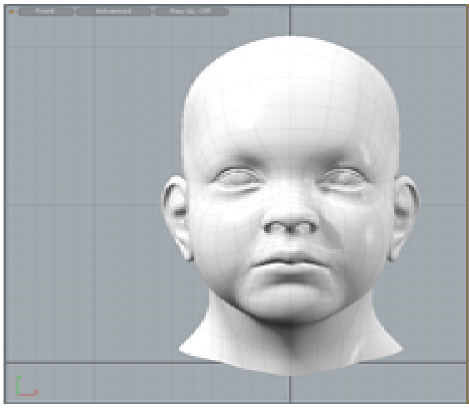
The following options are available:


Environments
Viewport Textu...
(none)
3 Point Beige.exr
3 Point Blue.exr
3 Point Green.exr
3 Point Orange.exr
Abstract Room 01.exr
Abstract Studio 03.exr
abstract studio.exr
Matcaps
Viewport Textu...
(none)
3PointBlue.png
3PointGreen.png
3PointPurple.png
3PointRed.png
3pointBrown.png
9b MC 002.png
9b MC 003.png
UVs
Viewport Textu...
(none)
cube512.png
fractal.png
grid512.png
grid1024.png
large.png
lines.png
Modo UV checker.jpg

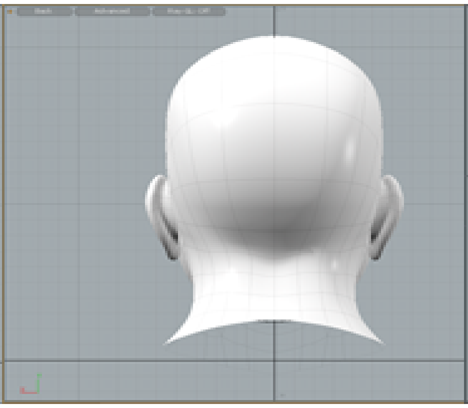
Option	Description
Environments	When an environment texture is set, Modo replaces the Environment item in the **Shader Tree**.
Matcaps	When a MatCap shader is set, Modo completely replaces the surface shader used in the **Shader Tree** with the selected MatCap shader.
UV Textures	When a UV texture is set, Modo replaces the Diffuse texture applied on the mesh surface and uses a default material. A number of provided UV Textures are used for UV Diagnostics. **Note:** When applying UV Textures to a Preset mesh, the UV Map name for the UV Texture must be the same as the UV Map name used by the Preset. For more information, see Preset Browser. For more information about renaming UV Maps, see Working with UV Maps.

View Types

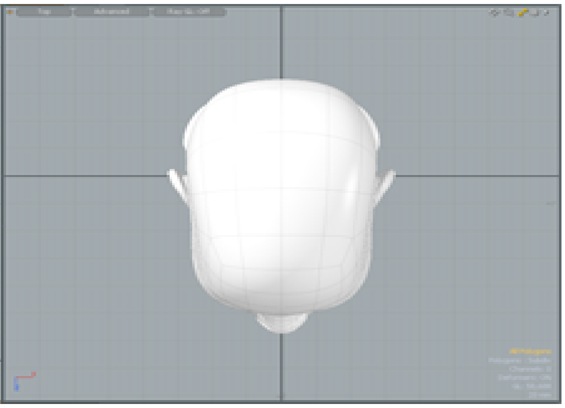
While viewports have no actual camera, Modo does provide you the ability to change how a scene is viewed through the viewports. The options can be selected from the **View Type** menu in the upper left corner of the viewport. Click on the button to open the menu where you can choose alternate options. The following orthogonal views provide viewpoints free from perspective distortion of all the cardinal viewing planes - front, back, top, bottom, left and right.



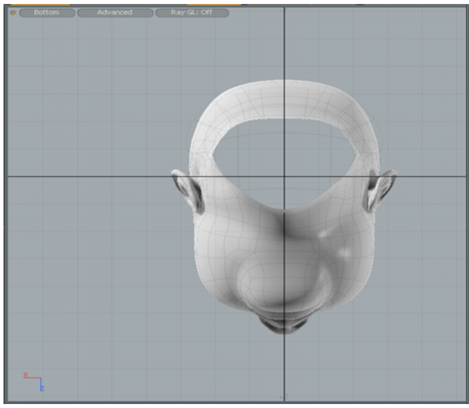
Front



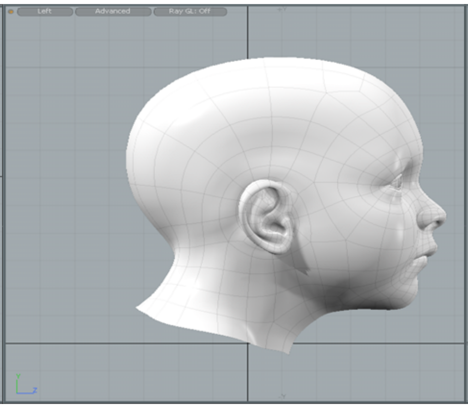
Back



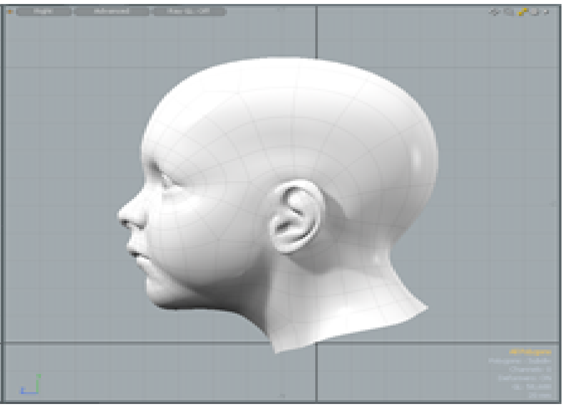
Top



Bottom



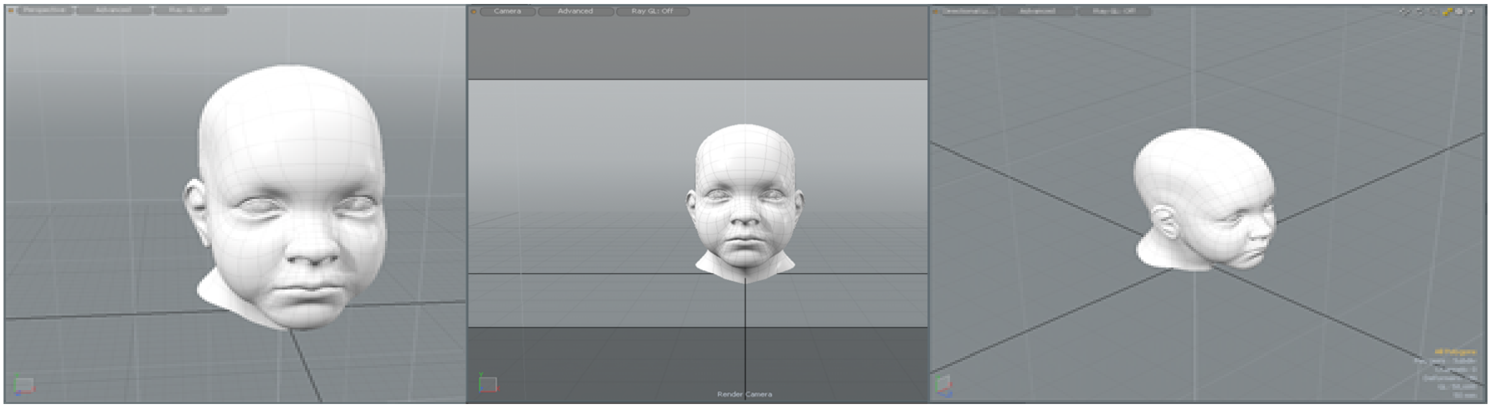
Left



Right

Perspective views are similar to viewing the scene through a real-world camera, where the displays have perspective and a vanishing point. The amount of perspective distortion can be controlled in the menu bar, on Windows, in **System** > **Preferences** > **Display** > **OpenGL** setting **Flatness of Perspective** and on Mac OS, **Menu** > **Preferences** > **Display** > **OpenGL** setting **Flatness of Perspective**in (This is for the **Perspective** view only, the **Camera** and **Light** views are controlled by the item's settings). Additionally, you can view a scene from the Camera's point-of-view, mimicking the field of view settings of the item. When in the **Camera** mode, translucent black bars appear over the frame to represent the rendered frames aspect ratio, for when it doesn't match that of the viewport frame. Finally, there is also the **Light** view that displays the scene from a particular light's point-of-view, providing an easy way to see what the light is pointed at and illuminating.

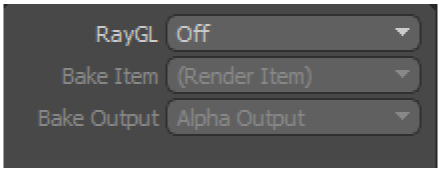
Note: When navigating in the **Camera** or **Light** views, you are actually moving the positions of the elements within the scene.



Perspective Camera Light

Ray GL

The **Ray GL** viewport option leverages Modo's render preview functionality overlaid with the standard viewport options to provide a hybrid display of the rendered scenes, where you can model and interact with the rendered scene directly. To use it, click the center button on the top left of the 3D viewport and select **Ray GL**.



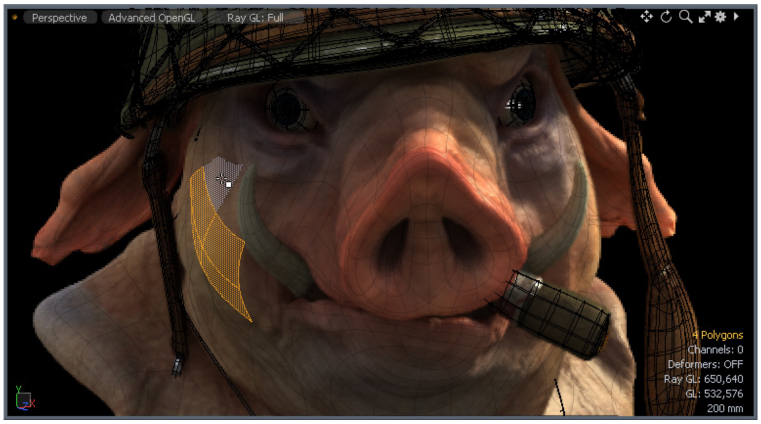
RayGL Off
Bake Item (Render Item)
Bake Output Alpha Output

There are three options to choose from:

- **Full** - Nearly approaches the quality level of a final render but with a reduced ray amount for faster interaction.
- **Fast** - Forgoes textures for a speedier preview of the scene.
- **Bake** - Provides a preview of a bake. When selected, choose from the following options:
  - **Bake Item** - Select the bake item to use for the bake:
    - **(Render Item)** - Bake via the Render Items.
    - **(Current Bake Item)**- The currently selected bake item.
    - One of the other bake items in the scene.
  - **Bake Output** - Choose from one of the available outputs for the currently selected bake item. Outputs can be render outputs or texture outputs from a bake item. Texture outputs are based on the selected texture effect.

    For more information see Working with Bake Items and Baking Tools

For simpler scenes, shading is near real-time, just like in the render preview, but for more complex scenes, there is a lag between edits, where the system temporarily reverts back to the standard **GL** view. Users can control how **Ray GL** updates based on options available within the RayGL **Preferences** available on Windows in **System** > **Preferences** > **Display** > **Ray GL** and , on Mac OS in **Menu** > **Preferences** > **Display** > **Ray GL**. Additionally, you can choose which elements are taken into account when rendering. For example this can be useful when you want to disable fur rendering to speed up the preview when modeling an object with complex fur.

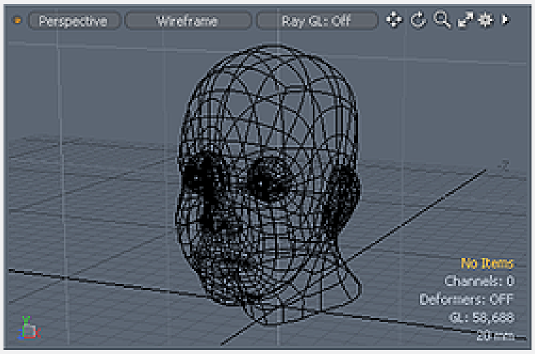
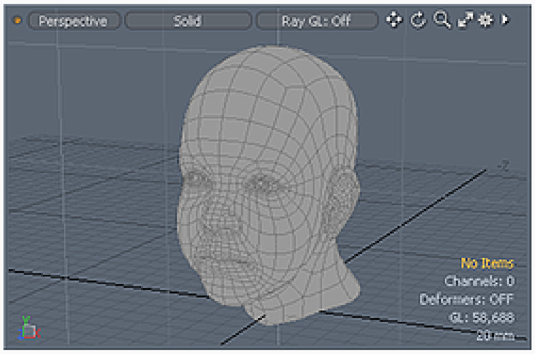
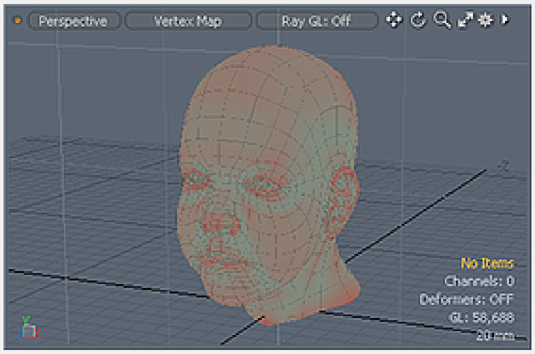
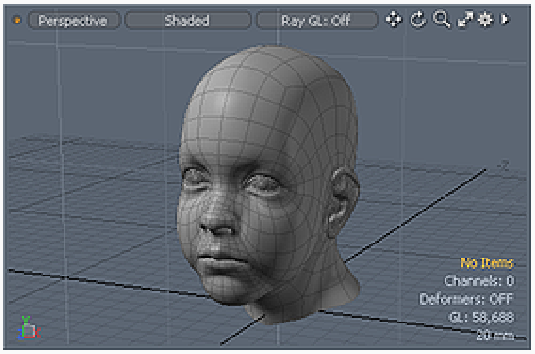


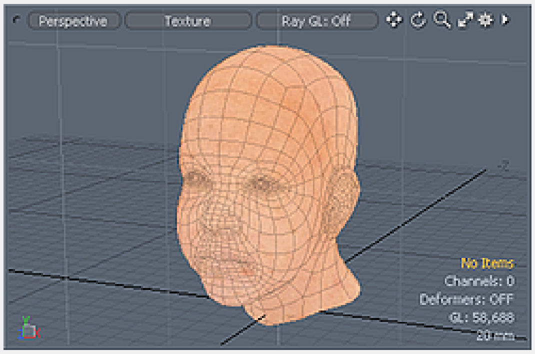
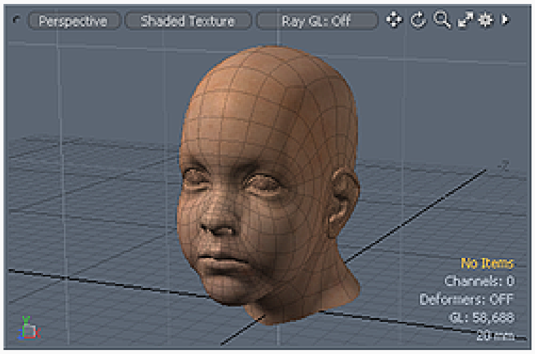
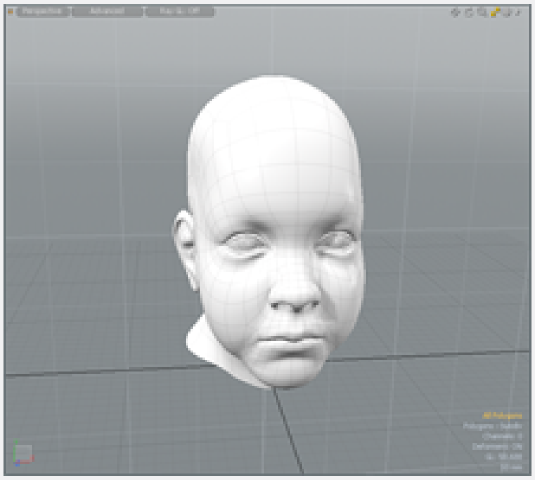
Model courtesy of John Hayes

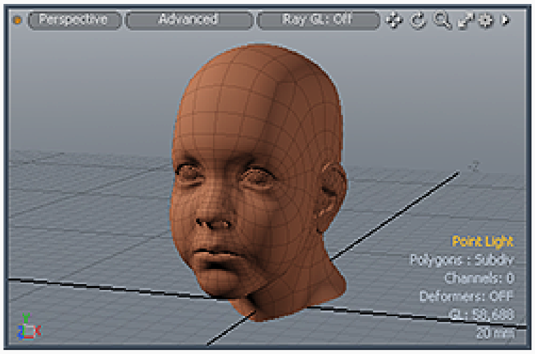
3D Viewport Styles

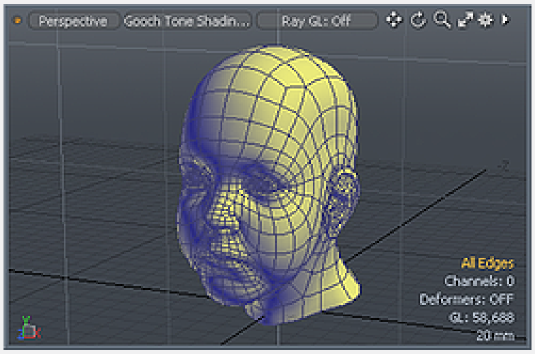
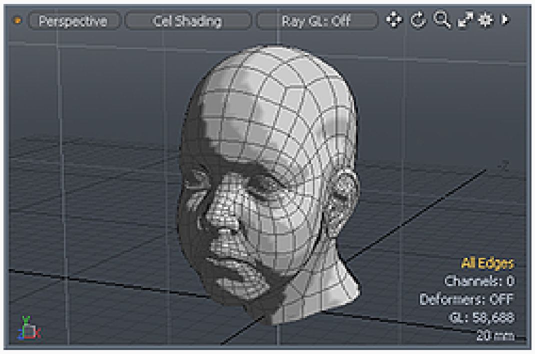
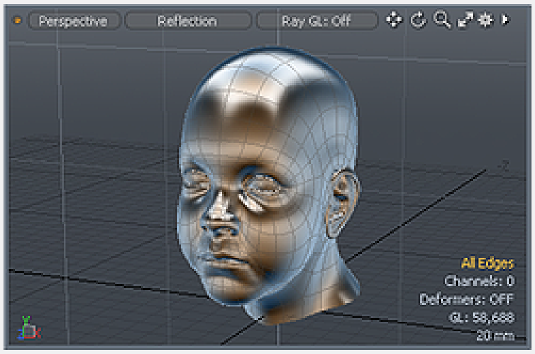
Modo provides a number of ways to visualize a scene using viewport styles. These are in-viewport shaders that display the geometry in specific ways, making it easier to visualize certain aspects of a model or scene. You can choose from the available options by using the **Viewport Options** button within the viewport (the second button from the left on the top).

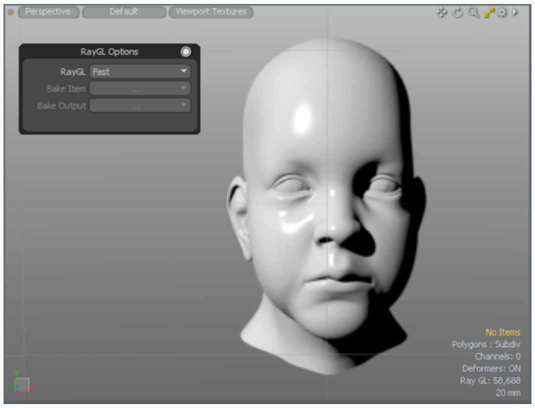
Tip: Click the **Viewport Options** button to set shading options for active (selected) meshes. Hold **Alt** and click the **Viewport Options** button to set shading options for inactive meshes.

Layout	Description
 Perspective Wireframe Ray GL: Off No Items Channels: 0 Deformers: OFF GL: 58,688 20 mm	In the **Wireframe** drawing style only vertices and the edges that connect them to make polygons are visible. The actual polygon faces are not drawn, and as such, you can see through the model to vertices and polygon edges on the backside as well.
 Perspective Solid Ray GL: Off No Items Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Solid** shading mode renders the geometry without shading. This mode is most useful when combined with a wireframe overlay, resulting in a flat rendered sketch mode.
 Perspective Vertex Map Ray GL: Off No Items Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Vertex Map** shade option uses the currently selected Vertex Map to shade the model. In the case of Vertex Color maps, the color of each vertex is blended with the colors of the neighbor vertices. For Weight Maps, positive values shade red while negative values shade blue. Any vertex with a value of 0 or no value at all is shaded green. If Color and Weight type Vertex Maps are selected, the Color Vertex Map takes precedence.
 Perspective Shaded Ray GL: Off No Items Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Shaded** style shows meshes with their basic material definitions such as Diffuse, Specular and Transparency. No image maps appear in the **Shaded** style.

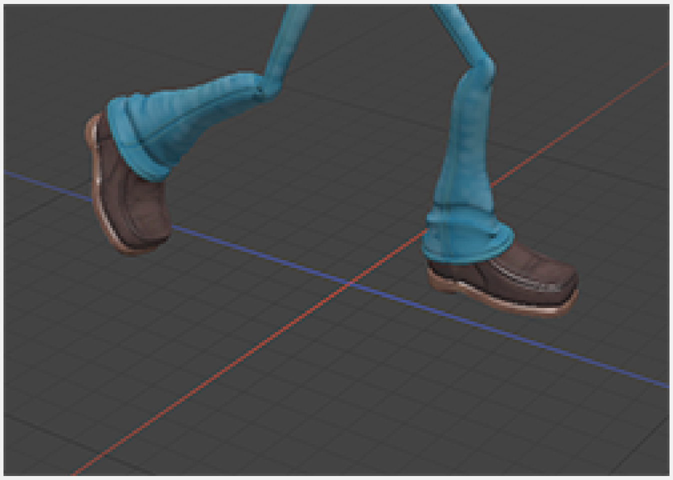
Layout	Description
 Perspective Texture Ray GL: Off No Items Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Texture** style shows the mesh surface with only the most basic material attributes including any image map textures. In **Texture** style, RGBA images can be seen with their alpha blending them into the material beneath. However, limitations of standard Open GL force the image map to draw without shading (that is, full bright). This is a reasonable view style for 3D painting or placing image decals onto a model. The **Shaded Texture** style can shade the image texture but cannot use alpha channel. The **Advanced** view correctly alpha blends multiple image texture layers and shades them appropriately.
 Perspective Shaded Texture Ray GL: Off No Items Channels: 0 Deformers: OFF GL: 58,688 20 mm	In the **Shaded Texture** view, image maps are visible as blended with the light shading effects. Due to OpenGL limitations this mode cannot show images with alpha channels, so any RGBA image appears as an opaque RGB image. To see alpha channels from the image, use the **Texture**, **Default** or **Advanced** view styles.
	The **Default** view allows you to see many more effects than the normal **Texture** view. With **Default** mode you can see the following types of image maps: Color, Bump, Specular, Luminous and Transparency. It blends multiple layers of each (though Bump *only* uses the first layer). However, the total number you actually see at a time is based on the resources of your graphics card. These resource limitations are per Shader Tree surface so if your card has 8 texture units, each Material can have up to 8 image maps. Aside from simple alpha blending, each layer's opacity is respected as well as the base material settings - so you can, for instance, set the material to be 50% transparent, then create a clip and apply it as a transparency map and paint in areas of complete transparency or complete opacity

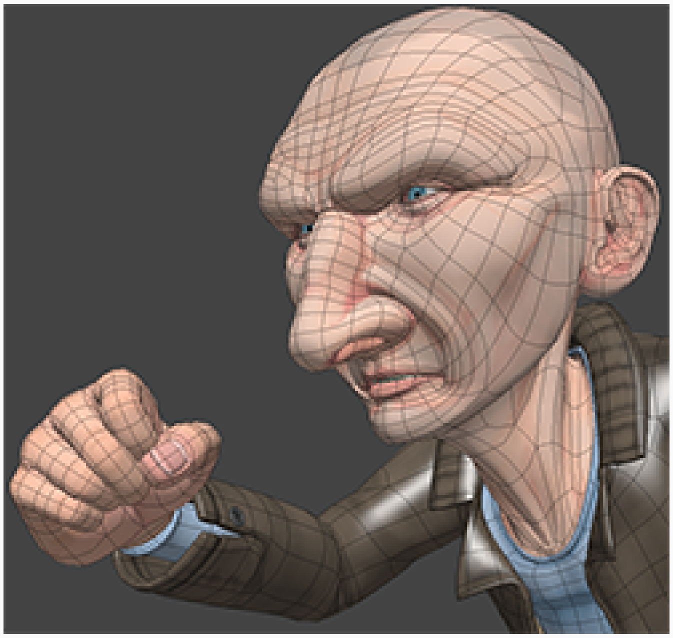
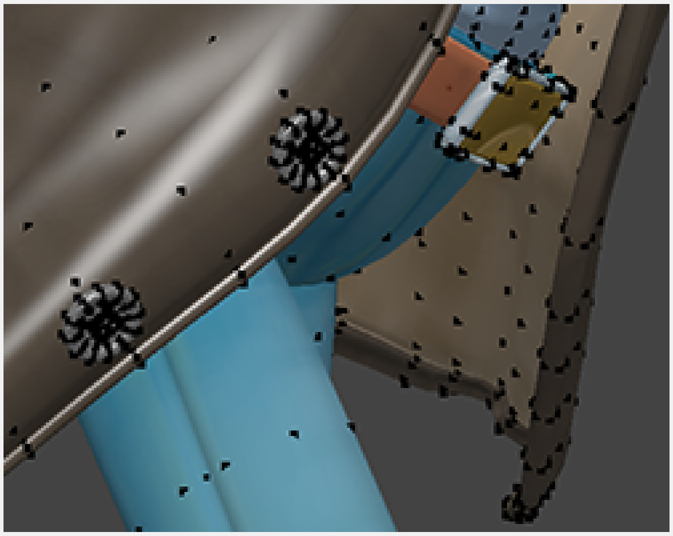
Layout	Description
	(*previously referred to as Advanced GL*).
 Perspective Advanced Ray GL: Off Point Light Polygons : Subdiv Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Advanced** viewport provides a more powerful rendering mechanism by using modern graphics technology. The purpose of this viewport is to be able to produce closer results to the final offline render of scenes, within an interactive manner. Supporting more functionality than the 'Default' view, the **Advanced** viewport supports reflections and fresnel of the Physical Shader, fur materials, as well as 'Ambient Occlusion' and Direct Light shadows and has its own viewport antialiasing. These functions, and more, can be enabled and controlled in the Viewport options panel (see **Advanced Options**). A graphics card with at least OpenGL 3.2 support is required for Advanced Viewport, and the VBO mode must be enabled. When activated, toggle buttons are displayed in the 3D viewport allowing you to select a shading mode. Select one for the following: • Simple Materials • Basic Materials • Full Materials **Note:** Group Masks are supported in the Advanced 3D viewport, allowing you to work on complicated material setups directly in the viewport and to make changes in real time. There is little difference between the representation of a material in the Advanced 3D viewport and the representation of a material in Modo's Preview Viewport.

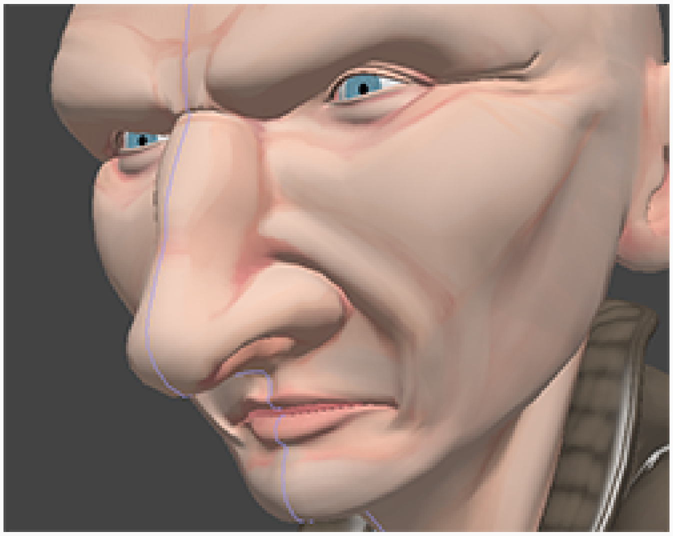
Layout	Description
 Perspective Gooch Tone Shadin... Ray GL: Off All Edges Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Gooch** shading mode is a vertex shading algorithm that displays the geometry using a technique known as tone shading. Rather than relying on the traditional light to dark shadow style shading, which can obscure parts of the model while working, tone mapping represents the curvature of the geometry, shading warm-to-cool where the warm areas are those that would traditionally be *lit* and the cool areas are those that would normally be in shadow. This shading mode allows you to check the surface of your model for proper contouring while still seeing the entire mesh evenly lit.
 Perspective Cel Shading Ray GL: Off All Edges Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Cel Shading** style creates a stepped shading on the model to appear as if it were a cartoon with a fixed number of shading zones.
 Perspective Reflection Ray GL: Off All Edges Channels: 0 Deformers: OFF GL: 58,688 20 mm	The **Reflection** shader displays your geometry as if it were fully reflective. This shader uses a cubic reflection map, and mixes in some of the original material properties with a small amount of shading to properly show object contours. This mode is very useful when performing visual review of a models surface as it assists in finding surface irregularities.

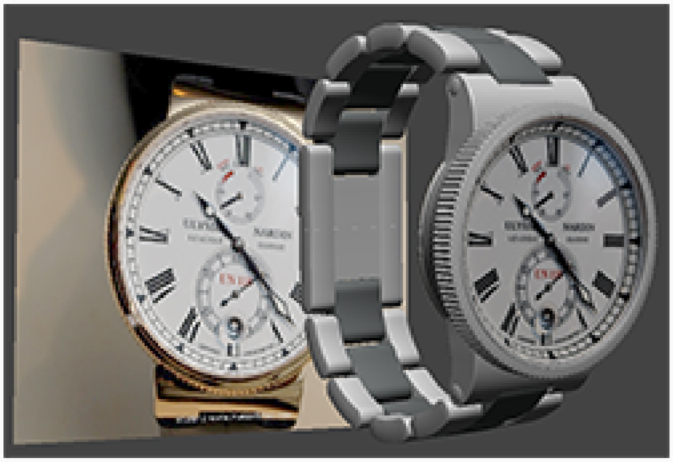
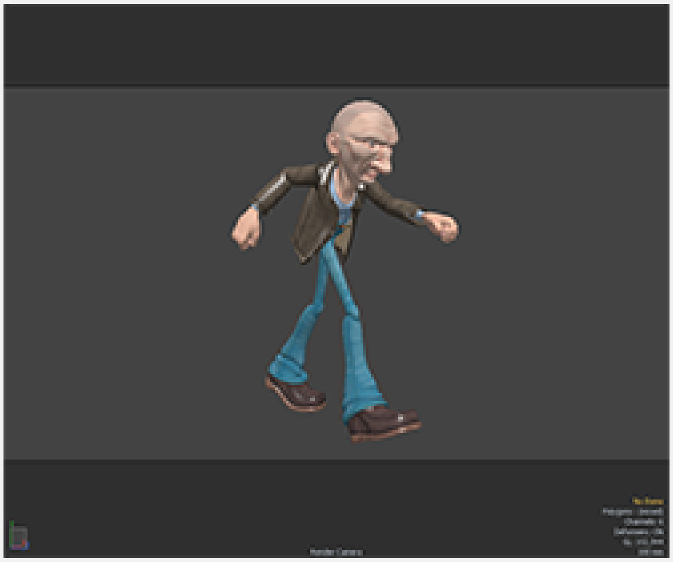
Layout	Description
 Perspective Default Viewport Textures No Items Polygons : Subdiv Channels: 0 Deformers: ON Ray GL: 58,688 20 mm Y X RayGL Options RayGL Fast Bake Item ... Bake Output ...	The **RayGL** shader leverages Modo's render preview functionality overlaid with the standard viewport options to provide a hybrid display of the rendered scenes, where you can model and interact with the rendered scene directly.

Viewport Options

Name	Enabled	Description
Show Grid		Toggles the Grid display in the viewport between the current setting and **OFF**.

Name	Enabled	Description
Show Wireframe in Item Mode		Determines whether wireframe drawing is shown for meshes when in **Item** selection mode.
Show Verts		Toggles between always displaying vertices and the current **Show Vertices** mode selected in the **3D Viewport Properties**.

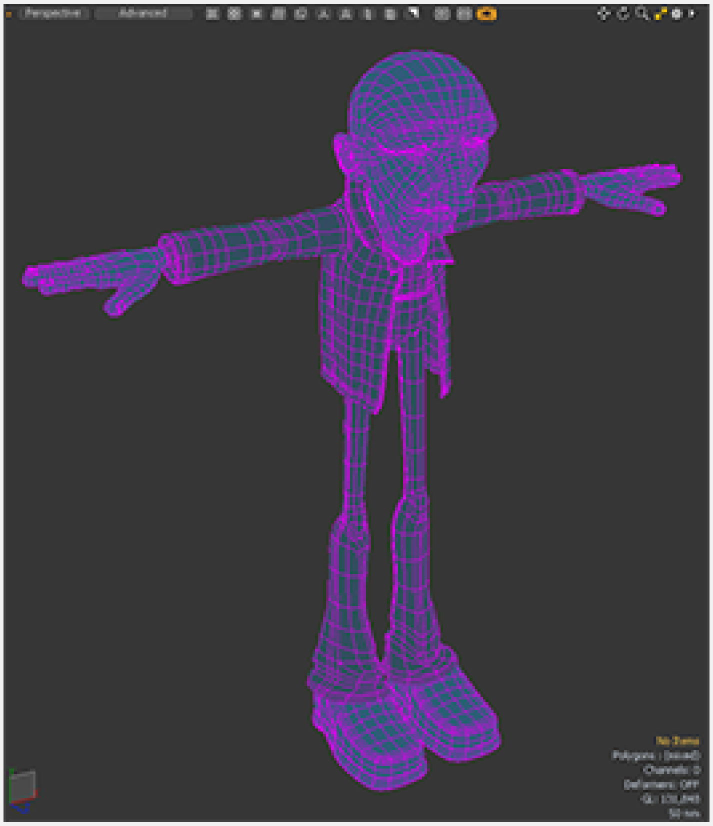
Name	Enabled	Description
Show UV Seam		Toggles the display of **UV Seam** edges. To view a **UV Seam**, you must first create a **Cut Map** and define the seam. For more information, see UV Unwrap.
Smoothing		Toggles smooth shading in the selected 3D viewport.

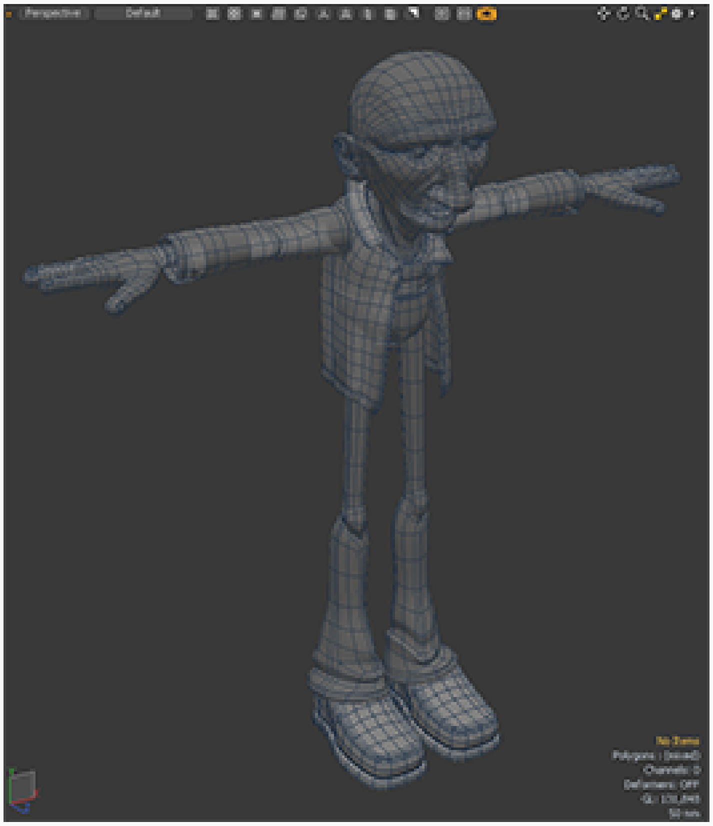
Name	Enabled	Description
Show Backdrop		Toggles the visibility of the **Backdrop item** in the selected 3D viewport. **Note:** For more information about using backdrop images, see Modeling with Images.
Show Camera Frame Region		Toggles the display of the area outside of the camera region in the selected 3D viewport. **Tip:** When the viewport is set to Camera view, you can also adjust the way the Framing Region overlay looks by changing the Camera Frame Color and Camera Frame Opacity.

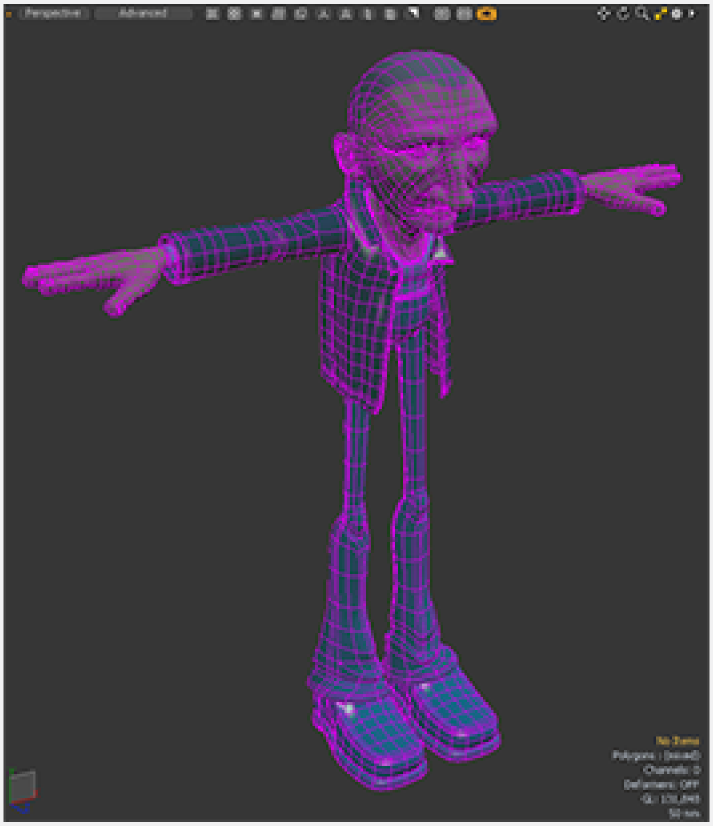
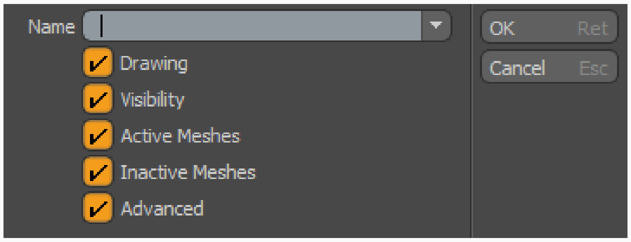
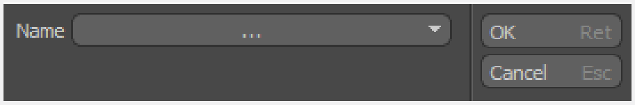
Viewport Presets

Layout	Description
	AVP Good Quality - a preset using the **Advanced** viewport. The following viewport options are enabled: • **Show Grid** • **Show Wireframe in Item Mode** • **Show Backdrop** • **Show Camera Frame Region**

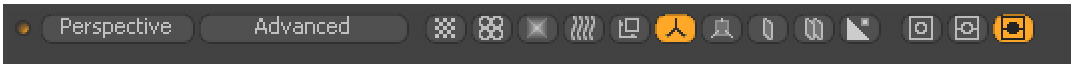
Layout	Description
	Model Fusion - this preset configured for **Mesh Fusion** workflows using the **Default** (OpenGL) viewport. The following viewport options are enabled: - **Show Wireframe in Item Mode** - **Show Backdrop** - **Show Camera Frame Region**

Layout	Description
	The **Topology (AVP)** viewport preset sets up the **Advanced** viewport for retopology workflows. Artists can use this preset for reconstructing the topology of a mesh, for example, to create a lower polygon count version of a model for a game asset, or to create usable topology from a high-density scanned mesh.

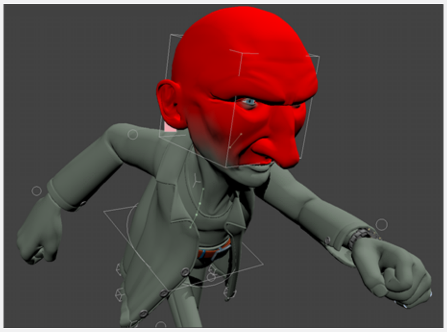
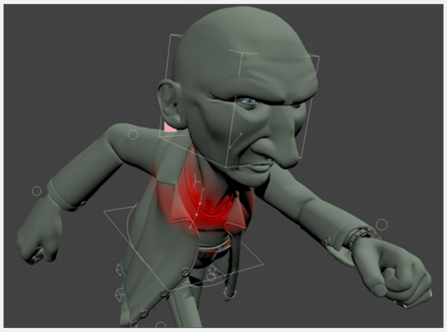
Layout	Description
	The **Topology (OGL)** viewport preset sets up the **Default** (OpenGL) viewport for retopology workflows. Artists can use this preset for reconstructing the topology of a mesh, for example, to create a lower polygon count version of a model for a game asset, or to create usable topology from a high-density scanned mesh.

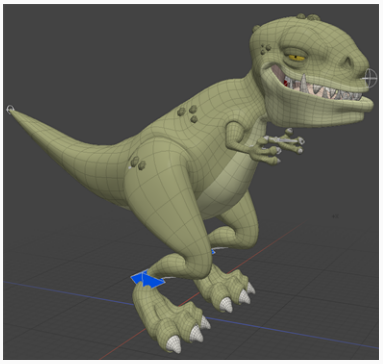
Layout	Description
	The **Topology Intersection (AVP)** viewport preset is tailored for topology workflows by giving artists a better visualization of which areas of a mesh are above or below a background mesh. This preset uses: - The **Advanced** viewport - **Ghost** transparency mode - **Overlay** - No **Flat Shading**
 Name ☑ Drawing ☑ Visibility ☑ Active Meshes ☑ Inactive Meshes ☑ Advanced OK Ret Cancel Esc	**Save Viewport Preset** allows you to save the 3D view as a preset and to specify if Drawing, Visibility, Active Meshes, Inactive Meshes and Advanced viewport options are included. Once done, this viewport is listed in the 3D Viewport Styles dropdown menu.
 Name ... OK Ret Cancel Esc	**Delete Preset** allows you to select a viewport preset to delete.

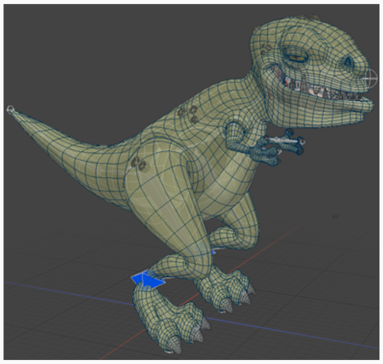
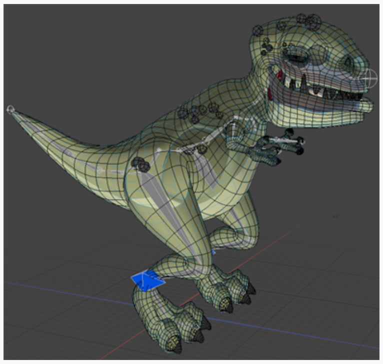
Viewport Toggle Buttons



Use any of the following toggle buttons to quickly enable common options while working on your scene.

▦	**Viewport Textures**	Applies a predefined texture to your scene to quickly see a preview in the 3D Viewport using any of the 3D Viewport Styles. A variety of environments, MatCaps shaders, and UV textures are available. For more information, see Viewport Textures.
▦	**Replicators**	Sets how the replicators are displayed in the 3D viewport. The following options are available: • **None** - Draws all replicators as a bounding box. Choose this option for best performance. • **Selected** - Displays shaded replicators for only those selected. • **All** - Shades all replicators in the 3D viewport.
☒	**Show Weightmaps**	Displays the selected weight map on the geometry in the 3D viewport. If this option is enabled, weightmaps can be displayed in the Vertex Map viewport, the Default viewport and the Advanced viewport. **Show Weightmaps** Off  **Show Weightmaps** On Head Weightmap  **Show Weightmaps** On Chest Weightmap
▨	**Show UV Distortion**	Displays the UV distortion on the geometry in the 3D viewport.

		Note: A UV map must be selected to be visible in the 3D viewport.
	Reference Item	Aligns the coordinate system to the selected item.
	Show Locators	Toggles the visibility of the locator items in the 3D viewport.
	Overlay Drawing	Toggles the visibility of locator items through objects.
	Make Inactive Invisible	Hides inactive layers in the 3D viewport.
	Make Inactive the same as Active	Draws the inactive layers in the same styles as the active layers.
	Transparency Overrides	Allows you to differentiate between background and foreground wireframes, so you can easily understand the topology you're working with. This option is especially useful for decluttering heavy wireframe models. Clicking any of the options sets the override for active meshes. **Alt**+clicking an option sets the override for inactive meshes. The following options are available: **Normal** The default setting. Transparency overrides are disabled. 

Ghost	Front-facing polygons are drawn semi-opaque with a specular highlight. Back-facing polygons are drawn 100% opaque.	
X-Ray	Front-facing polygons are transparent with a specular highlight. Back-facing polygons are drawn 100% opaque. This mode allows you to easily select the joints in a rigged character, for instance.	

Model courtesy of William Vaughan

Note: Transparency overrides are not available in **Wireframe** mode. If you want to use transparency overrides on inactive meshes, make sure that their shading style is NOT set to **Wireframe** under **3D Viewport Properties** > **Inactive Meshes** > **Shading** > **Shading Style**.
Transparency overrides are available in the **Advanced** viewport, as of Modo 14.1v1.

You can edit the Transparency Override properties, such as opacity values and wireframe color, for active and inactive meshes in the **3D Viewport Properties** under **Active Meshes** > **Transparency Overrides**, and **Inactive Meshes** > **Transparency Overrides**, respectively. For more information, see 3D Viewport Properties.

Advanced Viewport Toggle Buttons

When the **Advanced** viewport style is enabled, the following toggle buttons are displayed in the 3D viewport.

▣	**Simple Materials**	Displays the basic geometry shading and materials. Use this shading mode for basic modeling.
▣	**Basic Materials**	Displays basic shading, materials, shadows, and IBL lighting. Use this shading mode to balance speed performance and quality.
▣	**Full Materials**	Displays the highest quality shading and materials. Use this mode when working on textures and final render Playblast. **Note:** The complexity of the Shader Tree setup may affect performance. For example, if you have several layers under the parent shader item, it may take longer to display.

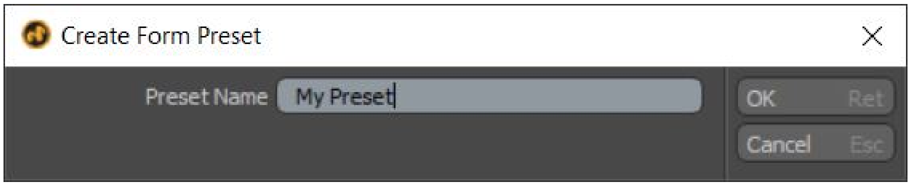
Viewport Light

You can adjust the default lighting in a 3D viewport using the options found under **View** > **Edit Viewport Lights**. Once the function is active, you can click and drag in the viewport to change the current light's position. Additional properties are available in the **Tool Properties** for **Color** and **Intensity**. **Current** defines the particular light being edited. There are two lights by default. You can use the commands **View** > **Add Viewport Light** and **View** > **Remove Viewport Light** to modify the number of lights visible.

Viewport Presets

Under the **Styles** menu, you can save a viewport **Preset**, which allows you to capture all the current viewport settings and easily recall them at a later time.

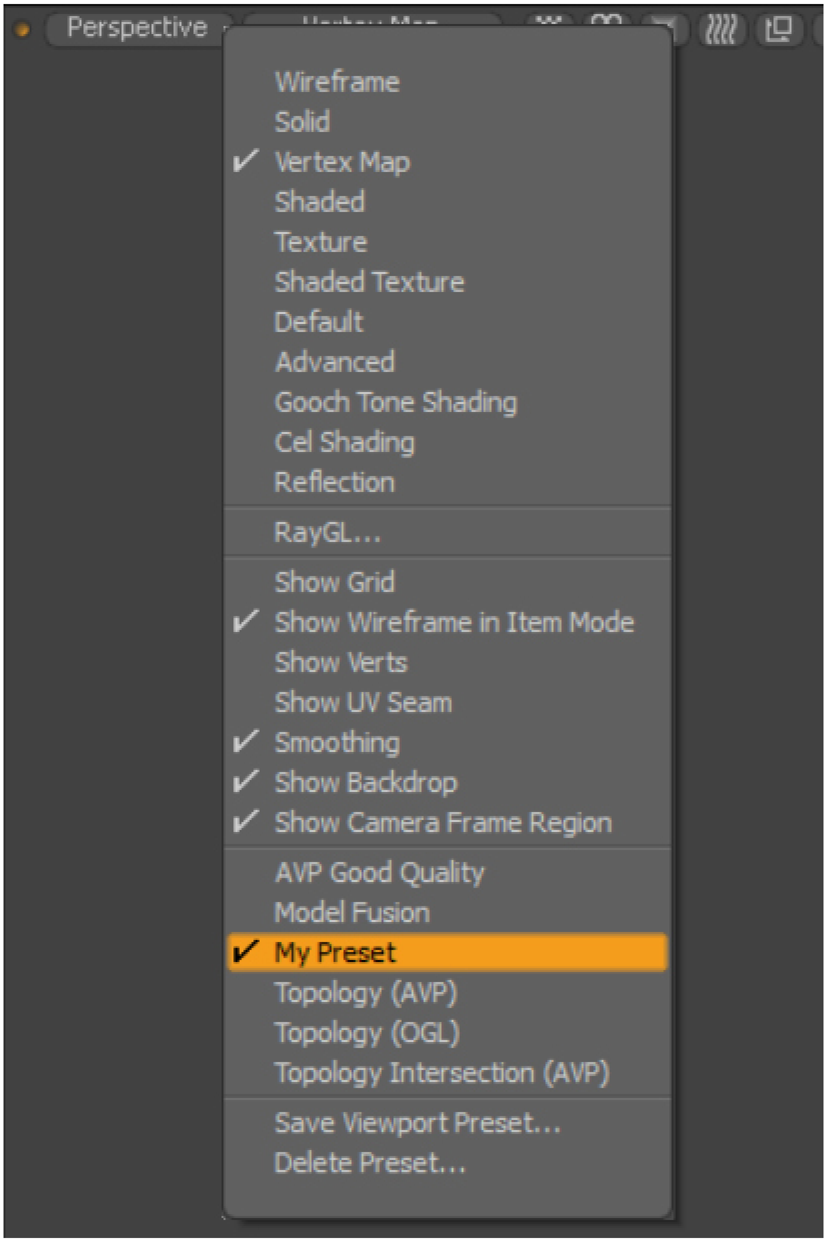
1. Click on the **Viewport Options** button.
2. Select **Save Viewport Preset...**
3. In the **Preset Name** input field, enter a name for the preset and click **OK**.


Create Form Preset
Preset Name
My Preset
OK Ret
Cancel Esc

The new preset name appears in the **Viewport Options** menu.

Note: To delete a viewport preset, select **Delete Preset...** from the **Viewport Options** menu, under the **Save Preset** option.

4. Open the **Viewport Options** menu and click on your new preset from the list.

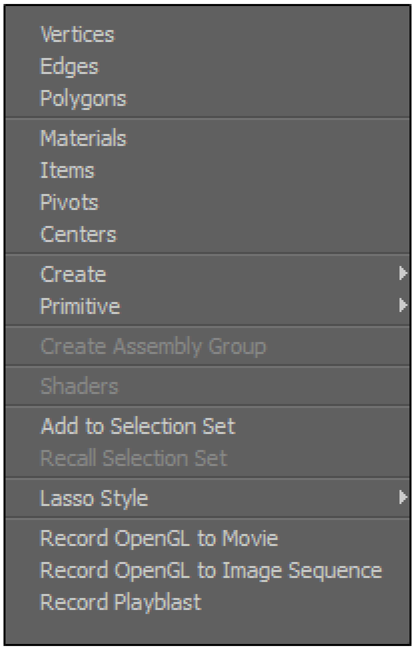

Perspective
Wireframe
Solid
✓ Vertex Map
Shaded
Texture
Shaded Texture
Default
Advanced
Gooch Tone Shading
Cel Shading
Reflection
RayGL...
Show Grid
✓ Show Wireframe in Item Mode
Show Verts
Show UV Seam
✓ Smoothing
✓ Show Backdrop
✓ Show Camera Frame Region
AVP Good Quality
Model Fusion
✓ My Preset
Topology (AVP)
Topology (OGL)
Topology Intersection (AVP)
Save Viewport Preset...
Delete Preset...

A check mark appears on the active preset.

5. If you change any of the viewport options, your active preset is marked with a red asterisk to let you know the settings differ to the preset.
6. Click the preset again to deselect it. Modo reverts the viewport to the state prior to activating the preset.

Contextual Menus

By right-clicking in the 3D Viewport in an empty scene, you are shown a contextual menu to assist you in your design workflow. As you select items that you have added to your scene, you can right-click on an item to view a contextual menu to display another set of related tools for your use. The following is an example of the contextual menu presented in an empty scene.


Vertices
Edges
Polygons
Materials
Items
Pivots
Centers
Create
Primitive
Create Assembly Group
Shaders
Add to Selection Set
Recall Selection Set
Lasso Style
Record OpenGL to Movie
Record OpenGL to Image Sequence
Record Playblast

Option	Description
Vertices	Change the selection mode to **Vertices**.
Edges	Change the selection mode to **Edges**.
Polygons	Change the selection mode to **Polygons**.
Materials	Change the selection mode to **Material** selection.
Items	Change the selection mode to **Items**.
Pivots	Change the selection mode to **Pivots**.
Centers	Change the selection mode to **Centers**.
Create	Create Backdrop Images, Camera Items, Locator Item, Mesh Item, and a number of other Modeling Operations.
Primitive	Add a number of Adding Primitives to your scene.
Create Assembly Group	Create an Assembly Group. For more information, see Assembly Inputs and Outputs
Shaders	Assign an image-specific Shader.

Add to Selection Set	Add the selected elements to a Selection Set. For more information, see Using Selection Sets.
Recall Selection Set	Recall any existing selection set. For more information, see Using Selection Sets.
Lasso Style	Change the Lasso style to **Lasso**, **Rectangle**, **Circle** or **Ellipse**. For more information, see Lasso.
Record OpenGL to Movie	Save your scene as a movie file. A number of file formats are available.
Record OpenGL to Image Sequence	Save your Image Sequence. A number of file formats are available.
Record Playblast	Set Playblast settings. For more information, see Playblast.

3D Viewport Properties

Allows you to further customize the display of elements within a viewport. The default interfaces apply varieties of these options to their custom workspaces, however, they have been fine-tuned to the degree that they rarely need adjusting.

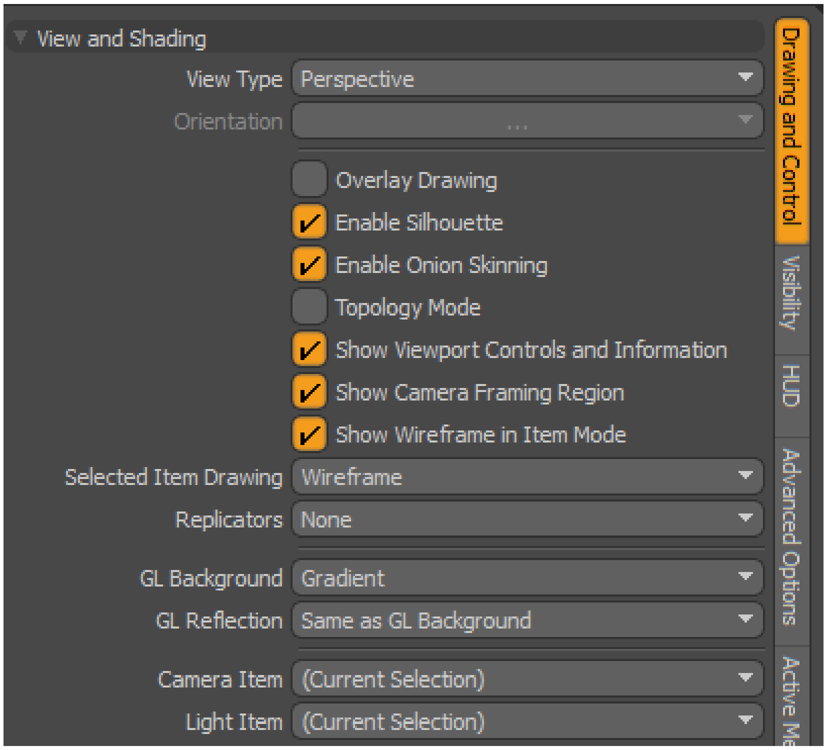
If required, you can easily access the display menu options by clicking on the gear icon in the upper right corner of the viewport, or moving the pointer over the target viewport and pressing **O** to open the pop-up menu.

You can dismiss the window by clicking anywhere in the interface.

Note: Edits are made per-viewport.

Drawing and Control

View and Shading



View and Shading

View Type: Perspective

Orientation: ...

☐ Overlay Drawing

☑ Enable Silhouette

☑ Enable Onion Skinning

☐ Topology Mode

☑ Show Viewport Controls and Information

☑ Show Camera Framing Region

☑ Show Wireframe in Item Mode

Selected Item Drawing: Wireframe

Replicators: None

GL Background: Gradient

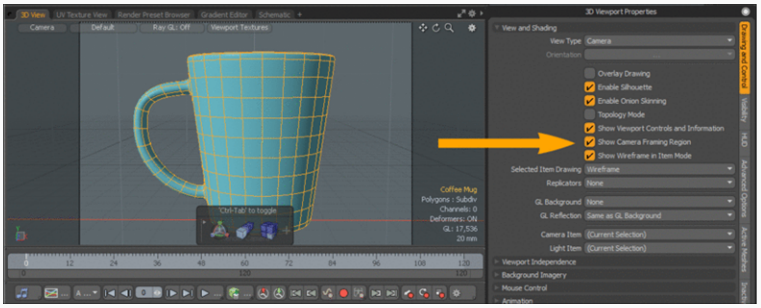
GL Reflection: Same as GL Background

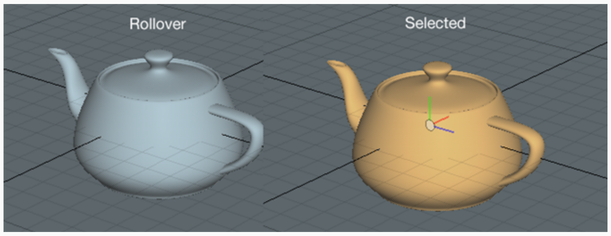
Camera Item: (Current Selection)

Light Item: (Current Selection)

Note: The **Enable Silhouette**, **Topology Mode** and **Show Weightmaps** options are all incompatible with each other and therefore cannot all be enabled at the same time. The **Show Weightmaps** option supersedes the **Enable Silhouette** option and the **Topology Mode** options supersedes both.

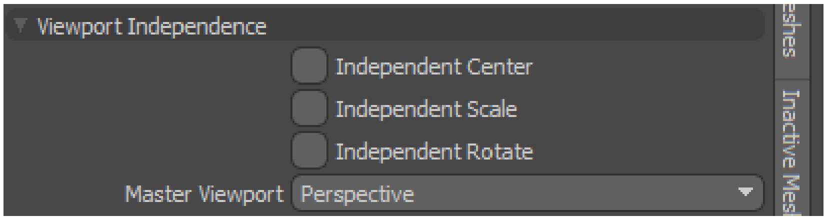
View Type	Choose from the various view types covered in the 3D (OpenGL) Viewport section: Top, Bottom, Back, Front, Left, Right, Perspective, Camera and Light.
Orientation	For the orthogonal views only, you can control upright rotation angle in 90° increments.
Overlay Drawing	For custom draw items, such as locators, cameras and lights (that have no inherent geometry) this option enables an X-Ray style of viewing, meaning

	you can see these elements through polygons when enabled.
Enable Silhouette	When this option is enabled (the default state), any Mesh Item designated in its display properties as **Show as Silhouette** displays as a flat solid shape with no interior details. This option should be thought of as a global enabler only, where the actual display of the silhouette itself is determined by the options of the item's **Display** properties.
Enable Onion Skinning	When this option is enabled, any Actor with **Onion Skinning** defined for it displays as designated in the 3D viewport (the default state).
Topology Mode	This option aids the process of retopologizing a mesh, sometimes also called simply **topo**. This is the process of converting a high resolution mesh to a low resolution mesh by creating new polygons against the high resolution mesh set as a background element. Generally, this process creates intersecting geometry that is difficult to see. Enabling the **Topology Mode**: This option forces the drawing of vertices and edges to always draw in front of background elements in the scene. Polygons are displayed as semi-transparent shapes with solid blue edges. Vertices are enlarged for easier visibility.
Show Viewport Controls and Information	This toggle controls the display of all the viewport element overlays, such as the axis widget, move, scale and Rotate tool's and the information overlays. When the controls are disabled, you can press **O** while hovering the mouse pointer over the target viewport to open the viewport options panel again.
Show Camera Framing Region	This option allows you to toggle the visibility of the camera framing regions in the 3D Viewport. Click the image below to view an animation.  3D Viewport Properties View and Shading View Type: Camera ☐ Overlay Drawing ☑ Enable Silhouette ☑ Enable Onion Skinning ☐ Topology Mode ☑ Show Viewport Controls and Information ☑ Show Camera Framing Region ☑ Show Wireframe in Item Mode Selected Item Drawing: Wireframe Replicators: None GL Background: None GL Reflection: Same as GL Background Camera Item: (Current Selection) Light Item: (Current Selection) 3D View UV Texture View Render Preset Browser Gradient Editor Schematic Camera Default Ray GL: Off Viewport Textures Coffee Mug Polygons : Subdiv Channels: 0 Deformers: ON GL: 17,536 20 mm Ctrl-Tab' to toggle Viewport Independence Background Imagery Mouse Control Animation Drawing and Control Visibility HUD Advanced Options Active Meshes
Show Wireframe in Item Mode	Toggle to enable wireframe drawing in **Item** mode, giving a cleaner display to differentiate between **Item** and **Component** modes.

	Note: Setting this option overrides the settings in the **Active Mesh** and **Inactive Mesh** tabs. For more information, see Active Meshes and 3D Viewport Properties.
Selected Item Drawing	Selected items are drawn as a wireframe (the default) with a transparent fill color over the mesh. The following options are available: • **Wireframe** - The selected items in the scene are displayed as a wireframe with a transparent fill color over the mesh. • **Filled** - The selected items in the scene are displayed with a transparent yellow mesh. Roll over an item in your scene to display the item in a transparent light blue mesh.  • **None** - The selected item in the scene is displayed in a fill color and the unselected items in the scene are drawn in wireframes.
Replicators	This option allows you to control the shaded display of any Replicator in a scene. You can adjust this setting to never (**None**), or always (**All**) display shaded Replicators, or only when a Replicator Item is active (**Selected**). Otherwise, the Replicator clones are displayed as bounding boxes. For more information, see Replicators.
GL Background	You can choose from several viewport-specific GL background options for controlling what is seen in the background of the viewport. • **None** - Displays no background, only the defined viewport color is visible. • **Gradient** - Displays a gray gradient shading that is multiplied over the viewport color. • **Environment** - Displays the settings as determined by the Environment Item of the **Shader Tree**, simulating the rendered output.

	• **Image** - You can define a custom image for use as a viewport background. When this option is selected, a file requester opens where you can select or navigate to the appropriate image. For best results, image should be a spherical projection.
GL Reflection	You can choose from several viewport-specific GL reflection options for controlling the display of reflective surfaces. • **None** - Displays no reflection, ignoring surface material settings. • **Gradient** - Displays a gray gradient shading on reflective surfaces. • **Environment** - Displays the settings as determined by the Environment item of the **Shader Tree**, simulating the rendered output. • **Image** - You can define a custom image for use as reflections (only applies to custom reflective surfaces, or **Reflection** viewport display mode. When this option is selected, a file requester opens, where you can select or navigate to the appropriate image. For best results, the image should be a spherical projection. • **Same as GL Environment** - Reflection options are determined by the **GL Background** settings.
Camera Item	In the case of multiple cameras in a scene, where the viewport is set to **Camera** view type, this option determines which camera item controls the viewport. Selecting alternate cameras changes the viewport to that camera's point of view. **Note:** Navigating in the viewport while in this mode physically moves the camera in the scene.
Light Item	In the case of multiple lights in a scene, where the viewport is set to **Light** view type, this option determines which light item controls the viewport. Selecting an alternate light changes the viewport to that light's point of view.

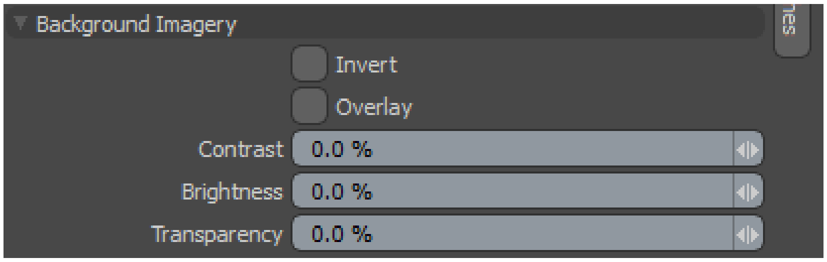
Viewport Independence



☐ Independent Center
☐ Independent Scale
☐ Independent Rotate
Master Viewport: Perspective

Independent Center/Scale/Rotate	To make viewport navigation easy, it is sometimes desirable to link or unlink grouped 3D viewports, so when changing the position, size and angle of the viewport view, associated viewports move in sync with the active viewport. These three toggles allow you to control the syncing of the view properties center, scale, and rotate.
Master Viewport	Determines the linking of viewports, on a per viewport basis, using the **Orthographic**, which links all associated orthographic viewports, and **Perspective**, which links associated perspective views (does not include camera and light views).

Background Imagery



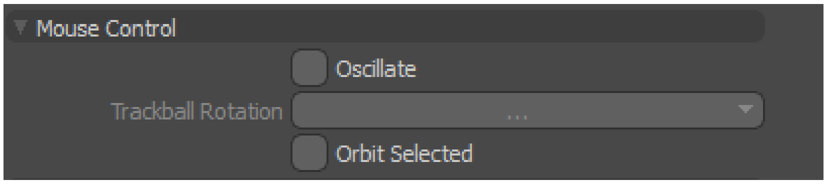
☐ Invert
☐ Overlay
Contrast: 0.0 %
Brightness: 0.0 %
Transparency: 0.0 %

For scenes using Front type texture projection method in the environment item, either from a camera or light, the background imagery options control the display of the projected environment image in the viewport. To use it, apply an image to the Environment item in the Shader Tree, and in its associated **Texture Locator**, assign the **Front** projection type, and specify a camera or light if necessary. For more information on Texture Locators, see Texture Locator.

Invert	Inverts the RGB values of the projected image.

Overlay	When enabled, draws the projected image on top of the viewport, obscuring its contents. When set as such, changing the **Transparency** allows you to see the viewports contents.
Contrast	Adjusts the contrast of the projected viewport image, modifying the visual difference between light and dark values in the image. Positive values increase the contrast, while negative values decrease it.
Brightness	Adjusts the brightness of the projected viewport image. Positive values brighten or lighten the image, while negative values darken it.
Transparency	Changes the opacity of the viewport projected image. When set to **Overlay**, it affects the visibility of items in the scene through the image. When overlay is disabled, this value reveals the settings of the background options.

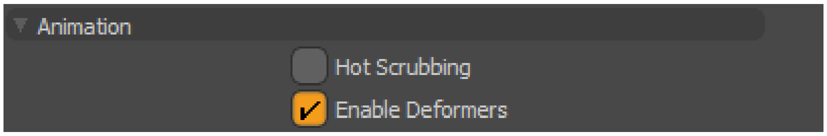
Mouse Controls



Oscillate	In the **Perspective** view type, you can enable the **Oscillate** option. Then, using **Alt**+right-click , drag the cursor, moving the view. When you let go of the mouse button, Modo continues to bounce back and forth, oscillating over the previous mouse movement. This is useful when you want to view a model in motion.
Trackball Rotation	You have several options for orbiting behavior when navigating the viewport. When trackball rotation is disabled, viewport rotations are limited to two axes. When enabled, the viewport can be rotated on multiple axes based on the mouse's viewport position, as if the viewport itself was a giant trackball. The options are: • **No** - Disables trackball style rotation in favor of standard upright orbiting. • **Yes** - Enables trackball rotation for the specific viewport. • **Default** - Uses the preference setting available on Windows and Linux

	in **System** > **Preferences** > **Input** > **Remapping** > **Viewport Rotation** and on Mac OS in **Menu** > **Preferences** > **Input** > **Remapping** > **Viewport Rotation**. **Note:** Trackball rotation is disabled by default in the **Preferences**.
Orbit Selected	When this option is enabled, Modo automatically orbits around the selected element, using the selection's bounding box center as the center of rotation.

Animation



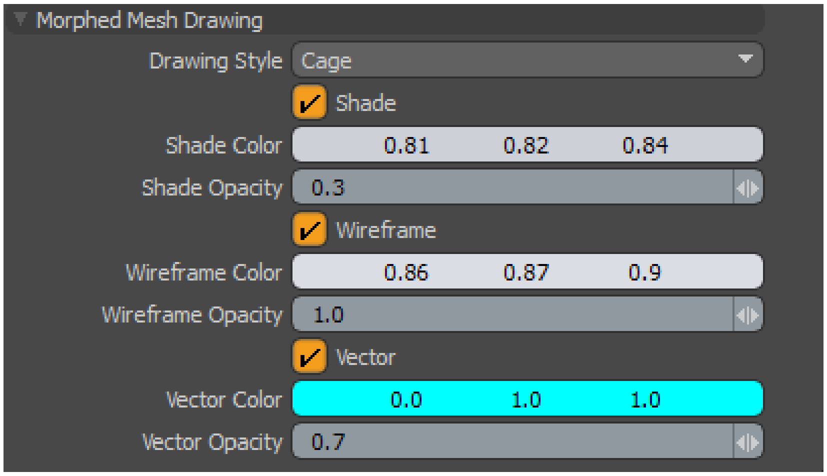
Animation

☐ Hot Scrubbing

☑ Enable Deformers

Hot Scrubbing	When enabled, animated items in the scene update continuously when dragging the Timeline, producing a smooth motion. When disabled, items only update when the mouse pauses, or isn't moving, providing slightly faster performance.
Enable Deformers	This toggle allows viewports to display the results of animated morph and Vertex Map deformers in the viewport. By default, this option is enabled in all built-in 3D viewports. **Note:** When this option is disabled, mesh operations are not displayed in the viewport.

Morphed Mesh Drawing



Morphed Mesh Drawing

Drawing Style: Cage

☑ Shade

Shade Color: 0.81 0.82 0.84

Shade Opacity: 0.3

☑ Wireframe

Wireframe Color: 0.86 0.87 0.9

Wireframe Opacity: 1.0

☑ Vector

Vector Color: 0.0 1.0 1.0

Vector Opacity: 0.7

Drawing Style	Specify how you would like to draw Morph Maps. The following options are available:
	• **Default** - Modo draws the traditional morphed mesh.
	• **Cage** - Modo draws the base mesh and the morphed mesh as a cage. When this option is selected, additional properties are revealed:
	  • **Shade** - Draws a transparent morphed mesh as a cage.
	  • **Shade Color** - Specifies the color of the morph cage. Note, only displayed when a Shader, Wireframe, or Vector is enabled.
	  • **Shade Opacity**- Specifies the opacity value of the morph cage.
	  • **Wireframe** - Draws the cage in wireframe mode.
	  • **Wireframe Color** - Specifies the color of the wireframe.
	  • **Vector** - Draws line segments from the vertices of the base mesh to the corresponding vertices of the morph mesh.
	Note: To display a morphed mesh, you need to select a morph map with **Enable Deformers** disabled.
	  • **Vector Color** - Specifies the color of the vector lines.
	  • **Vector Opacity** - Specifies the opacity value of the vector lines.

Visibility

Item Visibility

Show Lights	Toggles the visibility of the light item representations in the current viewport. Typically disabled for modeling views.
Light Illumination	When **Show Lights** is enabled, disabling this control turns off illumination from all lights in the scene to improve viewport response times. This is especially useful if you have multiple lights in the scene that you want to adjust. You can still see the light widgets in the viewport, but the illumination they provide is disabled so you can move the light positions quickly and easily.
Show Cameras	Toggles the visibility of the camera item representations in the current viewport. Typically disabled for modeling views.
Show Locators	Toggles the visibility of the locator item representations in the current viewport.
Show Texture Locators	Toggles the visibility of the texture locator item representations in the current viewport.
Show Meshes	Toggles the visibility of Mesh Items in the current viewport.
Show Instances	Toggles the visibility of instance items in the current viewport.
Show Pivots/Centers	Visibility for pivot and center representations can be determined by these criteria: • **None** - Only shows the pivot/center when in Pivots/Centers selection mode. • **Selected** - Shows the pivot/center when any item layer is selected in the Item List. • **All** - Always shows the pivot/center for all items.

General Visibility

Show Work Plane	Toggles the visibility of the Work Plane in the current viewport. (keyboard shortcut is Num Pad ***).** • **Always** - Set to always display the Work Plane. This is the default option. • **Active** - Set to display the Work Plane when it has been active. For example, moved or rotated. • **Never** - Set to never display the Work Plane.
Show Grid	Toggles the visibility of the ground plane grid at the 0 position of the up axis. • **Always** - Set to always display the grid. • **Inactive Work Plane** - Set the main grid to be drawn when the work plane is not active. This has the effect of there being only one grid drawn at a time. This is the default option. • **Never** - Set to never display the grid.
Draw Faded Grid	Toggles the visibility of the main grid and Work Plane to be drawn faded.
Faded Grid Color Axes	Toggles the visibility of the main grid and Work Plane to be drawn with colored axes.
Show Backdrop	Toggles the visibility of any Backdrop Image items in the scene. For more information on Backdrop Images, see Modeling with Images.

Selection Visibility

Show Selections	Toggles the visibility of selection highlighting.
Show Selection Normals	When enabled, this option displays normals for any selected polygon(s), displayed as a small dashed line pointing outward from the center of the polygon. Normals represent the facing direction for polygons.
Show Selected Filling	Toggles the visibility of the selection highlight overlay.
Show Selected Outline	Toggles the visibility of the selection highlighting outline.
Show Selection Rollovers	Toggles the visibility of selection rollover pre-highlighting.

Performance

By disabling some of these options, the performance is improved in the 3D viewport when you are working with complex models.

Fur	Toggles the visibility of fur material applied to your mesh.
Displacements	Toggles the visibility of Displacement maps.
Mesh Smoothing	Toggles the settings applied to smooth geometry. When enabled, polygons shade according to the material Smoothing setting.
Use Shader Tree	Toggles the materials and textures created in the Shader Tree.
Use Texture	When enabled, all viewport textures, except displacements, are displayed in the viewport.

HUD

Show Viewport HUD	Toggles the visibility of the Tool HUD. Press **Ctrl** + **Tab** to toggle the HUD off and on.
HUD Property Form	Specifies what property form the Tool HUD uses. You can customize the property form in the **Form Editor**, available from the menu bar. Click **System** > **Form Editor**. In the editor, the available property forms are listed under **Form HUD**.
HUD Style	Specifies how the Tool HUD is displayed. The following options are available: • **fixed** - Sets the Tool HUD to a fixed position. • **floating** - Allows the Tool HUD to be moved around. • **Close** - Transforms the Tool HUD into a moveable palette with a **Close** button.
Allow HUD Customization	When enabled, you can add new tools to the Tool HUD and remove existing ones. When disabled, the set of tools in the Tool HUD can't be edited.

Advanced Options

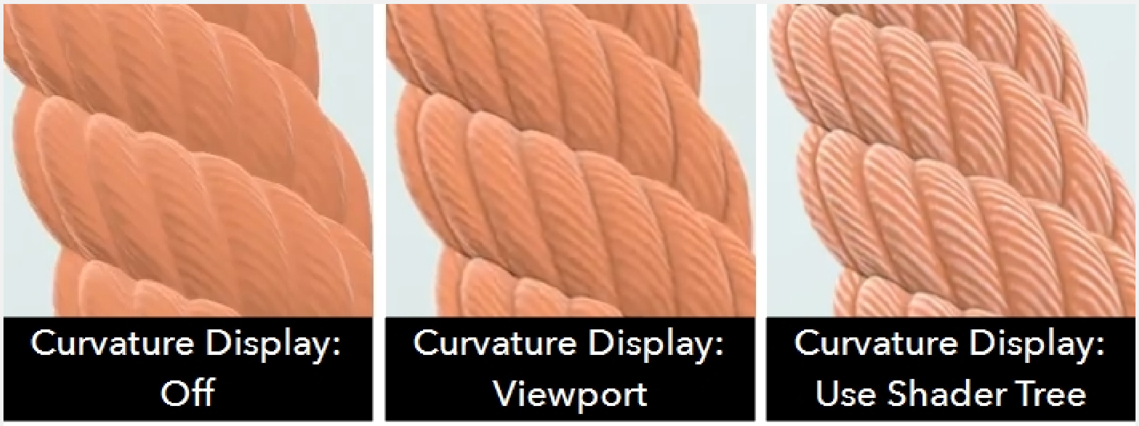
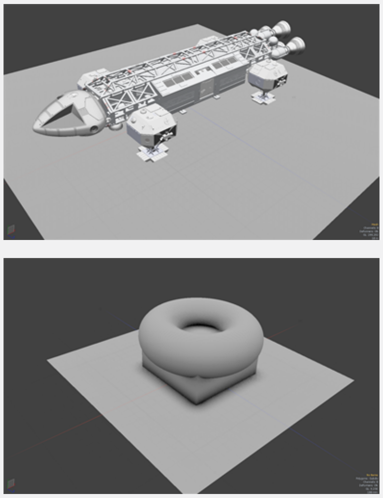
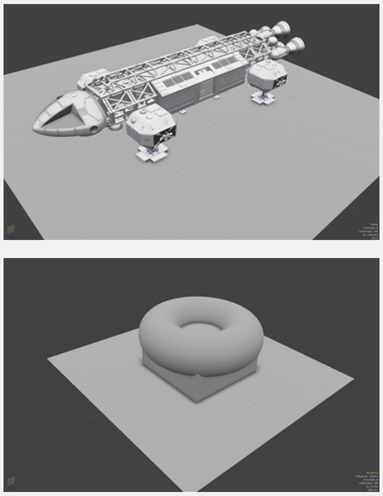
The **Advanced Viewport** options control the settings applied only to the **Advanced** viewport display style.

Effects

Ambient Intensity	Adds an even illumination globally to all diffuse surfaces in a scene. This is especially useful for simulating additional bounces of light.
Ambient Color	Allows you to specify a color value for the global light added by the **Ambient Intensity** setting. **Note:** **Ambient Intensity** and **Ambient Color** are also available in the Render Items's Global Illumination settings. The **Global Illumination** settings are applied at render time, and override the viewport settings.
Show Shadows	Toggles the visibility of Direct Light shadows in the viewport. Note that Point, Directional, and Spot Lights generate more accurate shadows than Area and Cylinder lights.
Shadows CSM	Increases or decreases the shadow quality displayed in the viewport. You can set the number of shadow buffer projections the Cascade Shadow Map system uses for the light source to view casting curved shadows on curved surfaces. The following options are available: **off**, **x2**, **x3**, and **x4**. The higher the value specified, the smoother the shadow edges are. By default, this option is set to **x2**. **Note:** The higher the level specified, the more memory and processing time are required. (Lower values render faster but produce less shadow detail.)
Shadows Resolution	Sets the texture resolution used when depth projecting the shadows. The following options are available: **Low**, **Mid**, and **High**. By default, this option is set to **High**.
Shadows Filter	Sets whether the shadows are filtered. By default, this option is **On**. When working in a complex scene, turn this option off to improve performance in the 3D viewport.
Use Normal Maps	Toggles all normal map visibility.

Use Bump Maps	Toggles all bump map visibility.
Materials	Overrides the **Shader Tree** materials being used in the scene. This setting provides faster rendering performance results in the **Advanced** viewport when rendering a number of different types of materials. These options are also useful when working with a Unity Material or Unreal Material. • **Simple** - Produces a fast, simple rendering in the **Advanced Viewport**. Note, this option does not support Image Based Lighting or shadows. • **Basic** - Produces a rendering in the **Advanced Viewport** using simple traditional Shader model materials. Most basic **Shader Tree** materials are supported. For example, Diffuse and Specular materials. Unity Material, Unreal Material, Image Based Lighting, and shadows are also supported. In most cases, selecting this option produces faster results than selecting the **Full** option. • **Full** - Produces a full rendering, using all of your specified materials, that is similar to the output of the Preview Viewport and Render Display. This option takes more time to render.
Groups	Controls how Shader Tree groups are handled in the 3D viewport: • **Flat** - Ignores groups and produces a flattened Shader Tree. • **Full** - All Shader Tree groups are rendered.
Lighting	Sets the light sources that are used to illuminate the geometry. • **Default Viewport** - Uses normal viewport lighting. This includes scene lights, Image Based Lighting, and the default Viewport Light. • **Scene** - Uses only lights in the scene. This excludes the **Viewport Light** and Image Based Lighting. • **Environment** - Uses only the **Environment** image for lighting. • **Scene + Environment** - Combines both **Scene** and **Environment** lighting. • **Default + Environment** - Combines both **Default Viewport** lighting, including scene lights, Image Based Lighting, and the default Viewport Light with **Environment** lighting. This is the default setting.
Background	Selects the viewport background type: • **Default Viewport** - Uses the default background. • **Environment** - Draws the background as defined by the Environments settings in the **Shader Tree**. This allows you to show **Environment** detail that was previously only available in the **Render Display**.

Display Override	Sets the option to remove wireframe lines or widgets from the viewport: • **(none)** - Draw all lines and widgets. • **Render without wireframe** - Removes wireframes but draw widgets, and overlay information. • **Render without wireframe and widgets** - Removes wireframes, widgets and overlay information.
Visibility	Controls the visibility of items in the viewport when set to **Advanced**. When set as the **Render** option, the Render visibility setting on each item determines viewport visibility, and when set to the **Viewport** option, an item's visibility is determined by its Items list visibility (as controlled by the eye column of the Items list).
Curvature Mode	Sets how the curvature shading is calculated when **Curvature Display** is set to **Use Shader Tree** or **Viewport**: • **Fast** - uses a derivative calculation to produce faster, lower quality shading. • **Screen Space (Geometric)** - uses a global screen space calculation based on the geometry to calculate shading, producing the curvature effect between each polygon, along each edge, making it easier to see how the geometry was constructed. • **Screen Space (Normals)** - uses a global screen space calculation based on the geometry normals to calculate shading, which smooths the curvature over the whole surface making it easier to see the curvature effect for larger, smoother shapes.
Curvature Display	Controls how curvature shading is displayed in the viewport: • **Use Shader Tree** - use the **Occlusion** type selected in the Shader Tree to calculate curvature shading for each mesh. The example image shows **Hard Light** occlusion. **Note:** The **Occlusion** layer **Type** property must be set to **Curvature** to apply the selected occlusion type. • **Viewport** - use the Viewport to calculate curvature shading equally on the entire scene. • **Off** - no curvature shading is added.

	 Curvature Display: Off Curvature Display: Viewport Curvature Display: Use Shader Tree
Curvature Intensity	Controls the intensity of curvature shading when **Curvature Display** is set to **Use Shader Tree** or **Viewport**.
Ambient Occlusion Mode	Sets the mode of rendering screen-space Ambient Occlusion, an effect that darkens recessed areas of a surface, producing a look similar to global illumination light scattering. You have the following options: • **Off** - The default option. Disables Ambient Occlusion in the **Advanced** viewport. • **Hybrid** - An improved, more robust version of the original Ambient Occlusion effect. It works better on large scale scenes than the original effect.  Original (deprecated)  Hybrid

	Note: This effect is completely independent of the **Ambient Occlusion** Shader Tree layer and not representative of how it renders, but adds an additional visualization possibility to enhance the look of the GL viewport models. In **Hybrid** mode, additional options become available to fine-tune the look of the Ambient Occlusion effect. • **Radius** - Determines the size of the effect. A large radius can affect performance. • **Bias** - Adjusts the shading falloff curve. Lower values darken the Ambient Occlusion, while higher values lighten the effect. This option is not used by **Hybrid** mode. • **Intensity** - Controls the overall strength (darkness) of the Ambient Occlusion effect. The higher the **Intensity** value, the darker the result. • **Blur Radius** - Applies a blur to soften the resulting **Ambient Occlusion** effect, producing a smoother overall result, but higher values, result in greater performance cost.
Antialiasing	Determines the amount of subsampling of viewport drawing that smooths the jagged display of the **Advanced** viewport. The default is set to **Off**.
Multisampling	With **Antialiasing** enabled, as you increase the **Multisampling** value, the amount of jaggedness reduces. The default is **1**.
Progressive	Another way of producing anti-aliased images. It specifies the maximum number of frames used to produce the final image in the **Advanced** viewport. Specifying higher values results in anti-aliased images. When interacting with the viewport, it runs at full non-aliased frame rates, then improves the quality and aliasing over time when there's no interaction with the viewport. The default value is **1**, which means the option is off.
Line Antialiasing	Renders an anti-aliased model wireframe displayed in the 3D viewport. The following options are available: • **Off** - No attempt is made to anti-alias the model wireframe rendering. This is always the fastest option. • **System** - If the present drivers support it, the 3D (OpenGL) viewport smooth line drawing is used. This is fast but has limited support. For more information on supported hardware, see the **Release Notes**. • **Full** - The 3D Viewport uses a special shader to render anti-aliased model wireframes. This is the slowest option but it provides the best visual quality.

	Note: The **Line Antialiasing** option is independent of the general **Antialisaing** option. **Note:** On macOS (especially ARM hardware) using the **Full** option can yield significantly lower performance.
Anisotropic Filtering	Filters all material textures using the Graphics Processing Unit (GPU) value. The **Max** level is different for all GPUs. When enabled, the GPUs hardware texture filters units to apply the best Anisotropic filtering the GPU can do. The default value is set to **Off**.
Transparency	Defines the display of transparent surfaces. The following options are available: • **Off** - Disables transparency drawing. • **Fast** - Does simple depth sorted transparency. • **Correct** - Does correct depth sorted transparency, producing a more accurate result. Use the **Mode** and **Depth** dropdowns to control how **Correct Transparency** is calculated.
Correct Transparency Mode	Sets how Modo sorts layers of transparency to render the final image: • **Depth Peeling** - uses the legacy Modo method to sort transparency. The results from **Depth Peeling** may not be as high quality as **Full** mode. **Note:** **Depth Peeling** is the only mode available on macOS, **Shallow** and **Full** are unsupported. • **Shallow** - only sort the top number of layers specified by the **Correct Transparency Depth** control, resulting in faster rendering of transparency. • **Full** - sort all layers of transparency, which produces the best result at a cost of processing time.
Correct Transparency Depth	When **Correct Transparency Mode** is set to **Shallow**, sets the number of layers of transparency to sort.
Screen Space Reflections	Shows surface reflections as you would in the **Render Display**. The reflection can be adjusted by adjusting the material properties, namely **Reflection Amount**, **Reflection Type**, **Blurry Reflection** and **Roughness**. See Material for more details.

	• **Off** - Disables reflections. • **Fast** - Enables fast (non-blurry) reflections. • **Blurry** - Enables full reflections with blur. **Note:** The frame rate will get worse with an increase in the number of reflected objects, especially if they are blurred.
Dither Mode	The advanced viewport has the following options for controlling color processing. • **Off** - Disables dithering, which may reveal banding in the image. • **Ordered** - Introduces simple ordered dither to reduce banding caused by color gradients. This option is enabled by default. • **HDR** - When using a HDR/XDR monitor, image intensity can go into High Dynamic Range. **Note:** The HDR option is currently only available on MacOS 10.15 or higher.
ColorSpace Mode	Use this setting to change the color space displayed in your viewport. • **OCIO** - Uses the colorspace loaded into Modo's global OCIO profiles. See Custom OCIO profiles for more information. • **sRGB** - Enabled by default. Sets the colorspace to sRGB, which is commonly used by computer displays. • **Linear** - Applies a linear calculation to pixel color and brightness values rather than a gamma curve. May improve performance slightly.

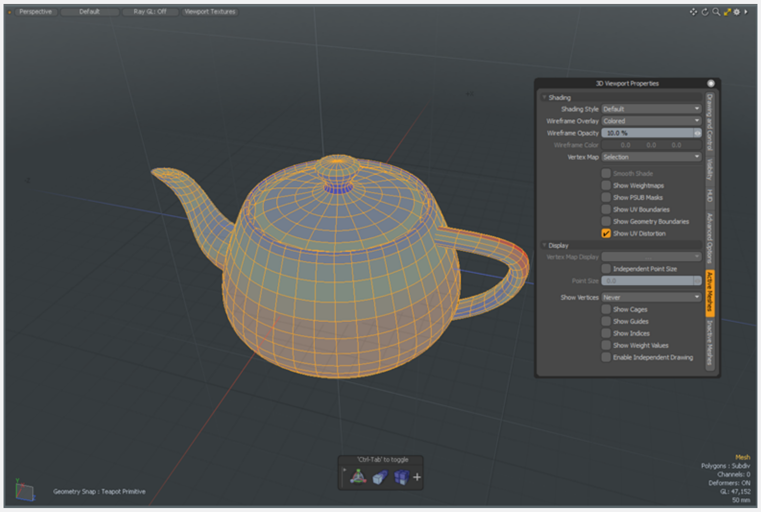
Active Meshes

The following options allow you to control the display of elements that are in the foreground, meaning they are selected in the Items list. Background layers are those that are visible, but not selected.

Shading

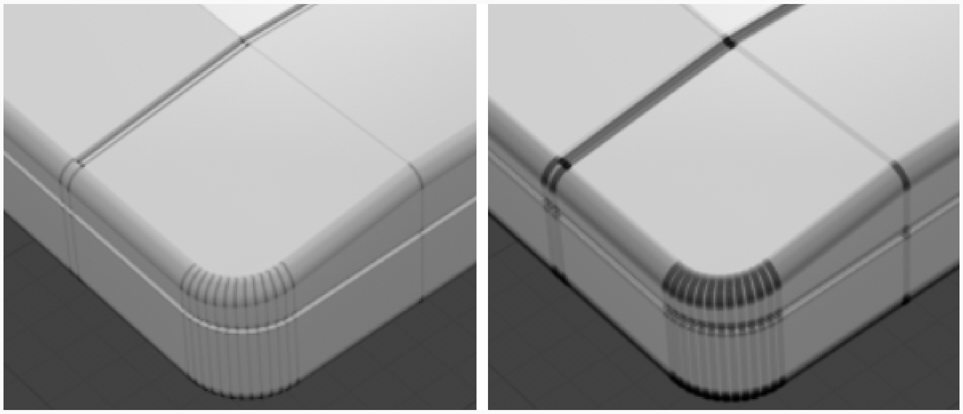
Shading Style	You can choose from the available view styles covered above: Wireframe, Solid, Shaded, Vertex Map, Texture, Shaded Texture, Default, Advanced,

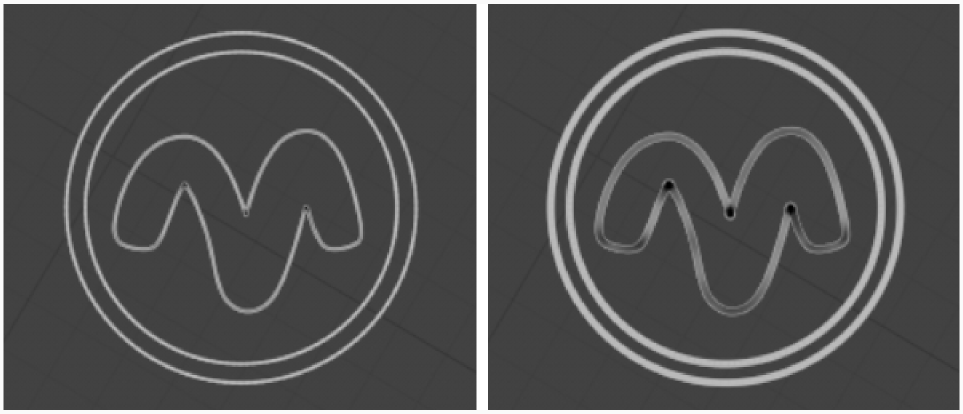
	Gooch, Cel and Reflection Shading
Wireframe Overlay	Determines the style of geometry wireframe display. You can adjust the colors of the wireframe using the preference setting in Windows in **System** > **Preferences** > **Display** > **Colors** and on Mac OS in **Menu** > **Preferences** > **Display** > **Colors**. Once a color scheme is created in the **Preferences** and saved as a named preset, it must be applied to the active viewport with the menu bar command **View** > **Viewport Color Scheme**. • **None** - No wireframe displayed. (**/**) • **Uniform** - By default this is a white wireframe outline. (**Shift**+**/**) • **Colored** - By default this is a black wireframe outline. (**Ctrl**+**/**)
Wireframe Opacity	Controls the opacity of the wireframe overlay display.
Wireframe Color	Specifies the color of the wireframe in the 3D viewport.
Vertex Map	Defines which type of Vertex Map is displayed when in the **Vertex Map** viewport. • **Selection** - Displays the most recent Vertex Map selection. • **Weight** - Favors Weight Maps for display. • **Vertex Color** - Favors vertex color maps for display.
Smooth Shade	This toggle overrides the material **Smoothing** setting, only changing how smoothing is displayed in the viewport, not the rendered output. When disabled, individual polygons shade flat, appearing faceted. When enabled, polygons shade according to the material Smoothing setting.
Show Weightmaps	When enabled, selected Weight Maps are visible as overlays to the shaded elements in the 3D viewport in **Default** viewport display style. This includes the automatic weights associated with bones and deformers.
Show PSUB Masks	Toggles the visibility of the sculpting masks in the viewport when working with multi-resolution sculpting. When a masks visibility is disabled, the mask still affects the target unless the **Enable Mask** option is inactive. For more information, see Using Sculpting Masks.
Show UV Boundaries	Toggles visibility of UV boundaries in the 3D viewport. These display as thin purple lines overlaid on the surface, representing the outline of each UV island. The default color can be changed in the **System** > **Preferences** > **Color** options.

Show Geometry Boundaries	Toggles visibility of Geometry boundaries. These display as a thin green line on any open edges of a surface. Especially useful in highlighting hidden surface discontinuities.
Show UV Distortion	Toggles visibility of the colors displayed for the geometry with signifying distorted areas. Enabling this option displays a color overlay on the geometry, fading toward red when polygons are distorted and smaller (relative to other polygons in the map) and fading toward blue when they are distorted and larger (relative to other polygons). Polygons that are the closest relative size to others in the map with the least distortion display a middle gray/green color.  3D Viewport Properties Shading Shading Style: Default Wireframe Overlay: Colored Wireframe Opacity: 10.0 % Wireframe Color: 0.0 0.0 0.0 Vertex Map: Selection ☐ Smooth Shade ☐ Show Weightmaps ☐ Show PSUB Masks ☐ Show UV Boundaries ☐ Show Geometry Boundaries ☑ Show UV Distortion Display Vertex Map Display: ... ☐ Independent Point Size Point Size: 0.0 Show Vertices: Never ☐ Show Cages ☐ Show Guides ☐ Show Indices ☐ Show Weight Values ☐ Enable Independent Drawing Geometry Snap : Teapot Primitive Mesh Polygons : Subdiv Channels: 0 Deformers: ON GL: 47,152 50 mm Perspective Default Ray GL: Off Viewport Textures Drawing and Control Visibility HUD Advanced Options Active Meshes Inactive Meshes 'Ctrl-Tab' to toggle
Show Locked Mesh Items	Displays locked polygons, edges, and vertices in the 3D viewport in a specified color. This option is enabled by default. For more information, see Locking.

Display

Vertex Map Display	Sets which color component of the selected RGBA map is drawn in the Vertex map mode. The following options are available: • RGB • Alpha • Red • Green

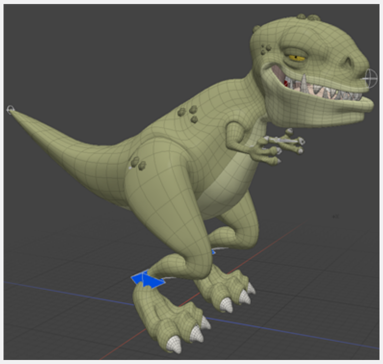
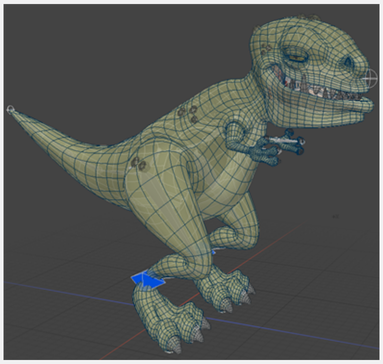
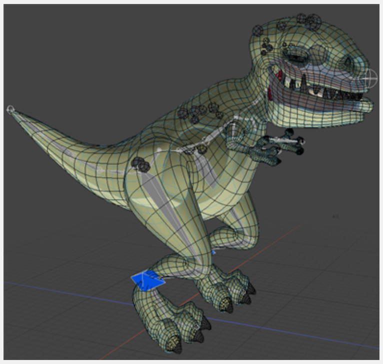
	• Blue **Note:** The **Shading** > **Shading Style** must be set to **Vertex Map** to set the Vertex Map display.
Independent Point Size	This is a per-viewport override for the display size of selected vertices. When enabled, you can specify the actual display size with the **Point Size** option.
Point Size	Determines the display size of selected vertices when the **Independent Point Size** option is enabled. Values are in screen pixels.
Show Vertices	Allows you to set visibility in the selected 3D viewport. The following options are available: • **Always** - Always show vertices. • **In Component Modes** - Only show vertices in Component mode. • **In Vertex Mode** - Only show vertices in Vertex Component mode. • **Never** - Never show vertices.
Edge Thickness	Controls the thickness of edges on active meshes. Increasing the default value of 1.0 can help you see edges more easily at different zoom levels. 
Curve Thickness	Controls the thickness of active curves. Increasing the default value of 1.0 can help you see curves more easily at different zoom levels.

	
Show Cages	Toggles the visibility of the Subdivision Surface Cage representation. This can be considered the original polygons that were smoothed.
Show Guides	Toggles the visibility of guide lines that draw between the subdivision limit surface and the cage geometry.
Show Indices	Toggles the visibility of the component index values display. Each number represents the particular component index. Visibility can be especially helpful when working with Geometry Constraints, where Constraints require specific index values for input.
Show Weight Values	Toggles the visibility of Weight Map value display. A numeric value representing the current vertex weight amount displays next to each selected vertex in the 3D viewport. See Working with Vertex Maps for Weight Map usage information.
Enable Independent Drawing	The Display Viewport allows you to define custom display options for each item layer. By default, the **Enable Independent Drawing** option is enabled and these custom drawing options display in all viewports. When enabled, all custom drawing options are ignored for active meshes.

Transparency Overrides

Transparency Mode	Allows you to differentiate between background and foreground wireframes, so you can easily understand the topology you're working with. This option is especially useful for decluttering heavy wireframe models.

The following options are available:

Normal	The default setting. Transparency overrides are disabled.	
Ghost	Front-facing polygons are drawn semi-opaque with a specular highlight. Back-facing polygons are drawn 100% opaque.	
X-Ray	Front-facing polygons are transparent with a specular highlight. Back-facing polygons are drawn 100% opaque. This mode allows you to easily select the joints in a rigged character, for instance.	

Model courtesy of William Vaughan

Note: Transparency overrides are not available in **Wireframe** mode. If you want to use transparency overrides on inactive meshes, make sure that their shading style is NOT set to **Wireframe** under **3D Viewport Properties** > **Inactive Meshes** > **Shading** > **Shading Style**. Transparency overrides are available in the **Advanced** viewport, as of Modo 14.1v1.

Polygon Color (front)	Specifies the color of front-facing polygons in **Ghost** or **Xray** mode.
Polygon Opacity (front)	Specifies the opacity value of front-facing polygons in **Ghost** mode.
Wireframe Color (front)	Specifies the color of front-facing wireframes in **Ghost** or **Xray** mode.
Polygon Color (back)	Specifies the color value of back-facing polygons in **Ghost** or **Xray** mode.
Polygon Opacity (back)	Specifies the opacity value of back-facing polygons in **Ghost** or **Xray** mode.
Wireframe Color (back)	Specifies the color of back-facing wireframes in **Ghost** or **Xray** mode.
Wireframe Opacity (back)	Specifies the opacity value of the wireframe in **Ghost** or **Xray** mode.
Overlay	When enabled, the visible front faces of the mesh are overlaid on top of everything else.
Link Wire Opacity	When enable, the front wireframe opacity is linked to the back wireframe opacity so that the two can be controlled together.
Flat Shaded	When enabled, meshes are rendered with self illumination and without lighting effects.

Inactive Meshes

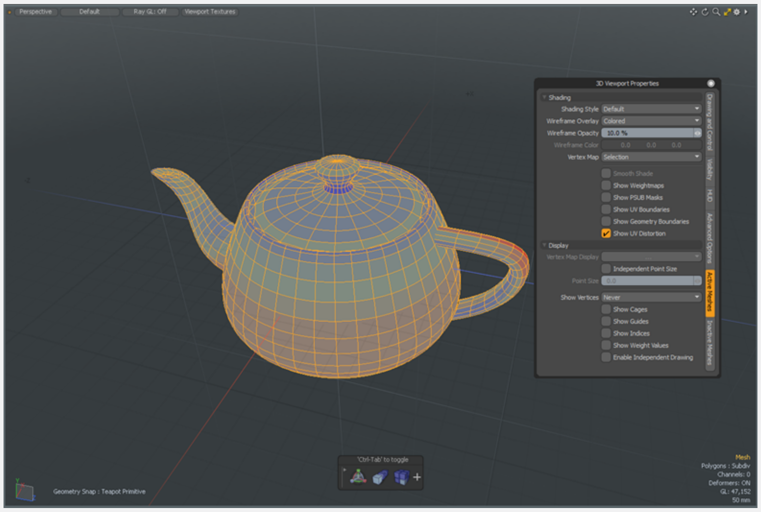
The following options allow you to control the display of item layers that are in the background, meaning that they are visible in the Items list, but unselected (not highlighted).

Overrides

Make Inactive Same as Active	When enabled, this option overrides all settings of the **Inactive Meshes** options panel and all background item layers display as foreground elements.
Make Inactive Invisible	When enabled, all item layers that are visible according to the visibility column of the Items list, but unselected don't show in the 3D viewport. This makes it easy to solo a single layer by simply selecting it without affecting the rendered image or requiring multiple layers' visibility to be toggled.

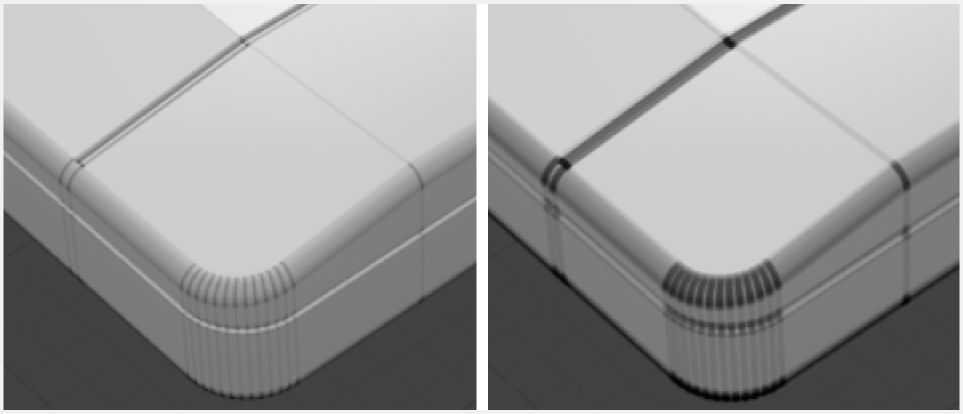
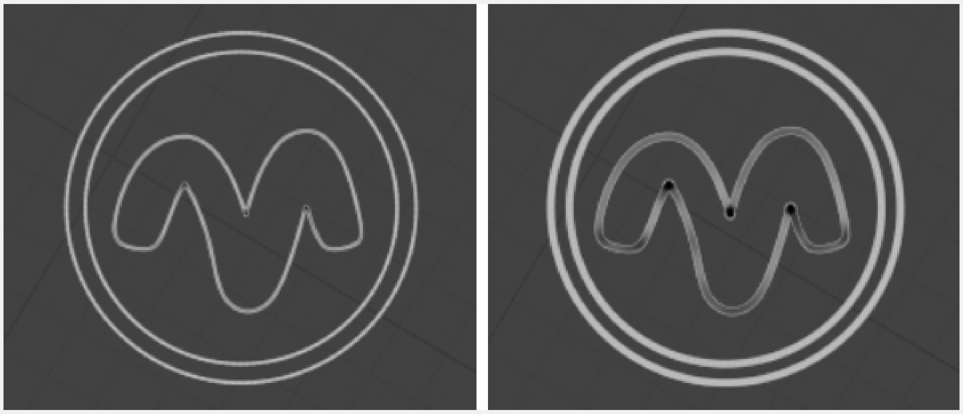
Shading

Shading Style	You can choose from the available view styles covered above: Wireframe, Solid, Shaded, Vertex Map, Texture, Shaded Texture, Default, Advanced, Gooch, Cel and Reflection Shading.
Wireframe Overlay	Determines the style of geometry wireframe display. You can adjust the colors of the wireframe using the preference setting on Windows and Linux in **System** >**Preferences** > **Display** > **Colors** and on Mac OS in **Menu** >**Preferences** > **Display** > **Colors**. Once a color scheme is created in the preferences and saved as a named preset, it must be applied to the active viewport with the menu bar command **View** > **Viewport Color Scheme**. • **None** - No wireframe displayed. (**/**) • **Uniform** - By default this is a white wireframe outline. (**Shift**+**/**) • **Colored** - By default this is a black wireframe outline. (**Ctrl**+**/**)
Wireframe Opacity	Controls the opacity of the wireframe overlay display.
Wireframe Color	Defines which color to display.
Vertex Map	Defines which type of Vertex Map is displayed when in the **Vertex Map** viewport. • **Selection** - Displays the most recent Vertex Map selection. • **Weight** - Favors Weight Maps for display. • **Vertex Color** - Favors vertex color maps for display.
Smooth Shade	This toggle overrides the material Smoothing setting, only changing how smoothing is displayed in the viewport, not the rendered output. When disabled, individual polygons shade flat, appearing faceted. When enabled, polygons shade according to the material Smoothing setting.
Show Weightmaps	When enabled, selected Weight Maps are visible as overlays to the shaded elements in the 3D viewport in **Default** viewport display style. This includes the automatic weights associated to bones and deformers.
Show PSUB Masks	Toggles the visibility of the sculpting masks in the viewport when working with multi-resolution sculpting. When a masks visibility is disabled, the mask still affects the target unless the **Enable Mask** option is inactive. For more information, see Using Sculpting Masks.

Show UV Boundaries	Toggles visibility of UV boundaries in the 3D viewport. These display as thin purple lines overlaid on the surface, representing the outline of each UV island. The default color can be changed in the Color Preferences section.
Show Geometry Boundaries	Toggles visibility of Geometry boundaries, These display as a thin green line on any open edges of a surface. Especially useful in highlighting hidden surface discontinuities.
Show UV Distortion	Toggles visibility of the colors displayed for the geometry with signifying distorted areas. Enabling this option displays a color overlay on the geometry, fading toward red when polygons are distorted and smaller (relative to other polygons in the map) and fading toward blue when they are distorted and larger (relative to other polygons). Polygons that are the closest relative size to others in the map with the least distortion display a middle gray/green color.  3D Viewport Properties Shading Shading Style: Default Wireframe Overlay: Colored Wireframe Opacity: 10.0 % Wireframe Color: 0.0 0.0 0.0 Vertex Map: Selection ☐ Smooth Shade ☐ Show Weightmaps ☐ Show PSUB Masks ☐ Show UV Boundaries ☐ Show Geometry Boundaries ☑ Show UV Distortion Display Vertex Map Display: ... ☐ Independent Point Size Point Size: 0.0 Show Vertices: Never ☐ Show Cages ☐ Show Guides ☐ Show Indices ☐ Show Weight Values ☐ Enable Independent Drawing Perspective Default Ray GL: Off Viewport Textures Drawing and Control Visibility HUD Advanced Options Active Meshes Inactive Meshes 'Ctrl-Tab' to toggle Geometry Snap : Teapot Primitive Mesh Polygons : Subdiv Channels: 0 Deformers: ON GL: 47,152 50 mm
Apply Current Morph	When a Morph Map is selected, the Morph is applied to all visible meshes, in active layers and inactive layers. The **Apply Current Morph** option toggles the display of Morphs for background (inactive) layers.

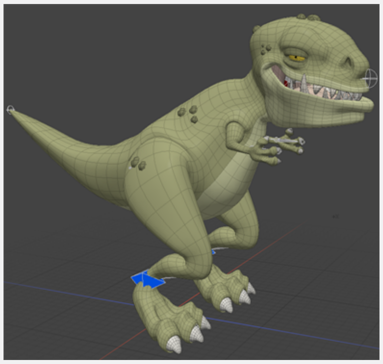
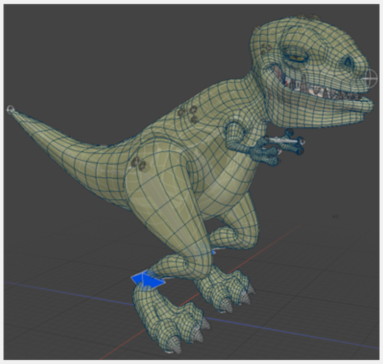
Display

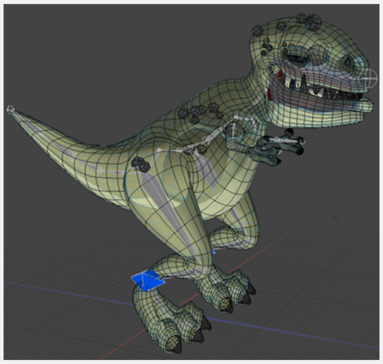
Independent Point Size	This is a per-viewport override for the display size of selected vertices. When enabled, you can specify the actual display size with the **Point Size** option.

Point Size	Determines the display size of selected vertices when the **Independent Point Size** option is enabled. Values are in screen pixels.
Show Vertices	Toggles the visibility of unselected vertices (selected vertices are always visible, unless **Show Selection** is disabled).
Edge Thickness	Controls the thickness of edges on inactive meshes. Increasing the default value of 1.0 can help you see edges more easily at different zoom levels. 
Curve Thickness	Controls the thickness of inactive curves. Increasing the default value of 1.0 can help you see curves more easily at different zoom levels. 
Show Cages	Toggles the visibility of the Subdivision Surface Cage representation. This can be considered the original polygons that were smoothed.
Show Guides	Toggles the visibility of guide lines that draw between the subdivision limit surface and the cage geometry.
Enable Independent Drawing	The **Display** viewport allows you to define custom display options per item layer. By default, these custom drawing options display in all viewports.

	However, when the **Enable Independent Drawing** option is enabled, all custom drawing options are ignored for active meshes.

Transparency Overrides

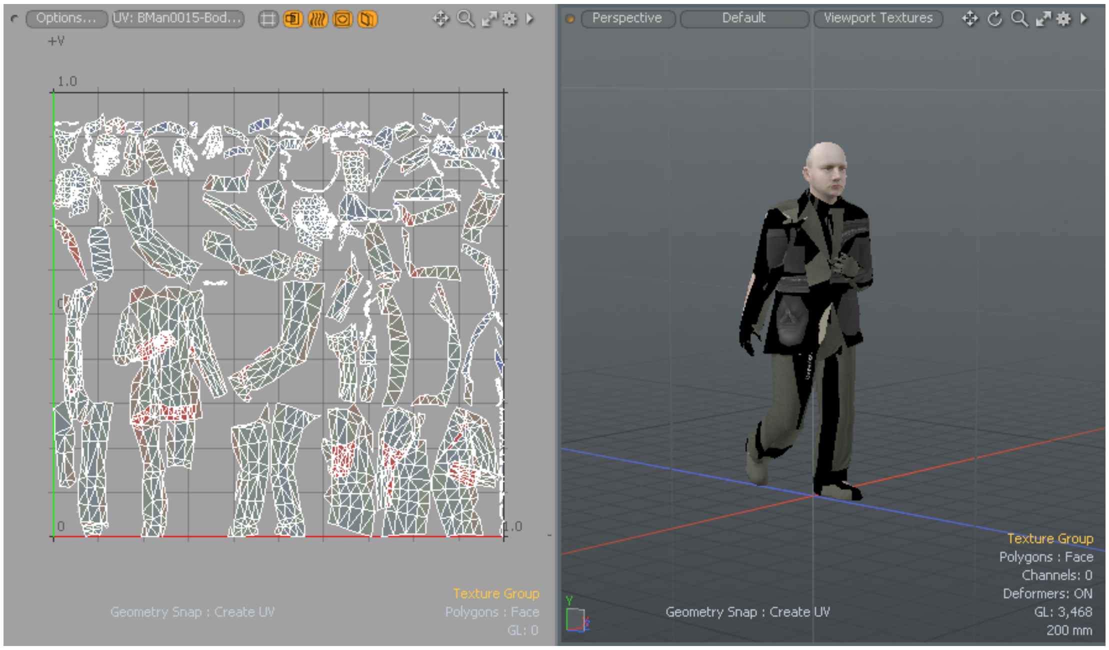
<table>
<tr>
<td>Transparency Mode</td>
<td>
<p>Allows you to differentiate between background and foreground wireframes, so you can easily understand the topology you're working with. This option is especially useful for decluttering heavy wireframe models.</p>
<p>The following options are available:</p>
<table>
<tr>
<td>Normal</td>
<td>The default setting. Transparency overrides are disabled.</td>
<td></td>
</tr>
<tr>
<td>Ghost</td>
<td>Front-facing polygons are drawn semi-opaque with a specular highlight. Back-facing polygons are drawn 100% opaque.</td>
<td></td>
</tr>
</table>
</td>
</tr>
</table>

	X-Ray Front-facing polygons are transparent with a specular highlight. Back-facing polygons are drawn 100% opaque. This mode allows you to easily select the joints in a rigged character, for instance.  *Model courtesy of William Vaughan* **Note:** Transparency overrides are not available in **Wireframe** mode. If you want to use transparency overrides on inactive meshes, make sure that their shading style is NOT set to **Wireframe** under **3D Viewport Properties** > **Inactive Meshes** > **Shading** > **Shading Style**. Transparency overrides are available in the **Advanced** viewport, as of Modo 14.1v1.
Polygon Color (front)	Specifies the color of front-facing polygons in **Ghost** or **Xray** mode.
Polygon Opacity (front)	Specifies the opacity value of front-facing polygons in **Ghost** mode.
Wireframe Color (front)	Specifies the color of front-facing wireframes in **Ghost** or **Xray** mode.
Polygon Color (back)	Specifies the color value of back-facing polygons in **Ghost** or **Xray** mode.
Polygon Opacity (back)	Specifies the opacity value of back-facing polygons in **Ghost** or **Xray** mode.
Wireframe Color (back)	Specifies the color of back-facing wireframes in **Ghost** or **Xray** mode.
Wireframe Opacity (back)	Specifies the opacity value of the wireframe in **Ghost** or **Xray** mode.
Overlay	When enabled, the visible front faces of the mesh are overlaid on top of everything else.
Link Wire Opacity	When enable, the front wireframe opacity is linked to the back wireframe opacity so that the two can be controlled together.

Flat Shaded	When enabled, meshes are rendered with self illumination and without lighting effects.

UV Viewport

The **UV viewport** is much like the 3D (OpenGL) Viewport. It displays the 3D space as well as a two dimensional view of UV space. UV maps are the flat, 2D representations of a model's 3D surface for the application of texture maps (images). For more information about texture maps, see Working with UV Maps.


Options...
UV: BMan0015-Bod...
+V
1.0
0
1.0
Texture Group
Polygons : Face
GL: 0
Geometry Snap : Create UV
Perspective
Default
Viewport Textures
Texture Group
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,468
200 mm
Geometry Snap : Create UV
Y

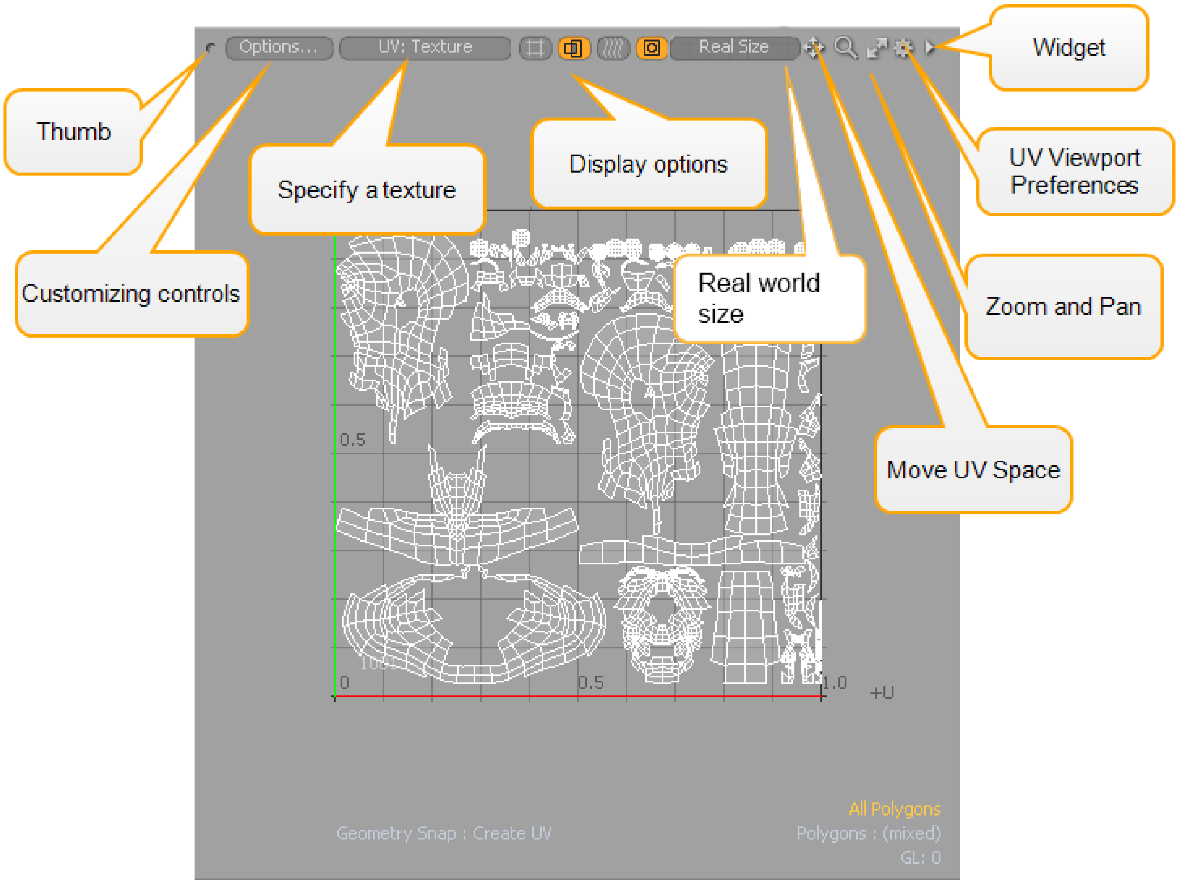
To open the UV Viewport:

- From the menu bar, click **Layout** > **Layouts** > **UV**.
- In the switcher bar, disable the Starred Only button and click **UV**.
- To work in a standalone UV Editor, click **Texture** > **Open UV Editor**.

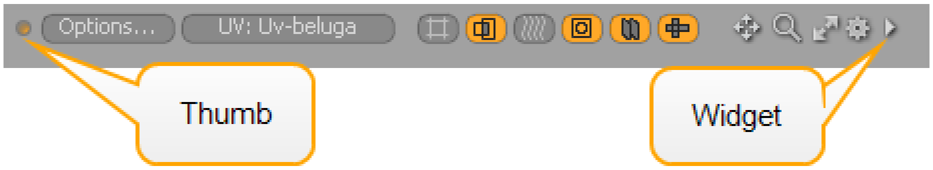
For more information, see Working in the UV Editor.

Working in the UV Viewport

The UV viewport displays the UV space in a 2D UV grid with a flat background color. You can customize the UV viewport by using the buttons and controls.


Options...
UV: Texture
Real Size
Thumb
Customizing controls
Specify a texture
Display options
Real world size
Widget
UV Viewport Preferences
Zoom and Pan
Move UV Space
0.5
0
0.5
1.0
+U
All Polygons
Geometry Snap : Create UV
Polygons : (mixed)
GL: 0

Thumb and Widget Controls


Options...
UV: Uv-beluga
Thumb
Widget

To detach, clone, split, or scale the viewport:

- Click and drag the **Thumb** control to detach.
- Right-click to select a viewport option.

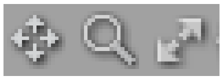
Tip: If the UV Viewport is not visible, double-click on the **UV** layout tab to restore the default settings.

To change the viewport preset:

- Click the **Widget** control and select another viewport style.

For more information, see Using Layout Controls.

Move, Magnify, and Scale Controls



- Click and drag to move, magnify, and scale the contents in the viewport, making it easier to see certain areas for detailed editing.

Available shortcuts:

- **Ctrl**/**Cmd** +**Alt**+click - zoom to mouse position.
- **Alt**+click - pan view.

Mouse wheel - zoom to mouse position.

- **Shift**+**Alt**+middle-click - pan horizontally (U direction).
- **Shift**+**Alt**+right-click - pan vertically (V direction).
- **Ctrl**/**Cmd** +**Alt**+right-click - box zoom (drag to define zoom area).
- **A** - fits and centers geometry to the viewport size.
- **Shift**+**A** - zoom to selection.
- **.** (period) or **,** (comma) - zoom in and out.
- **G** - center current mouse position.

Customize Viewport Controls

- Click **Options** to change the Customizing the UV Viewport.

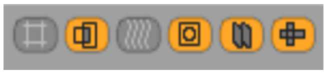
Replace or Create a Texture

- Click **UV: (none)** to replace or use a new UV texture on the mesh surface. The text in the button displays the name of the UV texture selected.

For information about how to create a UV Texture map, see Create UV Tool.

Note: When applying **UV Textures** to a **Preset** mesh, the UV Map name for the UV Texture must be the same as the UV Map name used by the Preset. For more information about presets, see Preset Browser. For more information about renaming UV Maps, see Working with UV Maps.

Change Display Options



Click a display icon to update the UV viewport.

- Auto Spans - draws the UV space with grids regardless to the size of the patches.
- Show Overlap - displays a red highlight to indicate the overlapping UV state.
- Show Distortion - display the distortion colors. For more information, see Displaying Distorted UVs.
- Show Image - displays a background reference image map. For more information, see Modeling with Images.
- Show Inactive UVs - toggles the visibility of UVs for items not selected in the Items list.
- Show Wireframe - toggles the visibility of wireframes.

For more detailed information, see Customizing the UV Viewport.

Adding Real World Size to UV Maps

Real world materials are often scanned and imported into 3D applications in real world sizes to ensure they are applied at the correct scale to the 3D model. For example, you could have a scanned square of material that is 10cm wide and high which you want to appear at the correct scale when applied to your models.

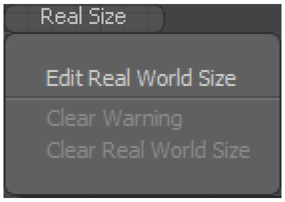
In Modo, you can define the real world size of your UV maps to ensure scanned materials and AxF materials are applied to your meshes at the right scale.

To define the real world size of a UV map:

1. Click the **UV** layout so you can see the UV viewport.
2. Select the UV map where you want to assign a real world size.

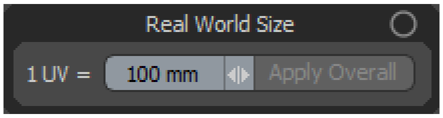
Note: See Replace or Create a Texture to learn how to create or select UV maps.

3. Click the **Real Size** button at the top of the UV viewport, and select **Edit Real World Size**.



4. Enter a value in millimeters for the real world size of each UV tile.

   For example, if you want each UV tile in the UV map to be 10cm, enter a value of **100mm**.



Note: If the **Apply Overall** button is disabled, it means you have not selected a UV map.

5. Click **Apply Overall**.

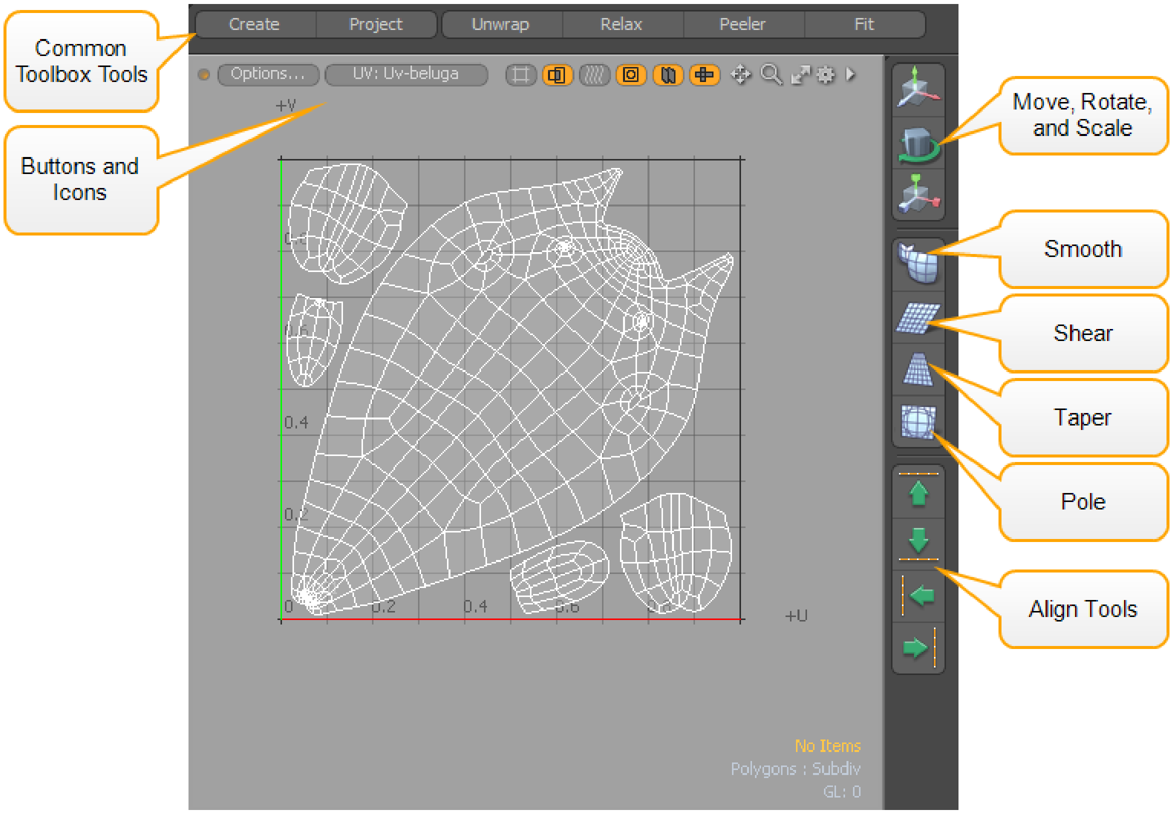
   The size is assigned to the UV map, and the size of the selected UV map appears at the top of the UV viewport.

To clear existing real world sizes on selected UV maps, click the **Real Size** button and select **Clear Real World Size**.

Note: See AxF Material and Texture Locator for more information about using the real world size on textures and materials in Modo.

Working in the UV Editor

Use the UV Editor as a standalone viewport while working in another layout, such as **Model**. Common tools are available in this viewport. To open the UV Editor, click **Texture** > **Open UV Editor**.


Common Toolbox Tools
Buttons and Icons
Create
Project
Unwrap
Relax
Peeler
Fit
Options...
UV: Uv-beluga
+V
+U
0.4
0.2
0
0.2
0.4
No Items
Polygons : Subdiv
GL: 0
Move, Rotate, and Scale
Smooth
Shear
Taper
Pole
Align Tools

For information about the common tools located on the right of the UV Editor, see Move, Shear, Smooth, Taper, and Align UV

Toggling Visibility

Generally, you can toggle visibility of items in the Items (Scenes) List, which changes visibility of the entire layer. Commands are also available to temporally toggle the visibility for selected items. Once hidden, geometry is uneditable, meaning it cannot be selected or moved.

- In the 3D viewport, select the geometry component you wish to hide and click **View** > **Hide Selected** or press **H**.
- To make the geometry visible again, click **View** > **Unhide** command or press **U**.

When a file is saved and re-opened later, all hidden polygons are visible.

Tip: By dragging over an object, you can paint a selection of one of more items. If you right-click and drag, you can draw a lasso around the selection area to select multiple items at once.

Visibility Shortcuts

- **H** - hide selected (hides all if nothing is selected).
- **Shift**+**H** - hide unselected (hides all unselected geometry).
- **Ctrl**/**Cmd** +**H** - hide invert (toggles visibility, inverting the present state).
- **U** - unhide (makes all hidden geometry visible).

Locking Geometry

Locking provides a similar function to hiding, allowing you to fix a selection of geometry so that it is immovable and un-selectable, effectively un-editable. However, it is still fully visible. Unlike hiding, this locking function is limited to the component selection modes.

- Select geometry and click **Edit** > **Lock Selected** or press **J**.
- You can unlock any locked element in an item layer with **Edit** > **Unlock** or by pressing **I**.

Locking Shortcuts

- **J** - lock selected (locks all if nothing is selected).
- **Shift**+**J** - lock unselected (lock all of the unselected geometry).
- **Ctrl**+**J** - lock invert (toggles locked state of all geometry).
- **I** - unlock (makes all locked geometry editable).

Displaying Flipped UVs

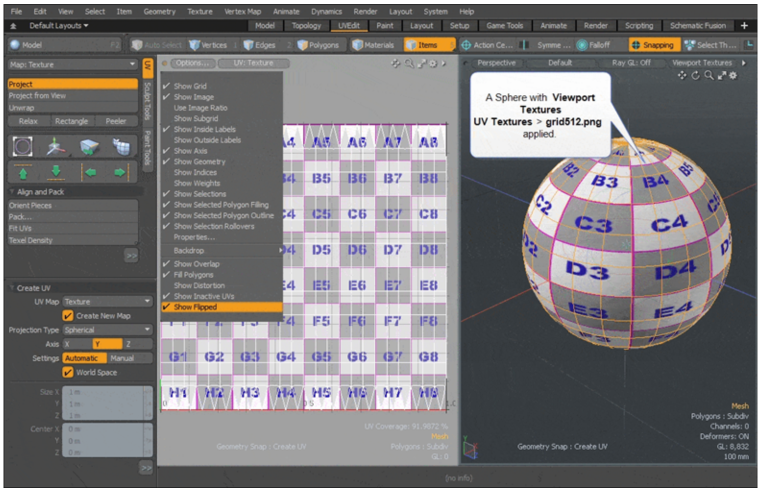
You can display the orientation of your UV shells within the UV viewport. Flipping a UV shell is useful if you need to correct a texture that is reversed.

From the **Options** button, enable the **Show Flipped** option in the context menu to display flipped UVs in yellow. You can also display flipped UVs with overlapped UVs. If both are present, the color is blended.

Tip: The **Show Flipped** option is also available in the UV Editor (**Texture** > **Open UV Editor**).

Flipped UVs may occur when you Mirror selected geometry. The following example displays flipped UVs and how to correct their orientation. For more information about how to use the Flip UV tool, see Flip UV.

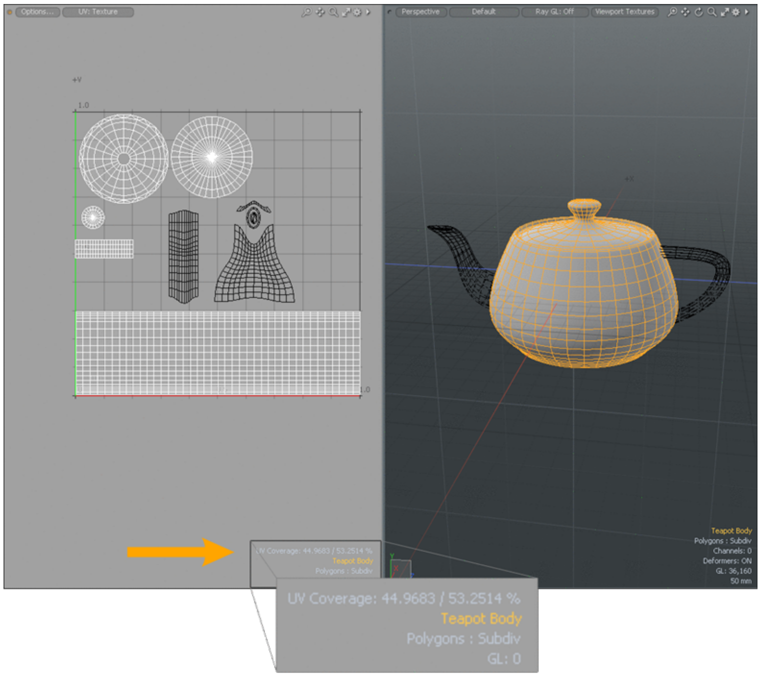
Click the image below to view an animation.



A Sphere with **Viewport Textures**
UV Textures > **grid512.png** applied.

Reviewing UV Coverage

When you create a UV map, the UV coverage indicator displays a percentage value in the lower right corner of the **UV viewport** and **UV Editor**, representing the amount of uniform UV space of the total region that is covered by geometry. This information is useful when trying to maximize the amount of texture space being used by the target surface.



UV Coverage: 44.9683 / 53.2514 %
Teapot Body
Polygons : Subdiv
GL: 0

If you select only a few geometry items in your scene, the UV coverage indicator displays a percentage value representing the coverage used for the selected items and a percentage value representing the coverage used by the remaining items on your UV map. For example, when viewing the background and foreground items in a UV map, the active selected geometry is displayed in white, while the inactive geometry is displayed in black. Clicking on either item displays the coverage usage for that selection.

Selecting Items

You can select individual UV maps in the **UV viewport** and in the **UV Editor** using paint selection and lasso selection while in Items Selection Mode selection mode.

> **Tip:** By dragging over an object, you can paint a selection of one of more items. If you right-click and drag, you can draw a lasso around the selection area to select multiple items at once.

If you have individual UV maps using the same name for each mesh item in your scene, the active UV map displays in white in the UV viewport and the non-active (not selected) UV maps display in black.

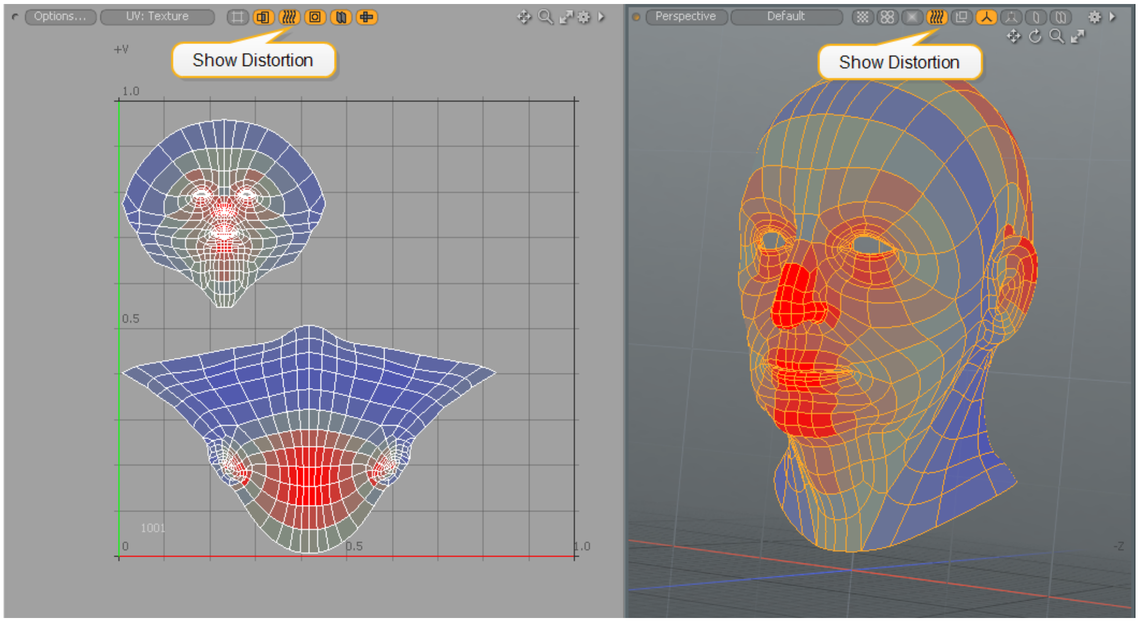
- You can use the draw selection or lasso selection to activate individual UV maps.
- You can also make your selection in the 3D viewport by clicking the UV wireframe.

Displaying Distorted UVs

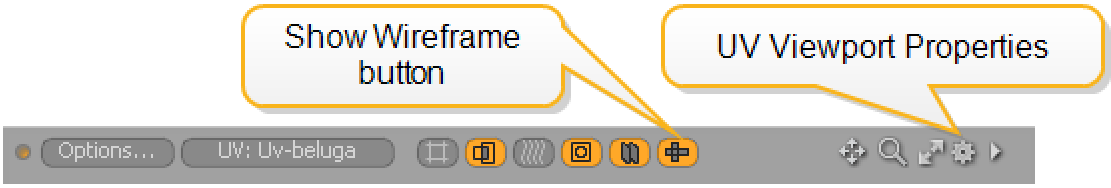
Two main goals when working with UV maps are to eliminate or reduce seams, and to minimize distortion from stretching. Unfortunately, these two are not mutually exclusive. Distortion often happens with UV mapping.

To enable distortion visibility, click the Show Distortion ▨ button or open the UV Viewport Properties, and enable **Visibility** > **Show Distortion**.

Enabling the UV Viewport option displays a color overlay on the geometry, fading toward red when polygons are distorted and smaller (relative to other polygons in the map) and fading toward blue when they are distorted and larger, relative to other polygons. Polygons that are the closest relative size to others in the map with the least distortion display a middle gray/green color. You can also display the distortion colors in the 3D viewport by clicking the Show Distortion ▨ button at the top of the viewport.


Options...
UV: Texture
Show Distortion
+V
1.0
0.5
1001
0
0.5
1.0
Perspective
Default
Show Distortion
-Z

When you're working with dense meshes, it might be difficult to see the distortion map. In such cases, you can disable the wireframe visibility by clicking the Show Wireframe button at the top of the viewport, or disabling **Show Wireframe** in **UV Viewport Properties** > **Visibility**. For more information on properties, see Customizing the UV Viewport.


Show Wireframe button
UV Viewport Properties
Options...
UV: Uv-beluga

Customizing the UV Viewport

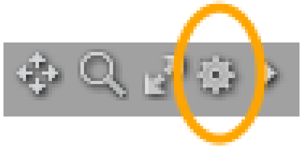
You can select from a variety of display customization options from the UV viewport **Properties** menu.

To open the properties:

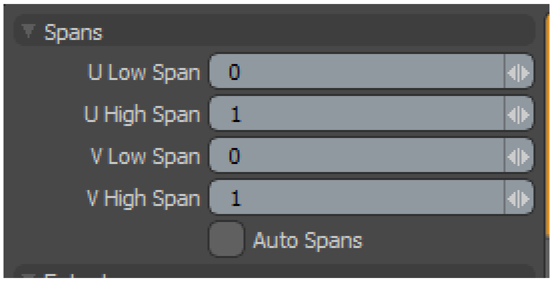
- Click the gear ⚙ icon in the upper-right corner.
- Hold the mouse pointer over the viewport and press **O**.
- Click Options... to set common viewport options.

Drawing and Control Options

Click the **Gear** icon on the top of the UV viewport to open the UV Viewport Properties.

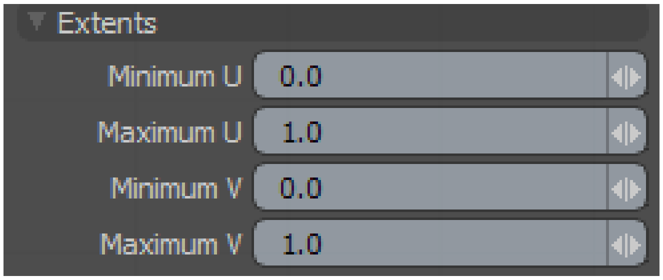


Spans


Spans
U Low Span 0
U High Span 1
V Low Span 0
V High Span 1
☐ Auto Spans

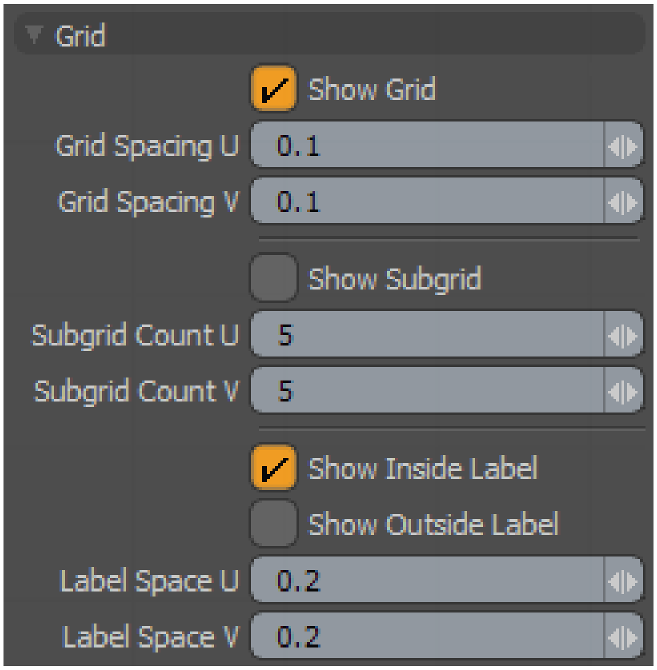
U/V High/Low Spans	Controls the number of UV sections or spans displayed in the viewport. Each span represents a complete UV grid section. The default value produce the grid found most commonly in 3D programs representing, the -1 to 1 UV space coordinates. You can place vertex data anywhere within UV space; this option simply determines the display extent of the UV grid.
Auto Spans	The **Auto Spans** option fills the UV space with grids.

Extents


Extents
Minimum U 0.0
Maximum U 1.0
Minimum V 0.0
Maximum V 1.0

Minimum/Maximum U/V	If you have **Show Image** enabled with an image selected, you can control the UV area where your image map is displayed.

Grid



Grid

☑ Show Grid

Grid Spacing U 0.1

Grid Spacing V 0.1

☐ Show Subgrid

Subgrid Count U 5

Subgrid Count V 5

☑ Show Inside Label

☐ Show Outside Label

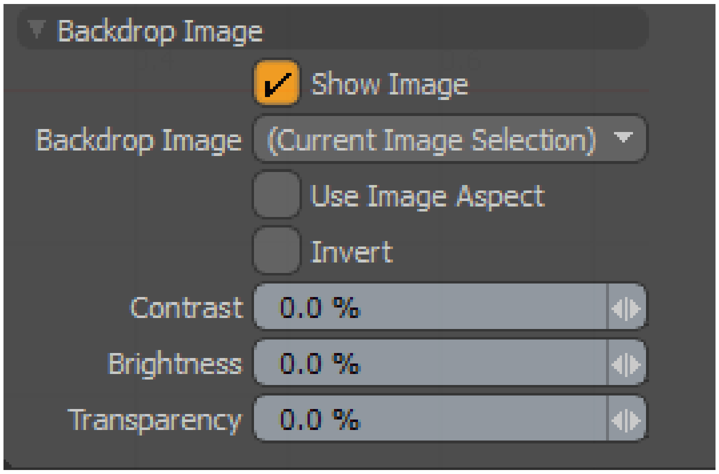
Label Space U 0.2

Label Space V 0.2

Show Grid	Toggles the visibility of the UV grid in the viewport.
Grid Spacing U/V	Sets the grid line spacing using these values, defined as decimals of the UV unit. The default value of 0.1 draws grid lines every 10th of a unit.
Show Subgrid	Displays a lighter secondary grid. Use this option for detailed maps, as the standard grid may not have enough sub-divisions to be visible when zoomed in.
Subgrid Count U/V	Determines the number of divisions between each **Grid** sub-division.
Show Inside Label	Toggle the grid value labels inside the grid area.
Show Outside Label	Toggle the grid value labels outside the grid area.
Label Space U/V	Determines the division level for the grid labels. Typically this value matches or is a division of the **Grid Spacing** value.

Grid Spacing U/V	Sets the grid line spacing using these values, defined as decimals of the UV unit. The default value of 0.1 draws grid lines every 10th of a unit.
Show Subgrid	Displays a lighter secondary grid. Use this option for detailed maps, as the standard grid may not have enough sub-divisions to be visible when zoomed in.
Subgrid Count U/V	Determines the number of divisions between each **Grid** sub-division.

Backdrop Image


Backdrop Image

☑ Show Image

Backdrop Image: (Current Image Selection)

☐ Use Image Aspect

☐ Invert

Contrast: 0.0 %

Brightness: 0.0 %

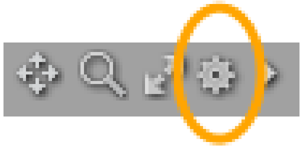
Transparency: 0.0 %

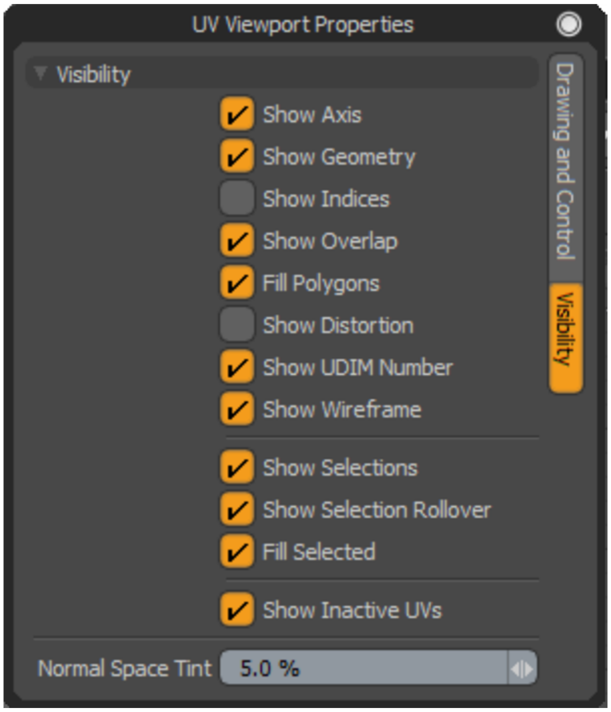
Show Image	Toggles the visibility of a background reference image map. The image displayed is determined by the **Backdrop Image** settings.
Backdrop Image	Select from a list of loaded images (those that are available in the Clips/Image list).] When the option is set as **Current Image Selection**, the current image selected is used as the backdrop. Images can be selected in the **Images** tab or within the **Shader Tree**.
Use Image Aspect	When enabled, the UV window is scaled horizontally to match the aspect ratio of the selected image, making it easier for you to edit UV maps for non-square aspect ratios.
Invert	Inverts the RGB color values of the backdrop image, producing a negative

	effect.
Contrast	Sets the visual difference between light and dark values in an image. Values greater than 0% increase the contrast of the backdrop image, while values lower than 0% decrease it.
Brightness	Sets the degree or amount of light or luminosity in an image. Values greater than 0% increase the brightness of the backdrop image, while values lower than 0% decrease it.
Transparency	Determines the opacity of the backdrop image. Values greater than 0% increase the transparency, ramping toward 100%, where the image becomes completely transparent.

Visibility Options

Click the **Gear** icon on the top of the UV viewport to open the UV Viewport Properties.





UV Viewport Properties

Visibility

- ☑ Show Axis
- ☑ Show Geometry
- ☐ Show Indices
- ☑ Show Overlap
- ☑ Fill Polygons
- ☐ Show Distortion
- ☑ Show UDIM Number
- ☑ Show Wireframe
- ☑ Show Selections
- ☑ Show Selection Rollover
- ☑ Fill Selected
- ☑ Show Inactive UVs

Normal Space Tint 5.0 %

Show Axis	Toggle the colored axes displayed in the viewport.
Show Geometry	Toggle the displayof all geometry in the viewport.
Show Indices	Toggle the display of vertex index values for selected items.
Show Overlap	When the area of a polygon sits over the area of another, enabling this option displays a red highlight to indicate the overlapping state.
Fill Polygons	Toggle the visibility of the faint fill coloring given to polygon faces.
Show Distortion	Toggle the display of the distorted UVs. When enabled, colors overlay onto the geometry signifying distorted areas of the UV map. For more information, see Displaying Distorted UVs.
Show UDIM Number	Toggle the displays the UDIM address at the center of each UDIM tile. The UDIM number is displayed in the bottom left corner of the UV space.
Show Wireframe	Toggle wireframe display. When working with dense meshes with UV distortion enabled it can be difficult to see the distortion map. This option is useful to make dense UV maps easier to see.
Show Selections	Toggle the display of highlighted polygon selections (by default, an orange color).
Show Selection Rollovers	Toggle the display of selection rollover.
Fill Selected	Toggle visibility of the orange fill color for selected polygons.
Show Inactive UVs	Toggle the display of UV maps for items in the background (visible but not selected in the Items list).
Normal Space Tint	Controls the amount of tint coloration applied to the areas in UV space that don't contain UV patches, identifying the central or main UV area of 0 to 1. This is where textures originate; outside of this area the texture simply tiles.

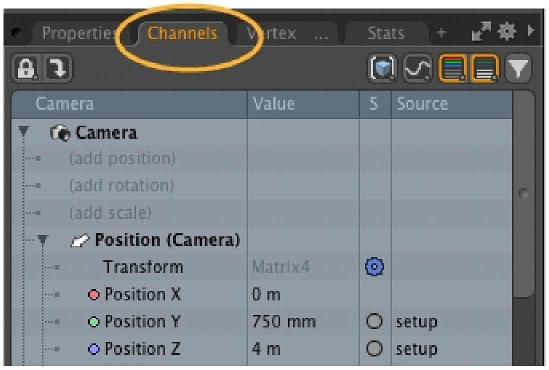
List and Info Viewports

The List and Info viewports provide an intuitive way for you to select, arrange, and group the various elements that make up a scene. Outside of simply selecting items, viewports, such as the **Item List**, provide important functions for setting up hierarchies for animation. The **Shader Tree** is the necessary component for assigning surfacing attributes.

List and Info Viewports	
Items (Scenes) List	The **Item List** has a number of functions, including listing all of the locator-type items within your project, as well as allowing you to create, edit, and organize the various item layers.
Shader Tree	The **Shader Tree** is the main interface for adding materials and textures to surfaces in your scene.
Vertex Map List	The **Vertex Map List** shows Vertex Maps for the current mesh in groups: **UV**, **Weight**, **Morph**, and **Other**.
Channels Viewport	The **Channel** viewport shows all channels of any item and provides a direct method for selecting channels, as well as creating, navigating, and managing any keyframes on those channels.
Images/Clips List	The **Clip List** (also referred to as the **Image List**) viewport provides direct access to all images loaded into Modo, organized by scene.
Clip Browser	The **Clip Browser** allows you to view and select images loaded into Modo, for use in a variety of situations. When re-using images that have been loaded into Modo, you can browse through them all in this viewport.
Properties Panel	The **Properties** viewport is basically an empty frame until an item or tool is selected but, once a tool is selected, all the editable attributes related to that item are listed in the panel. You can then make changes to the options available and, if applicable, see them update in the viewport or render previews.
Info and Statistics	The **Info and Statistics** viewport provides information about the currently-selected or active geometry in the scene. It also provides powerful selection and precise editing functionality.
Command History	Modo keeps a listing of every command used during a session in the **Command History** viewport. This view is useful for quick undo and redo operations, managing macros, assigning keyboard shortcuts to specific commands, and other utility functions.

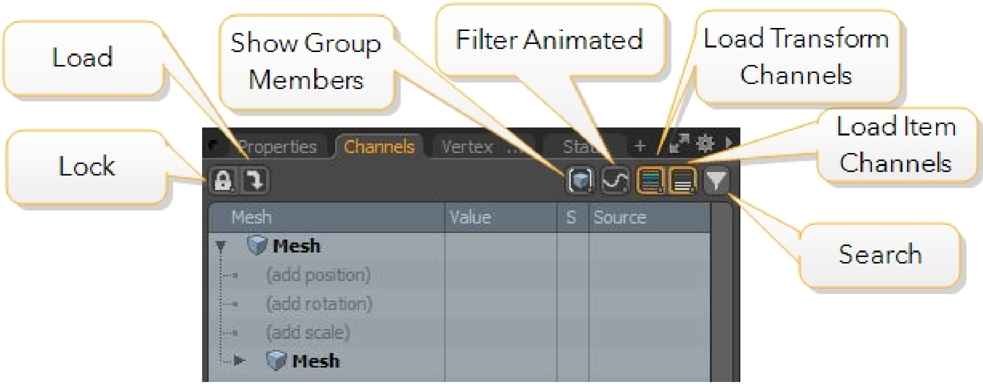
Channels Viewport

The **Channels** viewport provides an exhaustive window into the inner workings of Modo's animation system. All channels of any item can be viewed in the **Channel** viewport. Channels are any of the animateable attributes of an item (the identical attributes found in the Properties Panel). The **Channels** viewport provides a direct method for selecting channels, and creating, navigating, and managing any keyframes on those channels.



Camera	Value	S	Source
Camera			
(add position)			
(add rotation)			
(add scale)			
Position (Camera)			
Transform	Matrix4	◎	
Position X	0 m		
Position Y	750 mm	○	setup
Position Z	4 m	○	setup

At the top of the viewport, you can find a few controls.



Mesh	Value	S	Source
Mesh			
(add position)			
(add rotation)			
(add scale)			
Mesh			

From left to right, these controls are:

- **Lock** - Locks the contents of the Channels viewport allowing item selection to be changed without affecting the contents of the viewport.
- **Load** - When the viewport contents are locked this button will add any selected items to the viewport.
- **Show Group Members** - When a Group is selected this button loads the Group member items and/or channels into the viewport.
- **Filter Animated** - When active only channels that are animated will be loaded into the viewport.
- **Load Transform Channels** - Allows you to add or remove an item's transform channels (position, rotation, and scale).
- **Load Item Channels** - Adds or removes an item's non-transform channels.
- You can search the list of channels by clicking the **Search** button in the top-right corner of the viewport. This opens a **Search** field, allowing you to enter a search string. Clicking the arrow on the right of the

field allows you to do a **Simple Search** or search for a pattern using **Pattern Matching**. Click **Match Case** to do a case-sensitive search. Click **Flat** on the left to only display the list of channels without the hierarchy.

There are four columns in the **Channels** viewport:

Channel	The Channel column displays the channel names for the currently-selected item. For example, if a mesh is selected, the column name becomes **Mesh**.
Value	Displays the value of the channel at the current scene time.
Status (S)	Shows the state of the channel (be it animated, static, or having a keyframe) and this is communicated using the same dot method used on the channel keyframing control.
Source	Shows where the current channel value is stored. This is referencing the use of Render Passes and could be the base **Setup** state in a temporary **Edit** state, when the value is modified, or the named **Pass** state, when the value is applied to **Pass**.

Interacting with Channels

Similarly to the **Properties** viewport, the **Channels** viewport updates to reflect the current item selection. These can be items in the **Items** list, the Shader Tree, or even the **Groups** palette. Once selected, all animateable attributes appear in the **Channels** viewport. To affect an attribute's value, click the current value in the **Value** column. For Boolean values, which can be toggled, this changes the value field between true and false, representing their enabled or disabled states. For attributes with various properties, a menu appears presenting all the possible options. For numerical values, click and hold to activate the field so that you can enter a new numerical value. Once done, press **Enter** to set it, or simply click away from the field.

Setting Keyframes

You can set animation keyframes in all the usual ways: using the keyframe buttons, the **S** keyboard shortcut, from within the **Properties** viewport, or from within the Graph Editor. To set a keyframe for a value directly in the **Channels** viewport, you can right-click on the target value and select the **Add Key** option from the context menu. You can also remove keyframes using this method.

When any attribute has a keyframe, colored circles appear next to the value in the **S** column, referencing the attribute's current state. The possible states are:

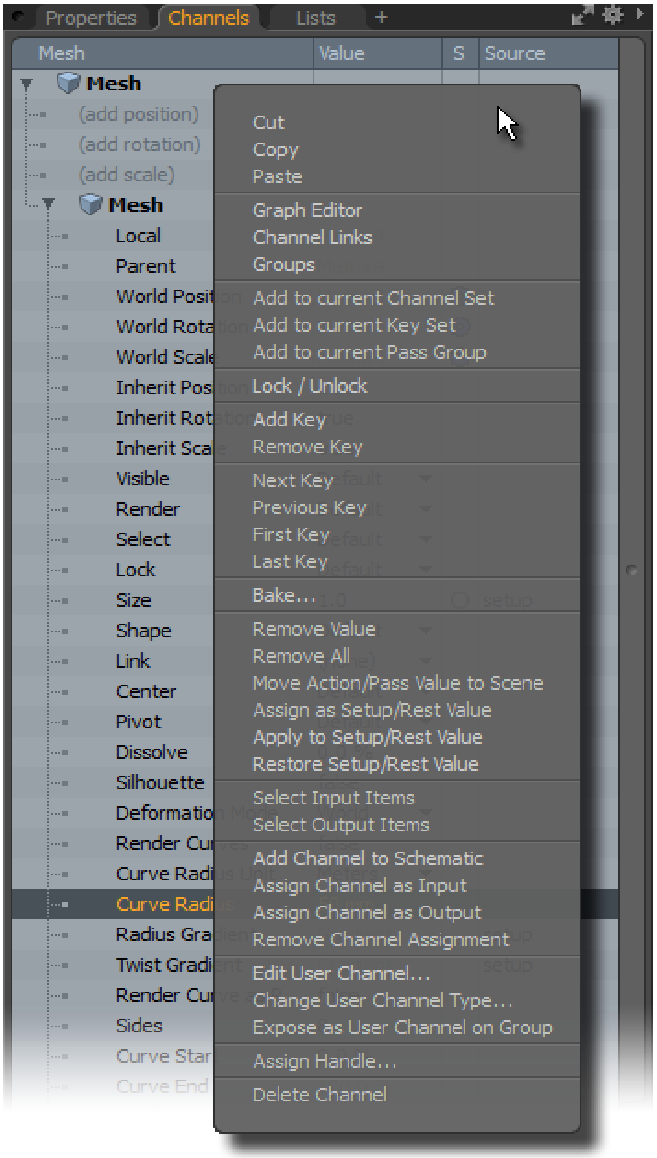
- ○ None - the channel is at the default value.
- ● Constant - the channel has a constant value.
- 🟢 Animated - there is no keyframe at the current time position.

- 🔴 Animated - there is a keyframe at the current time position.
- 🔵 Mixed - the channels have differing values or states.
- 🟣 Driven - the channel is driven by an external source.
- 🟡 Undefined - the channel is not defined.
- An arrow over the dot represents that channel as driving another for a channel link.
  - ➲ Driver - the channel is linked to another.
- A gear icon represents a channel that is being controlled by another for a channel link.
  - ⚙ Driven - the channel is linked to another.

Tip: You can automatically select input or output items by selecting the appropriate channels and pressing **I** (to select the input) or **O** (to select output).

Contextual Menu

Right-clicking in the **Channels** viewport opens up the context menu with options for modifying the keyframes of the various channels, as well as some shortcuts to access some often-used viewports.


Properties
Channels
Lists
Mesh
Value
S
Source
Cut
Copy
Paste
Graph Editor
Channel Links
Groups
Add to current Channel Set
Add to current Key Set
Add to current Pass Group
Lock / Unlock
Add Key
Remove Key
Next Key
Previous Key
First Key
Last Key
Bake...
Remove Value
Remove All
Move Action/Pass Value to Scene
Assign as Setup/Rest Value
Apply to Setup/Rest Value
Restore Setup/Rest Value
Select Input Items
Select Output Items
Add Channel to Schematic
Assign Channel as Input
Assign Channel as Output
Remove Channel Assignment
Edit User Channel...
Change User Channel Type...
Expose as User Channel on Group
Assign Handle...
Delete Channel

Options from the context menu include:

- **Cut**: Cuts the channel values and sets them back to their defaults.
- **Copy**: Copies the channel values.

> **Note:** Only channel values are copied, not the properties of the channel itself, such as its lock state or any minimum/maximum range settings.

- **Paste**: Pastes channel values. When values are pasted, the entire target channel is replaced by the source values.

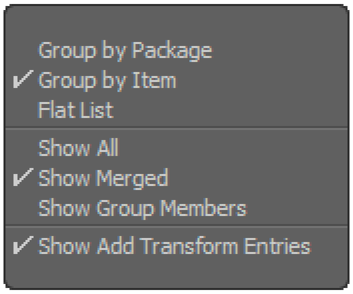
Note: The above operations require the source and target channels to be compatible. Such channels are found by matching the types of the source channels to the types of the destination channels. For a channel to match it must be of the same numerical type, for example, integer or floating point, and also the same data type, such as distance, angle, or percentage. If the channel has integer hints, such as the camera **Projection Type** channel, then the hints must also match.

- **Graph Editor** - opens a **Graph Editor** viewport as a pop-up palette.
- **Channel Links** - opens a Using Channels viewport as a pop-up palette.
- **Groups** - opens a Groups Viewport as a pop-up palette.
- **Add to current Channel Set** - adds the selected channel(s) to the current Channel set.
- **Add to current Key Set** - adds the selected channel(s) to the current Key Sets.
- **Add to current Pass Group** - adds the selected channel(s) to the current Pass group.
- **Add Key** - sets a keyframe at the current time with the current value.
- **Remove Key** - removes the keyframe at the current time.
- **Next/Previous/First/Last Key** - moves the Timeline to the appropriate time.
- **Bake** - for a modifier-driven Channel, this option opens a dialog, allowing you to convert the procedural values to actual keyframes at every frame and disconnect the modifier(s). This is useful when exporting animation.
- **Remove Value** - removes the current value, resetting the channel to its default state.
- **Remove All** - removes all keyframes and returns to the default (unmodified) value.
- **Move Action/Pass Value to Scene** - assigns the current channel value(s) to the scene. This is used to move a value inadvertently-assigned while in a Render Pass or Actor/Action/Pose.
- **Assign as Setup/Rest Value** -
- **Apply to Setup/Rest Value** - takes the current channel values associated to the selected item and moves them to the **Setup** action state.
- **Restore Setup/Rest Value** - replaces the current value of the selected channel(s) and replaces it with the **Setup** action state value.
- **Select Input Items** - for driven channels, this option selects the driver item or channel.
- **Select Output Items** - for driven channels, this option selects any items or channels being driven by the current channel.
- **Add Channel to Schematic** - this option adds the currently-selected channel to the **Schematic** viewport.
- **Assign Channel as Input/Output** - you can define various User Channels for Assembly nodes, which adds them to the node's channels list. The resulting channels, once created, can be selected and defined as either inputs or outputs of the appropriate Assembly node, allowing you to connect any elements external to the Assembly.
- **Remove Channel Assignment** - removes any channel assigned as an input or output from the Assembly node.

- **Edit User Channel** - opens a dialog that allows you to modify the settings of the Users Channel, such as **Name**, **Default Value**, and the minimum and maximum values.
- **Change User Channel Type** - opens a dialog to change the User Channel type. The **Type** determines the formatting of the resulting value, such as a scalar, float, or Boolean.
- **Expose as User Channel on Group** - when making presets for surfacing, often times the author only wants to show certain channels for you to edit, instead of giving you everything. This command allows preset creators to control which channels are exposed. By selecting specific Shader Tree item channels, the **Expose as User Channel on Group** command links the channels to a properly-defined User Channel on the Material Group item where they can be edited directly (under the **User Channels** sub-tab). The right-click context menu for the Material Group Item allows authors to also lock the preset, making it easier to direct the preset end-user to the appropriate controls. See Preset Browser for more information on creating presets.
- **Assign Handle** - works in conjunction with the **Channel Handles** function to assign the selected channel to the appropriately created handle. See the Channel Handles page for more information.
- **Delete Channel** - removes the selected user channel.

Viewport Contextual Menu

Right-clicking on the **Channels** tab itself opens the viewport menu as well as selecting the menu under the viewport Widget ▸. Select the **Viewport Settings** option to open the channel's viewport settings menu.



- **Grouping** - the first three options define how channels are grouped in the channel list.
  - **Group By Package** - organizes channels by package, sub-groupings of related elements by individual item.
  - **Group By Item** - organizes channels strictly by item.
  - **Flat List** - organizes channels as a straight list with no grouping.
- **Show All** - when enabled, shows all channels on an item; when disabled, certain less-used channels are hidden, such as matrix channels.
- **Show Merged** - when enabled, combines all the channel values for all selected items, instead of listing each item's channel one after another. This is useful when gang editing multiple items at once.
- **Show Group Members** - when enabled, show the members of any selected group.

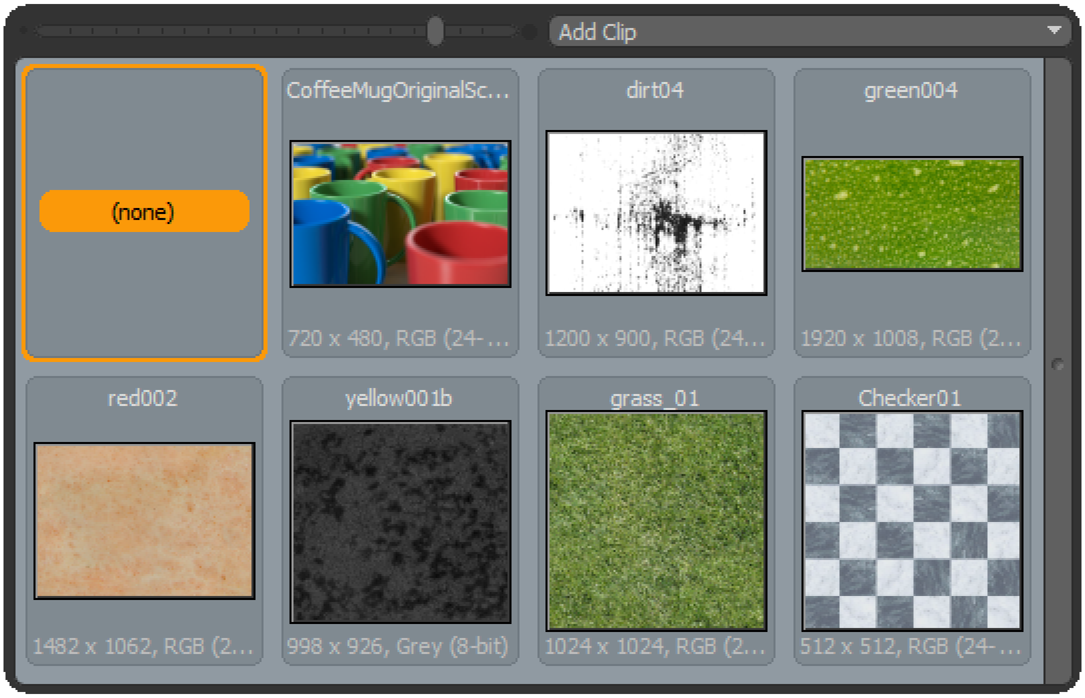
- **Show Add Transform Entries** - displays the **Add Position/Rotation/Scale** entries to the channel list. These options provide a shortcut for adding additional transform channels but, for smaller monitors, removing them can save some screen real estate.

Tip: To improve your workflow while working with User Channels, use the **Channel Haul** tool to toggle the visibility of all channels available for a selected item. For more information, see Displaying All Available User Channels in the 3D Viewport

Clip Browser

The **Clip Browser** provides an easy way to view and select images loaded into Modo for use in a variety of situations. Typically you can load images and sequences using OS-specific dialogs. When re-using images presently loaded, you can browse through them here.

The **Clip Browser** works like a combination of the Preset Browser and Images/Clips List view. The **Clip Browser** allows you to scale the thumbnails using the slider in the upper-left corner. Click and drag the slider to the right to increase the size or to the left to decrease the size of the thumbnails. You can access the **Clip Browser** in the **Shader Tree**, under **Add Layer** > **Image Map** > **Use Clip Browser**. It can also be found in the **Paint** interface layout tab for selecting images for **Image Ink** and **Brush Tips**. To select an image for use, simply click on the image thumbnail, dismissing the viewport.


Add Clip
(none)
CoffeeMugOriginalSc...
720 x 480, RGB (24-...
dirt04
1200 x 900, RGB (24...
green004
1920 x 1008, RGB (2...
red002
1482 x 1062, RGB (2...
yellow001b
998 x 926, Grey (8-bit)
grass_01
1024 x 1024, RGB (2...
Checker01
512 x 512, RGB (24-...

Add Clip

In the upper-right corner of the viewport, clicking on the **Add Clip** option offers some options to load additional images and sequences into Modo.

Load Image

Selecting the **Load Image** command opens an OS-specific dialog where you can browse and select images saved to the system drive(s). Modo accepts a variety of bitmap formats. Once an image is selected and you press **Open** to load it, the image is utilized as if it were selected in the browser, dismissing the window.

Load Sequence

Selecting the **Load Sequence** command opens an OS-specific dialog where you can browse and select still image formats, saved with sequential numbering for use as animated clips. Modo accepts a variety of bitmap formats. Once the first image in the sequence is selected and you press **Open** to load it, the sequence is utilized as if it were selected in the browser, dismissing the window.

New Image

Selecting the **New Image** command opens an OS-specific dialog where you first browse for a location and designate a name for the image file, selecting the format with the **Save as Type** options. Modo writes a variety of bitmap formats. Once an image location and name are defined and you press **Save** to create, Modo opens the **Create New Image** dialog, where you can define the image's pixel size, bit depth, and fill color, if desired. Once you select **OK**, the image is created and applied as if it were selected in the browser, dismissing the window.

Blank images are useful for painting, image-based sculpting, and baking. Once you have modified the image, it needs to be saved. Right-click on the image in the Images/Clips List to open the context menu and choose the **Save** option or, in the menu bar, use the command **File** > **Save Image** option.

Tip: To save an image as 16-bit, you must enable the **Floating Point** option or images default to 8-bit, regardless of the chosen file format.

New Sequence

Selecting the **New Sequence** command opens an OS-specific dialog where you first browse for a location and designate a name for the image file, selecting the format with the **Save as Type** options. Modo generates sequences as consecutively-numbered still image files, writing a variety of bitmap formats. Once a sequence location and name are defined and you press **Save** to create, Modo opens the **Create New**

Sequence dialog, where you define the initial image's pixel size, bit depth, and fill color, if desired. Once you select **OK**, the image is created and applied as if it were selected in the browser, dismissing the window.

Blank images are useful for painting, image-based sculpting, and baking. Once you have modified the image, it needs to be saved. Right-click on the image in the Images/Clips List to open the context menu and choose the **Save** option or, in the menu bar, use the command **File** > **Save File**. The length of the sequence is determined by the number of frames you create while in Modo.

For example, when painting a sequence, when the current frame is incremented, Modo duplicates the last frame. You can then add to the image frame. Each new increment duplicates the frame in its current state. You can also return to further edit previous frames. Modo only creates the length of frames that you've specifically generated by incrementing the Timeline. For a 300-frame sequence, if only 3 frames are generated and saved, then when rendered, Modo simply repeats the 3rd frame until the last frame of the rendered sequence.

Once you've modified the sequence, it needs to be saved. Right-click on the image in the **Clips** list to open the context menu and choose the **Save Sequence** option or, in the menu bar, use the command **File** > **Save Sequence** option.

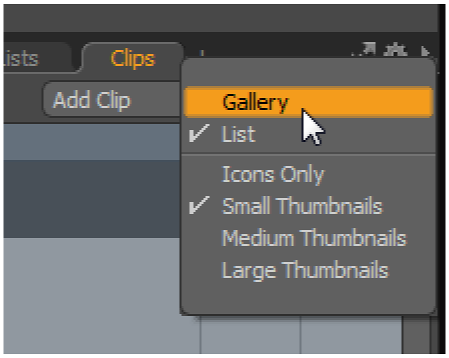
Note: To learn more about bitmap formats that Modo accepts, see Bitmap Image Formats.

Images/Clips List

The **Clips** or **Images** viewport provides access to all images loaded into Modo, organized by scene. The size of the viewport determines how the image information is displayed. In the compact view, you can select particular images, and add images to the list using the **Add Clip** button. Hovering the mouse over the edge of the frame and clicking and dragging the frame resizes it to reveal larger views. The small list only displays the current image, and clicking opens a gallery browser for navigating and adding images.

If the image is located relative to the scene directory, in the parent directory to the scene, or relative to the project directory (if any), then the file path is displayed as relative to that location. Otherwise, the absolute path is displayed. In both cases, you can see the absolute path in the tooltip when you hover the cursor over the image.

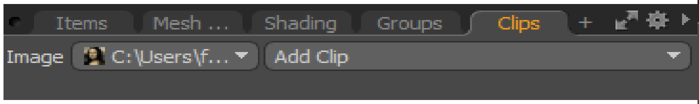
At full size, the list view shows the image by name, includes an icon of the image for visual reference, and has data about the resolution and format of the image. You can also select a **Gallery** view from the viewport widget, choosing the **Viewport Settings** menu.


Clips
Add Clip
Gallery
✓ List
Icons Only
✓ Small Thumbnails
Medium Thumbnails
Large Thumbnails

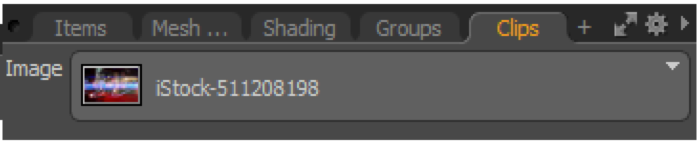
To access the Options menu you can either:

- Click on the Settings icon, or
- Right-click on the tab and select **Viewport Settings**.

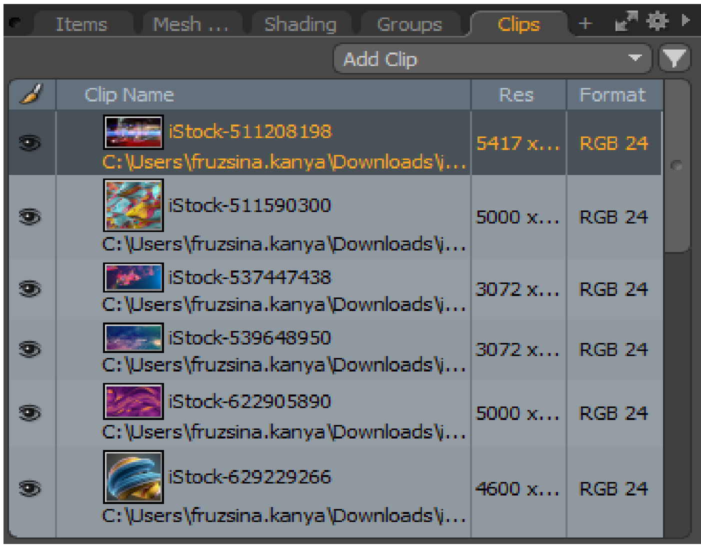
The gallery mode has a slider across the top of the viewport where you can resize the icon images interactively.


Items
Mesh ...
Shading
Groups
Clips
Image
C:\Users\f...
Add Clip

Clip List Compact

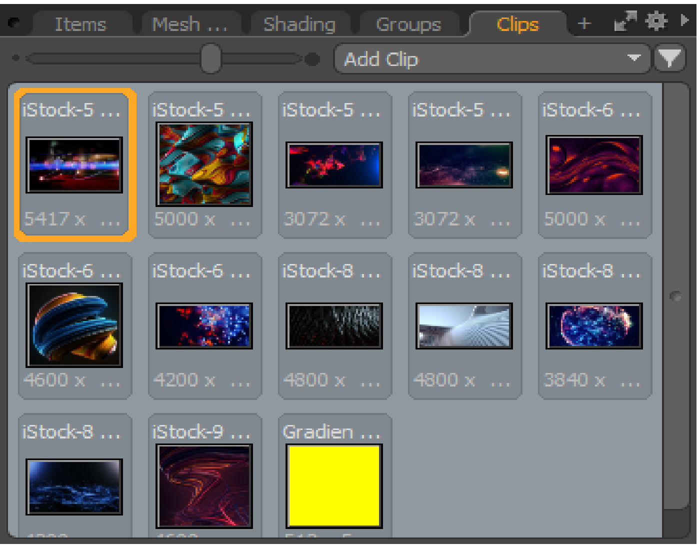

Items
Mesh ...
Shading
Groups
Clips
Image
iStock-511208198

Clip List Small


Items
Mesh ...
Shading
Groups
Clips
Add Clip

	Clip Name	Res	Format
	iStock-511208198 C:\Users\fruzsina.kanya\Downloads\j...	5417 x...	RGB 24
	iStock-511590300 C:\Users\fruzsina.kanya\Downloads\j...	5000 x...	RGB 24
	iStock-537447438 C:\Users\fruzsina.kanya\Downloads\j...	3072 x...	RGB 24
	iStock-539648950 C:\Users\fruzsina.kanya\Downloads\j...	3072 x...	RGB 24
	iStock-622905890 C:\Users\fruzsina.kanya\Downloads\j...	5000 x...	RGB 24
	iStock-629229266 C:\Users\fruzsina.kanya\Downloads\j...	4600 x...	RGB 24

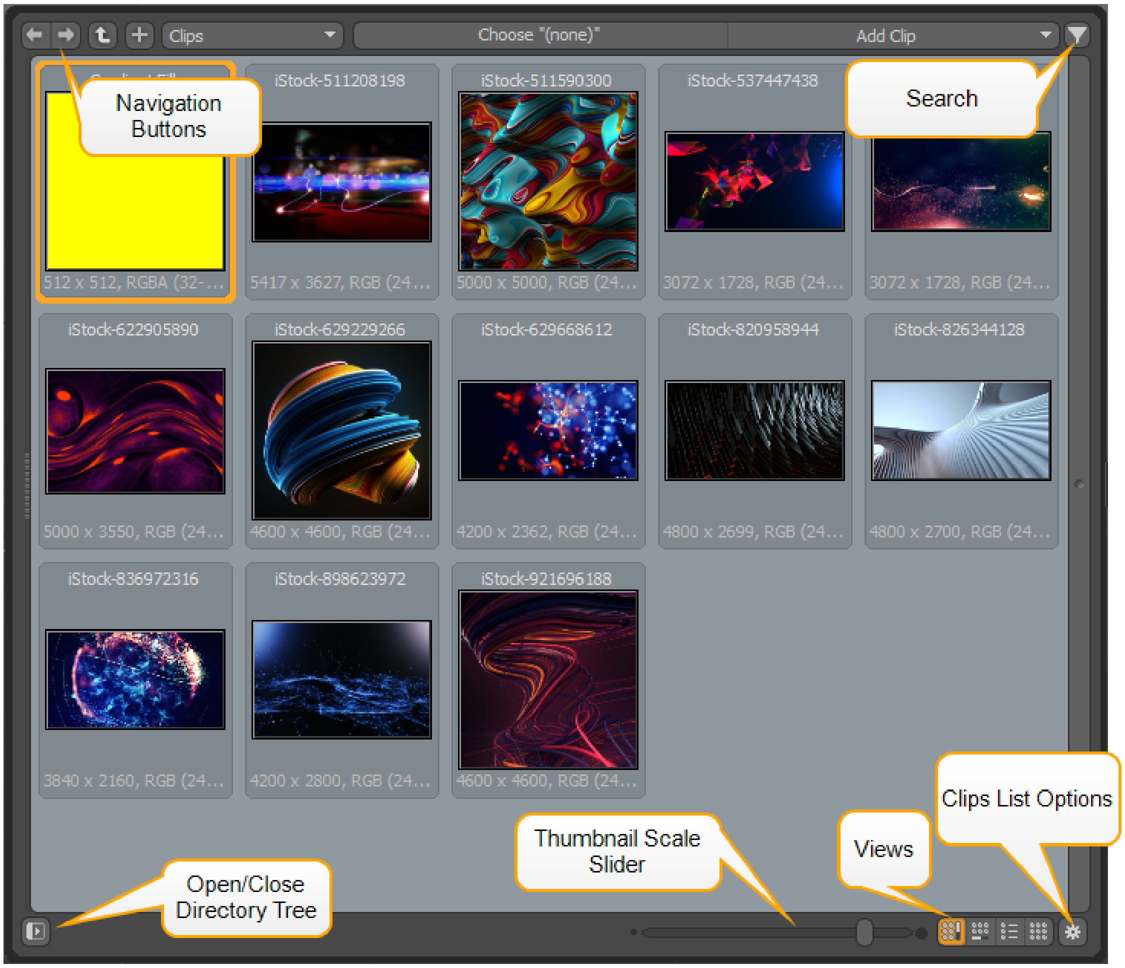
Clip List Full Size


Items
Mesh ...
Shading
Groups
Clips
Add Clip
iStock-5 ...
5417 x ...
iStock-5 ...
5000 x ...
iStock-5 ...
3072 x ...
iStock-5 ...
3072 x ...
iStock-6 ...
5000 x ...
iStock-6 ...
4600 x ...
iStock-6 ...
4200 x ...
iStock-8 ...
4800 x ...
iStock-8 ...
4800 x ...
iStock-8 ...
3840 x ...
iStock-8 ...
iStock-9 ...
Gradien ...

Clip List Gallery

When opening the **Clips** viewport from the **Texture Layer** properties, a full-size viewport appears with additional controls.

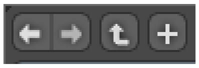
Tip: For more information on using images as textures, see Image Map.


Clips
Choose "(none)"
Add Clip
Navigation Buttons
Search
iStock-511208198
iStock-511590300
iStock-537447438
512 x 512, RGBA (32-…
5417 x 3627, RGB (24…
5000 x 5000, RGB (24…
3072 x 1728, RGB (24…
3072 x 1728, RGB (24…
iStock-622905890
iStock-629229266
iStock-629668612
iStock-820958944
iStock-826344128
5000 x 3550, RGB (24…
4600 x 4600, RGB (24…
4200 x 2362, RGB (24…
4800 x 2699, RGB (24…
4800 x 2700, RGB (24…
iStock-836972316
iStock-898623972
iStock-921696188
3840 x 2160, RGB (24…
4200 x 2800, RGB (24…
4600 x 4600, RGB (24…
Clips List Options
Views
Thumbnail Scale Slider
Open/Close Directory Tree

The **Add Clip** button provides you with commands for not only loading still image and image sequences, but for creating blank new images and sequences. Selecting any of the commands opens a file browser where you can navigate to a folder and either load or save the image, depending on the operation chosen.

Navigating in the Viewport

You can navigate the **Clips** list by using the buttons in the upper-left of the window:



Forward, **Back**, **Up to Parent**, and **Create New Folder** respectively. The **Forward** and **Back** options are only available when folders have been navigated (much like a web browser).

If you prefer to use the directory tree, you can open it by clicking the Directory Tree button in the bottom-left corner of the viewport. The directory tree is displayed on the left of the viewport.

You can scale the thumbnails using the slider at the bottom.

You can set how the **Clips** list displays information using the buttons in the lower right corner.

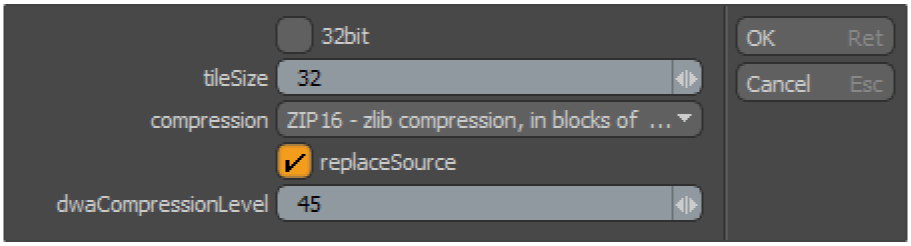
▥	Thumbnails view with vertical scrolling.
▤	Thumbnails view with horizontal scrolling.
☰	List view, showing assets in a vertically scrolling list.
▦	Thumbnail grid view. You can arrange the thumbnails arbitrarily.
⚙	Opens the **Clips** list options. For more information, see Clips List Options.

Double-clicking an image in the viewer opens it in a Preview Viewport set to **Clips**, so you can view the image at a larger resolution. Right-clicking an image provides the image item context menu, with a variety of options:

Rename	Renames the current image. This new name only affects the reference name in Modo and does not rename the file itself.
Editor Color	Provides you with the means to use color for organizational purposes.
Delete	Removes the image from the scene. If the image is applied to a surface, Modo pops up a requester stating that the image is in use and asking if you are sure you want to remove the image.
Reload	Reloads the image as saved to disk. This is useful for when edits, made in an external application, don't trigger Modo to reload it automatically.
Replace as Still	Replaces all instances of the current image or sequence in the scene with a new still image loaded from disc.
Replace as Sequence	Replaces all instances of the current image or sequence in the scene with a new image sequence loaded from disc.
Reload Layered	Similar to **Reload**, but works for layered images.

Image	
Replace layered Image	Similar to **Replace as Still**, but works for layered images.
Instance Sequence	Creates an instance of an image sequence that can be assigned different attributes from its source, without the need to have the entire sequence loaded multiple times.
Save	Saves the current image to disk, overwriting the previous version. Use this command to save any changes made to images using Modo's painting and sculpting tools.
Save As...	Saves the current still image to a new location and/or name.
Save Sequence	Saves the current image sequence to disk.
Export as Tiled EXR	Exports all the selected clips into the tiled EXR format. A memory-efficient format that only loads the applicable image tile as necessary. The resulting tiled EXR images are saved into the same folder as the source image, and if the **Replace Source** option is enabled the source image is replaced with the tiled EXR file in the active scene automatically. When invoked, the following dialog panel opens representing the export options for the resulting files.
Export as Tiled EXR to...	Similar to **Export as Tiled EXR** but allows the user to specify a specific location to save files before conversion takes place.
Export as LRL	Exports all the selected clips to **.lrl** files.
Open with Default Application	Opens the selected image in the default external image editor (as specified in the OS) for the associated file format.
Open Containing Folder	Opens the image location in a file browser.

Export as Tiled EXR Dialog

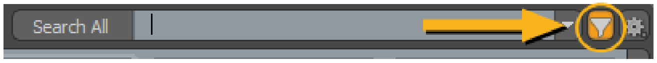


☐ 32bit
tileSize 32
compression ZIP 16 - zlib compression, in blocks of …
☑ replaceSource
dwaCompressionLevel 45

32 bit	When enabled, saves the files as 32 bit extended bit depth **.exr** format images.
Tile Size	Determines the smallest tile size generated in pixels. Depending on the image, should be in the area of 16x16 to 64x64 for best results.
Compression	Determines the type of compression assigned to the resulting images. Default option is for the ZIP 16 compression format.
Replace Source	When enabled, the selected image is automatically replaced by the generated tiled **.exr** in the associated scene.
dwaCompression Level	Determines the amount of compression when using Dreamworks DWAA or DWAB compression options. These lossy formats are variable in quality and can minimize the compression artifacts. Higher values will result in greater compression and likewise smaller file size, but increases the chance for artifacts.

Searching for Clips

You can search for specific images by using the Search feature. Search is located on the top-right of the viewport. Click the Search ▽ button to open the search string input field. Type a term into the field to see all images that contain the term in either the clip name or the clip's keyword tags. By default, the entire hierarchy is searched.



The first matching item is highlighted in the search results. You can scroll through the list using the cursor keys, and then press enter to open the image.

To change the search options:

- In the Search input field, click the More options ▾ button on the right of the search field, and select one of the following options:
  - **Simple Search** - Locates assets using the text entered. Ignores case.
  - **Pattern Matching** - Locates assets using the exact pattern of the text entered.
  - **Match Case** - Matches the case of the search term against the clip name.

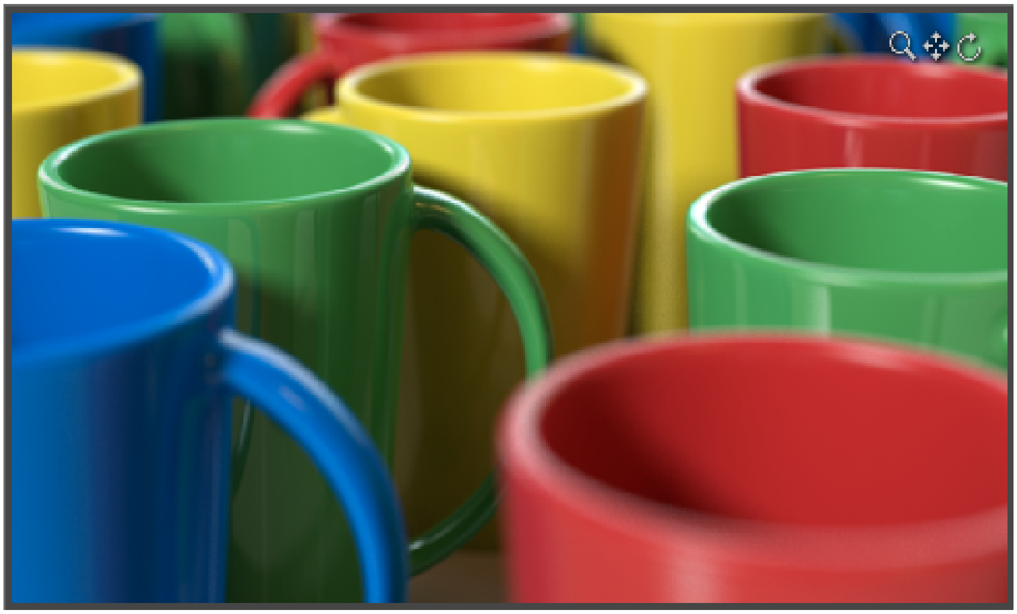
Note: If the cursor keys don't work on Mac OS X, try **Alt**+cursor to scroll through the options. This is due to a glitch in Modo's key handlers.

Preview

You can see a preview of any image in Modo by double-clicking on any image in the image list/clips browser. This opens an **Image Preview** pane. You can move around the image using the navigation widgets in the upper-right corner. Alternatively, you can use the following keyboard shortcuts:

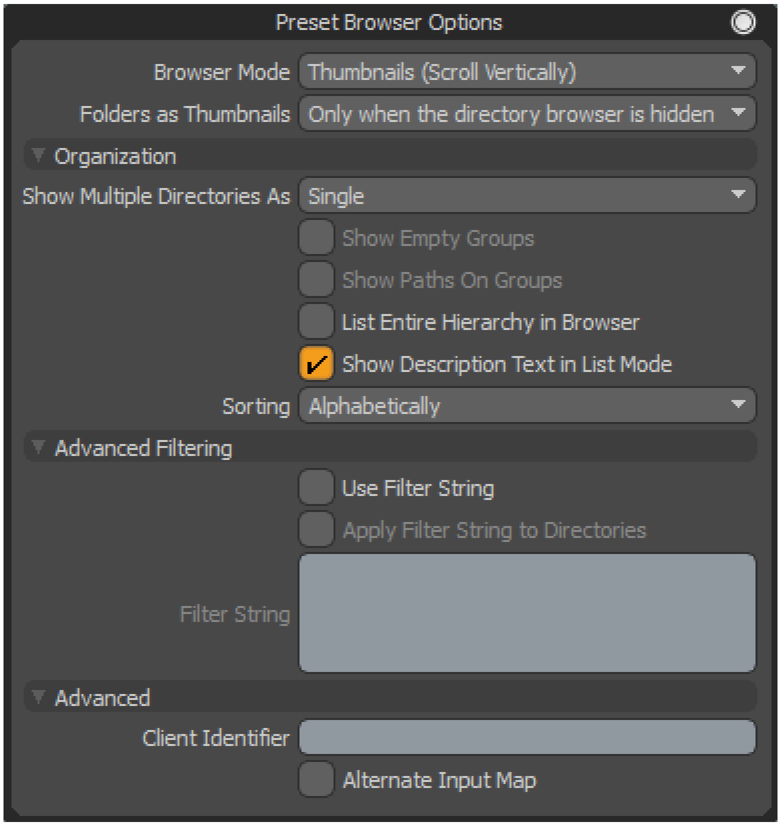
- **Alt**+click and drag to pan,
- **Shift**+**Alt**+click and drag to zoom

Tip: You can double-click the initial image again to close the window.



Clips List Options

The Settings button, in the bottom-right corner of the **Clips** list, opens the viewport options dialog, which allows you to further customize the information displayed in the viewport.



Preset Browser Options

Browser Mode: Thumbnails (Scroll Vertically)
Folders as Thumbnails: Only when the directory browser is hidden
Organization
Show Multiple Directories As: Single
☐ Show Empty Groups
☐ Show Paths On Groups
☐ List Entire Hierarchy in Browser
☑ Show Description Text in List Mode
Sorting: Alphabetically
Advanced Filtering
☐ Use Filter String
☐ Apply Filter String to Directories
Filter String
Advanced
Client Identifier
☐ Alternate Input Map

Browser Mode	How the images are displayed in the browser: • **List** - Displays icons next to the name of the clip, and two columns containing a star rating and comment text. You can choose to show/hide one or both of these columns by right-clicking on the column headers. You can also rearrange columns by dragging-and-dropping the column headers. • **Thumbnails (Scroll Vertically)** - Large icon view that scrolls vertically. • **Thumbnails (Scroll Horizontally)** - Large icon view that scrolls horizontally. • **Grid** - Similar to the **Thumbnail** view, but only displays the image thumbnail without the name.

Folders as Thumbnails	Controls whether the folders themselves, used to organize the imagess, display as thumbnails within the browser area. This is only active when either of the thumbnail options (above) are selected. Folders can be a helpful addition for navigation, more akin to regular OS file navigation.

Organization Options

Show Multiple Directories As	Controls how multiple folder selections are displayed (hold down **Shift** or **Ctrl** to select more than one folder in the **Directories** tree at a time): • **Single** - Displays the contents of single folders only (no multiple folder selections allowed). • **Multiple (Merged)** - Displays the contents of all selected folders simultaneously, as if they were a single folder. • **Multiple (Grouped)** - Same as above, but separates the contents of each folder with a named divider.
Show Empty Groups	Empty folders are displayed in the browser. Files can be dropped into empty groups when they are visible. **Note:** This option is only available when the Show Multiple Directories is set to **Multiple (Grouped)**.
Show Paths on Groups	Displays the full directory hierarchy in the group name in the clip list. This only applies if you are viewing multiple directories using the **Multiple (Grouped)** option.
List Entire Hierarchy In Browser	When this option is enabled, all child folders in the hierarchy are also displayed in the browser without being explicitly selected.
Sorting	Controls the display order of the clips in the browser. • **Alphabetically** - Displays in alphanumerical order. • **Manually** - Allows you to define the order by dragging-and-dropping within the browser (not available for multiple directories as merged). • **Star Rating** - Ordered by the number of stars assigned to a clip. • **File Extension** - Ordered by the clip's file extension, organized

	alphabetically. • **Modification Time** - Defines the order based on the last time the file was saved (or modified), starting with the most-recent file or directory. • **File Size** - Defines the order based on the actual size of the file as saved on the disc.

Advanced Filtering Options

The **Clips** list adds markup information to clips. Filtering procedurally refines what is visible in the browser, based on a scripting-like annotation

The filter options take into account both the user-defined markup, as well as metadata (information associated with the file, such as name and format).

Use Filter String	Toggles the filtering **on** or **off** for the browser.
Apply Filter String to Directories	When this option is enabled, the filtering is applied to the directory folders as well, otherwise all directories are shown.
Filter String	Enter the data string to filter the elements in the **Clips** list. **Note:** The **Use Filter String** and/or **Apply Filter String to Directory** option must be set before typing a filter string.

Advanced Options

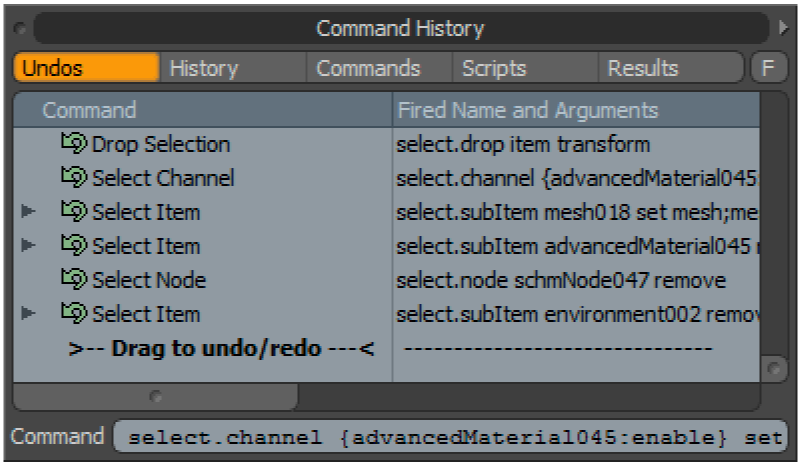
Client Identifier	Allows a **Preset Browser** to have its own unique selection. This is most useful when configuring a **Preset Browser** to be used with a specific tool. For example, this is how the **Polygon Bevel** tool knows which of many possible **Preset Browser** viewports to use (in the case of **Polygon Bevel**, the browser viewport is embedded in the **Tool Properties** viewport).

	Note: Each tool must be specially coded to look for a specific client identifier, so this feature is only really useful to tool developers. For more information, see our SDK documentation.
Alternate Input Map	An advanced setting for developers that controls the number of mouse clicks used to assign a preset to a tool. When enabled, they assign with a single click, and when disabled they assign with a double-click (the default behavior). For more information, see our SDK documentation.

Command History

Modo keeps a listing of every command used during a session in the **Command History** viewport. This view is useful for quick undo/redo operations, managing macros, quickly assigning keyboard shortcuts to specific commands, and many other utility functions.

To access the **Command History** viewport, look at the bottom-right corner of the interface where there is a single **Command** input field, then drag the divider above it upwards until the **Command History** header is displayed.



Command	Fired Name and Arguments
Drop Selection	select.drop item transform
Select Channel	select.channel {advancedMaterial045
Select Item	select.subItem mesh018 set mesh;me
Select Item	select.subItem advancedMaterial045
Select Node	select.node schmNode047 remove
Select Item	select.subItem environment002 remov
>-- Drag to undo/redo ---<	------------------------------

```
Command select.channel {advancedMaterial045:enable} set
```

There are five tabs in the **Command History** panel: **Undos**, **History**, **Commands**, **Scripts**, and **Results**. By default, the **Undos** list is active. Within this list are sections for **Command**, **Fired Name and Arguments**, and **Description**. The listings within these columns display the internal commands that are performed when you press any keyboard shortcut or interact with an interface element.

You can click and drag within the **Command** list to undo/redo actions. Locate the line that reads **Drag to undo/redo** in bold, and drag it up to undo multiple commands at once.

Note: You can control the maximum number of available undos in the **Preferences**. In the menu bar, click **System** > **Preferences** > **Defaults** > **Application** then adjust the value in the **Max Undo Level**.

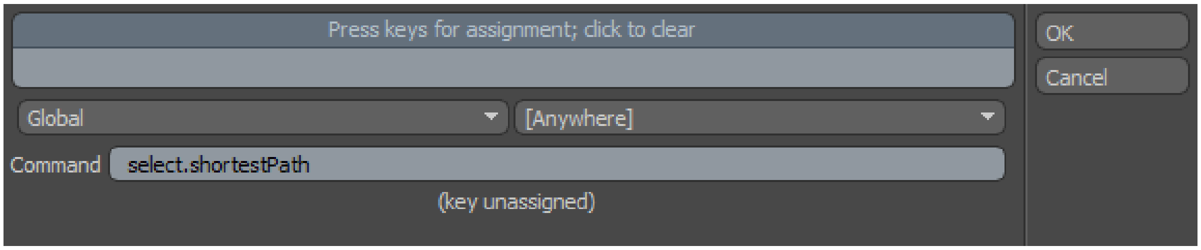
Clicking the subsequent section of the viewport opens the **History** tab, where a history of the performed commands are listed, with their possible attributes. The **Commands** tab supplies a complete list of all the commands available within Modo. The **Scripts** tab gives you access to all recorded macros stored in the configuration file. From here, it's possible to **Edit** or **Run** individual lines within these scripts. In the **Results** tab, the results of any command performed or queried is displayed here.

Creating Keyboard Shortcuts

You can assign keyboard shortcuts to commands in the **Command History** viewport's **Commands** tab.

1. Right-click a command item in the list and click the **Map Command to Key**.

   This opens the key assignment dialog. You can see the command the shortcut is being assigned to in the **Command** field.



Command select.shortestPath

(key unassigned)

2. Press a keyboard key, or multiple keys in combination, which appear in the assignment area.

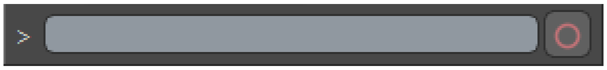
Note: If the intended combination is already in use by another command, Modo displays the associated command at the bottom of the dialog.

   By default, the context is set to **Global [Anywhere]**, but you can choose a specific viewport or UI component for the shortcut to work in using the dropdowns.

3. Click **OK** to save the shortcut.

Macro Recorder

You can use the **Macro Recorder** to record custom macros. It is situated at the bottom of the **Command History** viewport and it is visible even when the **Command History** viewport is collapsed.



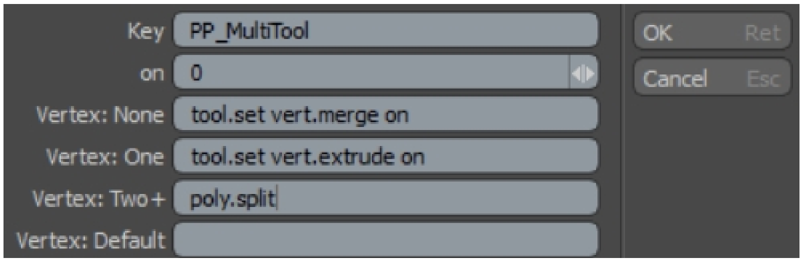
Click the red circle on the right of the command entry to reveal the macro recorder controls.

Icon	Description
●	Click to begin recording a macro, and click again to stop.
▶	Click to replay the macro. To replay a macro on repeat, click and hold the icon to reveal the additional **Replay Repeat** option.
⚙	Opens the macro recorder options, allowing you to save the macro as an **.lxm** file or in the user config, and to choose how UI commands are recorded.
✎	Opens the **Script Editor**.

To hide the controls again, click in the command entry field. This ensures that the maximum possible space is available for typing in new commands.

Creating Shortcuts with Multiple Conditional Functions

Conditional Commands allow you to add several different tools or commands to one shortcut based on the current selection mode and the number of selected components. In the first part of the video example, William creates a basic tool that behaves differently depending on how many vertices you have selected: no vertices selected enables Vertex Merge, one vertex selected enables Vertex Extrude, and more than two vertices selected enables Split Polygons.



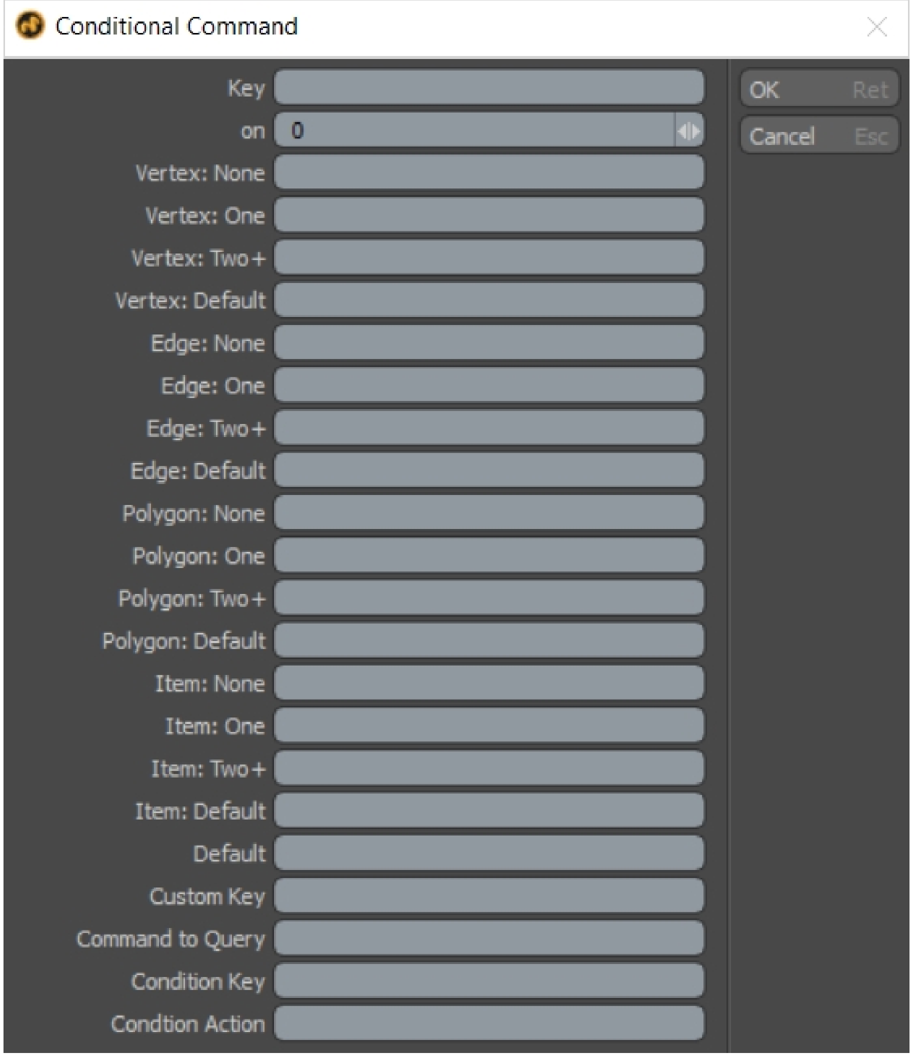
Key: PP_MultiTool
on: 0
Vertex: None: tool.set vert.merge on
Vertex: One: tool.set vert.extrude on
Vertex: Two+: poly.split
Vertex: Default:

OK Ret
Cancel Esc

To open the **Conditional Command** viewport:

1. In the **Edit** tab, click the **Conditional Commands** button and then click **Setup**.

   The **Conditional Command** viewport is displayed.



on: 0

Conditional Command
Key
on
Vertex: None
Vertex: One
Vertex: Two+
Vertex: Default
Edge: None
Edge: One
Edge: Two+
Edge: Default
Polygon: None
Polygon: One
Polygon: Two+
Polygon: Default
Item: None
Item: One
Item: Two+
Item: Default
Default
Custom Key
Command to Query
Condition Key
Condtion Action
OK Ret
Cancel Esc

2. Enter a descriptive name, or **Key**, so that you can easily identify the command later on.
3. The **on** field controls input key behavior:

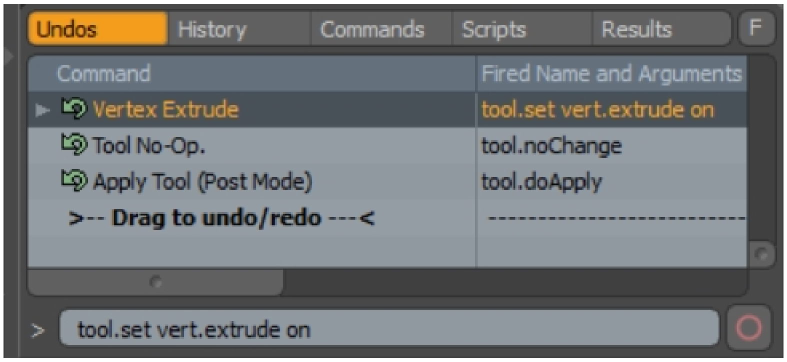
   • **0** - the default value, indicates that pressing the input key switches the command on or off.

- **1** - indicates that the command is a sticky key, meaning you hold the input key down to temporarily activate the tool.

4. Assign a command to activate in the component mode from which you want to trigger a command. For example, to add a command that enables Vertex Extrude if one vertex is selected, in the **Vertex: One** field enter:

```
tool.set vert.extrude on
```

If you're not sure of the syntax for the command you're looking for, perform the action manually in the direct modeling tools and then look in the **Command History** viewport. All actions in Modo are logged there.



Command	Fired Name and Arguments
Vertex Extrude	tool.set vert.extrude on
Tool No-Op.	tool.noChange
Apply Tool (Post Mode)	tool.doApply
>-- Drag to undo/redo ---<	-------------------------

```
> tool.set vert.extrude on
```

Commands called can be set depending on how many of each component are selected when the conditional command is activated. For example:

- **Edge: None** - when no edges are selected, run the specified command.
- **Edge: One** - when a single edge is selected, run the specified command.
- **Edge: Two+** - when two or more edges are selected, run the specified command.
- **Edge: Default** - when in edge component mode and no edges are selected, run the specified command.

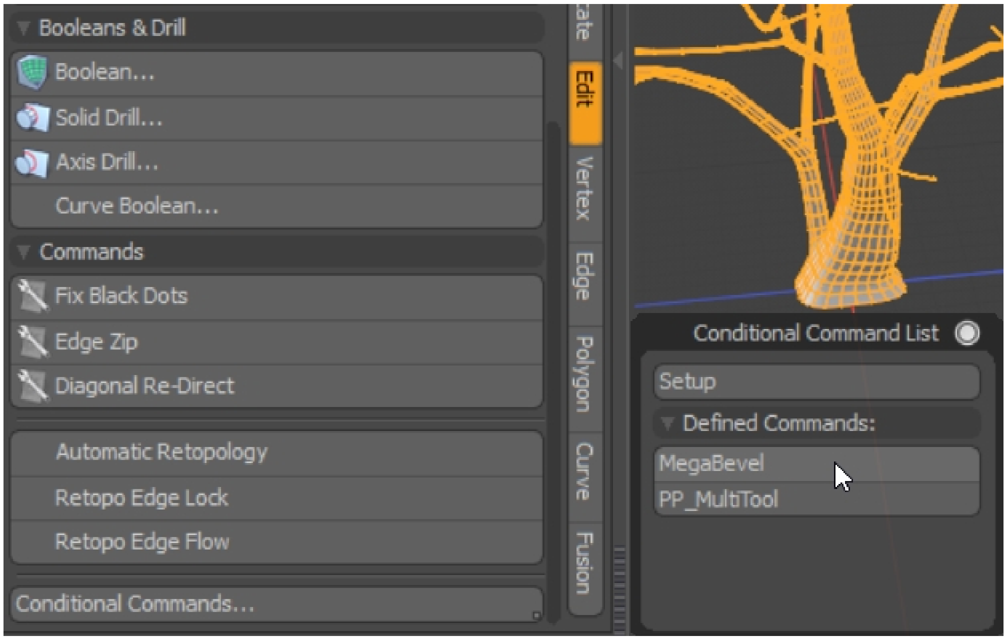
This is the same for the vertex, polygon and item component types listed in the conditional commands menu.

Tip: For more information on component selection modes, see Understanding Items vs. Components.

5. Click **OK** to save the command.

   The command is saved in the **Conditional Command List** viewport.

6. Activate a command by clicking the appropriate button in the list.


Booleans & Drill
Boolean...
Solid Drill...
Axis Drill...
Curve Boolean...
Commands
Fix Black Dots
Edge Zip
Diagonal Re-Direct
Automatic Retopology
Retopo Edge Lock
Retopo Edge Flow
Conditional Commands...
Edit
Vertex
Edge
Polygon
Curve
Fusion
Conditional Command List
Setup
Defined Commands:
MegaBevel
PP_MultiTool

Depending on the selected mesh type, you can configure conditional commands to call specific commands or custom tools. For example, specifying that a particular MeshOp is called instead of its direct modeling counterpart if a procedural mesh is selected.

Conditions are set via the **Custom key**, **Command to Query**, **Conditions Key**, and **Condition Action** parameters.

- **Custom Key** - the name of the custom command you want to run.
- **Command to Query** - the mesh type you want to apply the custom command to. For example, if you want your custom command to be applied to a procedural mesh, **input conditionalCommand.procMesh**.
- **Conditions Key** is the result that you are expecting to receive from the **Command to Query**. Expected results are treated as true, and the command specified in **Conditions Action** is run.
- **Conditions Action** specifies the command you want to run when the **Conditions Key** is true.

For example, you could use **conditionalCommand.selectedElementCount** as the **Conditions Key** which returns a number derived from the command. You can then use this number to drive the **Conditions Action** operation **selectedElementCount**. If **selectedElementCount** >= 2, then in Edge Selection mode enable **edge.loopSlice** and if **selectedElementCount** < 2, then in Edge Selection mode enable **edge.subdivide**.

You can also edit the configuration file directly to create Conditional Commands. The config file location differs by OS:

- **macOS:** /Users/<username>/Library/Preferences/MODO<version>.CFG/ConditionalCommands.cfg
- **Windows:** C:\Users\<username>\AppData\Roaming\Luxology\MODO<version>.CFG\ConditionalCommands.cfg
- **Linux:** /home/<username>/.luxology/MODO<version>.CFG/ConditionalCommands.cfg

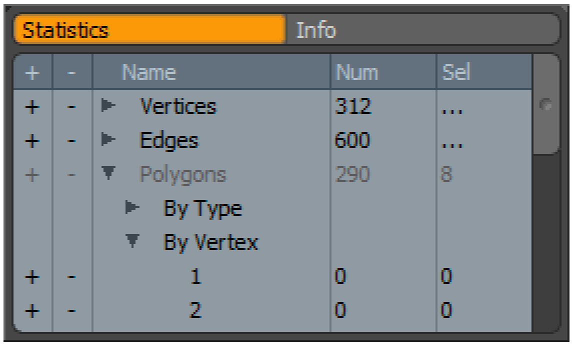
For example, to create a command to execute a conditional bevel or chamfer dependent on selection mode, a conditional cut, and automatically add MeshOps if the mesh is procedural, you could add:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
   <atom type="ConditionalCommands">
    <hash type="key" key="meshopBevel" val="meshopBevel">
      <list type="vert">
        <atom type="v_def">meshop.create vert.bevel.item</atom>
      </list>
      <list type="edge">
        <atom type="e_def">meshop.create edge.chamfer.item</atom>
      </list>
      <list type="poly">
        <atom type="p_def">meshop.create poly.bevel.item</atom>
      </list>
    </hash>

    <hash type="key" key="bevel" val="bevel">
      <list type="vert">
        <atom type="v_def">tool.set vert.bevel on</atom>
      </list>
      <list type="edge">
        <atom type="e_def">tool.set edge.chamfer on</atom>
      </list>
      <list type="poly">
        <atom type="p_def">tool.set poly.bevel on</atom>
      </list>
      <atom type="def">tool.set poly.bevel on</atom>
    </hash>

    <hash type="key" key="cut" val="cut">
      <list type="vert">
        <atom type="v_more">poly.split</atom>
        <atom type="v_def">tool.set edge.knife on</atom>
      </list>
      <list type="edge">
        <atom type="e_one">edge.subdivide</atom>
        <atom type="e_more">tool.set poly.loopSlice on</atom>
        <atom type="e_def">tool.set edge.knife on</atom>
      </list>
      <list type="poly">
        <atom type="p_more">tool.set poly.loopSlice on</atom>
        <atom type="p_def">tool.set poly.knife on</atom>
      </list>
    </hash>

    <hash type="key" key="megabevel" val="megabevel">
      <list type="custom">
        <hash type="command" key="proc" val="conditionalCommand.procMesh">
          <list type="conditions">
            <hash type="condition" key="true">conditionalCommand meshopBevel</hash>
            <hash type="condition" key="false">conditionalCommand bevel</hash>
          </list>
        </hash>
      </list>
    </hash>
  </atom>
</configuration>
```

Info and Statistics

The **Info and Statistics** viewport provides information about the currently selected/active geometry in the scene. It also provides you with powerful selection and precise editing functionality. The viewport is made up of two tabs: a **Statistics** tab and an **Info** tab.

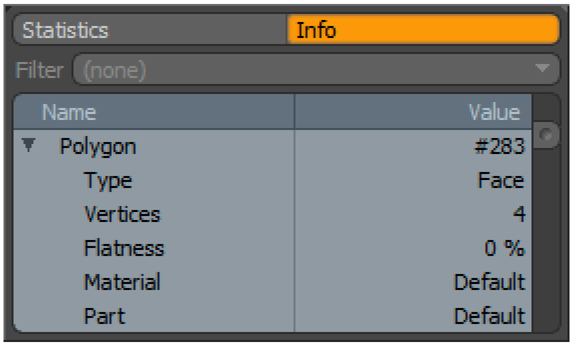
The **Statistics** tab is an integral part of the Modo workflow, giving you a wide range of tools to use for selection. This is a very powerful alternative to manually selecting vertices, edges, polygons, and materials in the 3D viewport using a mouse or tablet. Selections are made using the **+** and **-** columns. Simply navigate to the appropriate statistic, such as **Polygons**, by the number of vertices, and press the **+** to select (add) and **-** to deselect (subtract). When working with the various components (vertex, edge, or polygon) make sure to also be in the appropriate mode.



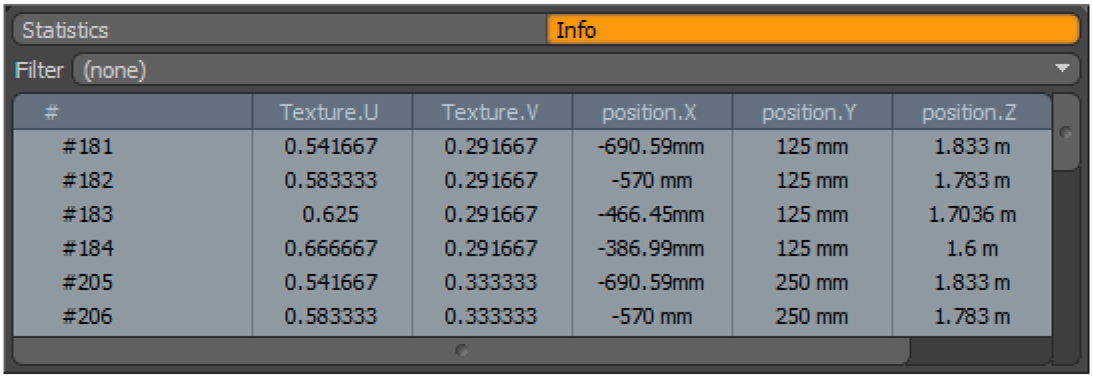
+	-	Name	Num	Sel
+	-	Vertices	312	...
+	-	Edges	600	...
+	-	Polygons	290	8
		By Type		
		By Vertex		
+	-	1	0	0
+	-	2	0	0

The **Statistics** tab also displays the total count and number selected for all vertices, edges, polygons, and materials. You can use this for very fast selection of the desired components in your scene. Expanding the entries in the **Name** column, by pressing the preceding arrow icons, gives you access to numerous methods for geometrical selection. Some of these include the ability to select **Vertices** by the number of edges or polygons they connect, by Vertex Map, or by named selection set. For edge selection, you can also select based on boundary. For polygon selection, you can select polygons by type, number of vertices, by named part or selection set, or by material.

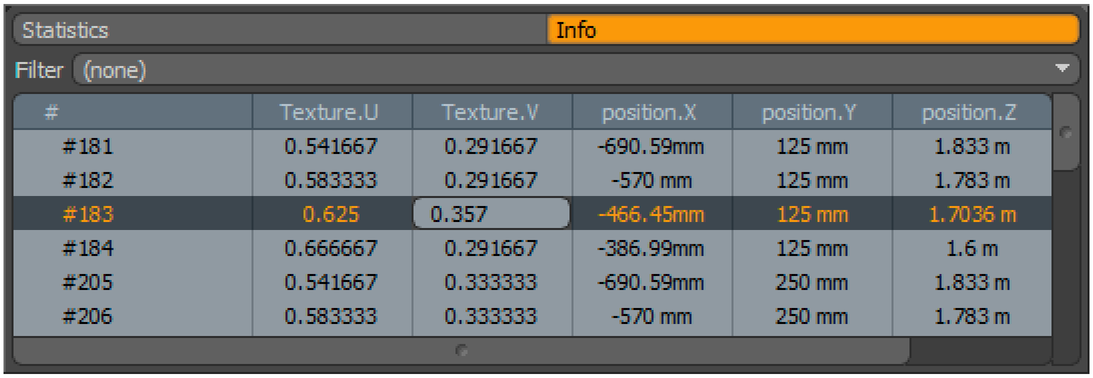
The **Info** tab displays vertex indices, subdivision weights, texture UVs, and position (in X,Y, and Z). The information displayed changes depending on the current mode. The **Info** tab allows inline editing of the displayed values for precise numerical editing. When the view is stretched horizontally it re-organizes into a spreadsheet view. Click on any cell you wish to edit, and type in new values. The filter option provides a means for only displaying a certain type of information, simplifying the list when editing large numbers of values.



Name	Value
Polygon	#283
Type	Face
Vertices	4
Flatness	0 %
Material	Default
Part	Default



#	Texture.U	Texture.V	position.X	position.Y	position.Z
#181	0.541667	0.291667	-690.59mm	125 mm	1.833 m
#182	0.583333	0.291667	-570 mm	125 mm	1.783 m
#183	0.625	0.291667	-466.45mm	125 mm	1.7036 m
#184	0.666667	0.291667	-386.99mm	125 mm	1.6 m
#205	0.541667	0.333333	-690.59mm	250 mm	1.833 m
#206	0.583333	0.333333	-570 mm	250 mm	1.783 m



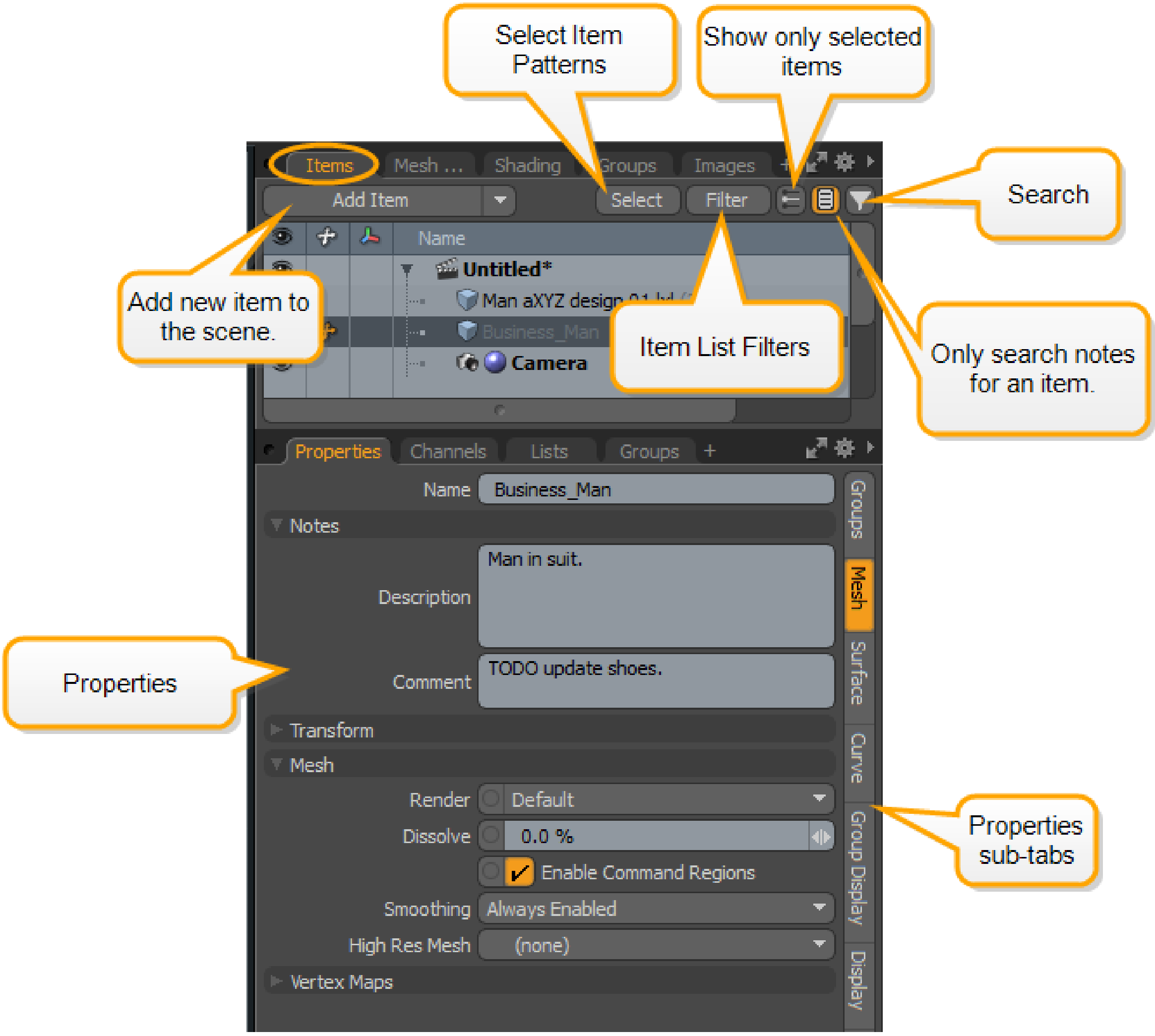
#	Texture.U	Texture.V	position.X	position.Y	position.Z
#181	0.541667	0.291667	-690.59mm	125 mm	1.833 m
#182	0.583333	0.291667	-570 mm	125 mm	1.783 m
#183	0.625	0.357	-466.45mm	125 mm	1.7036 m
#184	0.666667	0.291667	-386.99mm	125 mm	1.6 m
#205	0.541667	0.333333	-690.59mm	250 mm	1.833 m
#206	0.583333	0.333333	-570 mm	250 mm	1.783 m

When opened wide, the **Info** display changes into a spreadsheet view.

Items (Scenes) List

Note: In the Modo layout, the **Items List** is named **Scenes**, in all other layouts it is named **Items**.

The **Items List** has a number of functions: it lists all locator-type items within your project and allows you to create, edit, and organize various item layers (meshes, locators, cameras, for example). The functionality of the **Items List** includes selection, creation, deletion, and re-ordering of items and entire scene files, creating parent and child hierarchies, setting visibility, editing item properties, and creating basic groups for general organization.


Select Item Patterns
Show only selected items
Search
Add new item to the scene.
Item List Filters
Only search notes for an item.
Properties
Properties sub-tabs
Items
Mesh ...
Shading
Groups
Images
Add Item
Select
Filter
Name
Untitled*
Man aXYZ design 01.lxl
Business_Man
Camera
Properties
Channels
Lists
Groups
Notes
Transform
Mesh
Vertex Maps
Groups
Mesh
Surface
Curve
Group Display
Display

Name: Business_Man

Description: Man in suit.

Comment: TODO update shoes.

Render: Default

Dissolve: 0.0 %

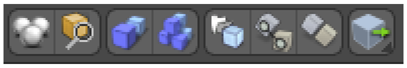
☑ Enable Command Regions

Smoothing: Always Enabled

High Res Mesh: (none)

The Scenes Viewport

If you are using the Modo layout, the **Items List** is named **Scenes**. The functions within the tab remain the same as the **Items List**, and the following buttons are also displayed above the **Scenes** viewport.



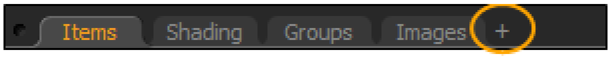
Item List Solo - When enabled, all unselected items are hidden.

	Note: You can also **Alt** + **Click** an item in the **Items List** to hide all other items and enable **Item List Solo**. See Items List Controls for more information. **Ctrl** + **Click** on the **Item List Solo** button to solo selected items and their children. **Shift** + **Click** on the **Item List Solo** button to hide just the selected item. **Note:** The **Item List Solo** button can also be accessed from the Shading and Groups viewports.
	Item List Find - Click to find the selected item in the **Items List**.
	Duplicate Item - Click to duplicate a selected item in the **Items List**. For more information, see Duplicate Commands.
	Duplicate Hierarchy - Click to duplicate a selected item and its children in the **Items List**. For more information, see Duplicate Commands.
	Group Selected Items - Click to group the selected items together under a **Group Locator** item. For more information, see Group Locator Item.
	Parent in Place - Click to set multiple items in the **Items List** as children, where the parent is the last item to be selected in the **Items List**. For more information, see Hierarchy Tools.
	Unparent in Place - Click to unparent children item(s) from their parent item in the **Items List**. For more information, see Hierarchy Tools.
	Export Selected - Click to save out selected items from the **Items List** to a file on disk. For more information, see Export Selected Layers.

Opening the Items Viewport

- On the right panel, click the **Items** tab.

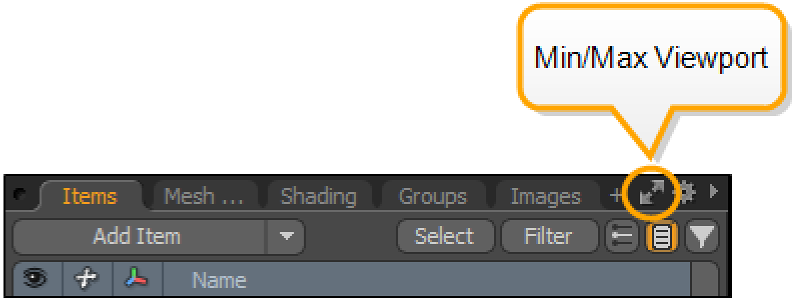
Tip: If the **Items** tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **Data List** > **Items List**.



Resizing, Reordering, and Repositioning

You can change the display size, order, and position of the **Items List** viewport.

The **Items List** has two viewport sizes: Maximize/Un-Maximize Viewport.



- **Maximize** - Shows all items and acts as a complete project manager displaying all of the Properties sub-tabs on the bottom panel.
- **Un-Maximized** - Converts to a **Mesh List**, allowing a very compact UI for quickly selecting the various meshes in your project.

To resize the viewport window:

- Click-and-drag any frame edge.

To reorder the viewport tabs:

- Click-and-drag the **Items** tab to a new position on the top right panel.

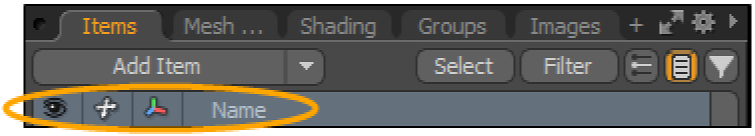
  An orange bar displays to indicate where the tab will be repositioned once you release your mouse button.

To reposition the Items List as a separate viewport:

- Click-and-drag on the **Items** tab and position it anywhere.

  The viewport acts as a separate palette.

Items List Controls



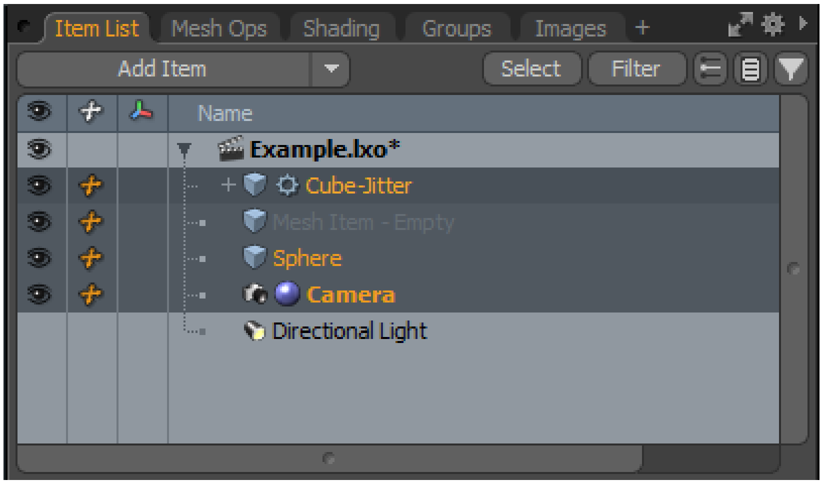
👁	Toggles the visibility of items displayed in the 3D viewport, as well as when rendered, unless specified otherwise by the Display Viewport or the item's **Properties** panel. **Tip:** **Alt** + **Click** the 👁 (eye) icon in front of an item to hide all other items. If the item itself is selected, this hides all unselected items. If all items are hidden, **Alt** + **Click** the icon to reveal them. **Shift**+**Alt**+clicking the icon hides or reveals the selected item itself. **Alt** + **Ctrl** + **Click** on a grouped item to solo within the group, and hide the children.
✥	Toggles editing to item mode to apply transforms (Move, Scale, and Rotate). **Note:** Only mesh items can be edited in component mode, using any of the tools Modo provides. Locators, cameras, lights, and backdrop items can only be edited in item mode. For more information, see Transforming Items and Selecting Items.
▦	Toggles editing in component mode (vertices, edges or polygons). **Tip:** To change the editing mode from Items to a component, under the layout menu bar, click **Vertices**, **Edges** or **Polygons**. The icon updates. **Note:** Component versus item transforms are an important concept to understand in Modo. If you think of Mesh Items as containers, item mode transforms are applied to the entire container, while component transforms modify the contents of the container only. Nearly any item can have an item-level transform applied, be it position, scale, or rotation.
⟂	Toggles the selected item between world space and local space.

Tip: When in local mode, an item's center point temporarily becomes the origin. This can make certain modeling tasks easier.

For instance, if you modeled a car's tire by radially sweeping around the origin and then re-positioned the tire at the axle of the car in item mode, if you needed to further modify the tire, you could place the item into the local space and re-sweep the tire around the origin. When you're finished, you can return to world space without ever needing to move a single item. This action may also be accomplished by invoking the menu bar command **Item** > **Set Reference System**.

Layers in the Items List

Items in the list can only be modified when selected. You can also edit multiple items at once. When selecting layers in this view, using **Ctrl** and **Shift** to select multiple items, you may notice there are some combinations of font styles and colors that indicate various layer states. Selected layer row(s) appear darker than non-selected row(s), and the layer name text becomes orange. Selected layers are considered foreground layers. Item layers that are visible, but not selected, are considered background layers. Background layer name text appears as plain black text. Mesh items that have been created, but contain no data, have light gray layer name text.



At the top of the list is the scene item, representing the scene file itself in bold text. Below it are the various item layers that make up the scene: Mesh Items, lights, cameras, and so on. Visibility of the various layers is controlled in the left-most column.

Finding Items

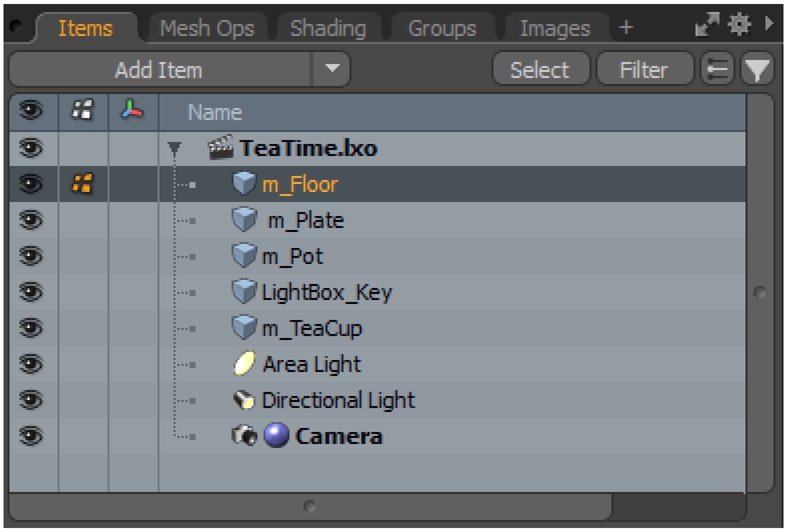
- Click on the item in the 3D view and press the **F** key to find the item in the **Items List**.

  The list auto-expands any necessary group locators or parents to reveal your item.

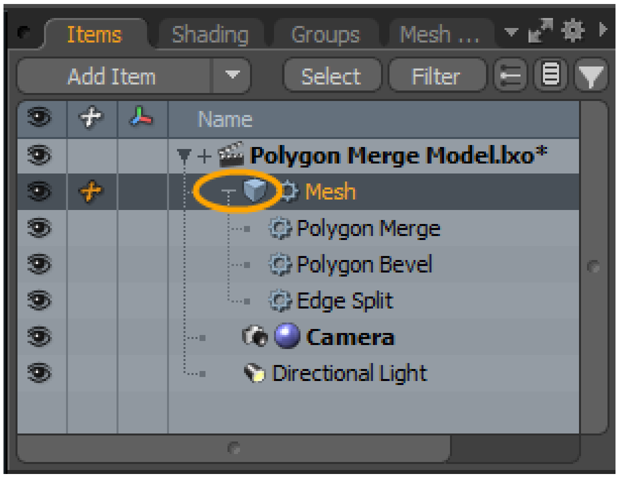
Tip: You can also quickly expand/contract an entire hierarchy branch with a single action. Holding **Shift** when clicking to expand or contract a hierarchy, expands/contracts the current item, as well as all items down the tree.

Setting a Layer to be a Parent

The full **Items List** view can also be called the "tree view", as the layers are displayed in a tree structure from top to bottom. It's called a tree because in this view you can also see the hierarchy of the layers, if any has been set. In most cases, the mesh layers are in a flat hierarchy, indicating that there is no parent-child relationship between any of the layers.


Items
Mesh Ops
Shading
Groups
Images
+
Add Item
Select
Filter
Name
TeaTime.lxo
m_Floor
m_Plate
m_Pot
LightBox_Key
m_TeaCup
Area Light
Directional Light
Camera

If a layer has been parented to another, that child appears beneath and is slightly indented to the right of the parent. The parent layer then has an **+/-** expand/contract widget to the left so that you can easily collapse entire hierarchies quickly.


Items
Shading
Groups
Mesh ...
Add Item
Select
Filter
Name
Polygon Merge Model.lxo*
Mesh
Polygon Merge
Polygon Bevel
Edge Split
Camera
Directional Light

To set a layer to be parented by another layer:

1. Drag-and-drop the intended child on top of the intended parent layer.

   The child layer appears beneath the parent and is slightly indented to the right.

2. If you want to move the child layer back to the same level as the parent, drag-and-drop the child layer back into the list between two layers.

   Be careful to ensure you don't drop it on top of another layer by mistake, or it is moved to be the child layer of a different parent.

Parent/Child Hierarchy Shortcuts for 3D Views

Select items in any 3D view and apply any of the following shortcuts.

- **P** - Creates a parent. Selected items are displayed in the 3D view in their original position and listed as a parent/child in the Items List.
- **Ctrl**+**P** - Creates a parent and moves the child to the position of the parent. A parent/child hierarchy is listed in the Items List.
- **Shift**+**P** - Un-parents.
- **Ctrl**+**Shift**+**P** - Un-parents.

Additional information on parent/child relationships is available under Hierarchy.

Using Auto Visibility

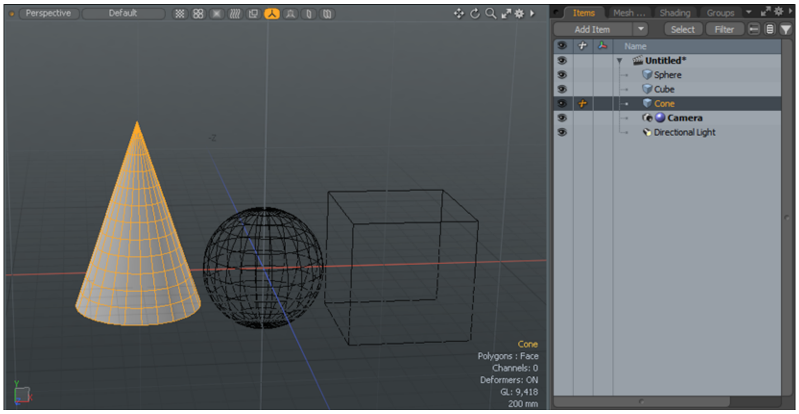
There are two very different workflows between modeling and layout/rendering processes. When modeling, you may find that you want to focus on a single Mesh Item (often referred to as a layer) but the layout process demands a complete view of the project.

Modo offers a solution using **Auto Visibility**, which turns visibility on for the currently-selected item and hides all de-selected items. In this way, you can easily set the focus to a specific Mesh Item. Once the item is visibly isolated, you can use the **Shift** or **Ctrl** keys to add to the selection.

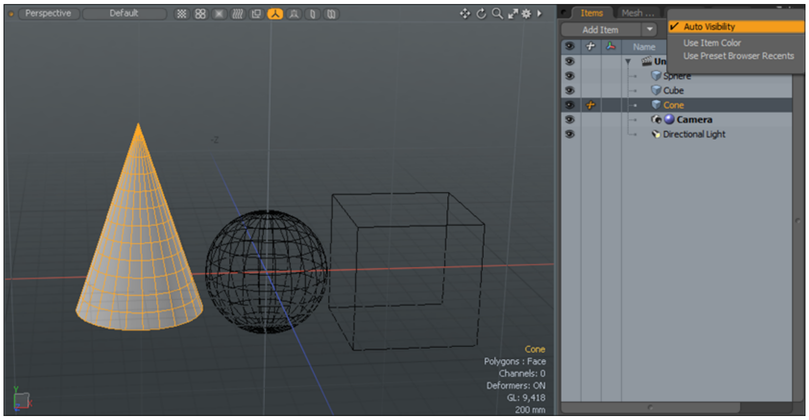
If you want to have one layer active but another layer visible and inactive, you can **Ctrl**+click on that item's visibility column to override the **Auto Visibility** setting at that time.

To apply Auto Visibility:

1. On the right panel **Items List**, select an item in your list.

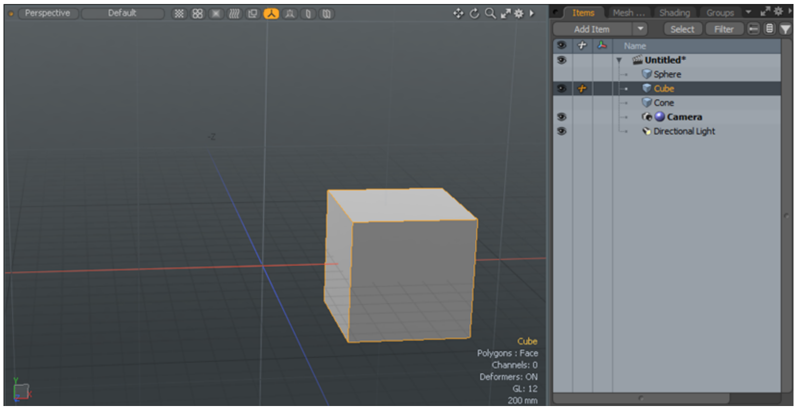

Perspective
Default
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Sphere
Cube
Cone
Camera
Directional Light
Cone
Polygons : Face
Channels: 0
Deformers: ON
GL: 9,418
200 mm

2. Click the gear icon on the top right panel and select **Auto Visibility**.


Perspective
Default
Items
Mesh ...
Auto Visibility
Use Item Color
Use Preset Browser Recents
Add Item
Name
Sphere
Cube
Cone
Camera
Directional Light
Cone
Polygons : Face
Channels: 0
Deformers: ON
GL: 9,418
200 mm

3. In the **Items List**, select another item.

   Only this item is displayed in the 3D viewport and the visibility icon is removed from the other items in the list.


Perspective
Default
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Sphere
Cube
Cone
Camera
Directional Light
Cube
Polygons : Face
Channels: 0
Deformers: ON
GL: 12
200 mm

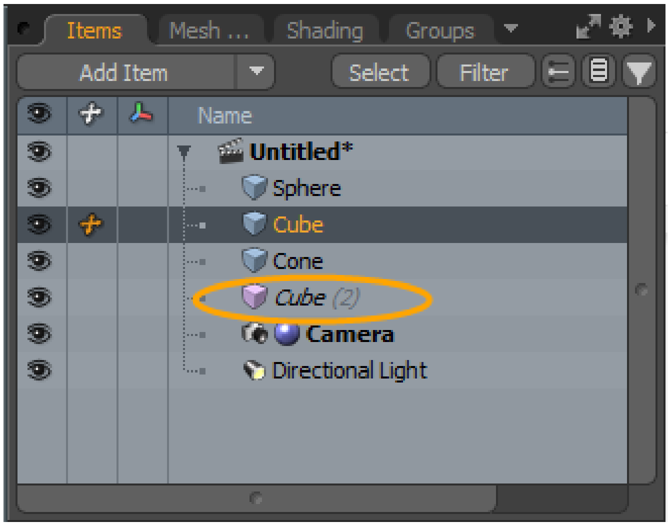
4. Select other items in the **Items List** to display them in the 3D viewport.

Tip: Use the **Shift** or **Ctrl** keys to add to the selection.

Creating an Instance

You can quickly create an instance from a selected item in the **Items List**. The instance is only relevant for the selected Items. The result is a duplicate of the selected item that is simply a reference to the original item. The advantage is that all item changes made to the original Item propagate to its instances.

Instances can be transformed at the item level and can have their own materials and textures applied to them using item-level masks. You can make an instance of all items in the **Items List**. Instances are displayed in *italic* and their icons are pink.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Sphere
Cube
Cone
Cube (2)
Camera
Directional Light

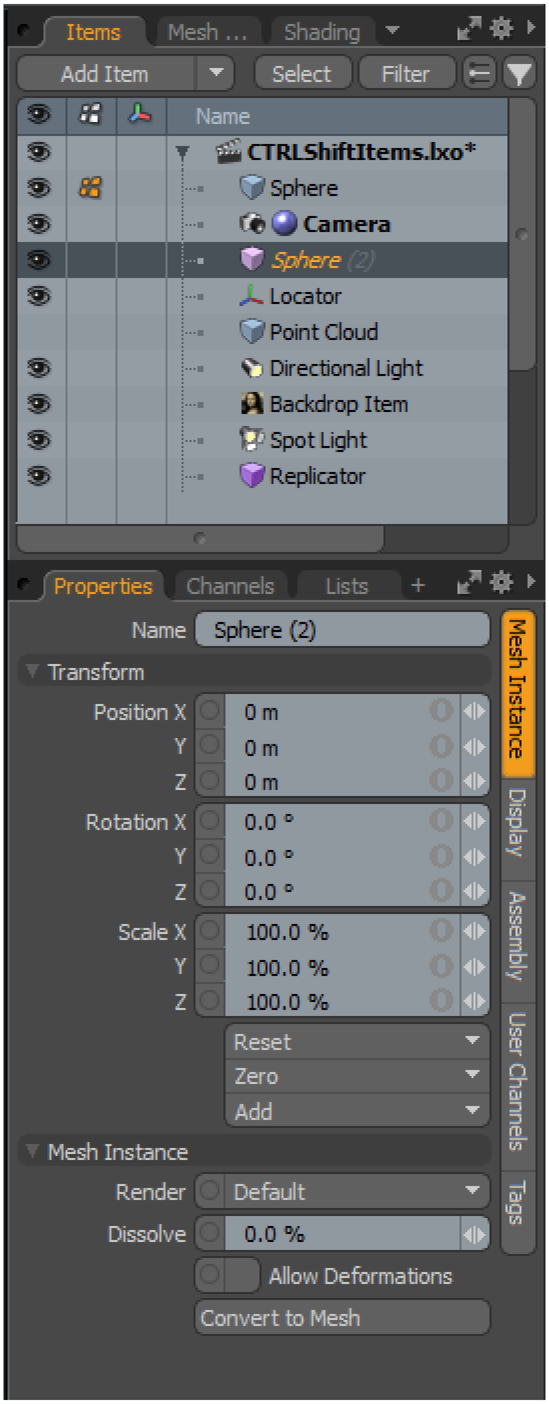
Note: Instances can not be edited at the instance-level, as they derive all information from the original Item.

Tip: You can also use the **Ctrl/Cmd** + **Shift** keys to create instances in the Mesh Operations List and add keyframes in the Graph Editor.

To create an instance:

1. In the **Items List**, press **Ctrl/Cmd** + **Shift**, click on a item, and drag the item to anywhere in the **Items List** tree.

An instance is created and the properties are displayed in the **Properties**. In this example, we have created an instance of the Sphere mesh.


Items
Mesh ...
Shading
Add Item
Select
Filter
Name
CTRLShiftItems.lxo*
Sphere
Camera
Sphere (2)
Locator
Point Cloud
Directional Light
Backdrop Item
Spot Light
Replicator
Properties
Channels
Lists
+
Mesh Instance
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Mesh Instance
Allow Deformations
Convert to Mesh

Name: Sphere (2)

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

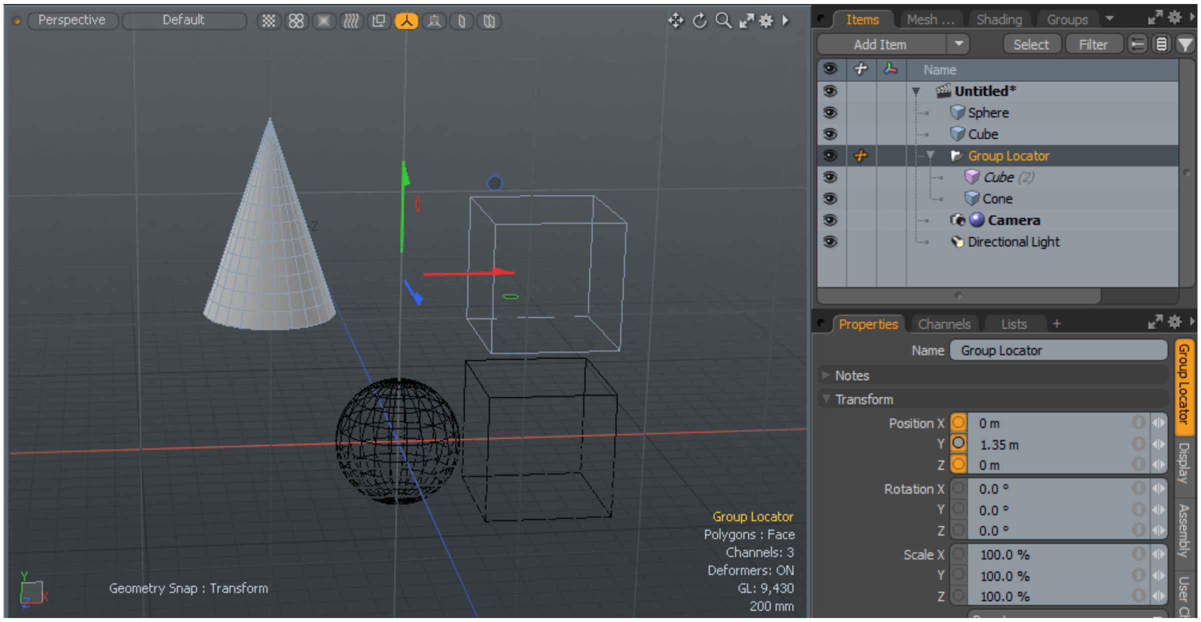
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Render: Default

Dissolve: 0.0 %

Creating Group Locators

As your projects become more complex, you may find a long flat list of items becomes unwieldy. Parenting items to a Group Locator Item helps organize a project. Different from a Groups Viewport, group locators are a special kind of locator item, as they do not draw anything in the 3D views (removing additional scene clutter) and can be easily-collapsed, which keeps the **Items List** clean and manageable. Group locators have an icon that looks like a folder in the **Items List** to distinguish them from regular locator items. In all other aspects they act the same as a normal locator; for example, they can be moved and rotated in the 3D views.


Perspective
Default
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Sphere
Cube
Group Locator
Cube (2)
Cone
Camera
Directional Light
Properties
Channels
Lists
Notes
Transform
Group Locator
Display
Assembly
User Ch
Geometry Snap : Transform

Group Locator
Polygons : Face
Channels: 3
Deformers: ON
GL: 9,430
200 mm

Name: Group Locator
Position X: 0 m
Y: 1.35 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

To create a group locator:

1. In the Items List, click **Add Item** > **Locators** > **Group Locator**.
2. Drag-and-drop selected items into a hierarchy to organize the list

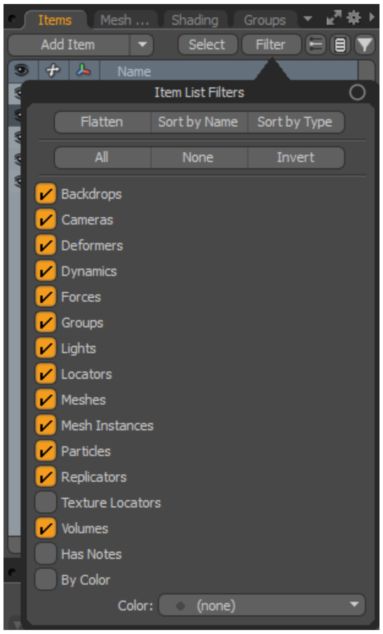
   Alternatively, select items in the list and press **Ctrl**+**G** to create a new group locator containing the selected items.

Filtering the Items List

You can filter the **Items List** to display certain types of items and order the items.

1. Click **Filter Items** to open the **Items List Filters** window.

The **Item Lists Filter** dialog contains a series of buttons for ordering and selecting items, plus a list of item types.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Item List Filters
Flatten
Sort by Name
Sort by Type
All
None
Invert

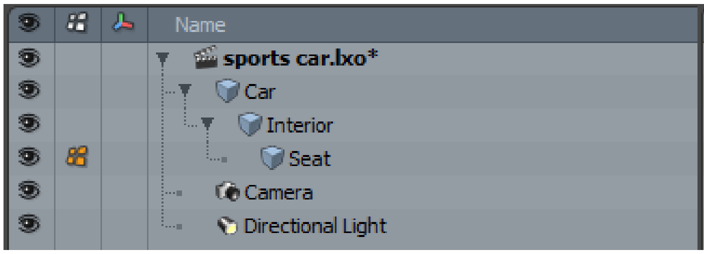
☑ Backdrops
☑ Cameras
☑ Deformers
☑ Dynamics
☑ Forces
☑ Groups
☑ Lights
☑ Locators
☑ Meshes
☑ Mesh Instances
☑ Particles
☑ Replicators
☐ Texture Locators
☑ Volumes
☐ Has Notes
☐ By Color

Color: (none)

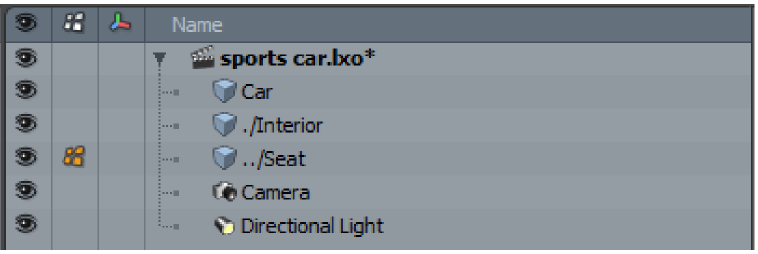
2. Uncheck the box in front of an item type to hide all items of that type in the **Items List**.

The following ordering buttons are available:

- **Flatten** - Displays the items in a flat list instead of showing nested relationships in a tree.


Name
sports car.lxo*
Car
Interior
Seat
Camera
Directional Light

A normal list with a hierarchical tree of nested meshes.


Name
sports car.lxo*
Car
./Interior
../Seat
Camera
Directional Light

A flattened list with each mesh at the base level of the list. The hierarchy is indicated by dots in front of the item name with one dot for each level of nesting.

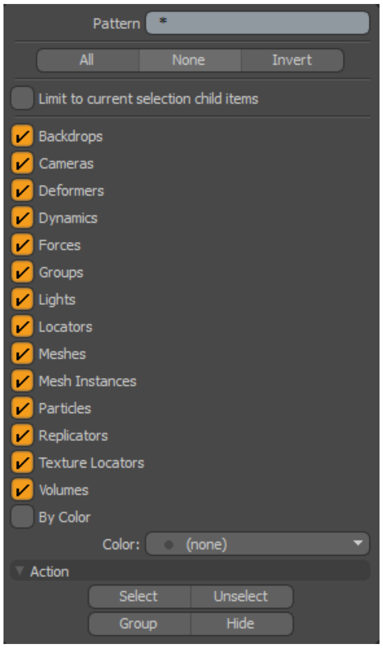
- **Sort by Name** - Sorts the list of items alphabetically.
- **Sort by Type** - Sorts and groups the items by their type.

The following buttons change the selection:

- **All** - Selects all item types.
- **None** - Deselects all item types.
- **Invert** - Reverses the selection.

Selecting Items

The **Select** button at the top of the **Items List** allows you to select items using a specific text pattern. For instance, typing **cam*** into the Pattern field, selects every item with a label starting with "cam".



Pattern: *

☐ Limit to current selection child items

☑ Backdrops
☑ Cameras
☑ Deformers
☑ Dynamics
☑ Forces
☑ Groups
☑ Lights
☑ Locators
☑ Meshes
☑ Mesh Instances
☑ Particles
☑ Replicators
☑ Texture Locators
☑ Volumes
☐ By Color

Color: (none)

Similarly to the **Filter** button, you can apply the selection pattern to several generic types, or color tags. You can uncheck the box in front of an item type to remove all items of that type from the selection. Controls at the top of the panel allow you to quickly select all or none of the item types, or to invert the existing selection.

Enabling the **Limit to current selection child items** option selects only the items parented under items in the active selection.

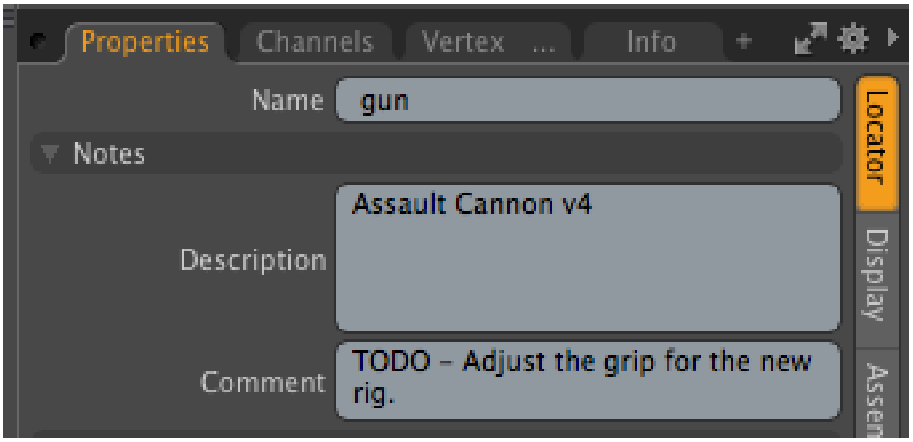
The following options are available at the bottom of the panel.

- **Select** - Selects items using the pattern (and removes the existing selection).
- **Unselect** - Removes items using the pattern from the existing selection.
- **Group** - Creates a group using the selected pattern.
- **Hide** - Hides items using the selected pattern.

Tip: You can hide or reveal selected items by **Alt**+clicking on the 👁 (eye) icon in front of an item in the list. This sets all the other items to hidden. **Alt**+clicking on the icon again reveals them.

Adding, Filtering, and Searching Notes

The Items List **Properties** tab has a **Notes** section (collapsed by default), which is useful to manage your project details.


Properties
Channels
Vertex ...
Info
Notes
Locator
Display
Assem

Name: gun

Description: Assault Cannon v4

Comment: TODO - Adjust the grip for the new rig.

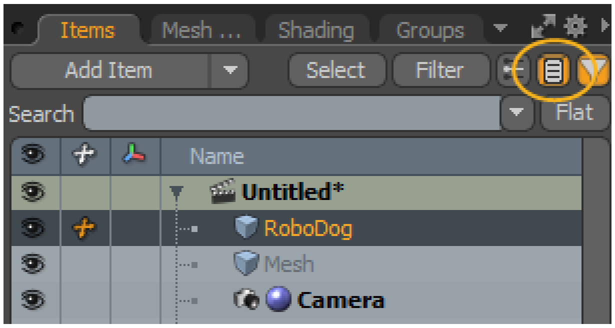
When an item has a description or comment, these are reflected in the tooltips in the Items List and 3D views.

Tip: You can use Modo's markup feature to add formatting to your notes, such as bolding, italics, bullets, and so on. For more information, see Modo Markup for Forms.

Items with descriptions and comments can be filtered in the Item List. For more information, see Filtering the Items List .

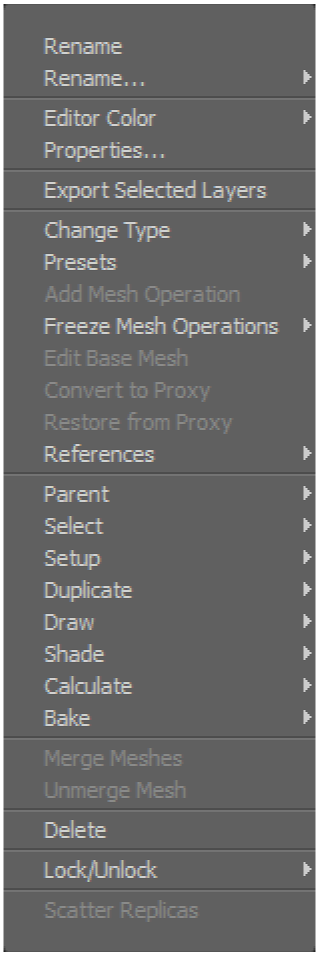
Comments and descriptions can also be searched by clicking a button next to the Filter button in the top-right corner of the Item List. With this enabled, Modo only looks for the search term in the **Notes** sections of items.

For more information, see Selecting Items.

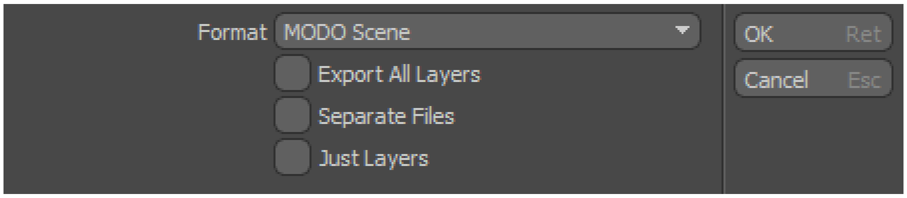

Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Search
Flat
Name
Untitled*
RoboDog
Mesh
Camera

Item Context Menu

There are several options available using the **Items List** context menu. Simply right-click on any item to open the menu.


Rename
Rename...
Editor Color
Properties...
Export Selected Layers
Change Type
Presets
Add Mesh Operation
Freeze Mesh Operations
Edit Base Mesh
Convert to Proxy
Restore from Proxy
References
Parent
Select
Setup
Duplicate
Draw
Shade
Calculate
Bake
Merge Meshes
Unmerge Mesh
Delete
Lock/Unlock
Scatter Replicas

- **Rename** - Allows you to rename the selected item using a pop-up dialog.
- **Rename...** - Allows you to rename multiple items based on defined rules. For more information, see Renaming Multiple Items.
- **Editor Color** - Provides you with the means to colorize item layers to help visually call out a layer, mostly for organizational purposes.
- **Properties** - Opens a panel with the currently-selected items available attributes.
- **Export Selected Layers** -Allows you to save out individual layers from a scene to a file on disk. You need to first select the target files in the **Items List** to export then right-click any of the selected item layers to open the context menu. Select the command **Export Selected Layers** to open the **Export Selected** dialog.



The dialog offers some options for how the selected layers are exported:

- **Format** - You can determine the save format of the exported items from the available options. For more information on import and export formats, see Importing and Exporting File Formats.
- **Export All Layers** - When enabled, all layers are exported regardless of selection; when disabled, only the selected layers are exported.
- **Separate Files** - When enabled, separate layers are saved as individual files; when disabled, all layers are saved as if they were a single, combined layer.
- **Just Layers** - When enabled, only Mesh Item layers are saved, ignoring cameras, lights, locators, and other non-Mesh Items. When disabled, all selected layers are exported. What can be exported is highly dependent on the chosen target format.

- **Save Assembly** - Allows you to save entire hierarchies of items of various types out to the preset file format (**.lxp**). The resulting preset can be later loaded from the **Preset Browser**, retaining all the settings of the original hierarchy. This command should always be applied to the top-most parent of the hierarchy.
- **Change Type** - Converts an item into another item type. For instance, you can convert an instance mesh into a real mesh so that you can edit the item at the component level. Another example is to convert a directional light into an area light. Not all items can be converted to all other types of items.

Note: Certain parameters may be lost if you switch back to the original item type, though common parameters should be preserved.

- **Presets** - Allows you to load, apply, and delete preset parameters to your objects.
- **Add Mesh Operation** - Opens the **Preset Browser**, where you can add mesh operations, deformers, and assemblies to your scene.
- **Convert to Proxy** - Proxies are a function that externalizes one or more layers of the **Items List** and replaces them with a simple bounding box. The resulting bounding box is only replaced by the full-resolution proxy at render time. For more in-depth coverage for proxies, refer to Render Proxy.
- **Restore from Proxy** - Restores any item that has been converted to an external proxy can be loaded back into the originating scene.
- **Parent/Parent In Place** - Creates a parent-child hierarchy of several items. To invoke, select multiple items within the **Items List**, right-click on any of them, and select **Parent** from the menu. The top-most item in the **Items List** becomes the parent, while all the subsequent items become children. The **Parent in Place** command retains an item's initial position prior to parenting.

- **Parent to Group Locator** - Creates a group locator item and parents all currently-selected items to it. The same can also be achieved by pressing **Ctrl+G**.
- **Unparent/Unparent In Place** - Removes parent/child relationships of items within any hierarchy. The **Unparent In Place** command retains the item's position when unparented, relative to its parent.
- **Select Item Hierarchy** - Selects the entire hierarchy down the chain from the current item.
- **Pick Walk** - Pick walking is used to make it easier to navigate complex hierarchies, as it allows you to easily create selection relationships up and down a hierarchy chain. These can be defined in the Assembly Viewport.
- **Clear Selection Redirect** - Selection redirects are useful to aid in selecting specific items in a 3D viewport. As an item, such as a locator, is selected in a 3D viewport, that selection can be redirected to another object that would be difficult to individually select otherwise. The **Clear Selection Redirect** command is an easy way to remove a selection redirect.
- **Apply to Setup/Rest** - When working with animation or deformers, the initial state of any item is defined while in **Setup** mode. This represents the zero position state, and everything else is an offset from the **Setup** value. This command takes the item's current values and pushes them to **Setup**.
- **Restore Setup/Rest** - Removes any modification to the item and restores the values to those of the initial **Setup** state.
- **Restore Setup/Rest (Transforms only)** - Restores only the transform values (move, scale, rotate) to the **Setup** state.
- **Duplicate** - Creates a duplicate of the current item(s).
- **Duplicate Hierarchy** - Creates a duplicate of the current parent item and all child items, while retaining the overall relationships of the hierarchy.
- **Instance** - Creates a duplicate of the current mesh that is simply a reference to the original mesh. Instances can be transformed at the item level and can have their own materials and textures applied to them using item-level masks. However, they can not be edited at the mesh-level, as they derive all mesh information from the original Mesh Item. The advantage is that all mesh changes made to the original Mesh Item propagate to its instances.

Note: Instance is only relevant for Mesh Items.

- **Instance Hierarchy** - Creates instanced versions of a parent item and all child items, while retaining the overall relationships of the hierarchy.
- **Duplicate** - Opens a pop-up dialog with a few options allowing you to choose which items to include or exclude from the duplication process.
- **Merge\Unmerge Meshes** - Works on separate clusters of polygons only while in **Items** mode.
  - **Merge** - takes all the geometry from multiple layers and combines it into a single Mesh Item layer. No vertices are merged and meshes remain separate.

  - **Unmerge** - takes all the separate mesh groups within a single layer and makes as many individual layers as necessary. Unmerged mesh layers still share a single **Center** location, which can be remedied with the menu bar command **Edit** > **Center to Bounding Box** > **Center**.
- **Select Source of Instance** - When working with instanced items, this command provides you with an easy way to select the source Mesh Item layer of any instance.
- **Select Instances** - Allows you to easily select all related instances from a single source item.
- **Select Input/Output Items** - When connecting items together, such as for animation rigging, **Select Input** or **Output Items** option allows you to automatically select the upstream or downstream items in the flow of the connection layout, as defined in the **Schematic** viewport.
- **Delete** - Deletes the currently-selected item's layer.
- **Lock/Unlock** - Locks and unlocks items. This is useful to control accidental manipulation, eliminating the need to do so manually through the **Assembly** viewport. Once an item is locked, a small padlock icon (🔒) appears next to the layer name. Once an item is locked, no item-level transformation modifications may be made. To make a change, you much first unlock the item. Note that this function won't affect or limit component-level transformations. A similar function is also available to lock geometry components. For more information, see 3D (OpenGL) Viewport.
- **Draw Style** - Allows you to set a specific draw style for an item, regardless of the 3D viewport display setting. This is useful for when you want to "fix" the way items within a scene are drawn in the viewport, essentially mixing view styles within one window. For example, you might have a wireframe mixed with shaded, mixed with weight shade.
- **Wireframe Color** - Allows you to change the default display color of an unselected item when displayed as a wireframe. Useful for differentiating multiple items in a complex scene.
- **Fill Color** - Allows you to change the display color of an object in any GL viewport, independent of its surface attributes.
- **Create Item Mask** - Creates a mask group in the **Shader Tree** with the current item as the filter. This allows you to quickly add a material/texture mask at the Mesh Item-level so that you can isolate material changes to that specific item. This is especially useful for applying surface attributes to an instanced object. If your original item is red, for instance, you could change the color of an instance of the item to blue by using an item mask, while adding no additional real geometry to the scene.
- **Create Item Shader** - Item shaders provide per-item overrides of any shading applied within the **Shader Tree**. The **Create Item Shader** option creates a special kind of **Items List** Shader item that is not present in the **Shader Tree**. When selected, the attributes of the shader item appear in the **Properties** viewport. You can select any of the **Control...** options. When enabled, Modo then uses the settings of the item shader for the associated item, superseding the evaluation of the **Shader Tree** for the selected options. This makes it very easy, for instance, to make a single item invisible to the camera. To do this, select **Create Item Shader**, then select the **Control Visibility** option, and then disable the **Visible to Camera** option.
- **Scatter Replicas** - Only applies to Replicators and acts as an easy way to apply a surface generator to an item, allowing finer control over duplicated variations through the **Shader Tree**.

- **Calculate Mesh Volume** - Reports the volume of the associated mesh in several way, displaying in a pop-up dialog
- **Calculate Center of Mass** - Generates a parented locator item to the mesh that represents its **Center of Mass**, a point that represents its median location for the mass distribution of the associated mesh. The item's center can be positioned here for dynamics purposes using the **Match** commands found in the **Setup** layout.

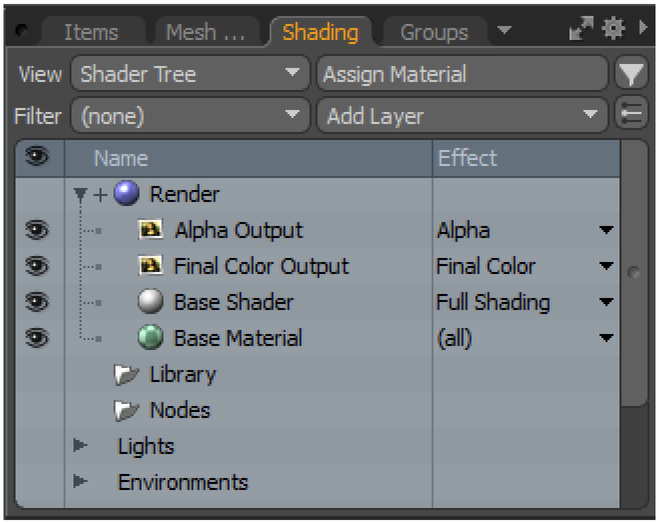
Project Item Context Menu

Right-clicking the file name itself in the **Items List** opens the file context menu. This is where you perform routine tasks, such as:

- **Save** - Saves the scene loaded in memory to disc, overwriting the previously-saved file.
- **Save As** - Saves the scene currently loaded in memory to a specified name and location.
- **Export As** - Exports the entire scene to a variety of different formats.
- **Close Scene** - Removes the scene and all related files, like images, MDD, and IES light files from memory. When the scene is closed, Modo defaults to a blank new scene, if no other scenes are loaded.

Items List Versus Shader Tree

You may have already noticed that some Modo items appear in both the **Items List** and the **Shader Tree**. The **Items List** is generally used to select locator-type items. Locator-type items are items that can be selected and edited directly in the 3D viewport, meaning that they have a location. There are items that are not locator-types. For instance, materials, shaders, and texture layers are also items but are not directly editable in the 3D viewport.



View Shader Tree

Filter (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Base Material	(all)
Library	
Nodes	
Lights	
Environments	

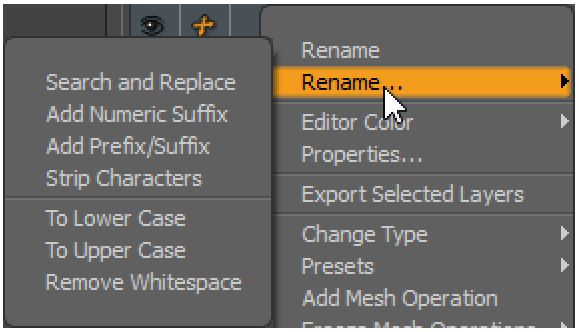
These items are accessed from the **Shader Tree** rather than the **Items List**. There are some crossover items that appear in both the **Items List** and the **Shader Tree**. This is because they may share common, linked items such as a texture layer and its associated locator item (texture positions are set by Texture Locators, which can be manipulated in 3D), or items that can be modified by textures (lights have a material and can be textured within the **Shader Tree**).

For more information, see Shader Tree and Shader Tree Item Layers.

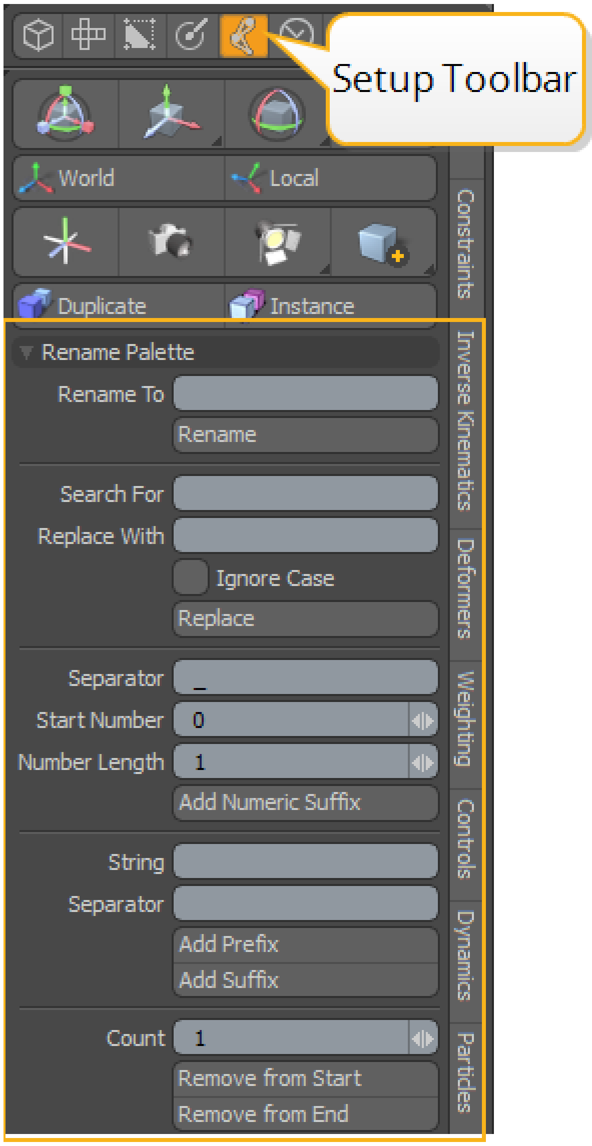
Renaming Multiple Items

You can rename multiple items at once, based on defined rules. Select the items, right-click them in the **Items** list, then click **Rename...**

Here you have multiple options.


Search and Replace
Add Numeric Suffix
Add Prefix/Suffix
Strip Characters
To Lower Case
To Upper Case
Remove Whitespace
Rename
Rename...
Editor Color
Properties...
Export Selected Layers
Change Type
Presets
Add Mesh Operation

You can also open the **Rename Palette** on the left panel by clicking the Setup Toolbar button.


Setup Toolbar
World
Local
Duplicate
Instance
Rename Palette
Rename To
Rename
Search For
Replace With
Ignore Case
Replace
Separator
_
Start Number
0
Number Length
1
Add Numeric Suffix
String
Separator
Add Prefix
Add Suffix
Count
1
Remove from Start
Remove from End
Constraints
Inverse Kinematics
Deformers
Weighting
Controls
Dynamics
Particles

Search and Replace

You can look for a string in the item names and replace every instance of it with another string. For example, if you want to replace every instance of 'Big' in the selected item names with 'Large', you can use this option. Clicking **Search and Replace** opens the **Search and Replace** dialog.

Enter the string you're looking for in the **Search** field, and the new string in the **Replace** field, then click **OK**. Your items are renamed.

Note: By default, **Ignore Case** is disabled, so the search is case-sensitive. This means that if you want to rename every instance of 'Big' in item names to 'Large', instances of 'big' aren't replaced, because they are lowercase. Enable **Ignore Case** if you want to turn off case-sensitivity.

Adding Numeric Suffixes

Add Numeric Suffix allows you to add numbers to the end of item names. If you have a long list of items of the same type, you can use this option to number them. Click this option to open the **Add Numeric Suffix** dialog and specify the following:

Separator	The character between the item name and the numeric suffix. If no separator is specified, the suffix is attached directly to the item name.
Start Index	The starting numeric value.
Minimum Length	The number of numeric characters the suffix contains. This setting is overridden if the number of items you are renaming is larger than what the minimum length allows. For instance, if the **Minimum Length** is '1', but you're renaming 12 items, some suffixes will contain two numeric characters.

Adding Prefixes or Suffixes

Add Prefix/Suffix allows you to add a string as a prefix or suffix to item names. Click this option to open the **Add Prefix/Suffix** dialog and specify the following:

String	The string to be added to the item names.
Separator	The character between the item name and the prefix/suffix. If no separator is specified, the suffix is attached directly to the item name.
Prefix or Suffix	Specify if the string should be attached to the beginning or the end of the item name.

Stripping Characters

Use this option to remove a certain number of characters from the beginning or end of the selected item names. In the **Strip Characters** dialog, specify the following:

Start or End	Specify whether to delete characters from the beginning or the end of the item name.
Count	The number of characters to be removed.

Changing Case and Removing Whitespace

To make the selected item names all lowercase or uppercase, click **To Uppercase** or **To Lowercase**, respectively.

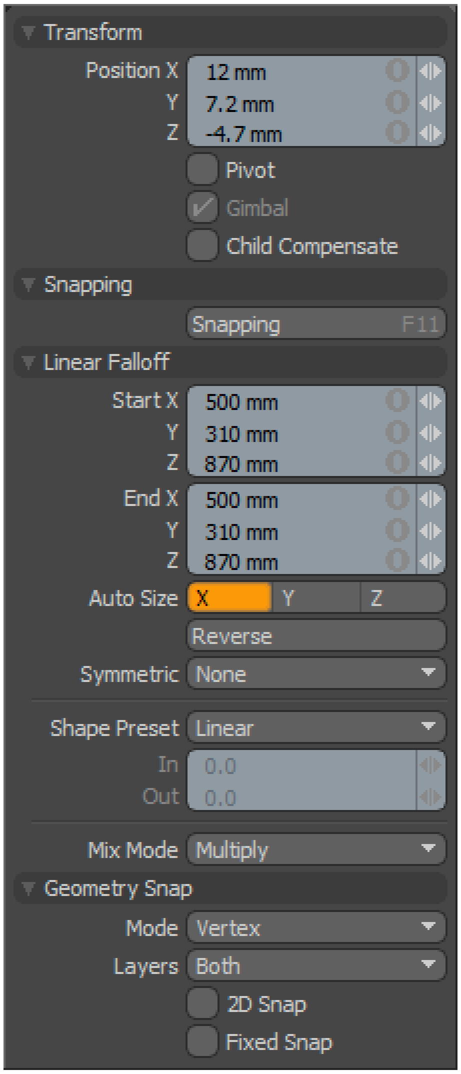
To remove all spaces from the item names, click **Remove Whitespace**.

Properties Panel

The **Properties** panels are the chameleons of the viewport family, as they are basically empty frames until an item or tool is selected. Once selected, all the editable attributes related to that item are listed in the panel, where you can make changes to the options available and, if applicable, see them update in the viewport or render previews.

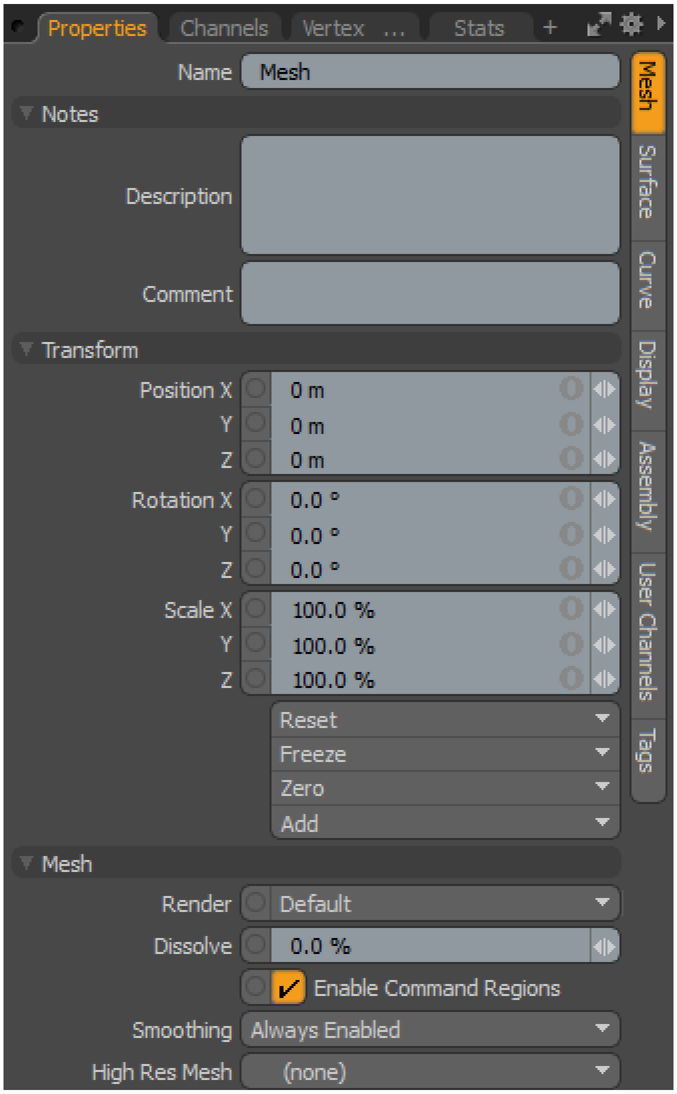
The main **Properties** panels are those of the toolbox and item properties. In the default **Model** interface, the toolbox **Properties** panel appears on the left side of the interface under the toolbox itself. Upon loading, the panel is empty, but as tools are selected, the attributes appear in it. Additionally, when falloffs, action centers, and snapping modes are enabled, additional sections of the **Properties** panel begin to fill, presenting all the editable settings related to the active tools and functions.

This is the same case with the item properties, located on the right side of the default interface. Unlike the tool properties, it is a tab, so you may need to select the particular **Properties** tab to make it visible. Upon loading Modo, the panel appears empty, but selecting items populates the panel with the various attributes related to the selected items. Unlike the tool properties, where the various additional attributes stack up, the item properties adds additional sub-tabs down the right edge of the panel that relates to all selected/active item layers. You can select from the available sub-tabs by simply clicking on them.



Transform
Position X 12 mm
Y 7.2 mm
Z -4.7 mm
☐ Pivot
☑ Gimbal
☐ Child Compensate
Snapping
Snapping F11
Linear Falloff
Start X 500 mm
Y 310 mm
Z 870 mm
End X 500 mm
Y 310 mm
Z 870 mm
Auto Size X Y Z
Reverse
Symmetric None
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply
Geometry Snap
Mode Vertex
Layers Both
☐ 2D Snap
☐ Fixed Snap

Toolbox Properties



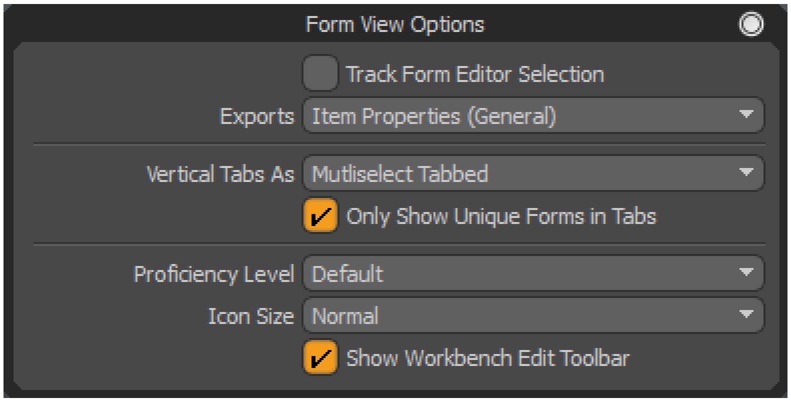
Properties
Channels
Vertex ...
Stats
+
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

Name Mesh
Notes
Description
Comment
Transform
Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Reset
Freeze
Zero
Add
Mesh
Render Default
Dissolve 0.0 %
☑ Enable Command Regions
Smoothing Always Enabled
High Res Mesh (none)

Item Properties

Form View Options

Pressing the **O** key in the **Properties** panel opens the **Form View Options**, where you can customize the appearance and content of the form.



Form View Options

☐ Track Form Editor Selection

Exports: Item Properties (General)

Vertical Tabs As: Mutliselect Tabbed

☑ Only Show Unique Forms in Tabs

Proficiency Level: Default

Icon Size: Normal

☑ Show Workbench Edit Toolbar

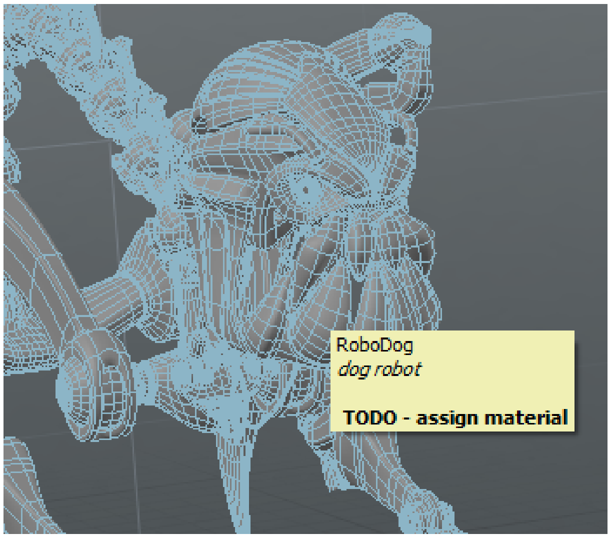
>

Track Form Editor Selection	When enabled, the form view displays what is currently selected in the Form Editor.
Exports	When a form's **Exported** option is enabled in the Form Editor, the form can be selected in a form view. This is also indicated by the X column in the tree. This dropdown lists the forms with **Exported** enabled. Selecting one displays it in the form view.
Vertical Tabs As	Select how vertical tabs are displayed. The following options are available: • **Tabbed** - Tabs are on the right, with one tab selectable at a time. • **Multiselect Tabbed** - Tabs are on the right, and you can select multiple tabs at the same time by **Ctrl**+clicking them. • **Merged Tabbed** - All tabs are merged into a single scrolling form. The tabs remain visible on the right and can be used to jump to different parts of the form. Tabs that are visible are drawn as selected, and change as you scroll the form. • **Merged** - All tabs are merged into a single scrolling from and the tab bar is hidden.
Only Show Unique Forms in Tabs	When enabled (the default), this hides subsequent references to the same sub-form when in a **Merged** or **Multiselect** mode. This helps to avoid redundant controls, such as the **Name** field at the top of many Item **Properties** forms.
Proficiency Level	This option is used to change the level of detail displayed in the form. • **Default** - Uses the **Base Form Proficiency Level** preference in the form. For more information, see Defaults Preferences. • **Basic** - A short list of the common core properties.

	• **Standard** - A list of all of the core properties. • **Advanced** - A list of all of the core properties plus custom functions for special behaviors.
Icon Size	Override's the form's default icon size with a custom size. Normal uses the icon size set in the form itself in the **Form Editor**. The following other options are available: • **Extra small Icons** • **Small Icons** • **Large Icons**
Show Workbench Edit Toolbar	Enabled by default, when there are any Workbenches in the form, this shows/hides their Edit toolbar.

Item Notes

All items' **Properties** tabs contain a **Notes** section where you can add notes and comments about the item. The text you enter here is displayed as a tooltip when you hover the cursor over the item in the 3D viewport or the **Items** list.



RoboDog

dog robot

TODO - assign material

The notes section is collapsed by default. Click the small arrow in front of its name to expand it. Fill in the **Description** and/or **Comment** field. The item's tooltip is populated with the new information.

You can also filter your items based on having notes, and search the notes in the Item List. For more information, see Adding, Filtering, and Searching Notes.

Tip: You can use Modo's markup feature to add formatting to your notes, such as bolding, italics, bullets, and so on. For more information, see Modo Markup for Forms.

Using Tool State Presets

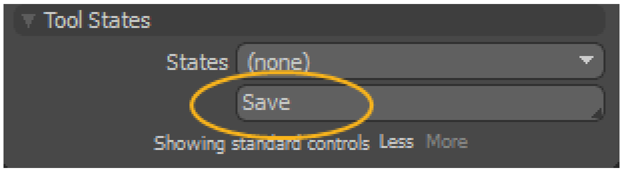
Modo allows you to save the state of a tool. This can be handy if you frequently use a tool for the same task, with the same settings. You can save the settings as a Tool State preset, and quickly recall the same state when needed.

The **Tool States** menu can be found in the **Tool Properties** tab when a tool is active.

Recalling a Tool State Preset

1. Select the required geometry.
2. Activate the tool.

   The tool's properties open on the lower left panel. In the Modo layout, a **Tool Properties** button appears. Click it or press **K** to open the panel.
3. Adjust the properties as needed, then under **Tool States**, click **Save**.

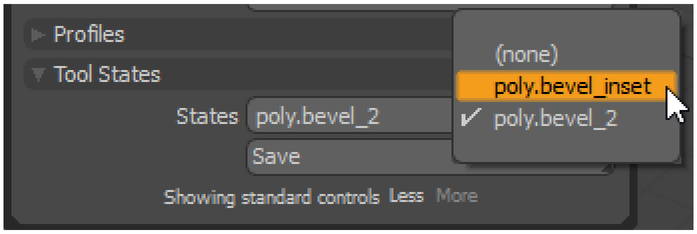


   Modo prompts you to name your preset.
4. Enter a name for the preset, then click **OK**.

   Your preset is saved and you can select it in the **States** field when you're using the tool.

Invoking a Saved Tool State

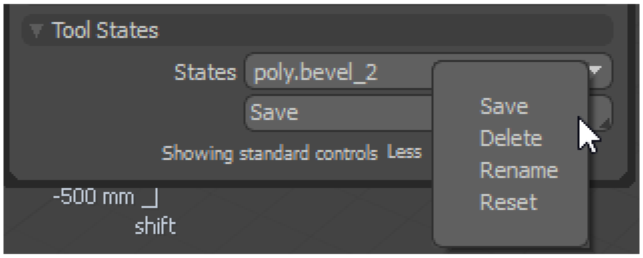
1. Activate the tool.
2. Open the tool properties. In the **Model** layout, they can be found in the lower left panel. In the **Modo** layout, click the **Tool Properties** button on the left panel, or press **K**.
3. Under **Tool States**, expand the **States** list, and select your preset.


Profiles
Tool States
States
poly.bevel_2
Save
Showing standard controls
Less
More
(none)
poly.bevel_inset
✓ poly.bevel_2

The settings of the selected preset are applied to the geometry.

Deleting, Renaming, and Resetting Tool States

With a Tool State preset selected in the **Tool Properties**, right-clicking the **Save** button gives you additional options.


Tool States
States
poly.bevel_2
Save
Showing standard controls
Less
-500 mm
shift
Save
Delete
Rename
Reset

Here you can do the following:

- **Delete** - Remove the preset.
- **Rename** - Edit the name of the preset. Clicking **Rename** opens a dialog where you can edit the preset name. Click **OK** to save the changes.
- **Reset** - Return the tool to its original state.

Shader Tree

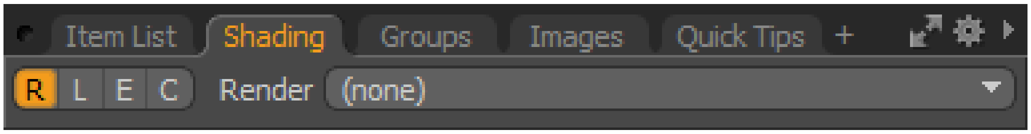
The **Shader Tree** is the main interface for adding materials and textures to surfaces in your scene. Modo uses a layered approach to texturing, allowing a simple to use, yet very powerful, method for assigning textures exactly where you want them, and creating complex looks for surfaces with ease. The visual impact on your scene is evaluated bottom to top. An image layer with 100% opacity that is placed above a procedural checker pattern obscures the checkers in the rendered image. Changing the image layer's transparency gradually reveals the checkered layer below, as it becomes less opaque. For specific information on working with **Shader Tree** layers, see Shader Tree.

The viewport itself, in the default configuration, appears in the upper-right viewport group. Click the **Shading** tab to make the viewport visible. Whenever a layer is selected within the viewport, its associated settings appear in the **Properties** viewport, found directly below. If the properties are not visible, one may need to also click the tab to reveal the viewport.

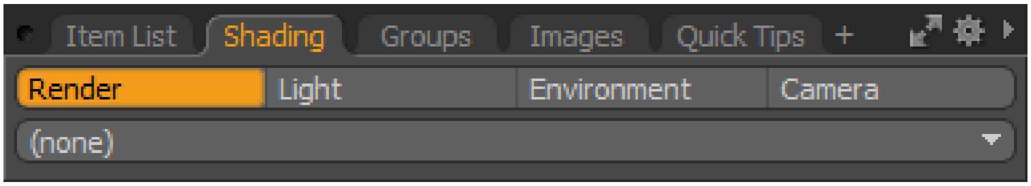
Shader Tree UI

The viewport has three UI modes: compact, small, and full. Depending on your amount of screen real estate, you can adjust the viewport by dragging any edge on it. The viewport dynamically scales, giving you the appropriate set of options based on the viewport's size.

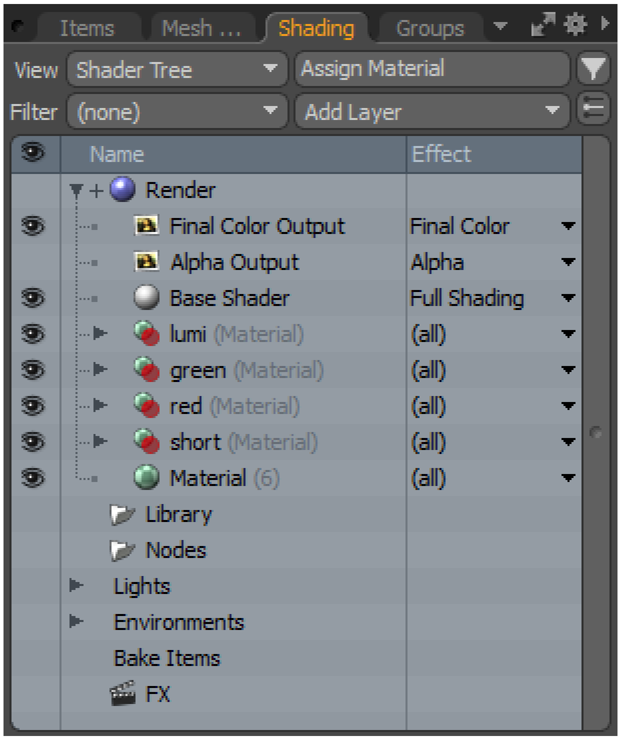
In compact mode, the first four buttons **R**, **L**, **E**, and **C** represent the render item, the lights, the environment, and the camera. The bar on the right, when clicked, opens a contextual menu, allowing you to choose individual layers specific to each **R**, **L**, **E**, and **C** setting. Additional layers can be added using the commands **Texture** > **Add Shader Tree Layer** > <layer type>.



In small mode, the **Render**, **Light**, **Environment** and **Camera** buttons become full sized and the layer selector appears below.



In full mode, each item is represented as a layer with a corresponding icon. Clicking on any item selects it, allowing you to edit the various properties. Multiple layers can be selected using the **Shift** and **Ctrl** key modifiers. Layers can also be dragged and dropped within the viewport, changing the layer order, which affects how layers feed into the render outputs. They can also be dragged into material groups to better organize scenes with many surfaces.



Name	Effect
Render	
Final Color Output	Final Color
Alpha Output	Alpha
Base Shader	Full Shading
lumi (Material)	(all)
green (Material)	(all)
red (Material)	(all)
short (Material)	(all)
Material (6)	(all)
Library	
Nodes	
Lights	
Environments	
Bake Items	
FX	

Additional controls appear above the viewport as well. The **View** option changes how the layers of the **Shader Tree** are displayed, with the **Materials** option offering a simplified look at the **Shader Tree**'s layers. The **Filter** option automatically shows/hides various type of layers. The **Assign Materials** button opens the Polygon Set Material/Part/Smoothing dialog and tags any selected polygons. The **Add Layer** button allows you to add additional layers to the **Shader Tree**. The Find ▽ button offers a search function that uses standard pattern matching to locate specific layers (detailed in Search Function Pattern Matching). The Show Only Selected Items ☰ button button toggles the visibility of shading layers, based on item selection in the **Item List**.

Within the **Shader Tree** window the 👁 icon column allows you to enable or disable the effect of a specific layer without removing it. Simply click the 👁 icon to toggle or drag over multiple eyes to toggle them simultaneously. The **Effect** column allows you to specify in what way the layer contributes to the rendered image. Right-click on the effect name to open a context menu with other additional options. Any layer name, once selected, can be right-clicked to provide a context menu with several options:

- **Rename** - allows you to assign a new name to the specified layer.
- **Editor Color** - 12 color options are provided for colorizing the **Shader Tree** layer itself. This is meant to be used as a visual organizational device. Choosing any of the colors from the supplied menu applies the selected color as the layer background.
- **Properties** - opens the **Properties** viewport with the selected layer's corresponding attribute settings.
- **Select Polygons** - this command makes all the polygons in the scene related to the current mask selected, switching the Modo interface into **Polygons** selection mode, if not already active.

- **Duplicate** - this command creates a copy of a layer with identical settings. Changes in the original item are not reflected in the duplicate layer, nor are modifications to the duplicate reflected in its original layer.
- **Create Instance** - creates a clone of a layer that references the original source. Changes to one layer automatically propagate to all instances. Instanced layers are denoted by their italicized name in the **Shader Tree**. This is especially useful when separate surfaces with identical properties reside within different masks. When instances are used, changing one item's attributes automatically clones the changes on the instance. Changes to the instance itself act as local overrides, modifying the attribute directly, while changes are not applied back to the source of the instance. Further changes to the source are propagated, except to those changes made individually on each instance on a channel-by-channel basis.
- **Lock/Unlock Preset** - when the **Lock** command is applied to any Library material, this prevents any accidental changes from being applied to the material group. The target layer need to be **Unlocked** for further editing. The **Lock**/**Unlock** commands have no effect on standard **Shader Tree** layers.
- **Select Instances** - if a layer has any instances of it in the **Shader Tree**, the **Select Instances** command selects all the associated instances of the target automatically.
- **Select Source of Instance** - the **Select Source of Instance** command selects the parent source items of any instance. Instances are denoted by their italicized name in the **Shader Tree**.
- **Reference/De-Reference** (only available when right-clicking over a Material Group Item) - a reference acts similar to an instance but works to clone all layers within a Material Group Item, instead of just a single layer. Referencing a material group clones the entire group hierarchy as a single layer that can be applied elsewhere in the **Shader Tree**. To create a reference layer, right-click on the target Material Group Item (it must contain at least one layer) and select the command **Reference**. Click on the resulting reference item and drag-and-drop the layer into another **Shader Tree** location to see the results. Changes made to the source are propagated to the reference layer(s). Reference layers cannot be modified directly, unless the **De-Reference** command is applied, converting the reference back to standard **Shader Tree** layers. **De-Reference** also converts library references to standard layers. (References essentially operate like library items, only outside of the library folder instead.)
- **Copy/Paste/Delete** - copy, paste, or delete entire layers, as well as the settings of individual layers themselves. When pasting values from one layer to another, the layers must be of the same type, that is to say, a material item to a material item.
- **Preset Browsers Palette** - this command opens a Preset Browser palette.
- **Save Preset** - a preset stores settings so they can be applied at a later time. In the case of materials, a preset stores **Shader Tree** layers, with an associated icon for visual browsing, in the **Preset Browser**. A material preset can be as simple as a single layer, or it can contain any number of layers with complex blending. For multi-layer presets, these must all be contained under a single material group 🗀 item. Right-clicking on the target layer and selecting the **Save Preset** option from the contextual menu presents you with an OS dialog, prompting where to save the preset itself. If you want the resulting preset to show up automatically within the default **Preset Browser** structure, it's necessary to navigate to the directory where presets are stored and save in the appropriate folder. You can also create a new directory for saving presets, but make sure to use the **(add path)** command in the **Preset Browser** to make it visible within the browser viewport. Once the File is saved, a default icon is generated that represents the settings of the

preset. If you prefer to produce your own custom icon, this can be generated ahead of time and assigned with the **Save Preset with Thumbnail** command.

- **Save Preset with Thumbnail** - the **Save Preset with Thumbnail** command works in the same way as the **Save Preset** command above, but once the preset is saved, a second dialog opens allowing you to specify any saved image that is then used as the browser icon.
- **Change Type** - this command offers the ability to convert a layer of one type, such as an image map, to another type, such as a procedural. Associated texture locators retain their settings, if applicable, but layers themselves lose their settings.
- **Bake to Texture** - this command offers the ability to generate a texture map image of a layer. Baking textures from inside the **Shader Tree** is useful for baking a single effect, such as Color or Diffuse, into an image, whereas the **Bake** command from the **Render** menu bakes all light and texture contributors into a final shaded image. It's reasonable to think of the **Shader Tree** level of baking as similar to flattening layers in Photoshop. Only layers of the same effect setting underneath the texture layer that is triggering the bake contribute to the baked layer. Once baked, this texture can remain in the **Shader Tree** as a pre-computed result of all contributors, which results in a savings of render time, and provides a real-time view of the effect. The **Bake to Texture** command requires a UV map on the baked surface, as well as a blank image map to bake into. For more on baking textures, see Baking.
- **Bake to File** - the **Bake to File** command is functionally identical to the **Bake to Texture** command above, but writes the resulting texture as a file to disc, with the location specified by you. When applying the command, an OS file requester opens to specify the file name, locations, and format. The second dialog is where the resolution and bit depth are specified. When saving to a 16-bit or greater format, the **Floating Point** option must be enabled.
- **Bake From Object to Texture** - this command offers the ability to use the UV map of a low-polygon count model to bake a texture map from a higher-polygon count model. Maps can be used for any of the various effects, including diffuse color, displacement, and normal. Like **Bake to Texture**, a UV map is required as well as a blank image map to bake into. Both surfaces must have different material names and cannot share the same UV map. When calculating a displacement map, the **distance setting** from the requester dialog should be used as the **Displacement Distance** setting in the material dialog box when applied. For more on baking textures, see Baking.
- **Reload Image** - this command reloads the image as saved to disk. This is useful when edits made in an external application don't trigger Modo to reload it automatically, or when you wish to undo a series of internal edits all at once.
- **Replace Image** - this command opens an OS file requester dialog allowing you to select an alternative bitmap image that replaces the selected image, while retaining its existing settings.
- **Open Image Folder** - this command opens the saved files' folder location on the internal hard disk using the systems default file manager.

Tip: For a full overview of working with the layers of the **Shader Tree**, see Shader Tree.

Search Function Pattern Matching

Clicking the Find button when in the **Shading** viewport, opens the search function for finding specific layers. below are the following

Syntax	**Description**	**Example**		
		Inputs	**Pattern**	**Matches**
*	Searches for any characters that might occur in place of the asterisk.	Apple, Apricot, Grape	Ap*	Apple, Apricot
#	Searches for any characters that might occur in place of the pound sign and, if there are none, continues searching for the next sub-pattern listed.	Ae, Ape, Appe, App, Axe	A#pe	Ae, Ape, Appe
?	Searches for any single character. For example, *a?* matches any two-characer strings starting with *a*. *#?* means "any number of any characters", and is functionally identical to *.	Axe, Ape, Asp	A?e	Axe, Ape
~	Negates the expression that follows its use. For example, #~x? matches any two letter string except those starting with *x*. Similarly, *~(x?)* matches anything except a two letter string beginning with x (so, *xxa* or *ab* but not *xa*).	Axe, Ape, Asp	A~xe	Ape, Asp
| (vertical bar)	Searches for any one of the items separated by the vertical bar character. Any number of vertical bars can be used.	Axe, Ape, Abe	A(x|b)e	Axe, Abe
[abc]	Searches for any one of the characters inside the square brackets.	Axe, Ace, Abe, Abc, ACE	A[bcBC]e	Ace, Abe, ACE
[a-z]	Searches for any of the range of characters within the square brackets. Multiple ranges cab be combined, such as *[a-dA-Dxyz]*.	Axe, Ace, Abe, Ate, ABE	A[a-kA-K]e	Ace, ABE
' (apostrophe)	Escape from the pattern matching search function.	A[1], A[2], A(3)	A'[?']	A[1], A[2]

Vertex Map List

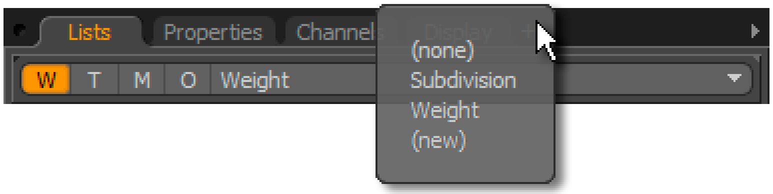
The **Vertex Map List** view displays Vertex Maps for the current mesh in groups: UV, Weight, Morph, and Other. The **Other Map** list includes the type of the Vertex Map as well as the name. The list of maps available is the union of the named maps in all the active (selected) meshes.

From within the **Vertex Map List** viewport, you can select a new map, regardless of the viewport size. If the viewport is compressed, such that you are presented with a single or multiple pop-ups, you may find the **(new map)** option inside the pop-up. If you accidentally choose **(new map)** from the wrong map type pop-up, you can simply change the map type on the new map requester that appears when creating a new map. When the viewport is expanded to its full size, expanding any of the Vertex Map types in the list present you with the **(new map)** option. Again, selecting this option in the list opens the **Create New Vertex Map** requestor.

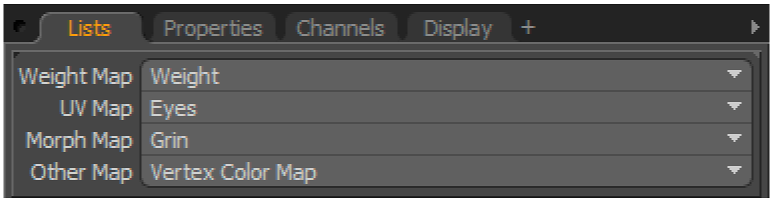
When working in the Vertex Map viewport, the current selection is maintained, even when selecting new mesh layers if each layer contains a Vertex Map with the same name. For instance, if you have a mesh layer with a texture map named "SteelMap" selected in the Vertex Map list, changing layers to another mesh that also has a texture map with the same name automatically selects that Vertex Map. This is very handy when working on objects that have been broken up against multiple layers.

When the **Vertex Map List** viewport's size is changed, it automatically changes its appearance so that the viewport is more-appropriately displayed for the space available. The viewport has three size settings: compact, medium, and expanded.

- **Compact** - the viewport is minimally displayed as a list of four buttons **W**, **T**, **M**, and **O** (weights, textures, morphs, and other) next to a pop-up list. The four buttons allows you to quickly set a filter for the pop-up list. If **W** is selected, the pop-up list displays all Weight Vertex Maps in the current mesh layer. To see the list of Texture maps, you would click on the **T** button, and so forth. The Other **Vertex Map List** contains color Vertex Maps, as well as Pick maps or custom Vertex Maps, added by 3rd party extensions or foreign data formats.

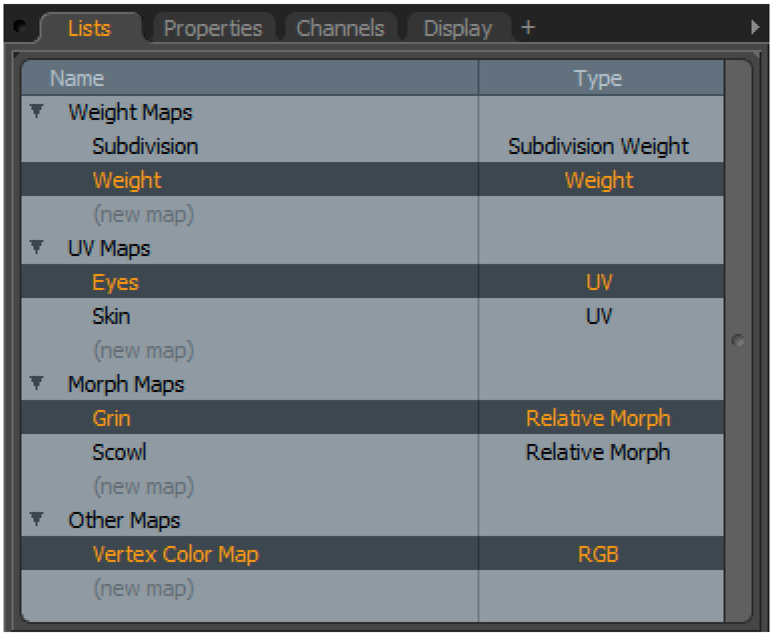


- **Medium** - the viewport is displayed as four pop-ups, stacked vertically. The type of Vertex Map is clearly labeled next to each pop-up.



Weight Map: Weight
UV Map: Eyes
Morph Map: Grin
Other Map: Vertex Color Map

- **Expanded** - the viewport is displayed as a tree with four categories. Expand/contract widgets are present on the left side of the text. In this view, it's possible to view and select multiple Vertex Maps from the four Vertex Map groupings.



Name	Type
▼ Weight Maps	
Subdivision	Subdivision Weight
Weight	Weight
(new map)	
▼ UV Maps	
Eyes	UV
Skin	UV
(new map)	
▼ Morph Maps	
Grin	Relative Morph
Scowl	Relative Morph
(new map)	
▼ Other Maps	
Vertex Color Map	RGB
(new map)	

Utility Viewports

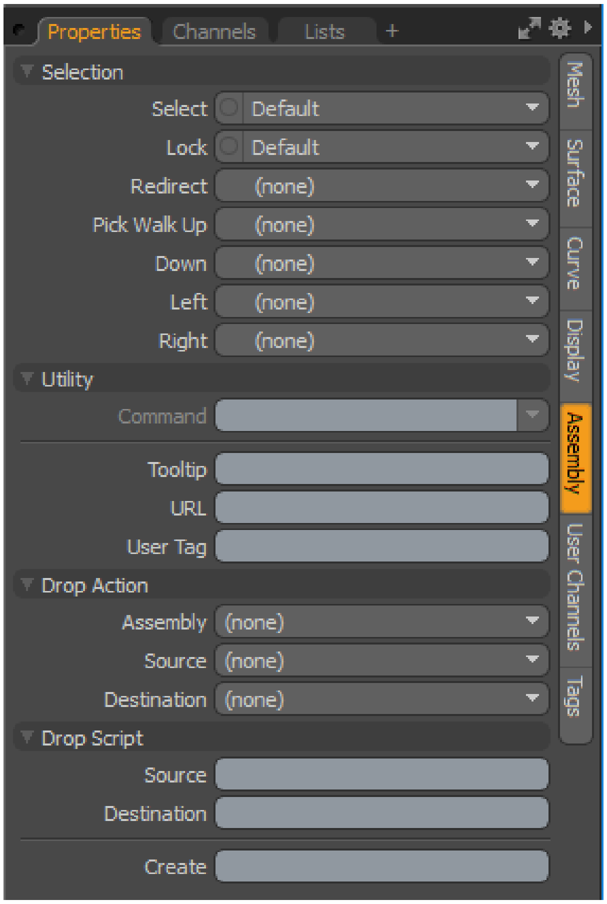
Utility viewports serve a number of purposes in Modo, whether it's providing a way for you to define and select color schemes in the **Color Picker**, editing the color values of a gradient in the **Gradient Editor**, or simply displaying a rendered image when pressing the **F9** key. These viewports are found throughout Modo, depending on the particular interface and utility you are looking at.

List and Info Viewports	
Assembly Viewport	The **Assembly** viewport provides functionality for selection, locking, and 'pick walking' up and down a hierarchy with the keyboard arrow keys.

List and Info Viewports	
Color Picker Viewport	The **Color Picker** viewport is a dedicated window for editing colors and creating schemes for scenes, which can then be easily applied when surfacing, setting up lights, and any other color-related operation.
Display Viewport	The **Display** viewport offers the ability to control how individual items appear within a 3D viewport, overriding the viewport style itself.
Gradient Editor	The **Gradient Editor** viewport offer the ability to control gradient textures in Modo.
Graph Editor	The **Graph Editor** viewport controls how the system interpolates key values.
Groups Viewport	The **Groups** viewport is a workflow streamlining tool, allowing the ability to easily control multiple items and/or channels within a scene. It also provides a number of functionalities to speed up common tasks, such as setting keyframes for complex hierarchies.
Preset Browser	The **Preset Browser** viewport is a visual reference to a library of pre-defined objects, surfaces, and settings, which are included with Modo.
Preview Viewport	The **Preview** viewport uses the Modo render engine in a progressive refinement mode that allows you to see all changes in a scene with a fully rendered image.
Render Display	The **Render Display** viewport provides a variety of useful options and feedback for working with the resulting rendered pixels for any Modo scene.
Tool Pipe	The **Tool Pipe** viewport allows the combination of the tools and tool modifiers, activating the proper functions automatically.

Assembly Viewport

The **Assembly** viewport provides functionality for selection, locking, and 'pick walking' up and down a hierarchy with the keyboard arrow keys. These options are essential to streamlined animation workflows. The **Utility** section in the viewport is related to items once saved as an assembly preset and defines how the item reacts to being placed from the **Preset Browser** into a given scene.



Selection

Select Default

Lock Default

Redirect (none)

Pick Walk Up (none)

Down (none)

Left (none)

Right (none)

Utility

Command

Tooltip

URL

User Tag

Drop Action

Assembly (none)

Source (none)

Destination (none)

Drop Script

Source

Destination

Create

Selection

- **Select** - controls whether or not items are selectable within any 3D viewport. Selecting **Yes** enables direct item selection, while **No** disables it. Items designated **No** can still be selected in the Items (Scenes) List.
- **Lock** - when lock is enabled by selecting **Yes**, items are locked from the application for any type of item transform. The item's **Shape**, **Link**, **Visible**, **Pivot**, **Center**, and **Size** channels also become locked.
- **Redirect** - when an item is selected in a 3D viewport, specifying an item for redirect triggers Modo to automatically select the alternate item instead. Helpful in easing item selections in complex hierarchies, locators with special Display Viewport properties can be used to visualize selectable items in the viewport that redirect to the actual item you wish to modify.
- **Pick Walk** - the arrow keypad can be used as a way to "walk" through selections in a hierarchy. By default, parent items are selected by pressing the up arrow, and child items are selected by pressing the down arrow. The **Pick Walk** function allows you to specify which items are selected by any of the four arrow keys. For instance, when you reach the bottom of a hierarchy, the foot, you could specify the opposing foot item by pressing the left or right arrows, and by pressing down you could select the head item and so on.

Utility

- **Command** - the **Command** option specifies a command to perform when the item is clicked in the 3D view. This can be any of Modo's commands. There are two commands that are especially useful when setting up assemblies:
    - **item.channelHaul**- this selects any user channels on the item and activates the **Channel Haul** tool.
    - **item.channelPopover**- this displays a pop-over form containing controls for any user channels on the item. If used in conjunction with a selection direction, this can also be used to display the channels belonging to a specific group.

> **Tip:** The **Command** function is disabled on Mesh Items. To enable the option, you can convert a mesh to a static mesh, by right-clicking on the item's layer name in the **Item List**, and selecting the **Change Type** > **Static Mesh** option from the context menu. Note that converting a mesh to a static mesh is a destructive action, so make sure you have a backup copy, should further editing be necessary to the Mesh Item.

- **Tooltip** - tooltips are the small blocks of text that appear, if assigned, when the pointer momentarily stays over an item. These can be notes or reminders to what an item is intended to do in a scene but tooltips can also be useful for an assembly author to provide some contextual instructions on how to work with an item in an assembly.
- **URL** - if a custom URL is assigned to an item, using the **F1** help feature then clicking the item in the 3D viewport opens that particular location. This is useful if you want to add custom documentation to an item.
- **User Tag** - this is a text string that can be used to identify particular items within an assembly or scene. These tags can be read by scripts with the **item.userTag** command. An example of use might be to identify items to be deleted or hidden when an assembly script has finished.

Drop Action

Assemblies offer you the ability to create a rig using Modo's many animation modifiers, then saving the setup as a Preset for easy application elsewhere. In the following descriptions, the item being dragged is referred to as the **source** item and the item that has received the drop is referred to as the **destination** item.

- **Assembly** - you can choose from a series of actions that are performed when the preset is initially dropped into the scene from the **Preset Browser**. The possible actions are as follows:
    - **Parent** - the source item is made a child of the destination item.
    - **Parent in Place** - the source item is parented, as above, but with **Compensation** applied.
    - **Match** - the source item is modified to match position, scale, and rotation of the destination.
    - **Match Position** - the source item is modified to match the position only of the destination item.
    - **Match Rotation** - the dropped item is modified to match the rotation only of the destination item.

  - **Match Scale** - the dropped item is modified to match the scale only of the destination item.
  - **Insert** - inserts the source item into the hierarchy of the destination item. The item is positioned and orientated to match the destination item and becomes the parent of the destination item.
  - **Insert at Parent** - similar to **Insert** but the source item matches its position and orientation to the destination item's parent (if a parent item is present).
  - **Insert in Place** - the source item is inserted into the destination item's hierarchy as its parent but retains its current position and orientation.
  - **Place** - positions the selected item at the intersection of the mouse pointer and the surface.
  - **Place and Align** - positions the selected item at the intersection of the mouse pointer and surface, rotating the item to match the surfaces normal direction.
- **Source** - the **Source** drop action is performed when the assembly item is dropped onto a scene item. It has the same action options as **Assembly**.
- **Destination** - as above, but the action is performed when a scene item is dropped onto the assembly item. It has the same action options as **Assembly**.

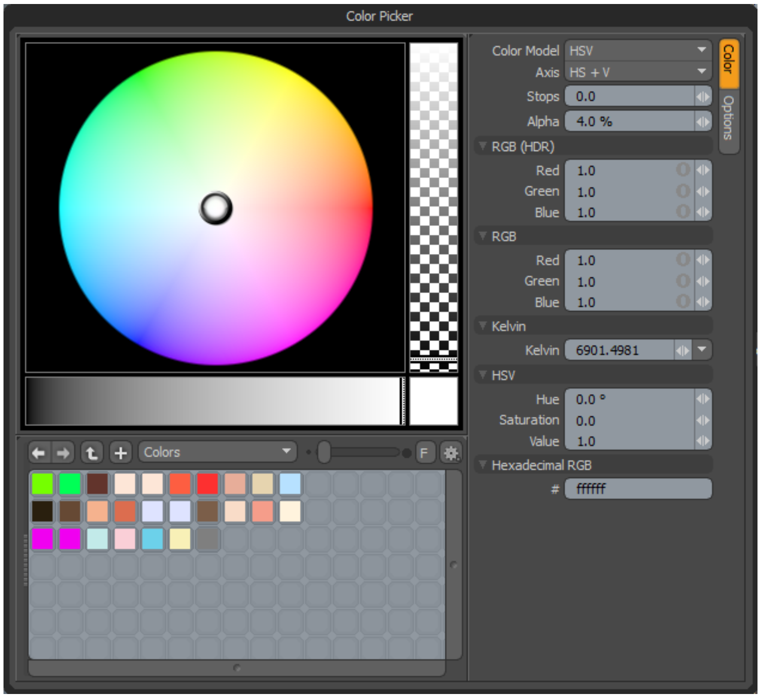
Drop Script

Additional to the above actions, scripts can be assigned to run when certain actions occur:

- **Source** - the assigned script (omitting the **@**) runs when this item is dropped onto another in the scene. The script gets the source item and destination item passed as arguments. If multiple items are dropped, then the script is called for each item in turn.
- **Destination** - as above, but the script runs when any item in the scene is dropped onto the item it's assigned to. As with the **Source** drop script, the source and destination items are passed to the script as arguments.

Color Picker Viewport

Color is an integral function when painting and surfacing for 3D. Color sets the mood, leads a viewer's eye, and provides a sense of cohesion to a scene. Modo provides a dedicated window for editing colors and creating schemes for scenes, which can then be easily applied when surfacing, setting up lights, and any other color-related operation. The **Color Picker** opens whenever you click on a color swatch in Modo. Being non-modal, the pop-up disappears as soon as you move your mouse cursor away from it.



Color Model HSV
Axis HS + V
Stops 0.0
Alpha 4.0 %

RGB (HDR)
Red 1.0
Green 1.0
Blue 1.0

RGB
Red 1.0
Green 1.0
Blue 1.0

Kelvin
Kelvin 6901.4981

HSV
Hue 0.0 °
Saturation 0.0
Value 1.0

Hexadecimal RGB
# ffffff

The viewport is dominated by the **Color Picker** with its color wheel on the left and the color options on the right, joined by a **Preset Browser** across the bottom. Combined with its many options, the color wheel is a full-featured color scheme generator, with options for creating a variety of automatically-coordinated schemes based on various rules that you can adjust.

The wheel itself is straight-forward. You only need to click the color you wish to choose, or click and drag to update the value in real-time. Releasing the mouse pointer when you've chosen a color, it's selected and the values in the panel to the right are filled in. Depending on the color model, the color strip below the wheel updates accordingly. You can click and drag in that strip to change it as well.

The right side of the **Color Picker** offers you the ability to define alpha (transparency) values, with the box in the lower-right corner displaying the current color swatch. Right-click the color swatch to save the color as a preset. Automatic preset saving can also be accomplished by dragging and dropping the swatch into the preset window.

For aspects of Modo that require two colors, such as when working with procedural textures, you can simultaneously define both colors by clicking the arrow to swap foreground/background colors. Once you define the colors you can click the arrows to swap back and forth. The **Properties** panel on the far-right side controls the display of the picker and allows you to define numerical values, as well as to set the scheme rules and save presets to the browser below.

Some of the options in the **Color** tab include:

- **Color Model** - this option allows you to choose the color model that suits your needs best. There are three current options:

  - **HSV**- standing for **Hue**, **Saturation**, and **Value**, it's considered the most intuitive color model. You can adjust colors by their **Hue**, the apparent color itself, **Saturation**, the intensity of the color, and **Value**, how light or dark the color is.
  - **Kelvin** - the **Kelvin** scale is based on thermal dynamics, and considers colors as temperatures. This single axis scale is useful in describing the color of light sources, as the hotter they burn, the brighter, and more white they become. The scale ranges from warm colors - reds and oranges - through yellow to white, extending to a slight blue as it increases to the maximum value.
  - **RGB** - standing for **Red**, **Green**, and **Blue**, the model itself is based on the additive property of light. If pure versions of each color are combined, they produce white light. While it is a common model, it's not very intuitive, as you might not have a good idea about what happens if you want to reduce **Red** by 50%. Still, it is the most common format for colors when exchanging data between applications.
- **Axis** - since the **Color Picker** is two dimensional and the various color spaces are multi-dimensional, the additional axis options modify how you interact with the picker, providing additional options for customizing the picker and color strip to suit your needs.
- **Stops** - this option is related to f-stops (familiar to photographers). It's intended to aid in previewing and selecting HDR (High Dynamic Range) colors, whose brightness goes beyond what typical computer monitors are capable of displaying. At the default 1.0 value, the **Color Picker** works like any other, maxing out at pure white. Changing the stops value to 1.4 allows you to define colors whose brightness is twice that of the 1.0 setting. A stop of 2.0 would again be twice as bright (or four times brighter than 1.0).

  These values work the same as those in photography, where each additional stop value lets in twice as much light as the previous. The standard photographic progression of stops is 1, 1.4, 2, 2.8, 4, 5.8, 8, 11, 16, and 22. It should be noted that the **Stops** value only changes how Modo displays the colors in the picker, allowing you the ability to choose over-bright values in the color wheel. It doesn't change the color value itself. Additionally, the stops value affects those colors selected using the OS-specific color pickers.
- **Alpha** - Use this to adjust color amounts on standard materials, for example, diffuse or specular color. You can also modify color amounts with this while painting.

Additionally, there are further options available under Models.

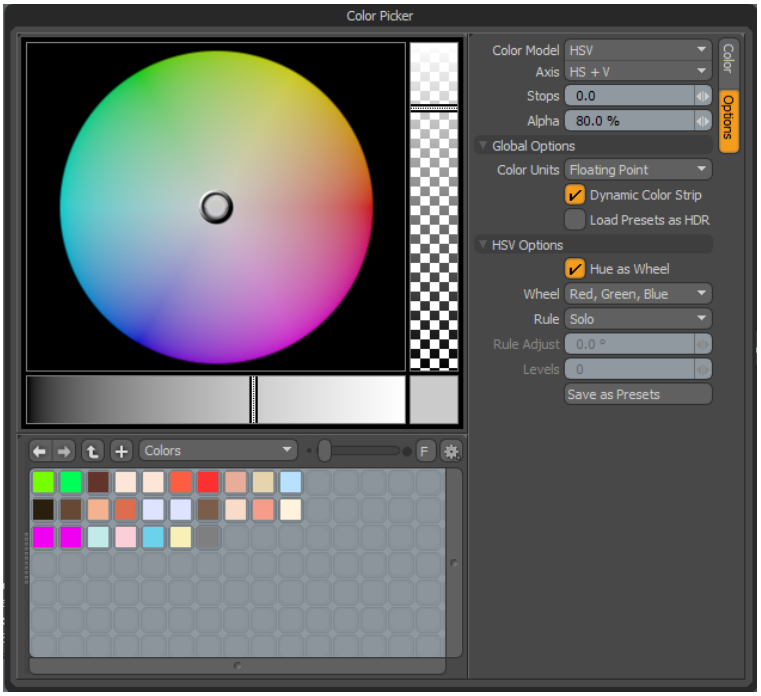
Models

The five following sections, **RGB(HDR)**, **RGB**, **Kelvin**, **HSV** and **Hexadecimal RGB**, are simply numeric value input fields for displaying and entering colors, using alternate color models. Displayed values always represent the current foreground color and update accordingly as you interact with the picker. All models, except **RGB (HDR)**, are not capable of defining or displaying HDR colors, and clip at the maximum values for the selected model (showing whatever the equivalent color is). The maximum value is different for each of the various **Color Units** options, as defined by you.

Global Options

When you click on the **Options** tab, a few new options are added to the panel.

- **Color Units** - changes the global preference for how color values are entered into the Modo color fields. There are four options for this control:
  - **Floating Point** - for this model, standard color values exist between 0 and 1, with HDR color values being those beyond 1.0.
  - **Percentage** - color values are defined as percentages, similar to **Floating Point** values, where up to 100% are normal color range, and values beyond 100% extend into the HDR colors.
  - **Integer** - this is the standard format for most image editing applications, where standard color values exist in the 0 to 255 range.
  - **Hexadecimal** - the standard for defining color on the web, as they are natural values for computers to understand. This is similar to binary, where the two values represent 255 different levels.
- **Dynamic Color Strip** - when this option is checked, the color strip updates dynamically, showing the color selection as a component of the bar itself. When unchecked, the bar simply shows its single color axis, as defined by the **Axis** option.
- **Load Presets as HDR** - when a value is saved, it's tagged as either a LDR (Low Dynamic Range) color or HDR (High Dynamic Range) color. Enabling the **Load Presets as HDR** allows you to only load HDR color presets from the browser, effectively hiding all the LDR colors from view (they are still there, only hidden). This is an easy way to determine if colors are HDR or LDR in a given scheme.



Color Picker

Color Model HSV
Axis HS + V
Stops 0.0
Alpha 80.0 %
Global Options
Color Units Floating Point
☑ Dynamic Color Strip
☐ Load Presets as HDR
HSV Options
☑ Hue as Wheel
Wheel Red, Green, Blue
Rule Solo
Rule Adjust 0.0 °
Levels 0

HSV Options

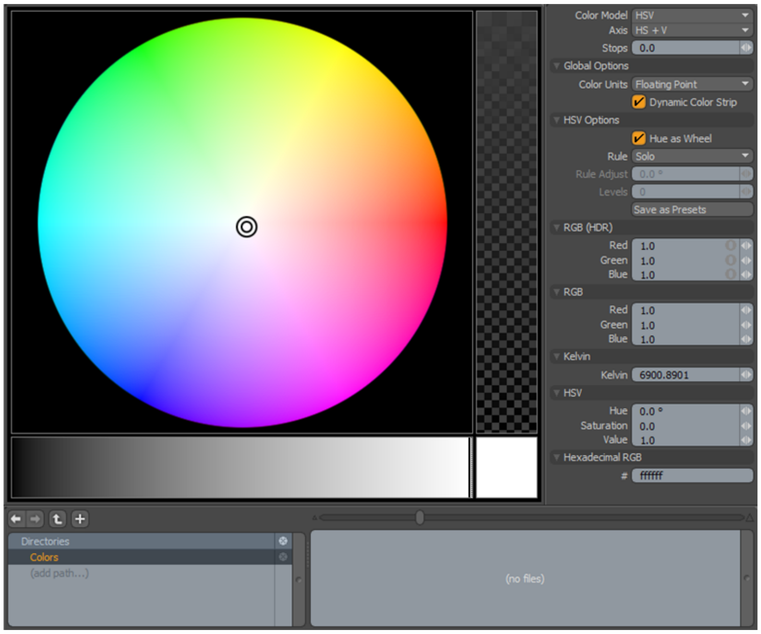
- **Hue as Wheel** - when the **Color Model Options** of **HSV** is selected, enabling the **Hue as Wheel** option displays the picker as a continuous wheel. When displayed as such, several options related to making rule-based color schemes becomes available.
- **Wheel** - this option defines the type of **HSV** wheel to display. You can choose the standard **Red, Green, Blue** (RGB) option common to many image editing applications, or the more familiar and artist-friendly **Red, Yellow, Blue** version. Unlike the RGB version, the RYB version works like the color wheel many artists learn in art school, where red and green, blue and orange, and yellow and purple oppose each other on the wheel.
- **Rule** - Modo allows multiple options for generating automatic schemes, based on common color rules:
  - **Solo** - the default mode, only a single color is defined.
  - **Complementary** - two colors whose positions oppose each other on the color wheel.
  - **Analogous** - three colors that contain a similar hue, appearing close to each other on the color wheel.
  - **Triadic** - a three-color scheme, where colors appear equidistant from each other on the color wheel.
  - **Tetradic** - a four-color scheme, where colors appear equidistant from each other on the color wheel (dual complementary).
  - **Compound** - a five-color scheme, with a main color and two split complementary colors.
  - **Tints** - appear as graduated value (brightness) variations of the foreground color towards white.
  - **Shades** - appear as graduated value (brightness) variations of the foreground color towards black.
- **Rule Adjust** - adjust the angle of degrees between scheme rules that modifies the scheme to suit your own taste.
- **Levels** - this option becomes active for the **Tints** and **Shades** rules, defining how many steps are in the scheme.
- **Save as Presets** - pressing this button saves the current foreground color into the **Presets** > **Colors** folder of the preset content. If the default content hasn't been installed, you need to manually define a directory and add it to the preset browser using the **Add Path...** option. When defining color schemes using the rule options, all the color defined in the swatch patch is added to the preset browser simultaneously.

Tip: You can access your system-level (OS) color picker at any time by clicking on the color swatch in the lower-right corner of the **Color Picker** itself. You should note that any colors selected in this way are also affected by the **Stops** value.

Color Theme Editor

Color is an integral function when painting and surfacing for 3D. Color sets the mood, leads the eye, and provides a sense of cohesion to a scene. Modo provides a dedicated window for editing colors and creating schemes for scenes, which can then be easily applied when surfacing, setting up lights, and any other color related operation. The window itself is really the standard color picker pop-up that appears when any color option is clicked, however, for the **Color Theme Editor**, it is persistent, instead of disappearing as soon as the cursor moves away from it, you can recall it from the menu bar, under **Texture** > **Open Color Theme Editor**.

The viewport is dominated by the color picker with its color wheel on the left, the color options on the right, joined by a preset browser across the bottom. Combined with its many options, the color wheel is a full-featured color scheme generator, with options for creating a variety of automatically coordinated schemes based of various rules you can adjust.



Color Model: HSV
Axis: HS + V
Stops: 0.0

Global Options
Color Units: Floating Point
☑ Dynamic Color Strip

HSV Options
☑ Hue as Wheel
Rule: Solo
Rule Adjust: 0.0 °
Levels: 0
Save as Presets

RGB (HDR)
Red: 1.0
Green: 1.0
Blue: 1.0

RGB
Red: 1.0
Green: 1.0
Blue: 1.0

Kelvin
Kelvin: 6900.8901

HSV
Hue: 0.0 °
Saturation: 0.0
Value: 1.0

Hexadecimal RGB
#: ffffff

The wheel itself is straightforward, you only need to click over the color you want to choose, while clicking and dragging, you update the value in real-time, to select the color, release the mouse pointer. Depending on the color model, the bottom color strip updates accordingly, but you can click and drag in that window as well. The right hand side of the color picker offers the ability to define alpha (transparency) value, with the box in the lower right hand corner displaying the current color. The properties panel on the far right side controls the display of the picker, allows you to define numerical values as well as set the scheme rules, and save presets to the browser below.

Color Theme Editor options:

Option	Description
Color Model	**The Color Model** option allows you to choose the color model that suits your needs best. There are three options: • **HSV** - Standing for Hue, Saturation and Value, it is considered the most intuitive color model. You can adjust colors by their Hue, the apparent color itself, saturation, which controls the intensity of the color, and value, which controls how light or dark the color is. This is also the only model that is displayed as the familiar color wheel. • **Kelvin** - The Kelvin scale is based on thermal dynamics, and considers colors as temperatures. This single axis scale is useful in describing the color of light sources, as the hotter they burn, the brighter, and more white they become. They go from warm colors, reds and oranges, through yellow to white, and then extend to a slight blue as it increases to the maximum value. • **RGB** - The Red, Green, and Blue (**RGB**) model, itself, is based on the additive property of light, and is the primary model for electronic displays and sensors (such as your monitor and digital camera). If pure versions of each color are combined, they produce white light. While it is a common model, its not very intuitive. For instance, it is hard to know how reducing the red color by 50% can impact the color before doing so. Still, it is the most common format for colors when exchanging data between applications.
Axis	Since the color picker is two dimensional, and the various color spaces are multi-dimensional, the additional axes options modify how you interact with the picker providing additional options for customizing the picker and color strip to suit your particular needs.
Stops	This option is related to the f-stops, familiar to photographers. It is intended to help in previewing and selecting HDR (High Dynamic Range) colors, whose brightness goes beyond what typical computer monitors are capable of displaying. At the default **1.0** value, the color picker works like any other, maxing out at pure white. Changing the stops value to **1.4** allows you to define colors whose brightness is twice that of the **1.0** setting. A stop of **2.0** is twice as bright (or four times brighter than **1.0**). These values work the same as those in photography, where each additional stop value lets in twice as much light as the previous. The standard photographic progression of stops is **1**, **1.4**, **2**, **2.8**, **4**, **5.8**, **8**, **11**, **16**, and **22**.

Note: The **Stops** value only changes how Modo displays the colors in the picker, allowing you to choose over bright values in the color wheel, and doesn't change the color value itself. Additionally, the stops value affects those colors selected using the OS specific color pickers.

Global options

Option	Description
Color Units	The color units option changes the global preference for how color values are entered into Modo color fields. There are four options: • **Floating Point** - For this model, standard color values exist between **0** and **1**, with HDR colors values being those beyond **1.0**. • **Percentage** - Color values are defined as percentages, similar to floating point values, where up to 100% would be normal color range, and values beyond 100% extend into the HDR colors. • **Integer** - This is the standard format for most image editing applications, where standard color values exist in the **0** to **255** range. • **Hexadecimal** - Hexadecimal colors are the standard for defining color on the web as they are natural values for computers to understand, similar to binary, where the two values represent 255 different levels.
Dynamic Color Strip	When this option is checked, the Color Strip bar updates dynamically showing the color selection as a component of the bar itself. When unchecked, the bar simply shows its single color axis, as defined by the **Axis** option.

HSV Options

Option	Description
Hue as Wheel	When the **Color Model** options of **HSV** is selected, enabling the **Hue as Wheel** option displays the picker as a continuous wheel. When **Hue as Wheel** option is enabled, several options related to making rule based color schemes become available.
Wheel	Defines the type of HSV wheel to display, you can choose the standard **Red Green**

Option	Description
	Blue (RGB) option common to many image editing applications, or the more familiar and artist friendly **Red Yellow Blue** version. Unlike the RGB version, the RYB version works like the color wheel many artists learn in art school, where red and green, blue and orange, and yellow and purple oppose each other on the wheel.
Rule	Modo allows multiple options for generating automatic schemes based on common color rules: • **Solo** - The default mode, only a single color is defined. • **Complementary** - Two colors whose positions oppose each other on the color wheel. • **Analogous** - Three colors that contain a similar hue, appearing close to each other on the color wheel. • **Triadic** - A three color scheme, where colors appear equidistant from each other on the color wheel. • **Tetradic** - A four color scheme, where colors appear equidistant from each other on the color wheel (dual complementary). • **Compound** - A five color scheme, with a main color and two split complementary colors. • **Tints** - Tints appear as graduated value (brightness) variations of the foreground color toward white. • **Shades** - Shades appear as graduated value (brightness) variations of the foreground color toward black.
Rule Adjust	You can adjust the angle of degrees between scheme rules that adjust the scheme to suit their own taste.
Levels	This option becomes active for the **Tints** and **Shades** rules, defining how many steps are in the scheme.
Save as Presets	Pressing this button saves the current foreground color into the **Presets** > **Colors** folder of the preset content. If the default content hasn't been installed, you need to manually define a directory and add it to the preset browser using the **Add Path** option. When defining color schemes using the rule options, all the color defined in the swatch patch are added to the preset browser simultaneously.

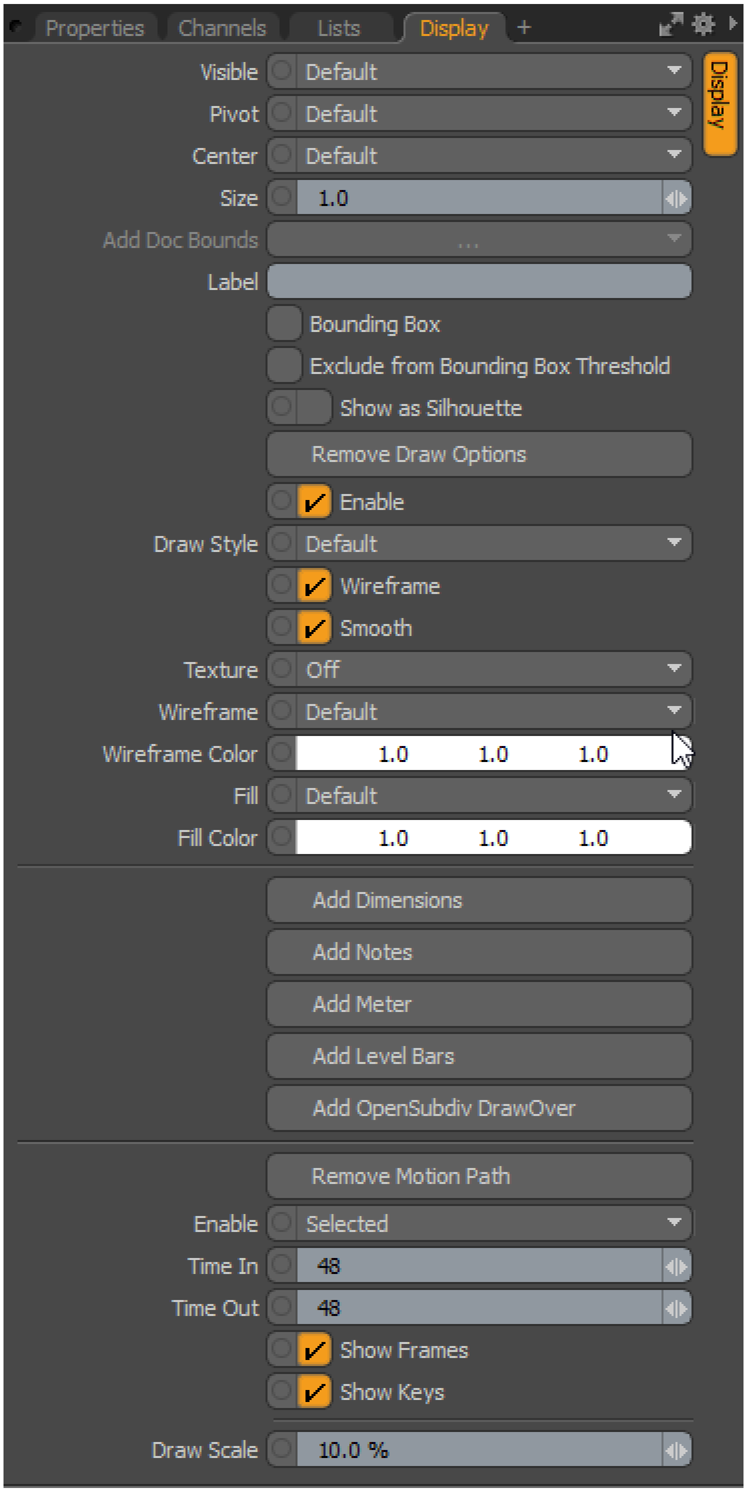
Models

RGB(HDR), **RGB**, **Kelvin**, **HSV** and **Hexadecimal RGB** are numeric value input fields for displaying and entering colors using alternate color models. Displayed values always represent the current foreground color and update accordingly as you interact with the picker. All models, except **RGB (HDR)** are not capable of defining or displaying HDR colors, and clip at the maximum values for the selected model (showing whatever the equivalent color is). The maximum value is different for each of the various **Color Units** options as defined by you.

Tip: You can access your system level (OS) color picker at any time by clicking on the color swatch in the lower right corner of the color picker itself. Any colors selected in this way are also affected by the **Stops** value.

Display Viewport

The **Display** viewport offers you the ability to control how individual items appear within a 3D viewport, overriding the viewport style itself. To modify any of the item display properties, simply select an item within a viewport, or the **Item List**, and open the **Display** viewport panel. Depending on the type of item layer selected, slightly different options appear.


Properties
Channels
Lists
Display
+
Display
Remove Draw Options
Add Dimensions
Add Notes
Add Meter
Add Level Bars
Add OpenSubdiv DrawOver
Remove Motion Path

Visible: Default
Pivot: Default
Center: Default
Size: 1.0
Add Doc Bounds: ...
Label:
☐ Bounding Box
☐ Exclude from Bounding Box Threshold
☐ Show as Silhouette
☑ Enable
Draw Style: Default
☑ Wireframe
☑ Smooth
Texture: Off
Wireframe: Default
Wireframe Color: 1.0 1.0 1.0
Fill: Default
Fill Color: 1.0 1.0 1.0
Enable: Selected
Time In: 48
Time Out: 48
☑ Show Frames
☑ Show Keys
Draw Scale: 10.0 %

3D Mesh Items Display Options

The following options are available when working with 3D mesh items:

- **Visible** - sets visibility for the item itself in the 3D viewport. **Default** respects viewport settings, **Yes** sets the item as always visible, and **No** disables visibility entirely.
- **Pivot** - sets visibility for the 3D item pivot. **Default** respects viewport settings, **Yes** sets the pivot as always visible, and **No** disables pivot visibility entirely.
- **Center** - sets visibility for the 3D item center. **Default** respects viewport settings, **Yes** sets the center as always visible, and **No** disables center visibility entirely.
- **Size** - a multiplier that adjusts the display size of the item's representations for camera, light, and locator icons. This does not affect the representations of mesh layers.
- **Label** - typing in text for the **Label** displays the text positioned next to the 3D item in the viewport for easy identification. It also adds two additional properties to control the display of the label in the viewport: **Show Label** and **Label Offset**.
- **Show Label** - this toggle temporarily enables/disables the display of the label in the viewport without losing settings. This option is only available when a label is specified.
- **Label Offset** - this value specifies the distance away from the locator where the label appears. This option is only available when a label is specified.
- **Bounding Box** - when enabled, Mesh Items display as a simple wireframe box around the geometry's total volume, instead of displaying the geometry itself.
- **Exclude from Bounding Box Threshold** - when enabled, the item is not affected by the **Preferences** > **Display** > **OpenGL** > **Bounding Box Threshold** option. For more information, see Display Preferences.
- **Show as Silhouette** - when enabled, items draw only as a solid shape, with no interior details. For the purposes of animation, it can be beneficial to view certain elements as a silhouette to ensure that key poses are expressing certain information. This works in conjunction with the **Enable Silhouette** setting (described in 3D (OpenGL) Viewport) of the viewport display options, which must be enabled (the default state) for the silhouette to be visible.

Tip: The **Show as Silhouette** display option is only applicable to the **Default** viewport display mode.

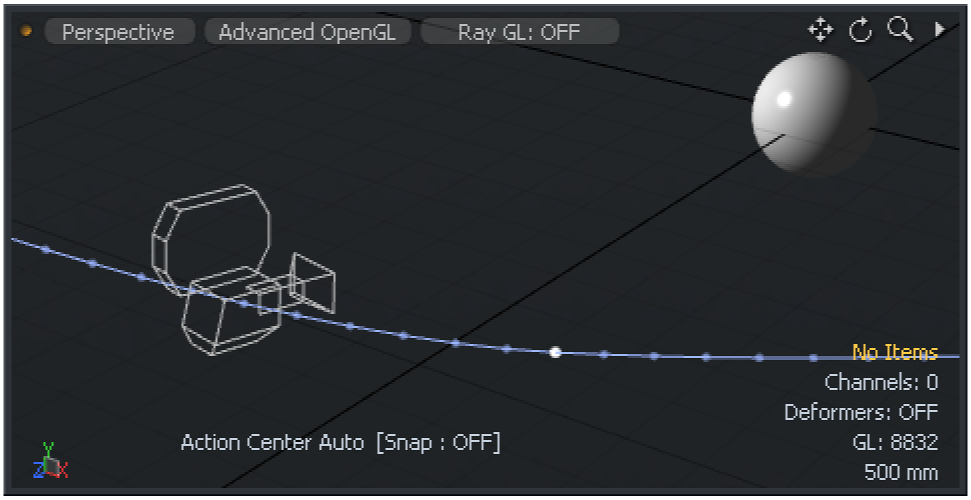
- **Add/Remove Draw Options** - the **Draw Options** open/remove additional properties for adjusting how items display in the 3D viewports, when selected additional properties appear.

Note: This option does not work in the **Default**, **Advanced**, **Gooch Tone Shading**, **Cel Shading**, and **Reflection** viewport modes.

- **Enable** - toggles the display of the **Draw Options** without losing values.
- **Draw Style** - allows you to choose a specific draw style for an individual Mesh Item. The **Default** respects viewport settings.

- **Wireframe** - disabling the **Wireframe** checkbox disables the wireframe overlay on 3D Mesh Items.
- **Smooth** - disabling the **Smooth** checkbox disables the smoothing applied to polygon surfaces, making polygons appear faceted.
- **Texture** - controls the display of texture layers on a Mesh Item. **Off** disables texture display, **Texture** draws the texture without shading, which can make it easier to see a texture for painting, and **Shaded Texture** takes into account the shading of the surface to which the texture is applied.
- **Wireframe** - adjusts the draw color of the wireframe overlay of the 3D item in the 3D viewport.
- **Wireframe Color** - when **Wireframe** is set to **Custom**, the **Wireframe Color** sets the color of the wireframe.
- **Fill** - if **Draw Style** is set to **Solid**, this value controls the color of the 3D item's fill.
- **Fill Color** - when **Fill Color** is set to **Custom**, this value specifies the RGB color of the fill of the **Solid** in the 3D viewport.
- **Add/Remove Dimensions** - when this option is enabled, dimension values for the overall bounding box size appear in the viewport around the Mesh Item. An additional sub-tab shows in the Properties Panel, allowing further display customizations.
- **Add/Remove Notes** - when this option is enabled, you can define multiple lines of text (up to nine) that show up when the item is selected. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Add/Remove Meter** - when this option is enabled, you can display an on-screen analog-style meter graph (this looks like a speedometer) that can be rigged to show individual values in the 3D viewport. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Add/Remove Level Bars** - when this option is enabled, you can display an on-screen equalizer-type graph for visualizing multiple on-screen values. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.

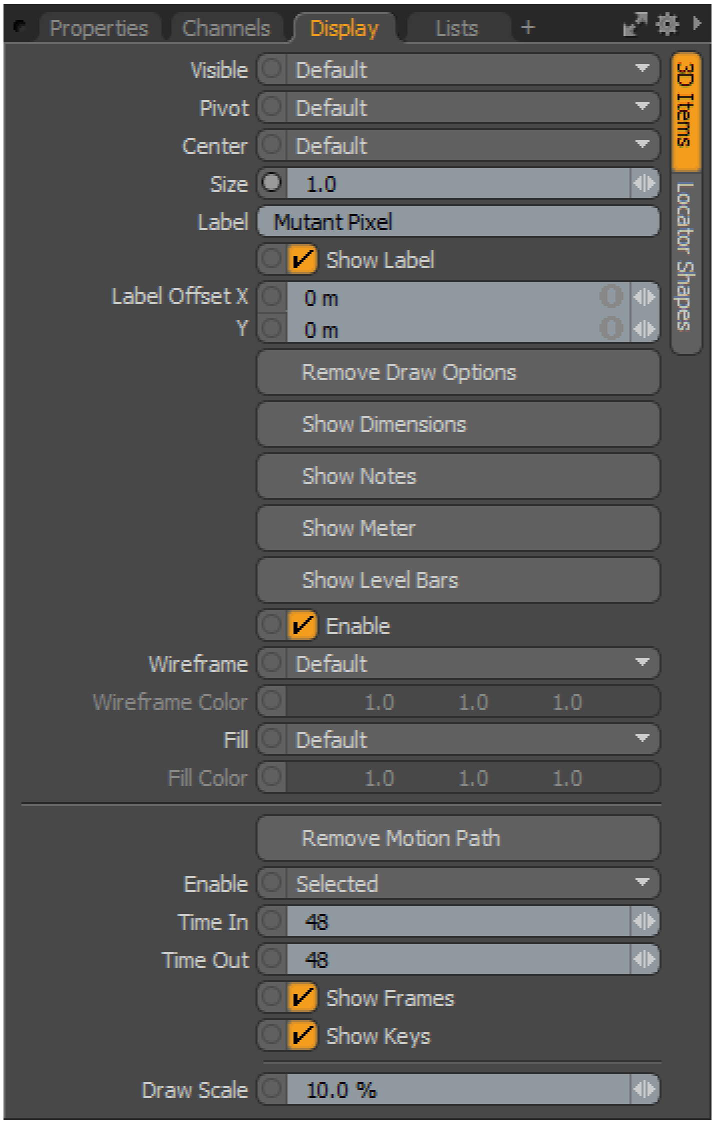
- **Add/Remove Motion Path** - displays the motion path for specific items within a scene (shown below for the camera item). You can select the **Add Motion Path** option, revealing additional properties controlling the display of the path in the 3D GL viewport.



- **Enable** - provides a means to toggle the display of the path **On** or **Off**, or to have the path only visible when the item is **Selected**.
- **Time In/Out** - **Time In** determines the number of frames before the current time in the Timeline that are displayed by the curve, while **Time Out** determines the number of frames after the current time that are displayed.
- **Show Frames** - when enabled, this option displays the position of the element at each frame as a small dot along the path. This can be useful to see how the element moves in time, but can also get busy for some types of motion and can, therefore, be disabled.
- **Show Keys** - when enabled, this option displays the position of the element at each keyframe as a white dot along the path.
- **Draw Scale** - determines the size of the resulting motion path in the 3D viewport.

Locator Display Values

Locators are instrumental in setting up controllers and modifiers for animation. Therefore, controllers have their own specific set of draw options, increasing their utility even further.


Properties
Channels
Display
Lists
+
3D Items
Locator Shapes
Remove Draw Options
Show Dimensions
Show Notes
Show Meter
Show Level Bars
Remove Motion Path

Visible: Default
Pivot: Default
Center: Default
Size: 1.0
Label: Mutant Pixel
☑ Show Label
Label Offset X: 0 m
Y: 0 m
☑ Enable
Wireframe: Default
Wireframe Color: 1.0 1.0 1.0
Fill: Default
Fill Color: 1.0 1.0 1.0
Enable: Selected
Time In: 48
Time Out: 48
☑ Show Frames
☑ Show Keys
Draw Scale: 10.0 %

The following options are available when working with controllers:

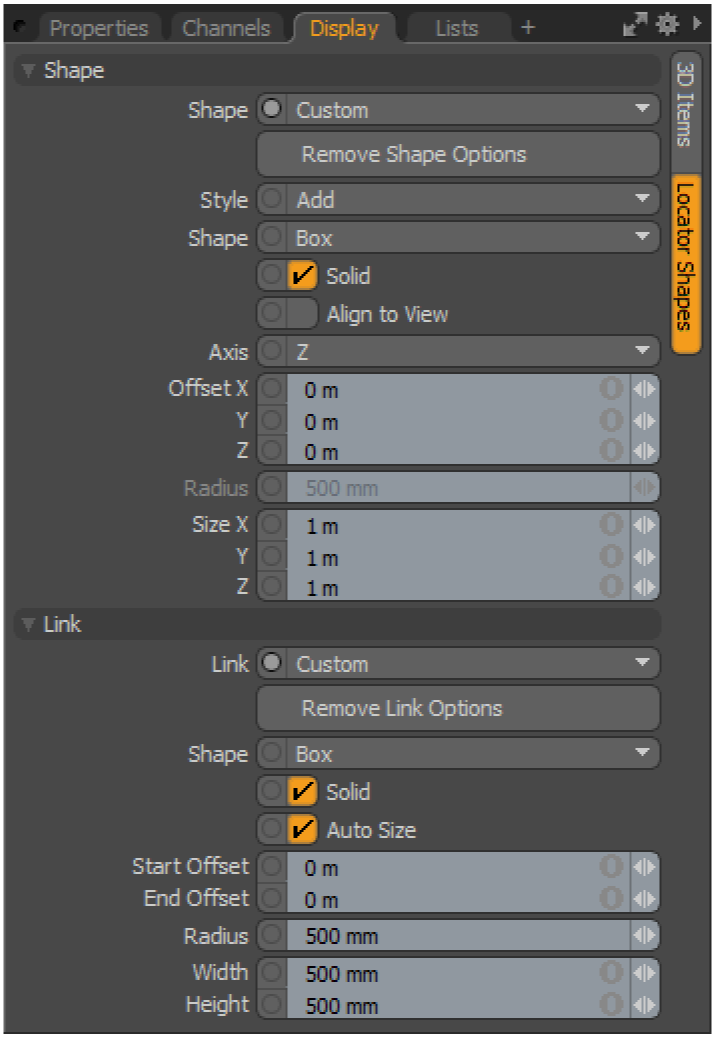
- **Visible** - sets visibility for the item itself in the 3D viewport. **Default** respects viewport settings, **Yes** sets the item as always visible, and **No** disables visibility entirely.
- **Pivot** - sets visibility for the 3D item pivot. **Default** respects viewport settings, **Yes** sets the pivot as always visible, and **No** disables pivot visibility entirely.
- **Center** - sets visibility for the 3D item center. **Default** respects viewport settings, **Yes** sets the center as always visible, and **No** disables center visibility entirely.
- **Size** - a multiplier that adjusts the display size of the locator.
- **Label** - typing in text for the **Label** displays the text positioned next to the locator in the viewport for easy identification. It also adds two additional properties to control the display of the label in the viewport: **Show Label** and **Label Offset**.
- **Show Label** - this toggle temporarily enables/disables the display of the label in the viewport without losing settings. This option is only available when a label is specified.

- **Label Offset** - this value specifies the distance away from the locator where the label appears. This option is only available when a label is specified.
- **Add/Remove Draw Options** - the **Draw Options** control the look of the locator in 3D Viewports, used mostly when **Custom Locator Shapes** is enabled.
- **Show/Hide Dimensions** - when this option is enabled, dimension values for the overall bounding box size appear in the viewport around the Mesh Item. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Show/Hide Notes** - when this option is enabled, you can define multiple lines of text (up to nine) that show up when the item is selected. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Show/Hide Meter** - when this option is enabled, you can display an on-screen analog-style meter graph (this looks like a speedometer) that can be rigged to show individual values in the 3D viewport. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Show/Hide Level Bars** - when this option is enabled, you can display an on-screen equalizer-type graph for visualizing multiple on-screen values. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Enable** - toggles the display of the **Draw Options** without losing values.
- **Wireframe** - adjusts the draw color of the locator in the 3D viewport.
- **Wireframe Color** - when **Wireframe** is set to **Custom**, the **Wireframe Color** sets the color of the wireframe part of the locator.
- **Fill** - if **Custom Draw** shape is set to **Solid**, this value controls the color of the fill itself.
- **Fill Color** - when **Fill Color** is set to **Custom**, this value specifies the RGB color of the fill of the **Solid** in the 3D viewport.
- **Add/Remove Motion Path** - displays the motion path for specific items within a scene. You can select the **Add Motion Path** option, revealing additional properties controlling the display of the path in the 3D GL viewport.
- **Enable** - provides a means to toggle the display of the path **On** or **Off**, or to have the path only visible when the item is **Selected**.
- **Time In/Out** - **Time In** determines the number of frames before the current time in the Timeline that are displayed by the curve, while **Time Out** determines the number of frames after the current time that are displayed.
- **Show Frames** - when enabled, this option displays the position of the element at each frame as a small dot along the path. This can be useful to see how the element moves in time, but can also get busy for some types of motion and can, therefore, be disabled.
- **Show Keys** - when enabled, this option displays the position of the element at each keyframe as a white dot along the path.

- **Draw Scale** - The **Draw Scale** option extrudes a ribbon shape outward from the motion path that represents the orientation angle of the animated element. The larger the scale value, the wider the ribbon representation in the viewport.

Locator Shapes

Locator shapes allow you to change the default draw style of the locators in the 3D viewport. Since locators have so many uses, it may be necessary to make each unique for easy identification and selection. Modo can automatically draw a non-rendering element between an item and its child (links always draw to child items in hierarchies) as well. Links are a useful in creating visual representations of item associations in complex hierarchies.



Shape

Shape: Custom

Style: Add

Shape: Box

☑ Solid

☐ Align to View

Axis: Z

Offset X: 0 m

Y: 0 m

Z: 0 m

Radius: 500 mm

Size X: 1 m

Y: 1 m

Z: 1 m

Link

Link: Custom

Shape: Box

☑ Solid

☑ Auto Size

Start Offset: 0 m

End Offset: 0 m

Radius: 500 mm

Width: 500 mm

Height: 500 mm

The following **Shape** options are applicable when working with locators:

- **Shape** - choose between the **Default** cross style, or choose **Custom** to enable additional properties for locator display customization.

- **Remove Shape Options** - this button removes the additional **Shape** options, reverting to the **Default** cross representation.
- **Style** - controls whether the **Shape** options draw in addition to the default value (**Add**) of if they replace its display (**Replace**).
- **Shape** - choose from multiple primitive shape options for the locator's representation.
- **Solid** - toggles between a filled or wireframe representation of the shape.
- **Align to View** - forces the representation to align with the viewport's camera.
- **Axis** - chooses the **Axis** for display of the shape representation.
- **Offset** - adjusts the offset values to position the locator representation.
- **Radius** - adjusts the size of circular locator representations.
- **Size** - adjusts the size of rectangular locator representations.

The following **Link** options are applicable when working with locators:

- **Link** - choose between the default value **None**, or a simple **Line**, or choose **Custom** to enable additional properties for link display customization.
- **Remove Link Options** - this button removes the additional **Link** options reverting to the default **None**.
- **Shape** - choose from multiple primitive shape options for the link representation.
- **Solid** - toggles between a filled or wireframe representation of the shape.
- **Auto Size** - dynamically scales the link shape, depending on the distance between the two items.
- **Start Offset** - offsets the start position of the link representation.
- **End Offset** - offsets the end position of the link representation.
- **Radius** - adjusts the size of circular link representations.
- **Width/Height** - adjusts the size of rectangular link representations.

Camera Display Options

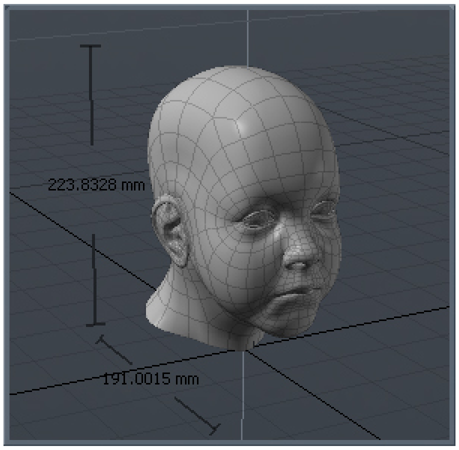
Cameras are a virtual lens into the 3D world inside Modo and everything rendered is seen from the point of view of a camera. The camera display options can be accessed from the **Display** viewport or the **Display** tab in the camera's **Properties** viewport.

For a full description of the camera **Display** parameters, see Camera Display Properties.

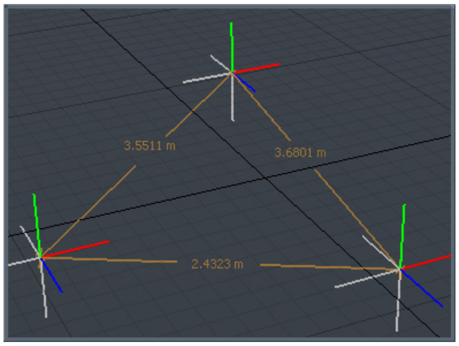
Additional Draw Options

For all items, there are additional GL drawing properties that allow you to customize the UI for rigging in a number of ways. Options are toggled on/off with associated buttons in the **Display** viewport when the target item is selected, with an associated sub-tab that appears in the **Properties** panel.

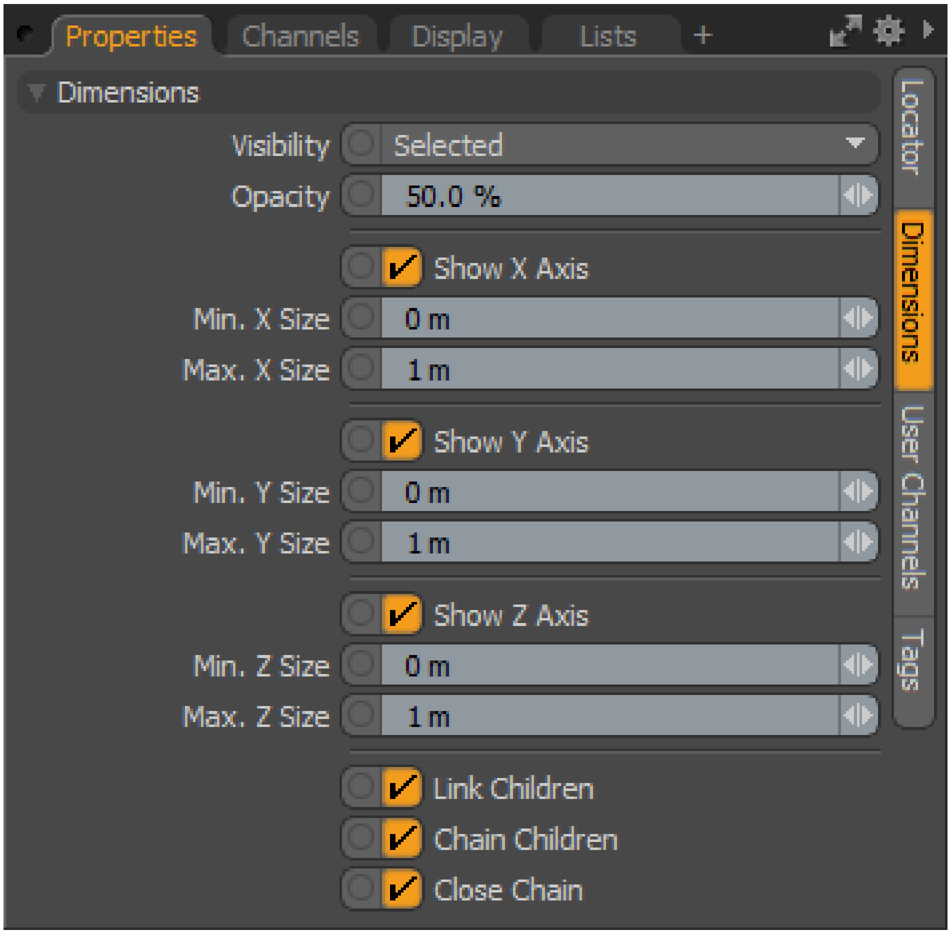
- **Show/Hide Dimensions** - when this option is enabled, an additional sub-tab appears in the **Properties** panel titled **Dimensions**. By default, dimension lines and values for the overall bounding box size appear in the viewport around a Mesh Item. The attributes in the **Properties** panel allows you to further customize the information that is displayed. Linking a series of items together through a hierarchy allows the display of distances between each object's **Center** point, from parent to child (and only when the parent item is selected do the dimension values display). Displayed dimensions are shown in the images below.
  - **Visibility** - determines the visibility of the dimension lines with **Off** being not visible, **On** being visible, and **Selected** being only visible when selected.
  - **Opacity** - controls the transparency of the dimension lines and numerical values displayed in the 3D viewport.
  - **Show X/Y/Z Axis** - with these channels, you can individually control the display for each axis.
  - **Min X/Y/Z Size/Max X/Y/Z Size** - the Min and Max values can be assigned for each axis and when the defined values for either **Min** or **Max** are reached, the dimension text displays in red in the viewport.
  - **Link Children** - when set to true (the default option), this option draws the link lines to the child items. This is used to disable the dimension line linking if the behavior is undesirable.
  - **Chain Child** - when set to true, this option draws dimension lines from child to child in hierarchical order, instead of from the parent to each individual child.
  - **Close Chain** - when set to true, this options draws an additional dimension line link back to the parent item from the last child item.



Dimensions shown on a Mesh Item



Dimension measurements



Dimensions

Visibility: Selected

Opacity: 50.0 %

☑ Show X Axis

Min. X Size: 0 m

Max. X Size: 1 m

☑ Show Y Axis

Min. Y Size: 0 m

Max. Y Size: 1 m

☑ Show Z Axis

Min. Z Size: 0 m

Max. Z Size: 1 m

☑ Link Children

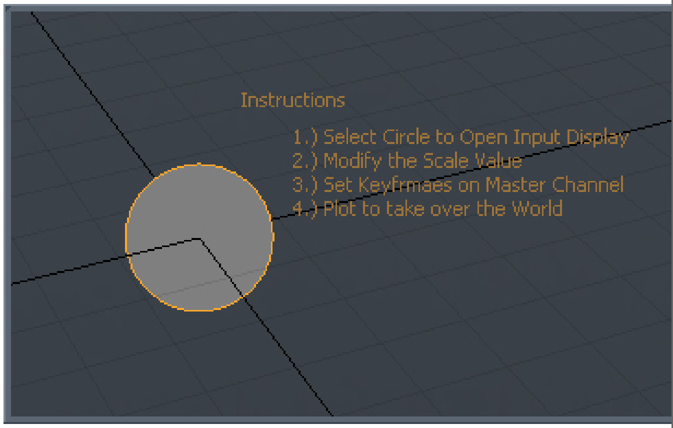
☑ Chain Children

☑ Close Chain

The **Dimensions** panel in the **Properties** viewport.

- **Show/Hide Notes** - when this option is enabled, an additional sub-tab appears in the **Properties** panel titled **Multi-Line Note**. This option allows for on-screen display of custom text strings for information such as instructions. The text is useful to Riggers and TDs, or just to add some helpful reminders for an item when selected.
  - **Title** - the first line of text that displays above the **Line** entries.
  - **Visibility** - determines the visibility of the dimension lines, with **Off** being not visible, **On** being visible, and **Selected** being only visible when selected.
  - **Opacity** - controls the opacity of the dimension line and numerical values displayed in the 3D viewport.
  - **Justification** - determines the justification of the text block so that text is set to **Left**, **Center**, or **Right**.
  - **Line 1-8** - text line entries that display in the 3D viewport.

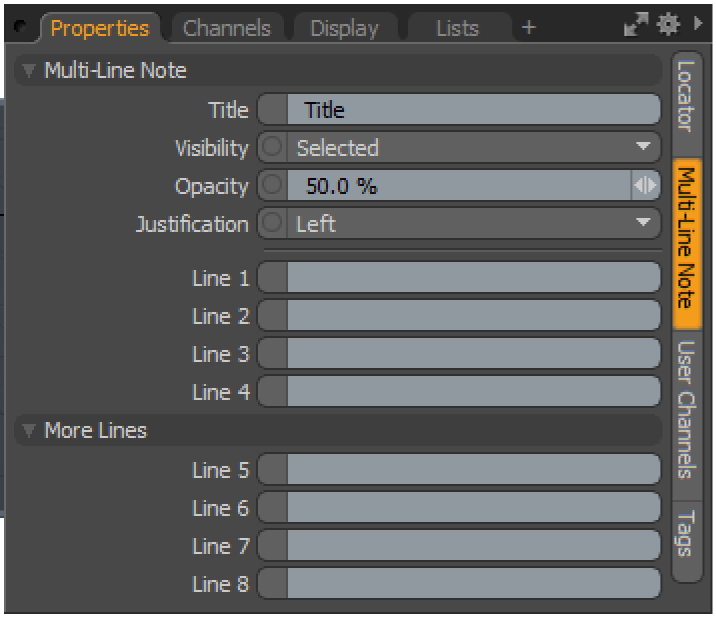
Note: Display text can also be positioned with the **Label Offset** control.



Instructions

1.) Select Circle to Open Input Display
2.) Modify the Scale Value
3.) Set Keyfrmaes on Master Channel
4.) Plot to take over the World

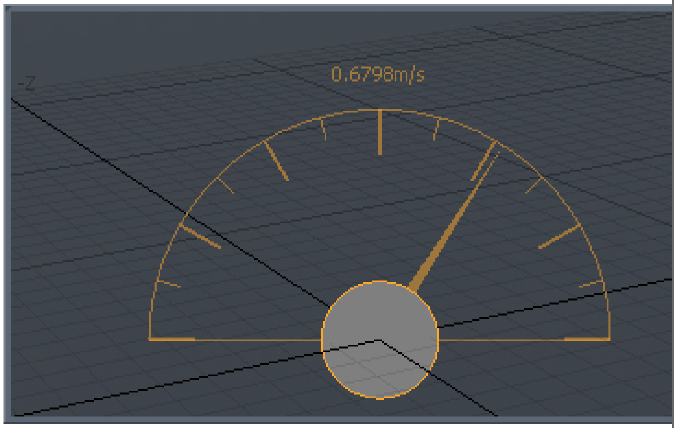
Dimension measurement labels.



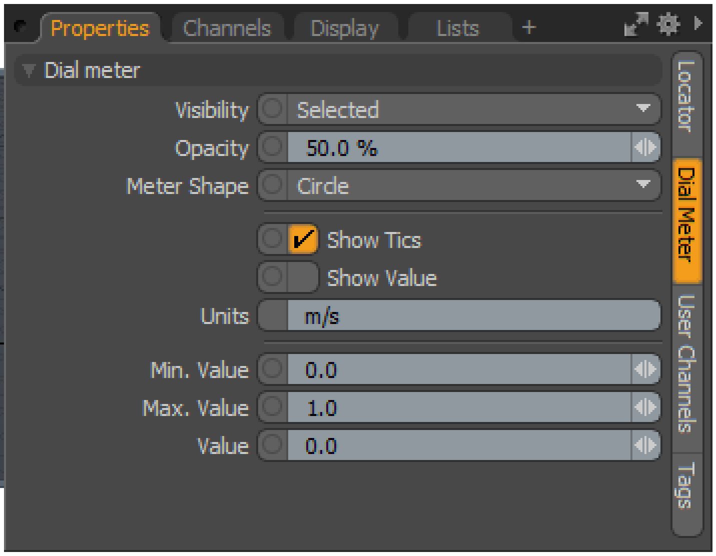
Multi-Line Note
Title: Title
Visibility: Selected
Opacity: 50.0 %
Justification: Left
Line 1
Line 2
Line 3
Line 4
More Lines
Line 5
Line 6
Line 7
Line 8

The **Multi-Line Note** panel in the **Properties** viewport.

- **Show/Hide Meter** - when this option is enabled, you can display an on-screen analog style gauge graph (looks like a speedometer) that can be rigged to show an individual value in the 3D viewport. An additional sub-tab appears in the **Properties** panel titled **Dial Meter** that allows some customization of the display.
  - **Visibility** - determines the visibility of the meter display, with **Off** being not visible, **On** being visible, and **Selected** being only visible when selected.
  - **Opacity** - controls the opacity of the meter readout and numerical values displayed in the 3D viewport.
  - **Meter Shape** - choose between a full **Circle** meter display and a **Half Circle** display.
  - **Show Tics** - toggles the visibility of 15° and 30° interval lines in the display.
  - **Show Value** - toggles the visibility of the **Value** information readout above the dial.
  - **Units** - a text display that is positioned next to the **Value** display when the **Show Value** option is set to true.
  - **Min/Max Value** - the Min and Max values determines the lowest value and the the highest value that is displayed in the on-screen meter, controlling the overall position of the **Value** needle.
  - **Value** - this is the input channel that determines the actual value displayed in the viewport, as well as the position of the meter needle (relative to the Min and Max values). Additional rigging is required in the Schematic Viewport in order to connect the **Value** channel input from an output.



Analog meter displayed.



Dial meter

Visibility: Selected

Opacity: 50.0 %

Meter Shape: Circle

☑ Show Tics

☐ Show Value

Units: m/s

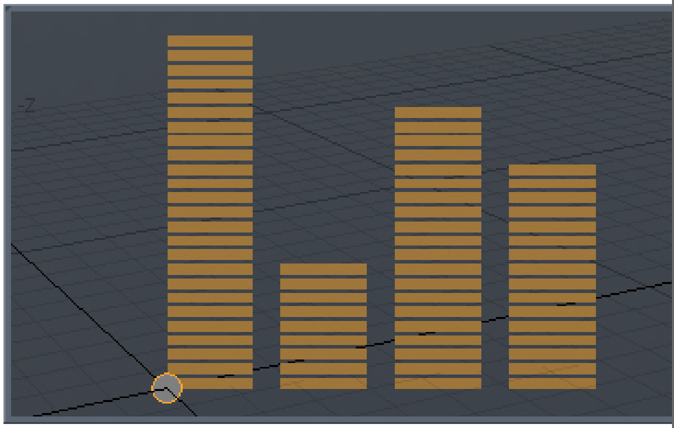
Min. Value: 0.0

Max. Value: 1.0

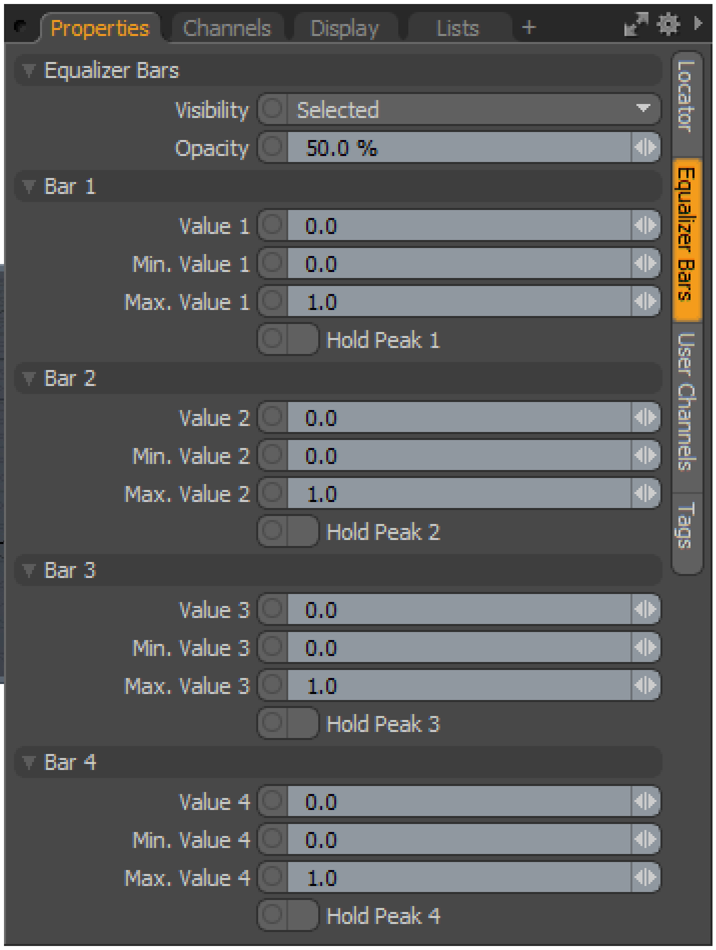
Value: 0.0

The **Dial Meter** panel in the **Properties** viewport.

- **Show/Hide Level Bars** - when this option is enabled, an additional sub-tab appears in the **Properties** panel titled **Equalizer Bars**. This option is used to display an on-screen equalizer-style bar graph for visualizing multiple on-screen values, relative to one another. Additional channels show in the Channels Viewport when enabled.
  - **Visibility** - determines the visibility of the level bars, with **Off** being not visible, **On** being visible, and **Selected** being only visible when selected.
  - **Opacity** - controls the opacity of the level bars readout displayed in the 3D viewport.
  - **Value 1-4** - these are the input channels that determine the actual values displayed in the viewport, as well as the scale of the individual level bars (relative to the Min and Max values). Additional rigging is required in the Schematic Viewport in order to connect the **Value** channel input from an output.
  - **Min Value 1-4/Max Value 1-4** - the Min and Max values determines the lowest value and the highest value, per readout, that is displayed in the on-screen level bar, controlling the overall scale of the bar displayed.
  - **Hold Peak 1-4** - like an audio readout, when the **Max** value is reached for any **Value** readout, an indicator is drawn in the viewport. Toggle the **Hold Peak** value to reset the display.



Level bars meter displayed.



Equalizer Bars

Visibility Selected

Opacity 50.0 %

Bar 1

Value 1 0.0

Min. Value 1 0.0

Max. Value 1 1.0

Hold Peak 1

Bar 2

Value 2 0.0

Min. Value 2 0.0

Max. Value 2 1.0

Hold Peak 2

Bar 3

Value 3 0.0

Min. Value 3 0.0

Max. Value 3 1.0

Hold Peak 3

Bar 4

Value 4 0.0

Min. Value 4 0.0

Max. Value 4 1.0

Hold Peak 4

The **Equalizer Bars** panel in the **Properties** viewport.

Note: Up to eight bars total can be displayed by referencing the additional Channels available in the **Channels** viewport.

Gradient Editor

The **Gradient Editor** viewport lets you control Gradient textures in Modo. Gradients are useful in modifying values gradually across a certain parameter such as along a surface, based on its length or angle. Gradients can also be used to drive other layers for complex texturing. It's all dependent on the Layer Effects that are selected and the input parameters. For more information on Gradients, see Gradient.

To open the Gradient Editor:

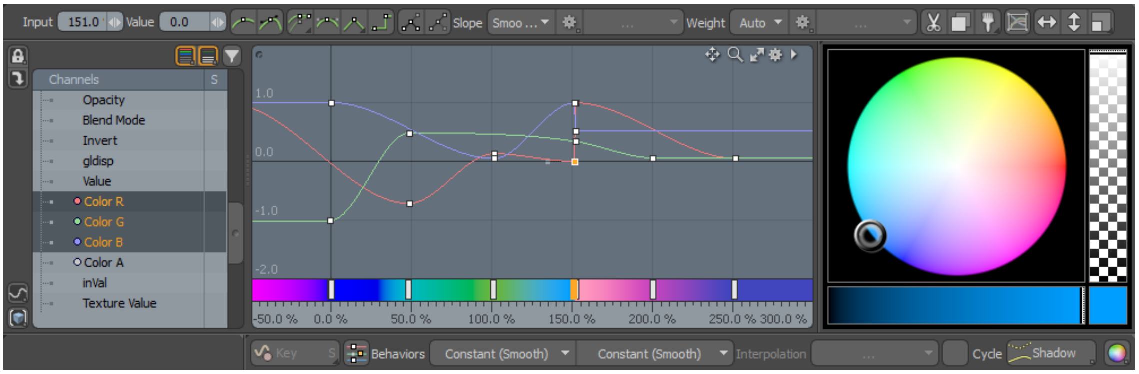
- With the gradient item selected in the **Items** list, in the **Properties** panel, click the **Edit Gradient** button

  Alternatively, click **Layout** > **Palettes** > **Gradient Editor**.

Usage

A gradient layer is added to the **Shader Tree** using the **Add Layer** button under **Processing** > **Gradient**. Once created, you need to assign the appropriate effect for the target surface. For more information on working with **Shader Tree** layers, see the Shader Tree topic.

The **Gradient Editor** is in many ways similar to the Graph Editor; the only difference being that the graph editor modifies values over time. It is helpful as well if you are familiar with the Color Picker Viewport as its use is integral for selecting color values.


Input
151.0
Value
0.0
Slope
Smoo ...
Weight
Auto
Channels
S
Opacity
Blend Mode
Invert
gldisp
Value
Color R
Color G
Color B
Color A
inVal
Texture Value
1.0
0.0
-1.0
-2.0
-50.0 %
0.0 %
50.0 %
100.0 %
150.0 %
200.0 %
250.0 %
300.0 %
Key
Behaviors
Constant (Smooth)
Constant (Smooth)
Interpolation
Cycle
Shadow

Channels and Channel Visibility

From the **Channels** section on the left side, you can select the appropriate value(s) you want to modulate. This depends on the desired effect. For diffuse color they are the **Color R**, **G**, and **B** channels, for specular amount it is the **Value** channel (any input that requires a single percent is a **Value** graph). Once selected, a line representing the numerical values of the channel appears in the graphing window for adding keys.

At the top and along the left side of the channel list are some toggle buttons that control the visibility of the channels.

▤	When enabled, this button shows transform channels for items.
▤	When enabled, this button shows standard item channels.
▽	Allows you to search for a specific channel. When enabled, a **Search** field appears where you

	can enter a search string.
∿	When enabled, this button only show channels in the channel view that have existing keys.
⬢	When enabled, this button only show channels in the channel view that are part of the selected group.

Key Creation and Selection

Keys can be added to the currently-selected curve(s) by middle-clicking either on the curve itself or anywhere on the background. Alternately, you can use the **Ctrl**+**Alt**+click to create a key. Key values are created at the location of the cursor. The **Input** and **Value** numeric fields in the upper-left corner allow for precise numerical editing of selected keys.

Selecting a key can be done by clicking with the cursor directly over the key itself. Holding the **Shift** key while selecting a key adds it to the selection. Holding the **Ctrl** key removes a key from the selection set. Clicking on the editor's background deselects all keys.

Clicking and dragging on the background starts a box marquis selection. Drag out the box to enclose the keys you want to select and release the mouse button. All keys within the box become selected. Any keys outside the box are deselected. Holding down the **Shift** key adds the keys within the box to the selection while the **Ctrl** key removes them.

Setting Colors for Keys

The color of selected keys can be set or adjusted by either clicking and dragging on the **Color Picker** viewport, or from within the right-click context menu. Selecting **Key Color** from this menu opens the **Color Picker**, from which a color may be chosen. In both cases, additional keys may be created on associated curves at the key **Input** value. Clicking the corner swatch for the color opens the system color picker.

Deleting Keys

Keys can be deleted from curves in two ways: to delete only the currently-selected keys, press the **Delete** key, to delete the currently-selected keys and keys at the same **Input** value on associated curves, select **Delete Key** from the right-click context menu.

Editing Keys

Clicking and dragging on a key selects the key, if it's not already selected, and allows quick editing of both **Input** and **Output** values at the same time. If there are several keys already selected and you click and drag on one of them, then all the selected keys are modified. Middle-clicking and dragging on a key works in a similar way except that editing is constrained to the **Output** value. Similarly, right-clicking and dragging constrains the edits just to the input parameter value.

Middle-clicking and dragging vertically on the background adjusts the **Output** value of all selected keys. Right-clicking and dragging horizontally on the background adjusts the **Input** value of all selected keys.

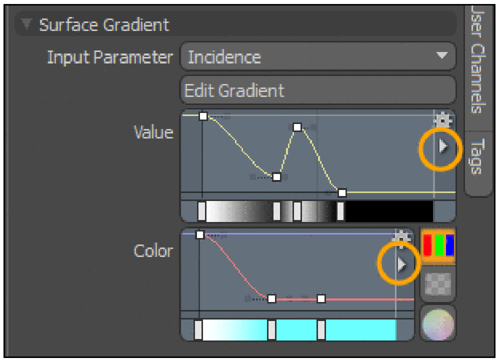
Ctrl+middle-click+drag on the background provides constrained editing of **Input** and **Output** values for selected keys, based on the initial direction of movement. Double-clicking on a key selects all the keys on the curve.

Context Menu

The **Gradient Editor** context menu is accessed by right-clicking. If the cursor is over an un-selected key when the menu is opened, the key is added to the selection. Operations performed from the context menu affect all the currently-selected keys.

Saving Presets

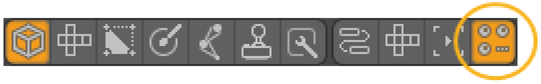
You can save your **Value** and **Color** as Presets for later use by clicking the down arrow and selecting **(save preset)**. The Preset is saved to the **Preset Browser** **Assets** > **Envelopes** directory. Once done, **Color Gradients** and **Percentage Gradients** directories are created.



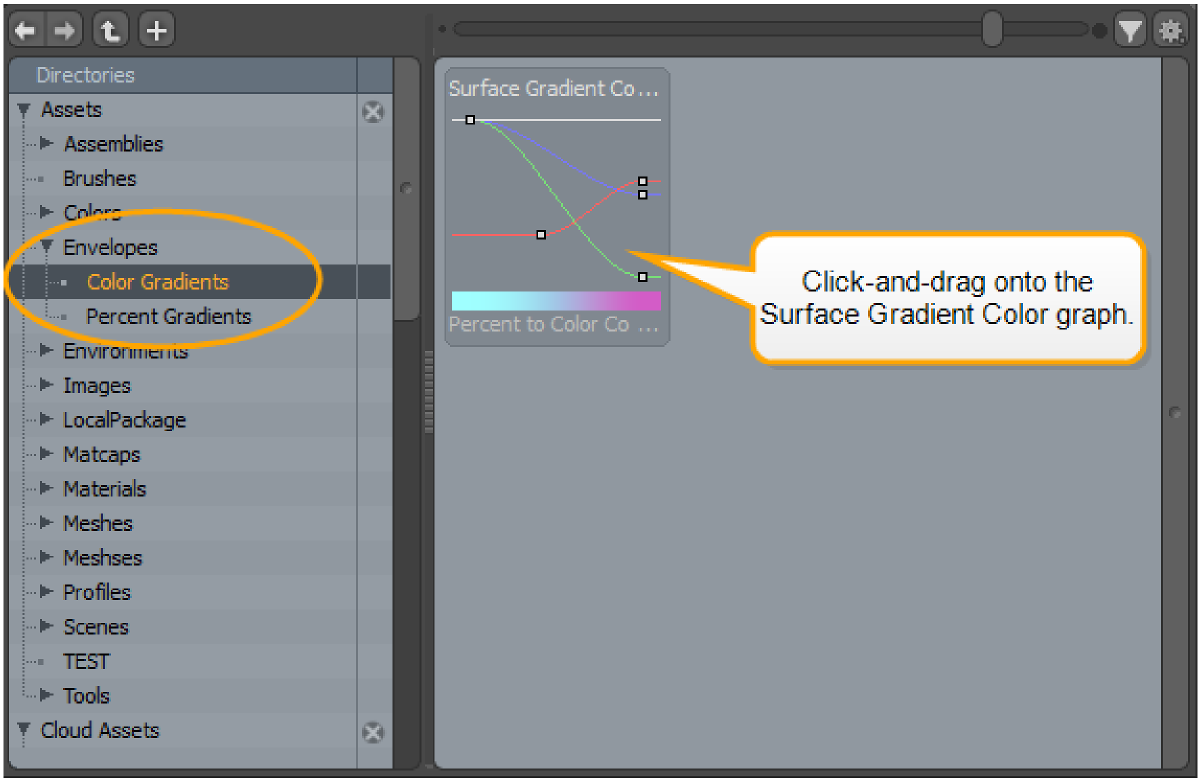
You can also drag-and-drop saved **Color Gradients** and **Percent Gradients** presets from the Preset Browser to the Surface Gradient **Value** and **Color** options in the **Properties** tab.

Video: Watch these videos: Mini Gradient Editor and Envelope Presets.

1. On the top left corner of the interface, click the **Open the Preset Browser** icon.

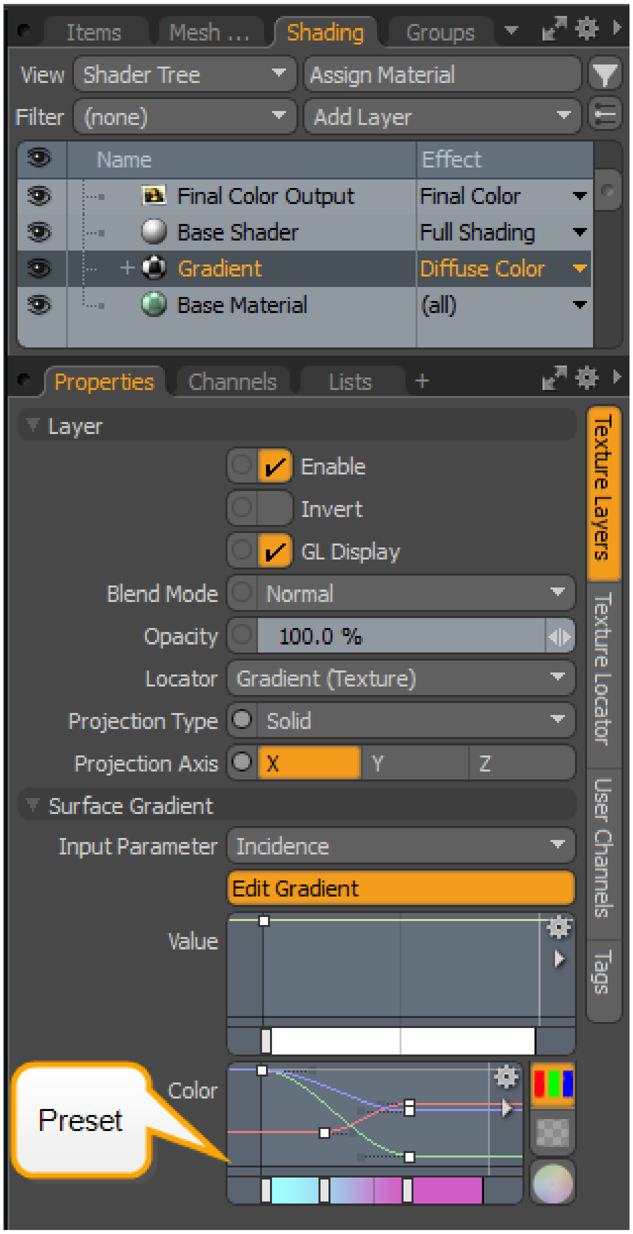


2. On the left panel of the Preset Browser, expand **Assets** > **Envelopes** > **Color Gradients**. Alternatively, expand **Assets** > **Envelopes** > **Percent Gradients**.


Directories
Assets
Assemblies
Brushes
Colors
Envelopes
Color Gradients
Percent Gradients
Environments
Images
LocalPackage
Matcaps
Materials
Meshes
Meshses
Profiles
Scenes
TEST
Tools
Cloud Assets
Surface Gradient Co ...
Percent to Color Co ...

Click-and-drag onto the Surface Gradient Color graph.

3. Click-and-drag a saved preset onto the **Surface Gradient Color** graph.



Items
Mesh ...
Shading
Groups
Assign Material
Add Layer
Properties
Channels
Lists
+
Texture Layers
Texture Locator
User Channels
Tags
Edit Gradient
Preset

View: Shader Tree

Filter: (none)

Name	Effect
Final Color Output	Final Color
Base Shader	Full Shading
Gradient	Diffuse Color
Base Material	(all)

Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Surface Gradient

Input Parameter: Incidence

Value

Color

Navigation

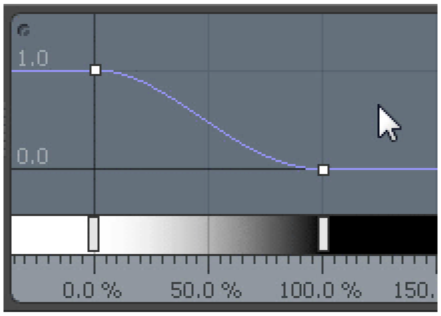
Panning and zooming behavior in the **Gradient Editor** is similar to that in the 3D Viewport, with a few additions:

- **Alt**+middle-click+drag adjusts the start value of the **Input Range** (horizontal), displayed in the editor, while **Alt**+right-click+drag adjusts the end value of the **Input Range**.
- **Alt**+**Shift**+middle-click zooms the **Output Range**, displayed in the editor, while **Alt**+**Shift**+right-click zooms the **Input Range**. In both cases, the zoom is centered on the initial cursor location.
- **Alt**+**Ctrl**+right-click provides box zooming.

- The mouse wheel zooms in and out around the cursor location.

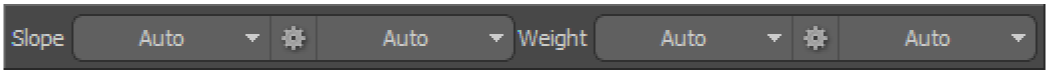
Modifying Curves

The **Gradient Editor** also offers a lot of control over how values interpolate from one key to the next. This is represented by the curve/line that connects each key. By default, the values gradually fade from one to the next: this is the **Auto** setting. You can adjust the key settings so incoming and outgoing lines are smooth, linear, stepped, or a variety of other shapes. Small Bezier-like handles appear next to selected keys that allow you to make these adjustments. Clicking and dragging the handle modifies it, putting it in to the **Manual** mode. The handles relate to the key in two ways - there is the angle of the handle called **Slope**, and then the length of the handle called **Weight**. These two options combine to produce the final shape of the curve. Some buttons quickly set the most common **Slope** and **Weight** combinations.



	Sets the selected keys to **Auto** for both **Slope** and **Weight**.
	Sets the selected keys to **Manual** for both **Slope** and **Weight**.
	Sets the selected keys to **Auto Flat Slope** and **Auto Weight**.
	Sets the selected keys to **Flat Slope** with **Auto Weight**.
	Sets the selected keys to **Linear** with additional sub-menu options (right-click) for **Linear In** and **Linear Out**.
	Sets the selected keys to **Stepped Slope** with **Auto Weight**.

The next group of buttons gives full control over both handles for each key: the incoming handle and the outgoing handle. You can think of the buttons representing the left side handle and the right side handle, where the setting is directed to each side, independent for the **Slope** and **Weight**. If the handle is unified, meaning adjusting one side moves the other in a straight line, then the outgoing settings are grayed out. The option gear in the middle allows you to "break" the handles or "unify" them, and to zero out the angle, which makes them perfectly horizontal again. There are a number of settings under the pop-up menu when clicking any of the buttons.



The **Slope** controls include:

- **Manual** - gives full control over **Slope** (handle angle).
- **Auto** - automatically adjusts **Slope** to produce a smooth ease-in and ease-out from the previous key value to the next.
- **Linear In** - angles the **Incoming Slope** to point toward the previous key, producing a straight line between keys.
- **Linear Out** - angles the **Outgoing Slope** to point toward the next key, producing a straight line between keys.
- **Flat** - sets the **Slope** to 0 degrees flat, regardless of the positions of the previous and next keys.
- **Auto Flat** - acts just like **Auto** except when the incoming or outgoing keys match the value, then it acts like **Flat**.
- **Stepped** - eliminates the **Slope** and **Weight** altogether, retaining the previous key value up until a new key is introduced.

The **Weight** control include:

- **Manual** - gives full control over **Weight** (handle length).
- **Auto** - automatically adjusts **Weight** to produce a smooth ease-in and ease-out from the previous key value to the next.

The following buttons also allow you to quickly break or unify the handles **Slope** and **Weight** in a single stroke.

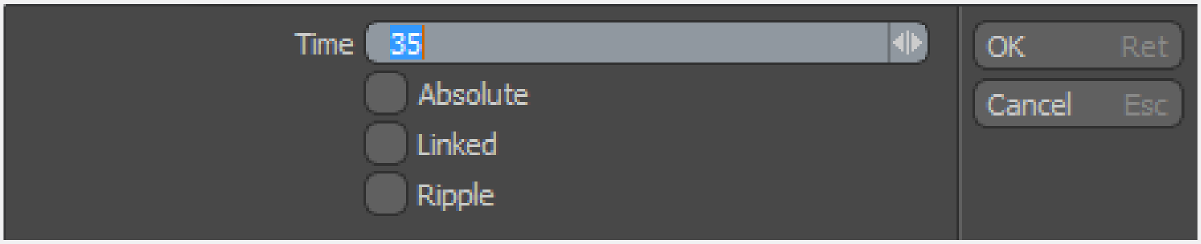
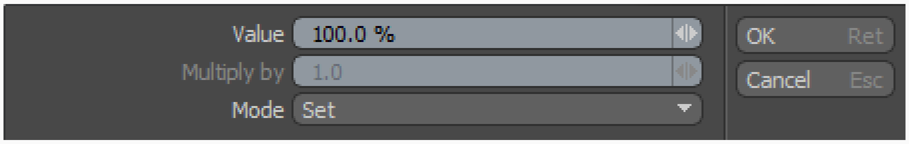
	Breaks the key handles for adjusting **Slope** and **Weight**.
	Unifies the handles for adjusting **Slope** and **Weight**.

Managing Keys

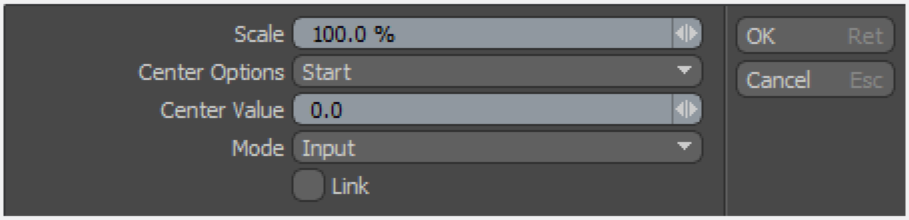
The **Gradient Editor** also offers some useful functions for working with key values. Working with a clipboard buffer, you can move keys around, and even move them from one channel to another. The **Paste** command simply adds the key values to the target, the **Insert** command adds them to the target and moves other keys as necessary, while **Replace** adds the keys, removing any existing keys it might overlap.

✂	Cuts the selected keys from the editor and stores them in a temporary buffer for later access.
⧉	Copies the selected keys from the editor and stores them in a temporary buffer for later access.
📋	Pastes the keys stored in the buffer with additional sub-menu options (right-click) for **Insert** and **Replace**.

The following buttons open dialog boxes that allow for easy gang editing of multiple selected values.

↔	Opens the **Set Input Value** dialog panel for adjusting the time position (not directly applicable to gradients).  Time 35 ☐ Absolute ☐ Linked ☐ Ripple • **Time** - determines the position in time to move the selected keys to. • **Absolute** - when enabled, **Absolute** time is used, where a specific time can be defined; when disabled, the time is relative to the current time position. • **Linked** - when enabled **Linked** keys (such as those of the R, G, and B channels) remain linked in the same relative position, even if only a single channel is edited; when disabled, keys can be moved freely. • **Ripple** - when enabled, all the following values of those selected are shifted by the same amount.
↕	Opens the **Set Key Value** dialog panel for adjusting the value of the selected keys.  Value 100.0 % Multiply by 1.0 Mode Set • **Value** - defines the new value that is applied to the selected keys when the mode is set as **Set**; defines what value is added when mode is set to **Add**. • **Multiple by** - acts as a multiplier of the selected values when mode is set to **Multiply**. • **Mode** - determines how the values affect the selected keys: **Set**, **Add**, and **Multiply**.

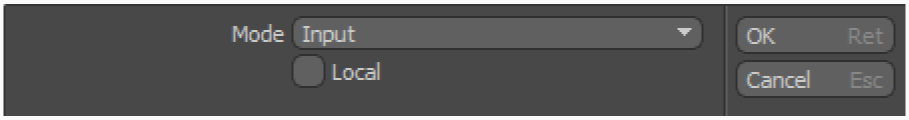
Opens the **Scale Keys** dialog panel for scaling the value of the selected keys.



Scale 100.0 %
Center Options Start
Center Value 0.0
Mode Input
☐ Link

- **Scale** - defines the amount of scaling applied to the selected keys.
- **Center Options** - determines the center location to scale the keys from- **Start**, **Center**, **End**, **Time**, and **User**. When **Time** or **User** is selected, the **Center Value** can be used to define the specific center.
- **Center Value** - determines the specific center locations when the **Time** or **User** options are selected as the **Center Option**.
- **Mode** - two modes to determine which direction gets scaled: **Input** scales the interval (horizontal) while **Output** scales the value (vertical).
- **Link** - when enabled, linked keys (such as those of the R, G, and B channels) remain linked in the same relative position, even if only a single channel is edited; when disabled, keys can be moved freely.

Opens the **Key Invert** dialog panel for adjusting the value of the selected keys.



Mode Input
☐ Local

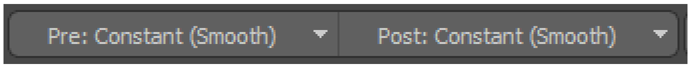
- **Mode** - two modes to determine which direction gets inverted: **Input** inverts along the interval (horizontal) while **Output** inverts along the value (vertical).
- **Local** - when the **Local** option is enabled, the inversion happens individually for each channel; when disabled, the inversion happens as a group for all that is selected.

Behaviors

The **Behavior** buttons define the incoming and outgoing curves that are outside the area of defined keys. The left button defines the per-behavior and the right button the post-behavior.

- **Constant** - values remain the same. With the **Auto Slope** and **Weight** options, ease-in or out as it transitions to the key values.
- **Stop** - values remain the same. For **Auto Slope** and **Weight**, transitions to the key values act as if linear - a sudden transition.
- **Linear** - values project outward in a linear fashion from the first or last key, providing a constant offset.

- **Repeat** - values of keys are repeated. When incoming and outgoing values match, a loop can be created.
- **Oscillate** - values are repeated forwards, and then reversed, like a ping-pong type effect.
- **Offset Repeat** - values of keys are repeated, but each subsequent repeat is offset by the amount of the last key.
- **Reset** - values are set to flat 0 (zero) outside the key area.



Pre: Constant (Smooth)

Post: Constant (Smooth)

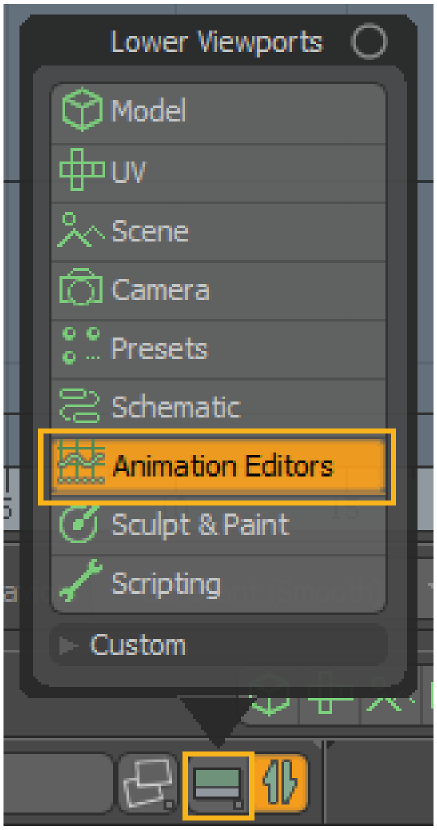
▦	When enabled, associated keys (such as the R, G, and B color channels) remain at the same interval relative to each other, even if only a single channel key is modified; when disabled, keys can be freely moved individually.
●	Opens the color picker for defining color-specific keys.

Graph Editor

Animating in 3D, in some ways, can be very similar to the art of stop motion animation, where you build a model, place it in a scene, light the scene for atmosphere, and then animate the model to bring it to life. However, unlike stop motion animation, in 3D animation, you can easily skip multiple frames and lock in a fixed value, called a keyframe. Between those skipped frames the computer interpolates from one value to the other.

To open the Graph Editor, click **Animate** > **Graph Editor**. Alternatively, click **Layout** > **Palettes** > **Graph Editor**.

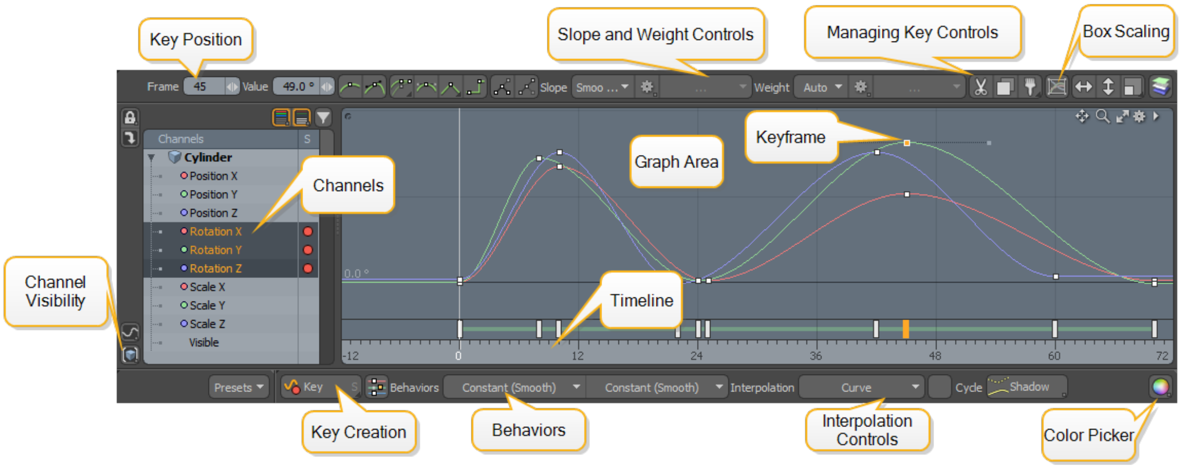
You can also view the **Graph Editor** using the **Animation Editors** accessible from the **Lower Viewports** Popover menu at the bottom of the Viewport.


Lower Viewports
Model
UV
Scene
Camera
Presets
Schematic
Animation Editors
Sculpt & Paint
Scripting
Custom

Lower Viewports Popover menu

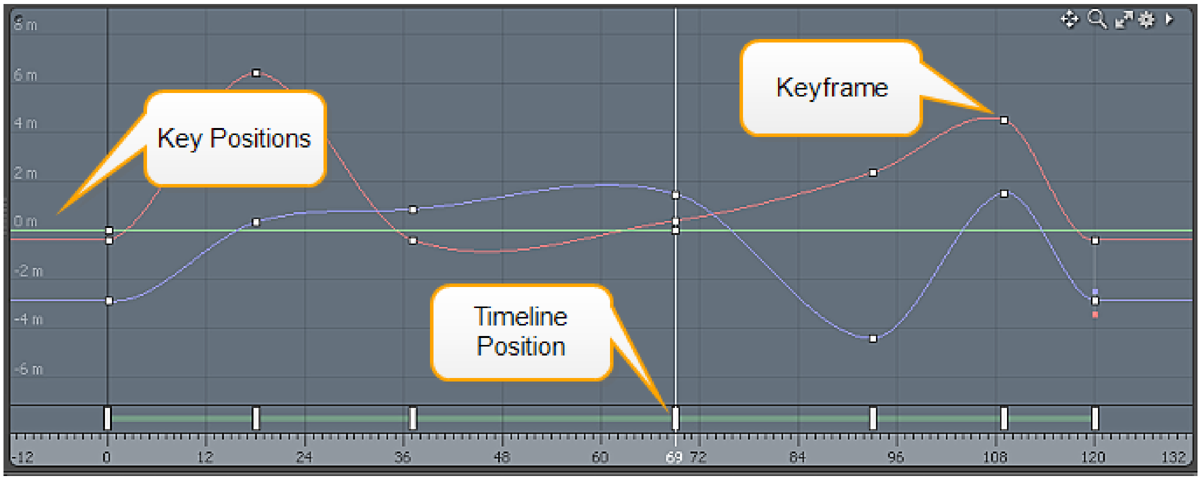
The **Animation Editors** is a layout containing the **Graph Editor** and **Dopesheet**, as well as the standard **Channels** viewport so you can easily select parts of your rig and channels and make adjustments.

The Graph Editor controls how the system interpolates those in-between key values.


Key Position
Slope and Weight Controls
Managing Key Controls
Box Scaling
Channels
Graph Area
Keyframe
Channel Visibility
Timeline
Key Creation
Behaviors
Interpolation Controls
Color Picker
Frame
Value
Slope
Weight
Cylinder
Position X
Position Y
Position Z
Rotation X
Rotation Y
Rotation Z
Scale X
Scale Y
Scale Z
Visible
Presets
Key
Behaviors
Constant (Smooth)
Constant (Smooth)
Interpolation
Curve
Cycle
Shadow

Nearly any property of any item can be animated by keyframing the values of an item's channel. Channels are the animateable properties of any item gathered together in list form. Generally, you interact with channels within the Channels Viewport, but since channels are so essential to the **Graph Editor**, the very same channels appear when an item is selected in the left-most column.

The graph area is where the actual values are manipulated. In this graph, left to right always represents time, but up and down changes depending on the channel type selected. The line that connects the individual keyframes represents the value's change over time. By default, Modo uses a smooth curve to interpolate between keyframe values, providing a nice ease-in/ease-out type of effect. There may be times when you want a very different interpolation style, or simply to apply finer control. Animation curves can only be changed by manipulating the keyframes and are very similar to working with Bezier curves in the modeling viewport. Keyframes are represented as tiny small squares along the line.



The Timeline across the bottom is functionally identical to the Timeline in the **Animation** layout. Clicking and dragging changes the current time. As you drag the current time value around, you can see the effect of the animation curve in the 3D viewport. Handles appear in the Timeline wherever a keyframe appears, and you can click and drag them in time forwards or backwards, modifying the position of the keyframes for selected channels.

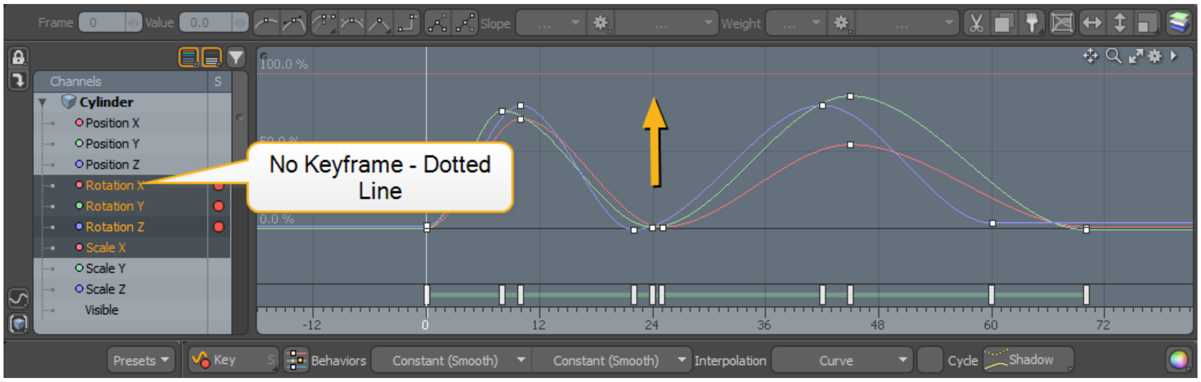
Channels

When you open the **Graph Editor**, any selected items appear with their associated channels in the **Channels** section of the viewport. From this section, you can select the appropriate channel(s) you want to adjust. Only the selected channels appear in the graph area.

To select channels:

- On the left panel, click an individual **Channel**.
- Use the **Ctrl** and **Shift** keys to add or subtract additional **Channels** from selection groups.
- Hold **Shift** and click the top and bottom-most channels to select sequential channels.

When a particular channel is selected, its animation curve is displayed in the graph area. If no keyframes are defined, it appears as a flat dotted line. Once a keyframe is created, it appears solid.


Frame
0
Value
0.0
Slope
Weight
100.0 %
0.0 %
Channels
S
Cylinder
Position X
Position Y
Position Z
Rotation X
Rotation Y
Rotation Z
Scale X
Scale Y
Scale Z
Visible
-12
0
12
24
36
48
60
72
Presets
Key
Behaviors
Constant (Smooth)
Constant (Smooth)
Interpolation
Curve
Cycle
Shadow

No Keyframe - Dotted Line

Channel Visibility Properties

Along the left side of the viewport are toggle buttons that filter the visibility of the channels:

∿	When enabled, only shows channels in the **Channels** view that have existing keys.
◈	When enabled, only shows channels in the **Channels** view that are part of the selected Groups Viewport.
☰	When enabled, shows transform channels for items.
☰	When enabled, shows standard item channels.
⤵	When pressed, loads the currently-selected items into the **Channels** view, replacing the current contents.
🔒	When enabled, locks the current items in the **Graph Editor**, so they remain, regardless of item selection, until disabled.

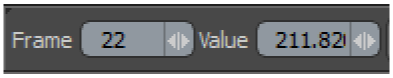
Key Creation

Keys can be added to the currently-selected curve(s).

- Middle-clicking either on the curve itself or anywhere on the background.
- Alternatively, you can use the **Ctrl**+**Alt**+click shortcut to create keys.

Key values are created at the location of the pointer.

Tip: When working in the Graph Editor, dragging a key with the right mouse button constrains it horizontally so that only the time is adjusted.



The **Frame** and **Value** fields in the upper-left corner allow for precise editing of selected keys.

There are also two buttons for assigning keys that utilize the location of the Timeline within the graph.

	Adds a keyframe to the selected channels at the current Timeline position.
	Adds a keyframe, same as the **Key** command, but retains the shape of the target curves by adjusting the **Slope** and **Weight** values along the target curve.

Key Selection

To select a key(s):

- Click over the key itself.
- Hold the **Shift** key while selecting to add to the selection.
- Hold the **Ctrl** and click on a key to remove it from the selection.
- Click on the graph background to deselect all keys.
- Click-and-drag on the background starts "box marquis" selection - drag out a box to enclose the keys you want to select and release the mouse button. All keys within the box are selected and any outside the box are deselected.

Editing Keys

- Once you have added a keyframe to your curve, you can quickly add additional keyframes. Select the original keyframe to activate it, press **Shift**+**Ctrl**, and then drag it to a new position on the curve. Repeat this process to add additional keyframes. Once done, drop the **Shift**+**Ctrl** keys. As a result, new keyframes are added to the curve at the positions you specified.
- Click and drag on a key selects the key if it's not already selected and allows quick editing of both **Input** and **Output** values at the same time. If there are several keys already selected, and you click drag on one of them, then all the selected keys are modified.
- Middle-click and drag on a key works in a similar way except that editing is constrained to the **Output** value. Similarly, right-click and drag constrains the edits just to the **Input** value.

- Middle-click and dragging vertically on the background adjusts the **Output** value of all selected keys.
- Right-click and dragging horizontally on the background adjusts the **Input** value of all selected keys.
- **Ctrl**+middle-click and drag on the background provides constrained editing of **Input** and **Output** values for selected keys, based on the initial direction of movement.
- Double-clicking a key selects all the keys on the curve.

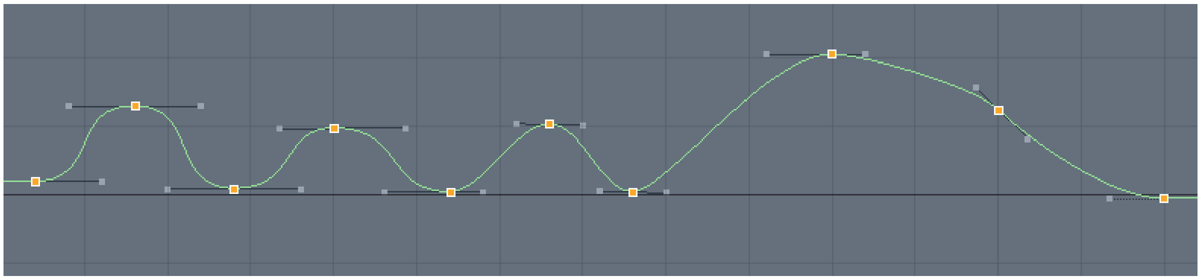
	When **Link related keys** is enabled, associated keys (such as the R, G, and B color channels) remain at the same time interval, relative to each other, even if only a single channel key is modified; when disabled, keys can be freely moved.

Inverting Keys

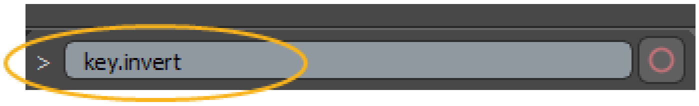
You can flip the horizontal and vertical positions of selected keys using either of the following two methods.

Key.invert Command

1. Select two or more keyframes you want to invert by **Shift**+clicking on them.

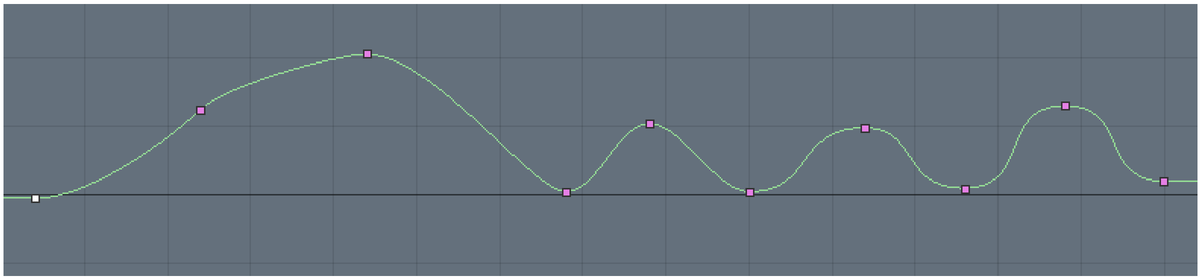


2. In the Command line in the bottom-right corner, enter "key.invert", then press **Enter**.



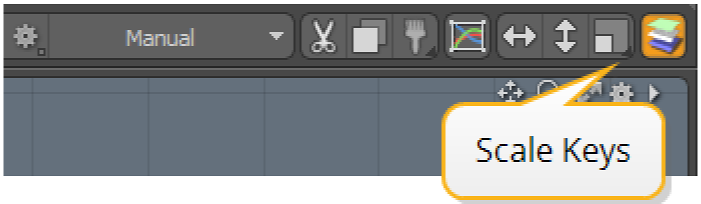
```
> key.invert
```

The positions of the selected keys are flipped.



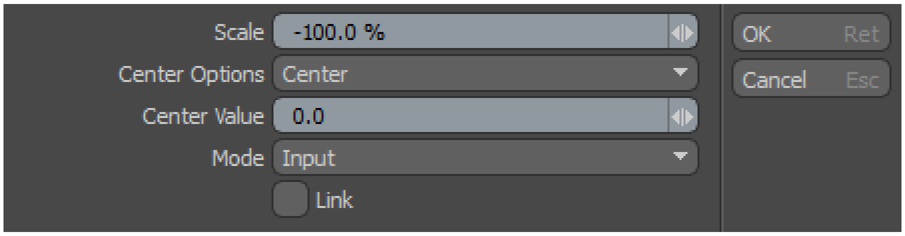
Scale Keys

Another way of inverting keys is to use the Scale Keys ◪ button at the top of the Graph Editor.



1. With the keys selected, click the Scale Keys ◪ button.

   This opens the **Scale Keys** dialog.
2. Set **Scale** to **-100.0%** and **Center Options** to **Center**, then click **OK**.



Scale -100.0 %
Center Options Center
Center Value 0.0
Mode Input
☐ Link

The positions of the selected keys are flipped.

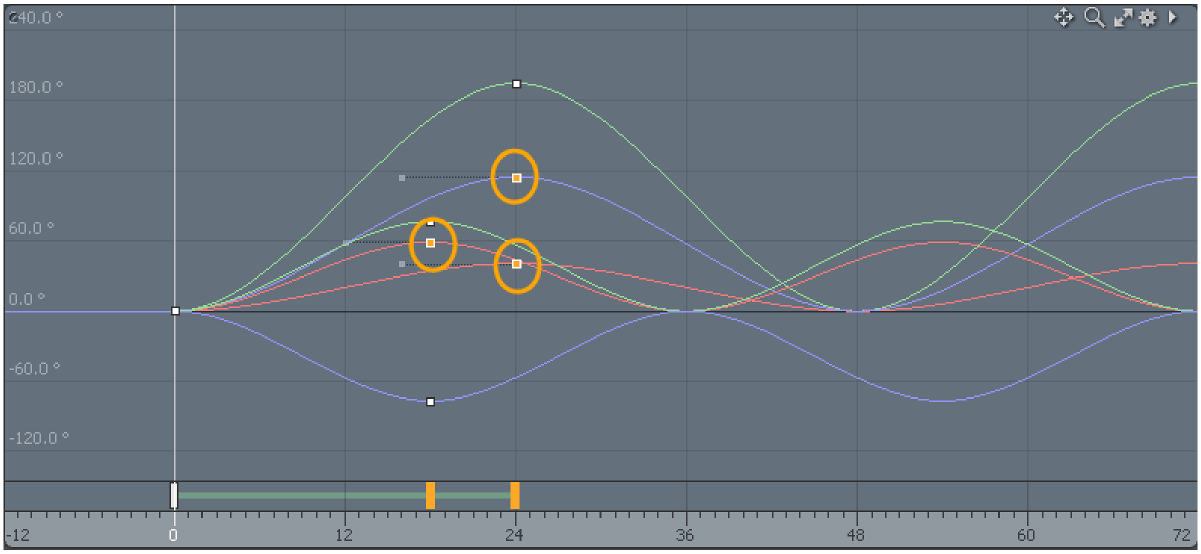
Note: For more information on the **Scale Keys** options, see Managing Keys.

Scaling Multiple Keyframes

You can scale multiple selected keyframes by their frame numbers and keyed values using the Box Scaling Tool.

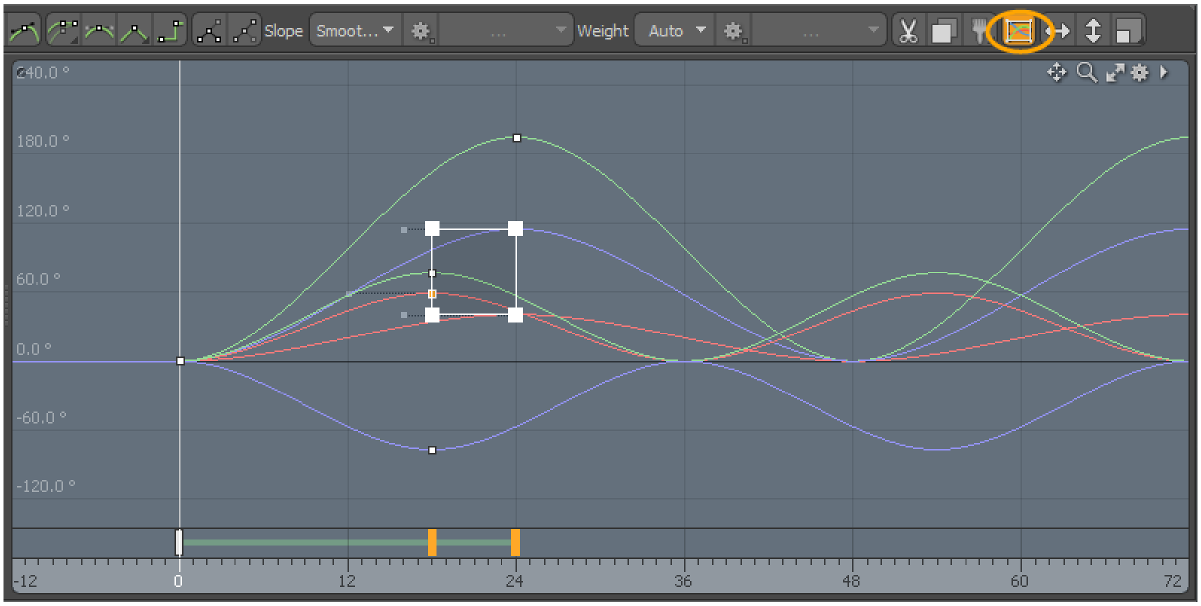
1. On the right panel, in the **Items** list, select the mesh item you want to edit.
2. Press **F7** to open the **Graph Editor**.
3. Hold **Shift** or **Ctrl** and click on individual keyframes to make a selection.

Note: At least two keys with different input and output values must be selected.


240.0 °
180.0 °
120.0 °
60.0 °
0.0 °
-60.0 °
-120.0 °
-12
0
12
24
36
48
60
72

4. On the top right corner of the interface, click the **Box Scaling** Tool icon.

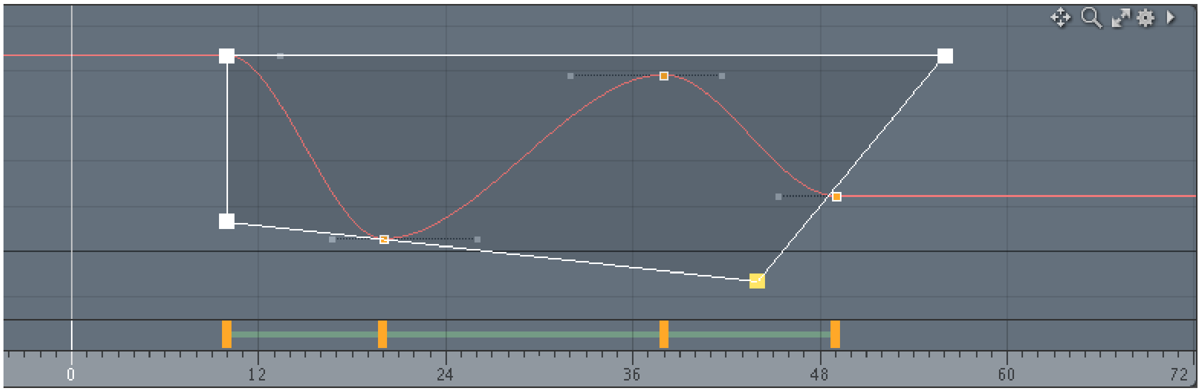
   A highlighted box displays around the selected keyframes on the graph.


Slope
Smoot...
Weight
Auto
240.0 °
180.0 °
120.0 °
60.0 °
0.0 °
-60.0 °
-120.0 °
-12
0
12
24
36
48
60
72

5. Move or scale your keyframe selection by doing any of the following:

- **Move** - Move your selection of keyframes.
  - Click and drag in the middle of the highlight box to move all selected keys.
  - Hold **Shift** and click and drag to constrain the selection movement up and down.
  - Hold **Ctrl**/**Cmd** and click and drag to constrain the selection movement left and right.
- **Scale** - Scale your selection of keyframes.
  - Click and drag from any of the sides of the box to scale all keys from the sides.
  - Click and drag any of the corner points to scale your selections on the X and Y axis at the same time.

- Hold **Ctrl**/**Cmd** on one of the corner points and click-and-drag to scale and warp your selection from that point.



To reset the tool, click on the viewport outside the handle area.

Deleting Keys

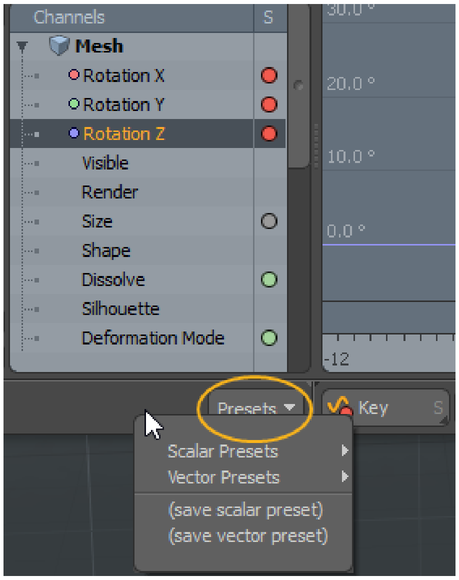
Keys can be deleted from curves in two ways:

- To delete only the currently-selected keys, press **Delete** on your keyboard.
- To delete the currently-selected keys and keys at the same **Input** value on associated curves, select **Delete Key(s)** from the right-click context menu.

Saving Presets

You can save vector and scalar Envelope Presets for later use from the Graph Editor. Vector presets are a combination of multiple channels and contain direction information, while scalar presets are the values of a single channel.

To save a preset, select the channel to save and click the **Presets** button at the bottom of the viewport.

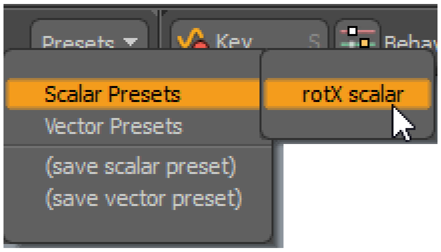

Channels
S
Mesh
Rotation X
Rotation Y
Rotation Z
Visible
Render
Size
Shape
Dissolve
Silhouette
Deformation Mode
30.0 °
20.0 °
10.0 °
0.0 °
-12
Presets
Key
Scalar Presets
Vector Presets
(save scalar preset)
(save vector preset)

The preset is saved to the **Preset Browser**. Scalar presets are saved under **Assets** > **Envelopes**.

Note: The **Envelopes** folder is only created when you save your first Envelope Preset.

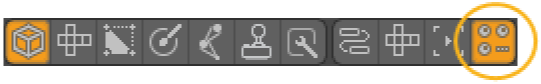
To apply a preset:

Select the channel to apply the preset to, then in the **Graph Editor**, click the **Presets** button, and click one of the available scalar or vector presets.

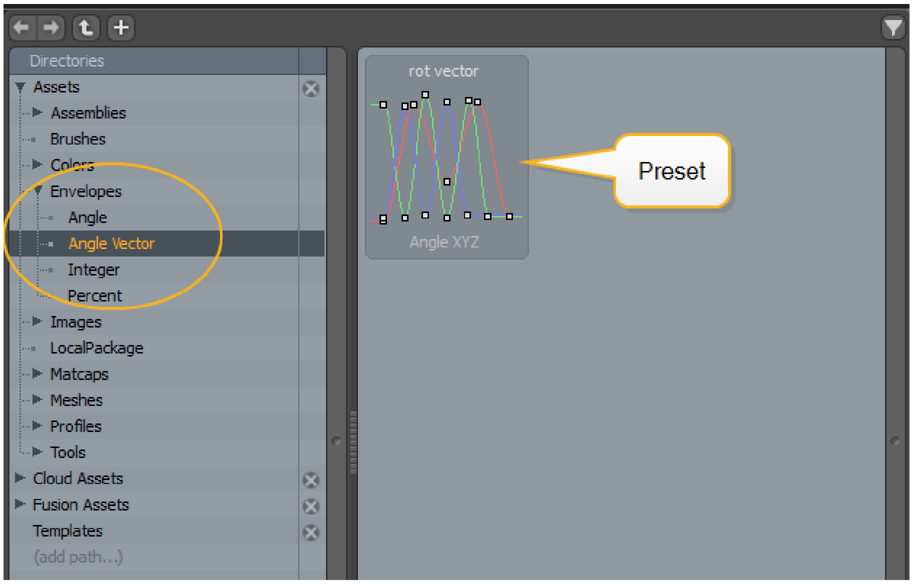

Presets
Key
Behav
Scalar Presets
Vector Presets
(save scalar preset)
(save vector preset)
rotX scalar

Alternatively, you can drag the preset from the Preset Browser and drop it onto the **Graph Editor:**

1. On the top left corner of the interface, click the **Open the Preset Browser** icon.

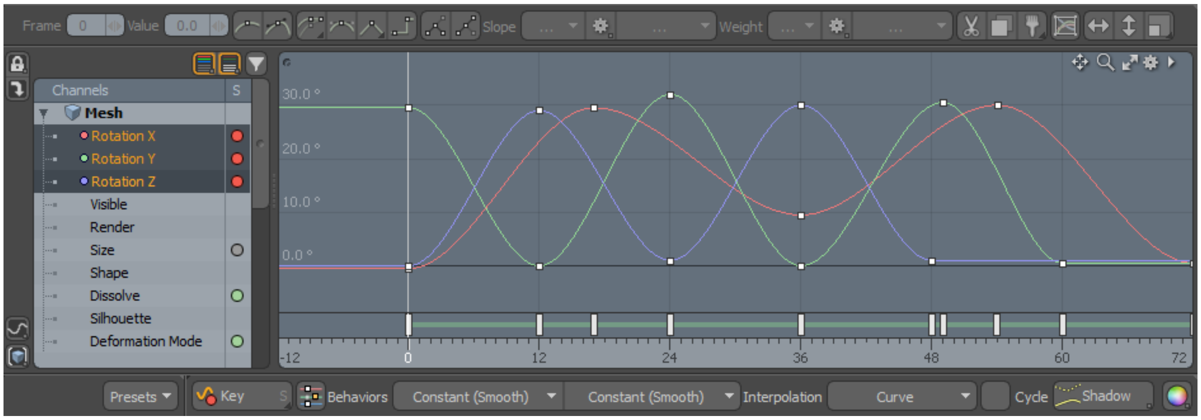


2. On the left panel of the Preset Browser, expand **Assets** > **Envelopes**, then open the appropriate folder and click the preset you need.


Directories
Assets
Assemblies
Brushes
Colors
Envelopes
Angle
Angle Vector
Integer
Percent
Images
LocalPackage
Matcaps
Meshes
Profiles
Tools
Cloud Assets
Fusion Assets
Templates
(add path...)
rot vector
Angle XYZ
Preset

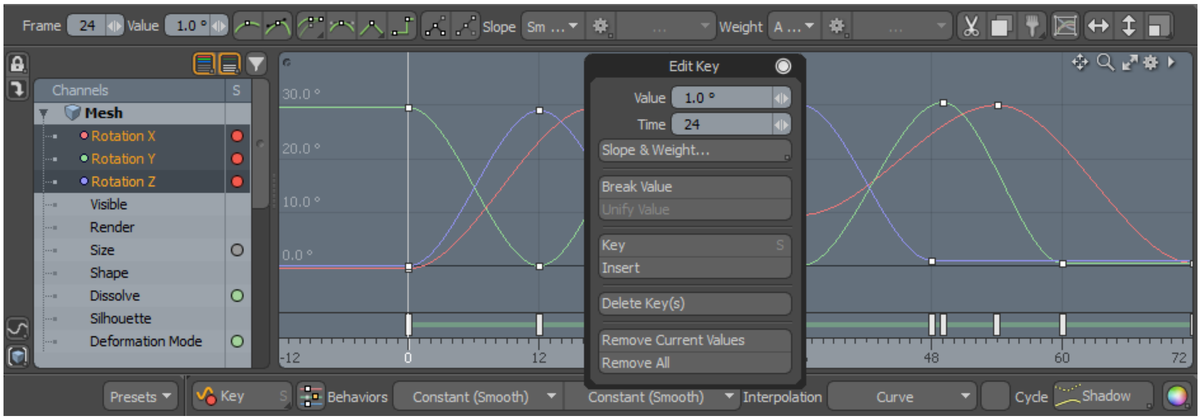
3. Click-and-drag a saved preset onto the **Graph Editor**.

   The channel's value is added.


Frame
0
Value
0.0
Slope
Weight
Channels
S
Mesh
Rotation X
Rotation Y
Rotation Z
Visible
Render
Size
Shape
Dissolve
Silhouette
Deformation Mode
30.0 °
20.0 °
10.0 °
0.0 °
-12
0
12
24
36
48
60
72
Presets
Key
Behaviors
Constant (Smooth)
Constant (Smooth)
Interpolation
Curve
Cycle
Shadow

Context Menu

The **Graph Editor** context menu is accessed by right-clicking. If the cursor is over an un-selected key when the menu is opened the key is added to the selection. Operations performed from the **Edit Key** context menu affect all the currently selected keys.


Frame
24
Value
1.0 °
Slope
Sm ...
Weight
A ...
Channels
Mesh
Rotation X
Rotation Y
Rotation Z
Visible
Render
Size
Shape
Dissolve
Silhouette
Deformation Mode
30.0 °
20.0 °
10.0 °
0.0 °
-12
0
12
48
60
72
Edit Key
Value
1.0 °
Time
24
Slope & Weight...
Break Value
Unify Value
Key
Insert
Delete Key(s)
Remove Current Values
Remove All
Presets
Key
Behaviors
Constant (Smooth)
Constant (Smooth)
Interpolation
Curve
Cycle
Shadow

Navigation

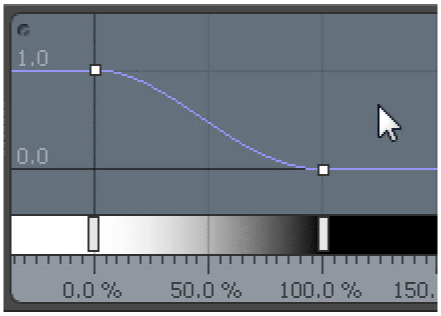
Panning and zooming behavior in the **Graph Editor** is similar to the 3D Viewports with a few additions:

- **Alt**+middle-click+drag adjusts the start value of the **Input** (horizontal) range displayed in the editor, while **Alt**+right-click+drag adjusts the end value of the **Input** range.
- **Alt**+**Shift**+middle-click zooms the **Output** range displayed in the editor, while **Alt**+**Shift**+right-click zooms the **Input** range. In both cases, the zoom is centered on the initial cursor location.
- **Alt**+**Ctrl**+right-click provides box zooming.
- Use the mouse wheel to zoom in and out at the cursor location.

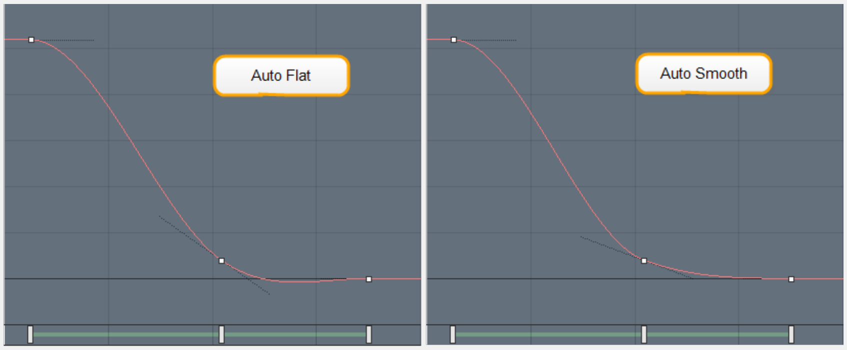
Modifying Curves

There are a number of ways to control interpolation from one key to the next. This is represented by the curve/line that connects each key. By default, the values gradually fade from one to the next - this is the **Auto** setting. You can adjust the key settings so incoming and outgoing lines are smooth, linear, stepped, or a variety of other shapes. Small Bezier-like handles appear next to selected keys that allow you to make these adjustments. Clicking and dragging the handle modifies it, putting it in to the **Manual** mode.

The handles relate to the key in two ways, there is the angle of the handle called **Slope**, and then the length of the handle called **Weight**. These two options combine to produce the final shape of the curve.

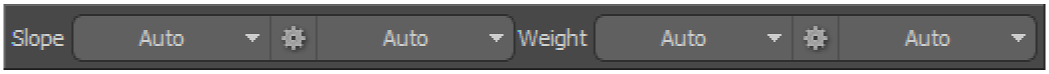

1.0
0.0
0.0 %
50.0 %
100.0 %
150.

Use the following buttons to quickly set the most common **Slope** and **Weight** combinations.

	Sets the selected keys to **Auto** for both slopes and weights.
	Sets the selected keys to **Manual** for both slopes and weights.
Smooth Flat Auto Flat Smooth Flat Auto Flat	Sets the selected keys to one of the following:  • **Auto Flat** - Sets selection to **Auto** flat slopes. • **Smooth Flat** - Adjusts the tangents as the key value approaches surrounding keys to avoid an over shoot. This is the default slope type for new keys. The **Smooth Flat** slope type also gives an improved Interpolation between keys. The image below shows a comparison with **Auto Smooth** and **Auto Flat** slope types in the background.
	Sets the selected keys to **Flat** with **Auto** weights.
	Sets the selected keys to **Linear** with additional sub-menu options (right-click) for **Linear In** and **Linear Out**.
	Sets the selected keys to **Stepped Slope** with **Auto Weight**.

The next group of buttons gives full control over both handles for each key - the incoming handle and the outgoing handle. You can think of the buttons representing the left side handle and the right side handle, where the setting is directed to each side, independent of the **Slope** and **Weight**. If the handle is unified, meaning adjusting one side moves the other in a straight line, then the outgoing settings are grayed out. The option gear in the middle allows you to "break" the handles or "unify" them, and to zero out the angle that makes them perfectly horizontal again. There are a number of settings under the pop-up menu when clicking any of the buttons.

Slope Controls



Manual	Provides full control over **Slope** (handle angle).
Auto	Automatically adjusts **Slope** to produce a smooth ease-in and ease-out from the previous key value to the next.
Linear In	Angles the **Incoming Slope** to point toward the previous key, producing a straight line between keys.
Linear Out	Angles the **Outgoing Slope** to point toward the next key, producing a straight line between keys.
Flat	Sets the **Slope** to 0 degrees flat, regardless of the positions of the previous and next keys.
Auto Flat	Acts just like **Auto** except when the incoming or outgoing keys match the value, then it acts like **Flat**.
Stepped	Eliminates the **Slope** and **Weight** altogether, retaining the previous key value up until a new key is introduced.

Weight Controls

Manual	Provides full control over **Weight** (handle length).
Auto	Automatically adjusts **Weight** to produce a smooth ease-in and ease-out from the previous key value to the next.

The following buttons also allow you to quickly break or unify the handles **Slope** and **Weight** in a single stroke. The color of the key itself represents the state of the two handles: unified handles draw as a white square while broken handles draw as a magenta square.

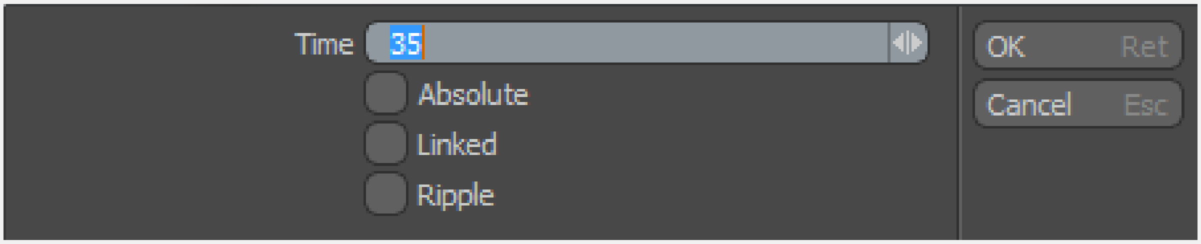
	Breaks the key handles for adjusting **Slope** and **Weight**
	Unifies the handles for adjusting **Slope** and **Weight**.

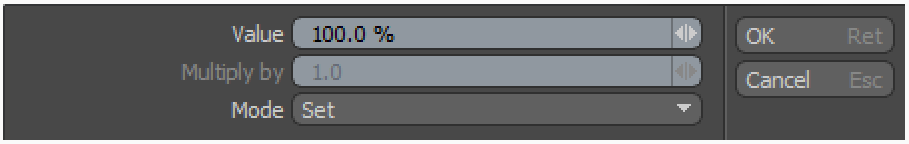
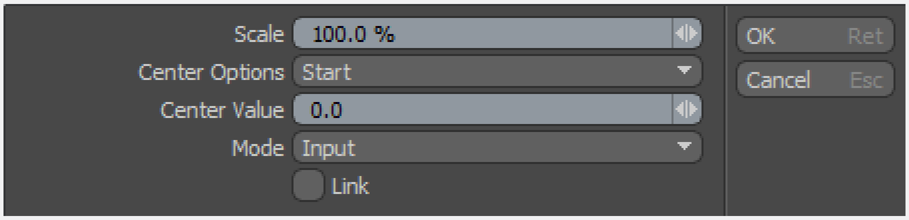
Managing Keys

The **Graph Editor** also offers useful functions for working with key values. Working with a clipboard buffer, you can move keys around and even move them from one channel to another. The **Paste** command simply adds the key values to the target, the **Insert** command adds them to the target and moves other keys as necessary, while **Replace** adds the keys, removing any existing keys it might overlap.

✂	Cuts the selected keys from the editor and stores them in a temporary buffer for later access.
⧉	Copies the selected keys from the editor and stores them in a temporary buffer for later access.
📋	Pastes the keys stored in the buffer.

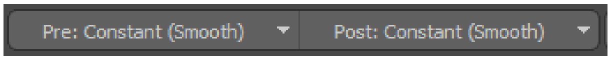
The following buttons open dialog boxes that allow for easy gang editing of multiple selected values.

↔	Opens the **Set Frame** dialog panel for adjusting the time position (not directly applicable to gradients).  Time 35 ☐ Absolute ☐ Linked ☐ Ripple • **Time** - Determines the position in time to move the selected keys to. • **Absolute** - When enabled, **Absolute** time is used, where a specific time can be defined; when disabled, the time is relative to the current time position. • **Linked** - When enabled **Linked** keys (such as those of the R, G, and B channels) remain linked in the same relative position, even if only a single channel is edited; when disabled, keys can be moved freely. • **Ripple** - When enabled, all the following values of those selected are shifted by the same amount.
↕	Opens the **Set Key Value** dialog panel for adjusting the value of the selected keys.

	 Value 100.0 % Multiply by 1.0 Mode Set • **Value** - Defines the new value that is applied to the selected keys when the mode is set as **Set**; defines what value is added when mode is set to **Add**. • **Multiple by** - Acts as a multiplier of the selected values when mode is set to **Multiply**. • **Mode** - Determines how the values affect the selected keys: **Set**, **Add**, and **Multiply**.
◱	Opens the **Scale Keys** dialog panel for scaling the value of the selected keys.  Scale 100.0 % Center Options Start Center Value 0.0 Mode Input ☐ Link • **Scale** - Defines the amount of scaling applied to the selected keys. • **Center Options** - Determines the center location to scale the keys from- **Start**, **Center**, **End**, **Time**, and **User**. When **Time** or **User** is selected, the **Center Value** can be used to define the specific center. • **Center Value** - Determines the specific center locations when the **Time** or **User** options are selected as the **Center Option**. • **Mode** - Provides two modes to determine which direction gets scaled: **Input** scales the interval (horizontal) while **Output** scales the value (vertical). • **Link** - When enabled, linked keys (such as those of the R, G, and B channels) remain linked in the same relative position, even if only a single channel is edited; when disabled, keys can be moved freely.

Behaviors

The **Behavior** buttons define the incoming and outgoing curves that are outside the area of defined keys. The left button defines the per-behavior and the right button the post-behavior.



Pre: Constant (Smooth)

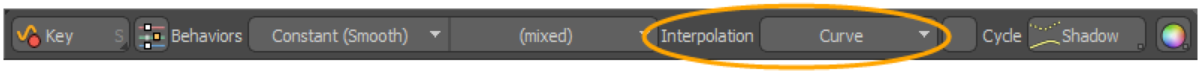
Post: Constant (Smooth)

Constant	Values remain the same. With the **AutoSlope** and **Weight** options, ease-in or out as it transitions to the key values.

Constant (Smooth)	Same as **Constant**, but the keyframe slope values are adjusted to smoothly transition to the **Constant** behavior.
Linear	Values project outward in a linear fashion from the first or last key, providing a constant offset.
Repeat	Values of keys are repeated. When incoming and outgoing values match, a loop can be created.
Oscillate	Values are repeated forwards, and then reversed, like a ping-pong type effect.
Oscillate (Smooth)	Same as **Oscillate** but slope V values are adjusted to smoothly transition to the next oscillation.
Offset Repeat	Values of keys are repeated, but each subsequent repeat is offset by the amount of the last key.
Offset Repeat (Smooth)	Same as **Offset Repeat** but slope values are adjusted to smoothly transition to the repeated sections.
Reset	Values are set to flat 0 (zero) outside the key area.

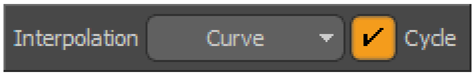
Interpolation

Interpolation controls the way in which all values are calculated between user-defined keyframes and can be a useful shortcut to modifying all the **Slope** and **Weight** values automatically for all keys.



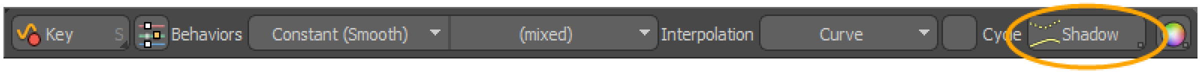
Interpolation Options:

- **Curve** - Allows you to smoothly control the easing-in and -out of the values. This is the default.
- **Linear**- Draws a straight line to each key that produces a jarring, mechanical kind of transition to the keys.
- **Stepped**- Stair-steps the values from one key to the next. This can be useful to see specific key positions or poses for a character without the distraction of the in-between frames.



When the **Cycle** option is enabled, the last keyframe is aligned to match the value of the first keyframe. The original value of the last key is shown as a colored square with a line connecting it to its new value. The last keyframe in the sequence defines the cycle length. The slope and weight values of the last keyframe are also adjusted to match, so the incoming and outgoing curves flow smoothly together. The first key's handle can be broken to allow the slope and weight values to be changed independently.

Shadow



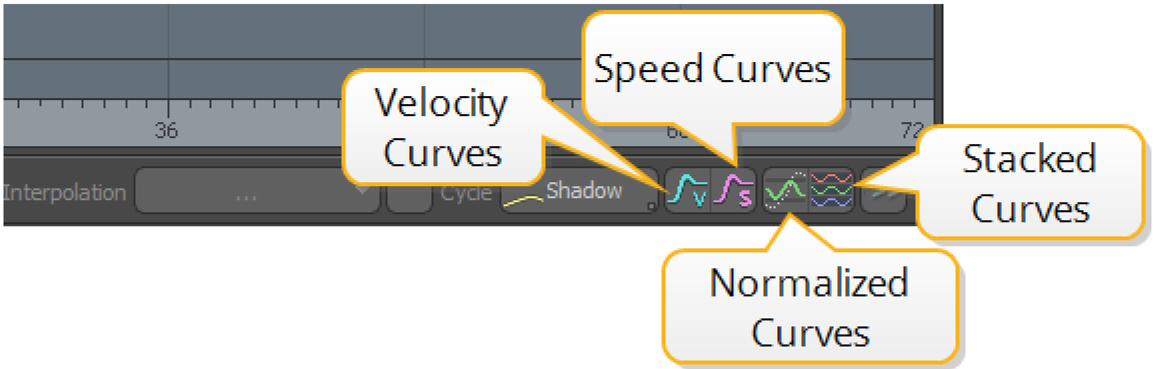
The **Shadow** function works to save alternate versions of a key graph for all currently-selected channels, allowing for convenient and non-destructive editing of the curves. Selecting the **Copy to Shadow** option adds them into a faded background view where you can then continue to adjust the keys visually, and see the current changes relative to the previous shadow values. You can then adjust keys, and scrub or play the Timeline to see the results. Pressing the **Shadow** button opens the pop-up menu with the following option:

Copy to Shadow	Stores the currently-selected channels into the shadow. Shadow channels are saved with the scene and are persistent across Modo sessions for the assigned channels.
Replace with Shadow	Completely replaces the current channel values with the stored shadow values.
Swap	As the name implies, the current values are replaced with the shadow values, and the shadow values are replaced with the current graph values.
Clear Shadow	Deletes the shadow data from the selected channels.

Animation Curve Views

There are multiple curve options you can use to optimize your way of working with animation curves. Adding Velocity and Speed curves for specific channels can help you visualize a specific channel's values, while normalized and stacked curves can make keyframe editing easier.

You can find the controls at the bottom of the Graph Editor.


Velocity Curves
Speed Curves
Stacked Curves
Normalized Curves
36
72
Interpolation
Cycle
Shadow

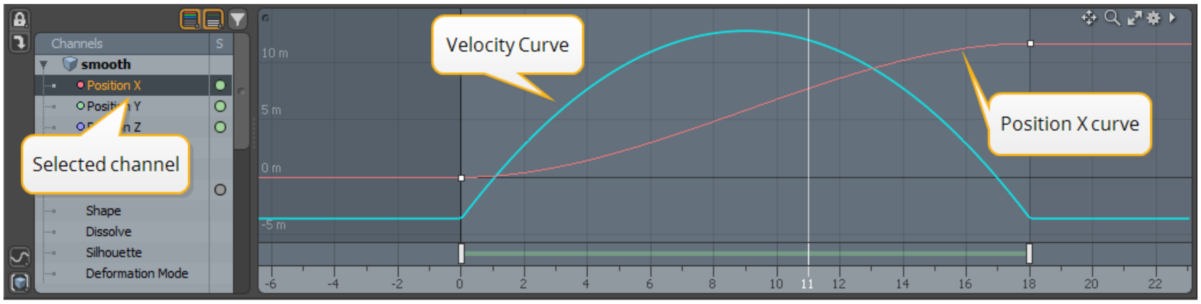
Velocity Curves

Velocity curves represent the selected channel's change in value over time, helping you visualize and smooth out the movement of animated items. You can display a Velocity curve for a single floating point type channel.

To work with Velocity curves:

1. Select a single channel from the list on the left.
2. Click the Velocity Curves button at the bottom right of the Graph Editor.

   A blue Velocity Curve appears in the editor. The image below shows the Velocity curve for the selected **Position X** channel.


Channels
S
smooth
Position X
Position Y
Position Z
Selected channel
Shape
Dissolve
Silhouette
Deformation Mode
Velocity Curve
Position X curve
10 m
5 m
0 m
-5 m
-6
-4
-2
0
2
4
6
8
10
11
12
14
16
18
20
22

Speed Curves

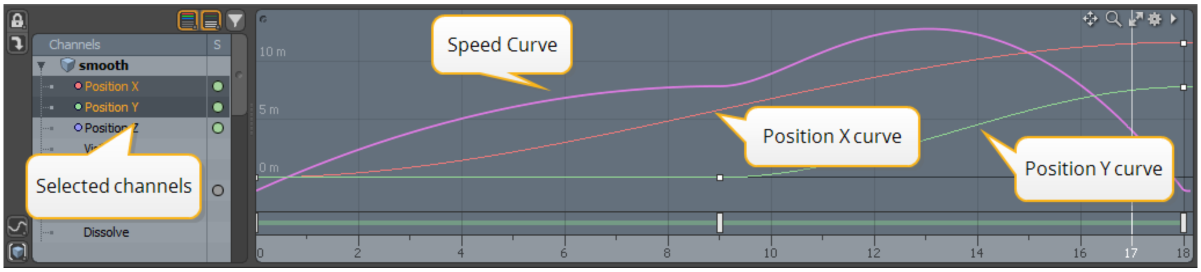
Speed curves are similar to Velocity curves, but they are only drawn when one or more Position channels are selected. They represent the distance the item moves over time.

> **Note:** Speed curves refer to the local space of the item the channels belong to. It does not represent the world speed of the item in the scene if the item is parented to another moving item, has modifiers applied, or the item has other Position transform items.

To enable Speed curves:

1. Select one or more Position channels from the list on the left.
2. Click the Speed Curves button at the bottom right of the graph editor.

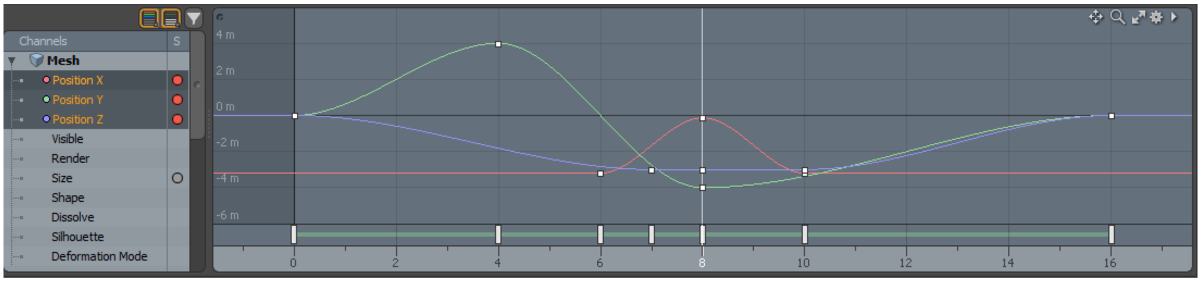
   A purple Speed Curve appears in the editor. The image below shows the Speed curve for the selected **Position X** and **Position Y** channels.


Channels
S
smooth
Position X
Position Y
Position Z
Dissolve
Speed Curve
Position X curve
Position Y curve
Selected channels
10 m
5 m
0 m
0
2
4
6
8
10
12
14
16
17
18

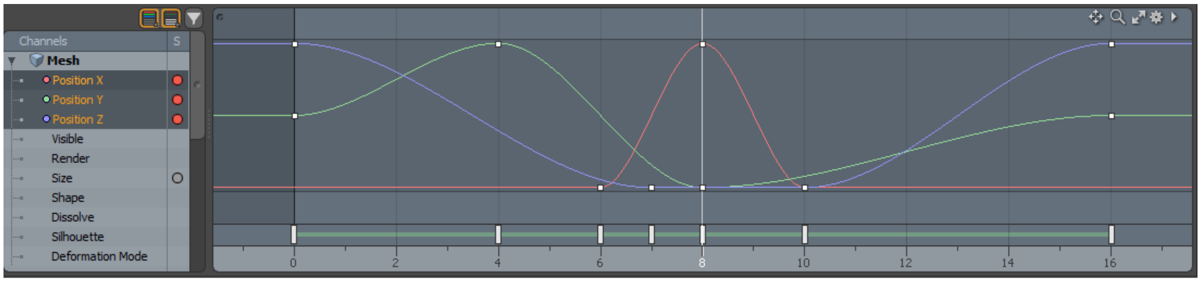
Normalized Curves

When multiple channels are selected in the Graph Editor, the view is scaled to fit the curves. When there is a significant difference in the ranges that these curves are animated over, they can become flattened vertically in the viewport, making editing more difficult and the animation harder to visualize.

Normalized curves allow each of the selected channels to be drawn in the same view space, each being scaled to a common vertical range. This makes editing and visualizing the curves easier.


Channels
S
Mesh
Position X
Position Y
Position Z
Visible
Render
Size
Shape
Dissolve
Silhouette
Deformation Mode
4 m
2 m
0 m
-2 m
-4 m
-6 m
0
2
4
6
8
10
12
14
16

Regular view


Channels
S
Mesh
Position X
Position Y
Position Z
Visible
Render
Size
Shape
Dissolve
Silhouette
Deformation Mode
0
2
4
6
8
10
12
14
16

Normalized view

Note: In this mode the vertical ruler on the left of the Graph Editor is not drawn, since each curve has it's own range.

To enable or disable Normalized curves, click the Normalized Curves button in the lower-right of the Graph Editor.

By default, curves are normalized when selected and keep their scaling during editing. The Normalized Curves button rescales selected curves to fit the normalized range.

Note: Middle-clicking on the background to add keys is disabled when multiple normalized curves are displayed. You can still middle-click a specific curve to insert a key.

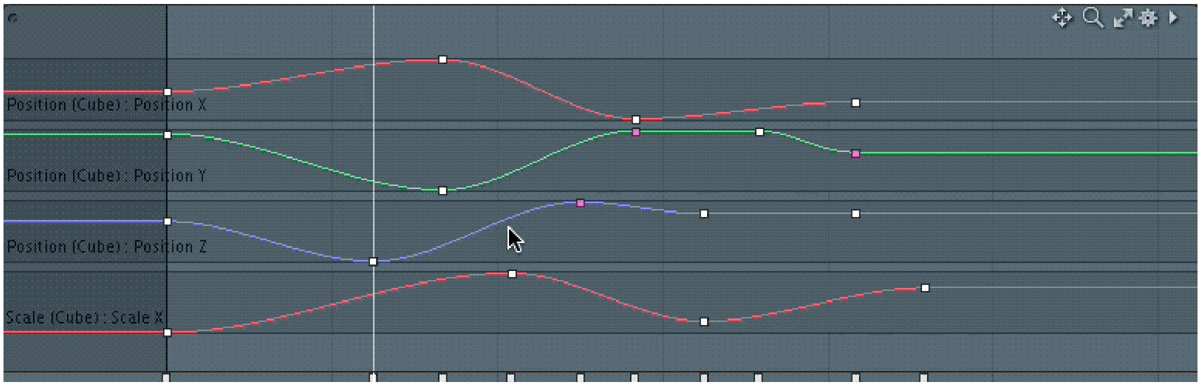
Stacked Curves

Stacked curves are similar to Normalized curves, but each channel is drawn individually in selection order from top to bottom.

To enable or disable Stacked curves, click the Stacked Curves button in the lower-right of the Graph Editor.

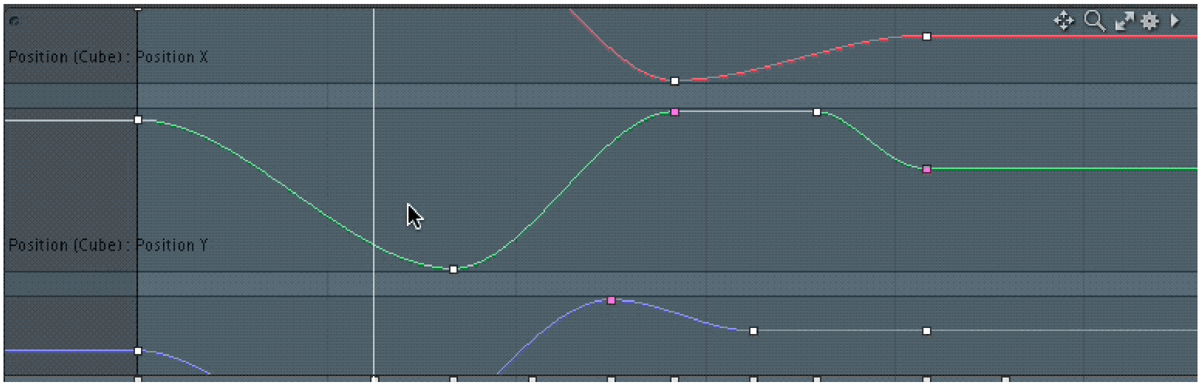
When enabled, the view automatically fits to a maximum of six channels, and further channels are drawn below. You can pan and zoom the viewport to look at these channels.

You can adjust the spacing between the rows by holding **Ctrl** and scrolling with the mouse wheel.


Position (Cube) : Position X
Position (Cube) : Position Y
Position (Cube) : Position Z
Scale (Cube) : Scale X

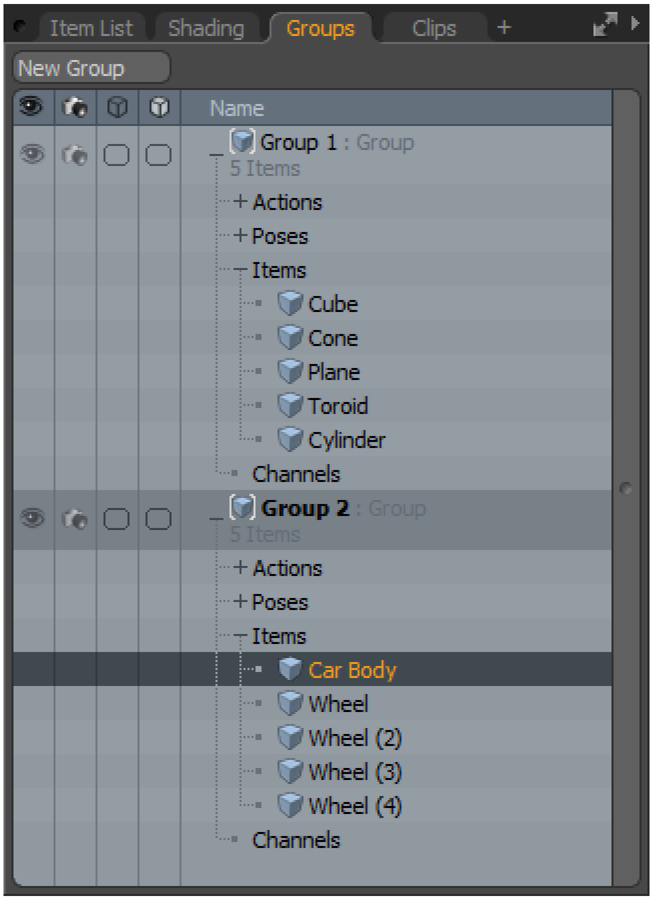
The spacing is reset when no channels are selected.

To step through the stacked rows, hold **Shift** and scroll the mouse wheel.


Position (Cube) : Position X
Position (Cube) : Position Y

Groups Viewport

The **Groups** viewport is a great workflow streamlining tool, allowing you to easily control multiple items and/or channels within a scene. It also provides a number of functionalities to speed up common tasks, such as setting keyframes for complex hierarchies. In addition, many functions of Modo also use groups as specialized containers, assembling together items, channels, values, and settings for a variety of purposes.


Item List
Shading
Groups
Clips
New Group
Name
Group 1 : Group
5 Items
Actions
Poses
Items
Cube
Cone
Plane
Toroid
Cylinder
Channels
Group 2 : Group
5 Items
Actions
Poses
Items
Car Body
Wheel
Wheel (2)
Wheel (3)
Wheel (4)
Channels

Note: Groups used in this context should not be confused with the Group Locator Item, which is a purely organizational construct, and Material groups, which define masks for surface shading as well as to organize the many **Shader Tree** layers.

Working with Groups

To work with groups, you first need to create a new group using the **New Group** button at the top of the viewport. You can name the group, define the type of group desired, and set the mode for how items and/or channels are added automatically. Groups can be made up from any number of scene items, including Mesh Items, cameras, lights, and even locators. Channels can also be part of a group to speed up animation workflows.

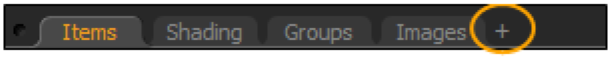
For most of these specialized functions, you are using the specific interface options, rather than directly creating a group and populating it with the desired settings manually. Groups can be used to extend the functionality of the Light Linking and Replicators features.

For more information, see Creating a New Group

Opening the Groups Viewport

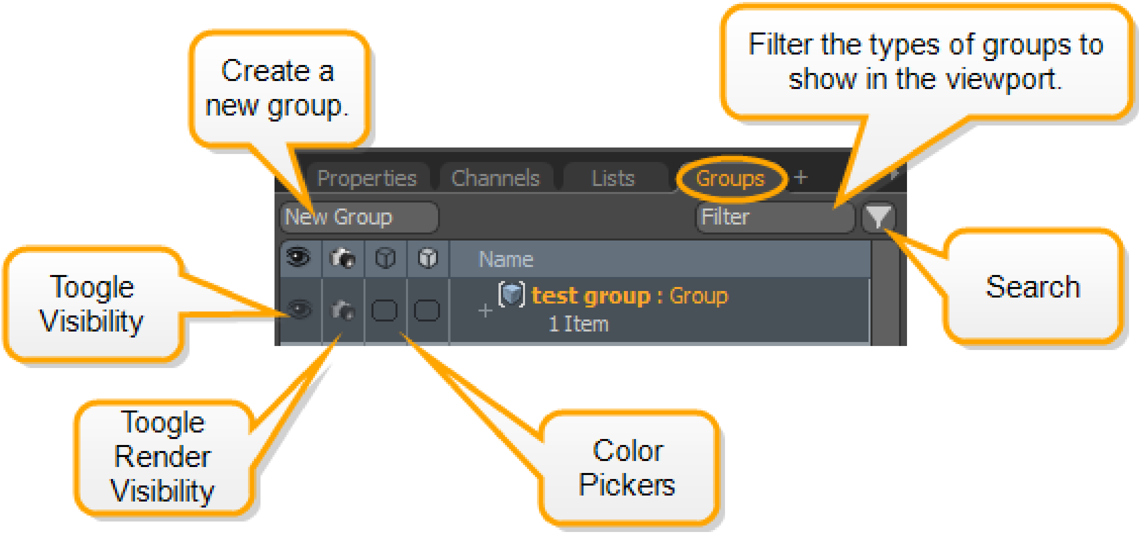
- On the bottom of the right panel, click **Groups**.

> **Tip:** If the **Groups** tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **Utility** > **Groups**.


Items
Shading
Groups
Images
+

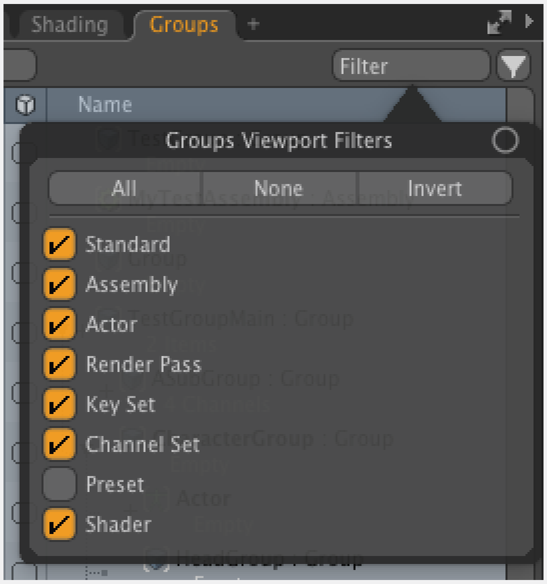
Groups Viewport Controls

The **Groups** viewport itself has a number of controls.

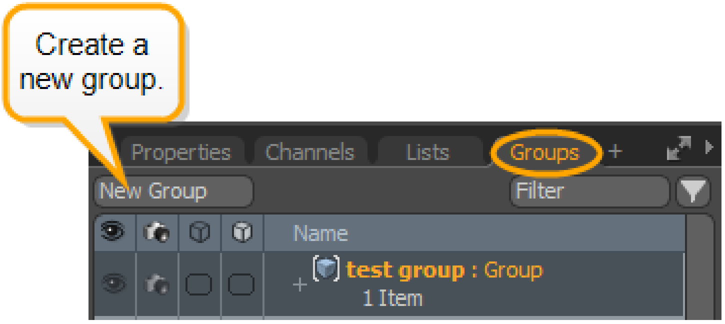

Create a new group.
Filter the types of groups to show in the viewport.
Properties
Channels
Lists
Groups
+
New Group
Filter
Name
test group : Group
1 Item
Toogle Visibility
Search
Toogle Render Visibility
Color Pickers

New Group	Allows you to define a new group. For more information, see Creating a New Group.
👁	Toggles the visibility for items within a group.

	Tip: This works the same way the eye column toggles visibility in the Items (Scenes) List. **Note:** The 👁 toggle visibility control can be overridden by the same settings on the individual item's Properties Panel. For example, if you have a group containing 20 items, of which 5 were listed as **Yes** for visibility in the individual item's properties, then toggling the groups visibility only toggles 15, as the other 5 items' settings would override those of the group. Listing the 5 individual items' settings back to **Default** returns the items to a state where they would follow the group's settings.
📷	Toggles render visibility within a group.
⬡	When clicked, presents a color picker that changes the wireframe color for all items within the group (only visible when the 3D viewport is set to **Wireframe**) and the fourth column changes the fill color for all items in the group (visible in the **Solid**, **Shaded**, **Texture**, and **Texture Shaded** display modes). **Tip:** Changing the display colors of items can be an easy way to visually tell which items are members of which group in scenes with many items. Individually, item settings can be adjusted within the Display Viewport when selected.
Filter	Allows you to filter the types of groups to show in the **Groups** viewport.

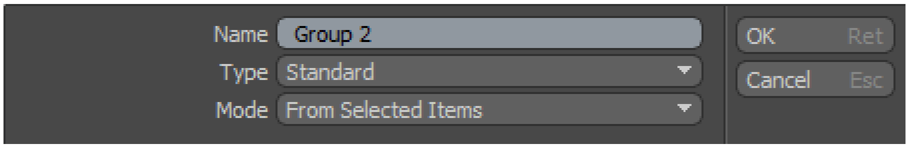
	 Groups Viewport Filters All None Invert ☑ Standard ☑ Assembly ☑ Actor ☑ Render Pass ☑ Key Set ☑ Channel Set ☐ Preset ☑ Shader
Search	When you click the Search ▽ button, a **Search** field appears, allowing you to enter a search string. The search is performed on the names of the groups, their items, and channel members. Clicking the arrow ▼ on the right of the field allows you to do a **Simple Search** or search for a pattern using **Pattern Matching**. Click **Match Case** to do a case-sensitive search. Click **Flat** on the left to only display the list of channels without the hierarchy.

Creating a New Group


Create a new group.

Pressing the **New Group** button near the top of the viewport opens the **New Group** dialog, which allows you to define the initial group. Though a variety of groups can be generated, you most likely want to use the specific UI controls located elsewhere in Modo for generating specialized groups.

When in **Item** mode, additional items can be added to a group by dragging and dropping them; simply select the item in the 3D viewport you wish to add then click-and-drag it to the group window before releasing it over the group to which you want to add the items. An orange highlight appears in the list to denote where the item lands when dropped. Channels can be added in the same fashion from the Channels Viewport.



Name Group 2
Type Standard
Mode From Selected Items

- **Name** - this text input field allows you to define the desired name of the group when generated.
- **Type** - allows you to determine the type of group generated. Types available are described below:
  - **Standard** - used for general groupings of items and for collecting together items for Replicators or light linking.
  - **Assembly** - used for collections of nodes within the Schematic Viewport, used to organize and compartmentalize special functions.
  - **Actor** - used generally for collections of the multiple elements that make up a single character in a scene, but they can be collections of any type of item and aid in animation keyframing. Additionally, they allow some specific functions related to the Timeline.
  - **Render Passes** - used to collect channel settings together that allow you to vary the values in a single pass. The resulting passes are rendered in succession when using the **Render Passes** option.
  - **Key Sets** - used to collect together various channels so keyframes can be assigned simultaneously.
  - **Channel Sets** - used for collections of channels that can be viewed and modified directly in the 3D viewport as an overlay.
  - **Preset** - created when working with presets - collecting the items into a container where their associated settings can be stored. You can define presets using the various Presets options found throughout Modo.
  - **Shader** - when assigning Item Shaders to multiple items, a shader group is generated to store the associated values.
- **Mode** - determines how items or channels are initially added to the generated group. Additional selections can be added and removed at any time:
  - **Empty** - generates a new group with no items or channels.
  - **From Selected Items** - generates a group that includes any items that are currently selected in the scene.
  - **From Selected Channels** - generates a group that includes any channels that are currently selected in the scene.
  - **From Item Edits** - generates a group that includes any items that have been edited from their default scene action state.
  - **From Channel Edits** -generates a group that includes any channels that have been edited from their default scene action state.

Channels in Groups

Groups can contain any number of individual channels from any combination of items in the scene. These individual channels can also be members of as many different groups as you want. If the visibility of a group containing channels is turned off, then those channels are hidden in the various channel lists, that is to say, the main Channels Viewport, Graph Editor and channel links viewports, though properties are not affected by this setting. If a group containing channels is locked, then the channels cannot be edited anywhere, including the channel list, Properties Panel, or interactively in a 3D viewport with the transform tools.

Group Parenting

Groups can be parented together, similar to 3D items; just drag a child group and drop it onto a parent item, working the same way parenting is performed in the **Items** list. Instead of inheriting transforms from their parents though, they inherit the group settings from their parent groups. Groups do not need to have any member items or channels either. They can be used as organizational tools. An example would be multiple groups, each containing many items. If parented to an empty group, and the visibility is switched off on that group, then all the child groups and, hence their member items, are hidden in the 3D views.

Group visibility and render channels can also override in the same way as their counterparts on Mesh Items. So in the above example, setting one of the child group's visibility channel to 'Yes' overrides the parent setting making all members of the group visible. Conversely if the parent group visibility is set to show member items, then setting a child's visibility channel to 'No' overrides that and hide its members. All other children of the parent group still get their settings from the parent group. This also applies to the group's 'Render', 'Selection' and 'Locking' settings available by the right-click context menu.

Group Drawing Settings

The **Wireframe** and **Fill** columns of the **Groups** viewport allow easy access to basic display settings. However, groups also support the full custom drawing settings available for 3D items. If you look in the Display Viewport with a group selected, you'll see an **Add Group Draw Options** button, which, when clicked, displays the custom drawing channels used for 3D items. (If you've already set a wire or fill color for the group the custom drawing channels are already available.) The difference with setting these channels on a group, as opposed to individual items, is that the group settings apply themselves to all items within the group.

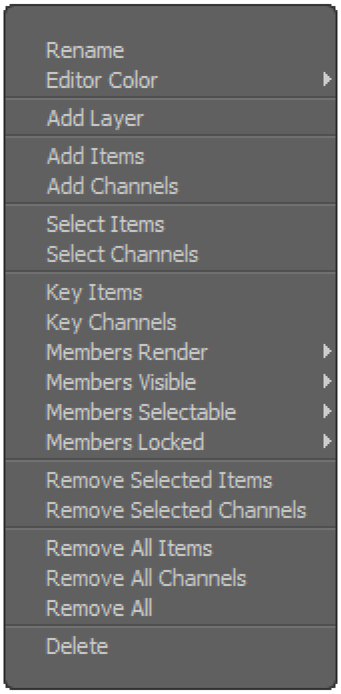
Group custom drawing is layered from parent groups to child groups, so settings on parent groups override any settings on child groups. In order for settings on child groups to be visible, custom drawing on any parent groups needs to be disabled in the display properties viewport by switching off the **Enable** channel. It is not currently possible for child group custom drawing to override parent settings in the same way as the visibility settings for example.

For all settings, the order in which the group appears in the **Groups** viewport is also taken into account. Groups are searched from the top down. When an item is found in a group hierarchy Modo looks at the settings on the group containing the item. If that setting is **Default** it looks up the group hierarchy for a non-default setting. If one is found, then that setting is used, otherwise it continues to look through the groups in the viewport from the top down.

Since multiple groups can contain the same items, this can be used to very quickly change settings. For example, if two groups contain the same items, one is set to lock member items and make them un-selectable, and has it's custom drawing set to a dark color to indicate they can't be changed. The other group has the inverse settings applied. Just by dragging either group above the other in the viewport, it changes the settings applied to the items in the groups. For example, if the locked group is top-most, then all the items are drawn dark, and are un-editable and not selectable, otherwise the items are drawn in the other color, and can be edited and selected.

Contextual Menu

Right-clicking a group title presents you with a number of options.

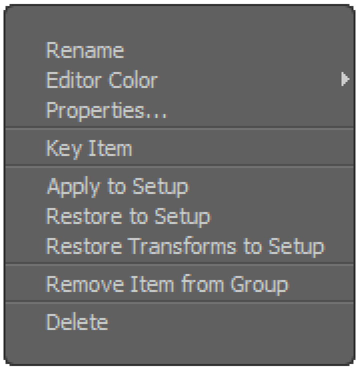

Rename
Editor Color
Add Layer
Add Items
Add Channels
Select Items
Select Channels
Key Items
Key Channels
Members Render
Members Visible
Members Selectable
Members Locked
Remove Selected Items
Remove Selected Channels
Remove All Items
Remove All Channels
Remove All
Delete

- **Rename** - allows you to rename the highlighted group.
- **Editor Color** - provides the means to colorize group layers, to help visually call out a specific group, mostly for organizational purposes.

- **Add Layer** - adds additional layers for the group, such as those used in Actions and Passes. The **Transfer** option copies any scene action state values for the related element when enabled.
- **Add Items/Channels** - adds all the currently-selected items from the item list or 3D viewport, or channels from the channels viewport, to the selected group.
- **Select Items/Channels** - selects (through **Items Mode**) all the associated items/channels within the group.
- **Key Items/Channels** - adds a keyframe at the current Timeline position for all associated items or channels within the group.
- **Members Render** - sets whether group members are visible in rendered images: **Default** respects item list visibility settings (or parent group settings if applicable), **Yes** enables member items' visibility in rendered images, and **No** disables member items' render visibility. When used in conjunction with group parenting, child group settings override parent settings.
- **Members Visible** - sets whether group members are visible in 3D Viewports: **Default** respects item list visibility settings (or parent group settings if applicable), **Yes** enables member items' visibility in 3D Viewports, and **No** disables member items' visibility in 3D Viewports. When used in conjunction with group parenting, child group settings override parent settings.
- **Members Selectable** - sets whether group members are selectable in 3D Viewports: **Default** respects item list settings (or parent group settings if applicable), **Yes** enables member items as selectable in 3D Viewports, and **No** disables selecting of member items in 3D Viewports. When used in conjunction with group parenting, child group settings override parent settings.
- **Members Locked** - sets whether group members are locked from property changes: **Default** respects item list settings (or parent group settings if applicable), **Yes** setting enables member property modifications, and **No** disables member property modifications. When used in conjunction with group parenting, child group settings override parent settings.
- **Remove Selected Items** - select an item in the 3D viewport or **Ctrl**+click to select multiple items and use the **Remove Selected Items** command to remove the selected items from the group.
- **Remove Selected Channels** - select a channel in the channel viewport or **Ctrl**+click to select multiple channels and use the **Remove Selected Channels** command to remove the selected channels from the group.
- **Remove All Items** - removes all items from the group.
- **Remove All Channels** - removes all channels from the group.
- **Remove All** - removes everything (all items and channels) from the group.
- **Delete** - deletes the group, removing it and its contents, from the **Group** viewport.

Items Contextual Menu

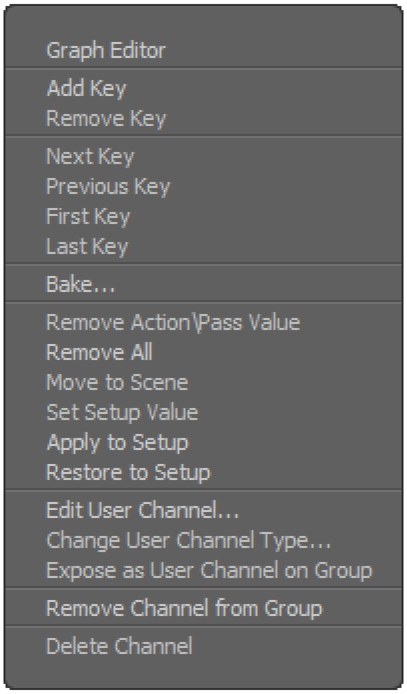
Right-clicking on any item within the group presents item-specific options:


Rename
Editor Color
Properties...
Key Item
Apply to Setup
Restore to Setup
Restore Transforms to Setup
Remove Item from Group
Delete

- **Rename** - allows you to rename the highlighted item.
- **Editor Color** - provides the means to colorize item layers, to help visually call out a specific layer (visible in the **Items** list as well), mostly for organizational purposes.
- **Properties** - opens a **Properties** dialog for quickly editing item attributes.
- **Key Item** - assigns a keyframe at the current time for all animated channels associated to the item.
- **Apply to Setup** - takes the current channel values, associated to the selected item, and moves them to the **Setup** action state.
- **Restore to Setup** - replaces the current channel values, associated to the selected item, with the setting from the **Setup** action state.
- **Restore Transform to Setup** - replaces the current channel values with the setting from the **Setup** action state for only the transform (position, rotation, and scale) values.
- **Remove Item from Group** - removes the item from the related group, but the item remains in the scene.
- **Delete** - removes the item from the scene entirely.

Channels Contextual Menu

Right-clicking on any channel within the group presents you with some channel-specific options:


Graph Editor
Add Key
Remove Key
Next Key
Previous Key
First Key
Last Key
Bake...
Remove Action\Pass Value
Remove All
Move to Scene
Set Setup Value
Apply to Setup
Restore to Setup
Edit User Channel...
Change User Channel Type...
Expose as User Channel on Group
Remove Channel from Group
Delete Channel

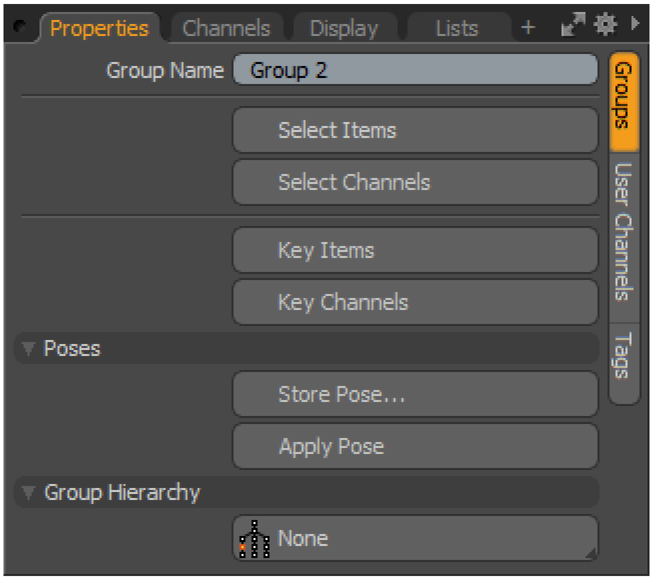
- **Graph Editor** - opens the selected channel in the Graph Editor for further refinement.
- **Add Key** - sets a keyframe at the current time with the current value.
- **Remove Key** - removes the keyframe at the current time.
- **Next/Previous/First/Last Key** - moves the Timeline position to the selected location.
- **Bake** - for a modifier-driven channel, this option opens a dialog allowing you to convert the procedural values to actual keyframes at every frame and disconnect the modifier(s). This is useful when exporting animation or network rendering.
- **Remove Action/Pass Value** - removes the current value from the current action/pass.
- **Remove All** - removes all keyframes and returns to the default (unmodified) value.
- **Move to Scene** - assigns the current channel value(s) to the scene. This is used to move a value inadvertently assigned while in a Render Pass or Actor/Action/Pose.
- **Set Setup Value** -
- **Apply to Setup** - takes the current channel values associated to the selected item and moves them to the **Setup** action state.
- **Restore to Setup** - replaces the current value of the selected channel(s) and replaces it with the **Setup** action state value.
- **Edit User Channel** - opens a dialog that allows you to modify the settings of the Users Channel, such as **Name**, **Default Value**, and the **Minimum** and **Maximum** values.
- **Change User Channel Type** - opens a dialog to change the user channel Type. The **Type** determines the formatting of the resulting value, such as a Scalar, Float, or Boolean.
- **Expose as User Channel on Group** - when making presets for surfacing, often times the author only wants to show certain channels for you to edit, instead of giving them everything. This command allows

preset creators to control which channels are exposed. By selecting specific **Shader Tree** item channels, the **Expose as User Channel on Group** command links the channels to properly-defined user channels on the Material Group Item, where they can be edited directly (under the **User Channels** sub-tab). The context menu for the Material Group Item allows authors to also **Lock** the preset, making it easier to direct the preset end-user to the appropriate controls.

- **Remove Channel From Group** - removes the selected channel from the group.

Keyframing and Selecting Group Members

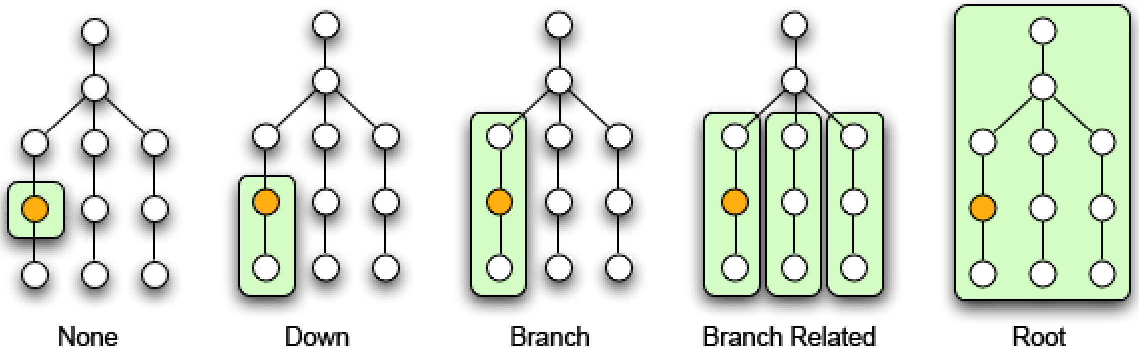
Groups can also be used to help with keyframing workflows, as well as with item selections. On the **Groups Properties** viewport there are buttons to select, and key items and channels within the group (also available from the right-click context menu).



Group Name Group 2

When keying Group Items, only channels belonging to the items that are already keyed are affected. Keying channels within a group adds a key at the current time to each member channel, regardless of whether they have been keyed already or not. This becomes more powerful with parented groups and the use of the Group Hierarchy buttons, also displayed on the **Groups Properties** form. By default, only the contents of the currently-selected groups are keyed (or selected), but this behavior can be changed with the Hierarchy-type buttons:

- **None** - the default; this option only keys the contents of the current group(s).
- **Down** - keys the contents of the current group(s) and all child groups.
- **Branch** - keys the contents of the hierarchy branch that the current group resides in.
- **Branch Related** - as above, but also keys the branches related to the current one, that is to say, branches that share a common parent. (An example might be fingers on a hand.)
- **Root** - keys the contents of the entire hierarchy that the group belongs to.



Light Linking

When a group contains one or more light items (directional, spot, point, or any of the other light types), they can also be used to control the lights overall affect on shading on a surface-by-surface basis, using a shader in the **Shader Tree** with the light linking function.

Replicators

Multiple items can act as base prototypes for Replicators when set into a group, allowing for greater variety in replicated objects. To use multiple Mesh Items as prototypes for a single Replicator, first create a group that contains all the different mesh layers. Once a Replicator has been created, the group name appears in the Replicator's **Prototype** dropdown list. When specified, the Replicator randomly grabs objects from the group for each **Replica**.

Preset Browser

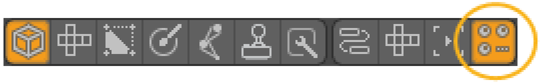
The Preset Browser is a visual library of pre-defined objects, surfaces, kits, and settings available in Modo. Modo's share site presets and some Modo kits are stored in the cloud on a shared site for easy access. These presets are organized and categorized for quick visual browsing. Basic sets of Cloud Assets, Fusion Assets, Templates, and pFusion Assets are available when you install Modo.

You can create your own presets and add them to the Preset Browser or download shared assets from the Foundry community website.

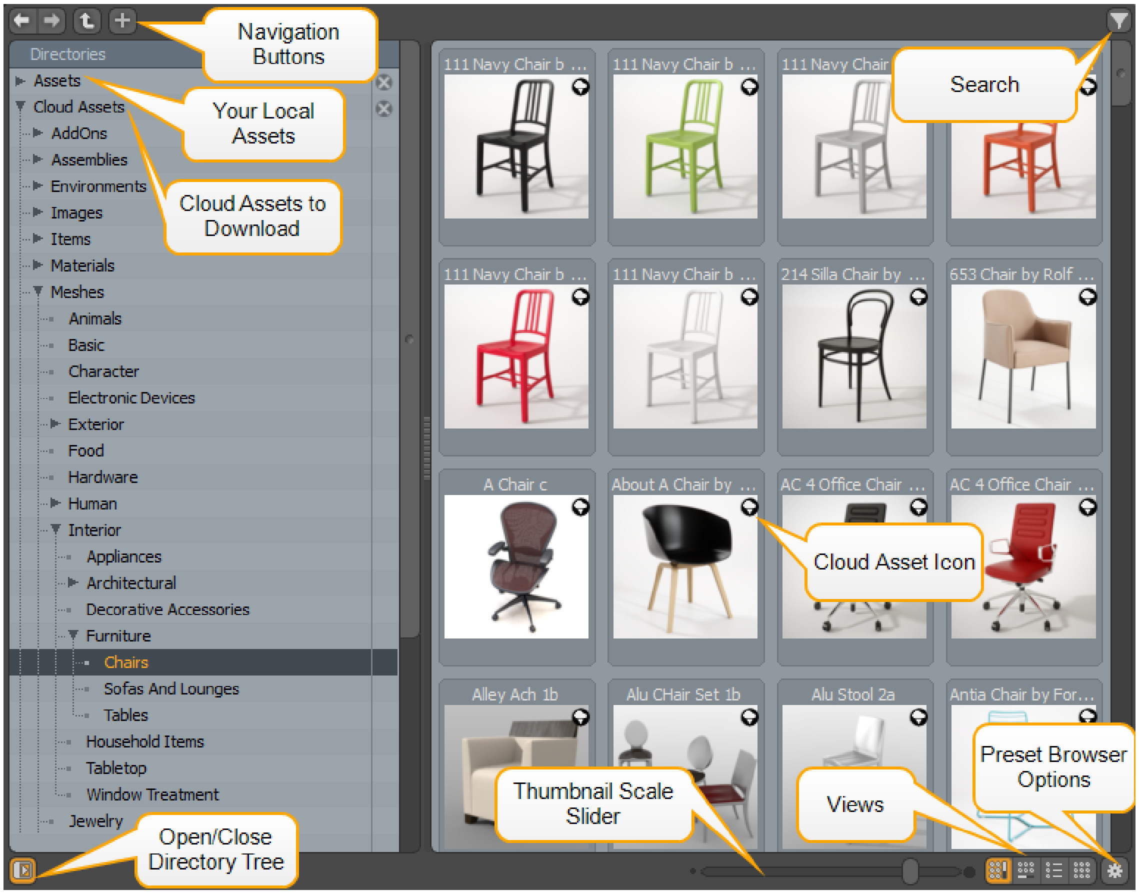
Note: If you require direct downloads for the Modo 11 Content packages, links are provided for all operating systems from our Support Portal. For details, see Q100062: Help with the "Standard Content Wasn't Found" message.
As of Modo 12, all assets are available from the cloud and are accessible from the Preset Browser.

To open the Preset Browser:

- Click the Preset Browser palette icon found on the top-left corner of the **Model** layout.

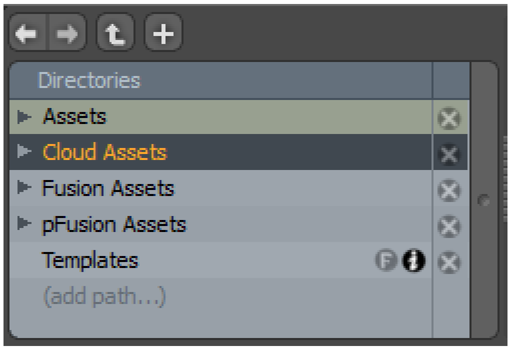


- Alternatively, press **F6** while working in any layout to open the pop-up palette.
- The Preset Browser is also found on the bottom panel of the **Layout** view. Under the main menu, click the **Layout** tab and on the bottom of the view, open the **Items** tab, and expand **Cloud Assets**.


Navigation Buttons
Your Local Assets
Cloud Assets to Download
Search
Cloud Asset Icon
Thumbnail Scale Slider
Views
Preset Browser Options
Open/Close Directory Tree
Directories
Assets
Cloud Assets
AddOns
Assemblies
Environments
Images
Items
Materials
Meshes
Animals
Basic
Character
Electronic Devices
Exterior
Food
Hardware
Human
Interior
Appliances
Architectural
Decorative Accessories
Furniture
Chairs
Sofas And Lounges
Tables
Household Items
Tabletop
Window Treatment
Jewelry
111 Navy Chair b ...
111 Navy Chair b ...
111 Navy Chair
111 Navy Chair b ...
111 Navy Chair b ...
214 Silla Chair by ...
653 Chair by Rolf ...
A Chair c
About A Chair by ...
AC 4 Office Chair ...
AC 4 Office Chair ...
Alley Ach 1b
Alu CHair Set 1b
Alu Stool 2a
Antia Chair by For...

Navigating the Preset Browser

The Preset Browser uses a familiar tree structure of nested folders.



Browsing the Directory Tree

The directory tree is located on the leftmost column and contains a list folder names of the grouped presets. For instance, in the case of the **Assets** > **Materials** folder, there are a number of sub-folders that organize the material presets by finish.

Tip: If the Directory tree is not displayed, you can make it visible by left-clicking and dragging on the inside edge of the left side of the viewport (over the tiny row of stacked squares). The cursor changes to a small arrow when active. See Collapsing the Directory Tree.

To expand or collapse any sub-folder within the tree:

- Left-click the triangle to the left of each folder name.

To display the content of a folder in the main browsing window:

- Click the folder.

To add a new directory:

- Click **(add path...)**, navigate to an appropriate directory, and then click **Select Folder**.

The Preset Browser displays the name of that folder in the navigation tree. Use this custom directory to save your own project assets to.

To remove parent directories from the tree:

You can remove parent directories displayed in your Preset Browser. Assets are not removed from your local drive.

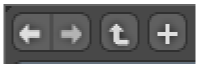
- Click the ☒ icon on the column to the right of the folder names.

  A confirmation dialog opens to confirm the removal of the parent directory from the Preset Browser navigation panel.

Collapsing the Directory Tree

If you prefer not to use the directory tree, you can collapse it by clicking the Directory Tree ◧ button in the bottom-left corner of the viewport.

When closed, navigate using the buttons in the upper-left of the window:



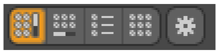
Forward, **Back**, **Up to Parent**, and **Create New Folder** respectively. The **Forward** and **Back** options are only available when folders have been navigated (much like a web browser).

Scaling the Thumbnails

You can scale the thumbnails using the slider at the bottom.

Switching Views

You can set how the Preset Browser displays information using the buttons in the lower right corner.



▥	Thumbnails view with vertical scrolling.
▦	Thumbnails view with horizontal scrolling.
☰	List view, showing assets in a vertically scrolling list.
▩	Thumbnail grid view. You can arrange the thumbnails arbitrarily.

Adding a Preset Directory

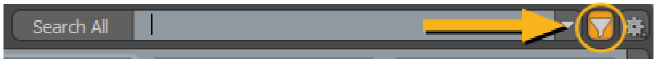
You can customize the Preset Browser. This is useful if you want to create a repository for project specific presets.

To create a preset directory:

1. Navigate to the Preset Browser.
2. Click **(add path...)**, navigate to an appropriate directory, and then click **Select Folder**.

Searching for Presets

You can search for specific presets by Sorting or by using the Search feature. Search is located on the top-right of the preset interface. Click the Search ▽ button to open the search string input field. Type a term into the field to see all presets that contain the term in either the preset name or the preset's keyword tags. By default, the entire hierarchy is searched.



The first matching item is highlighted in the search results. You can scroll through the list using the cursor keys, and then press enter to open the preset.

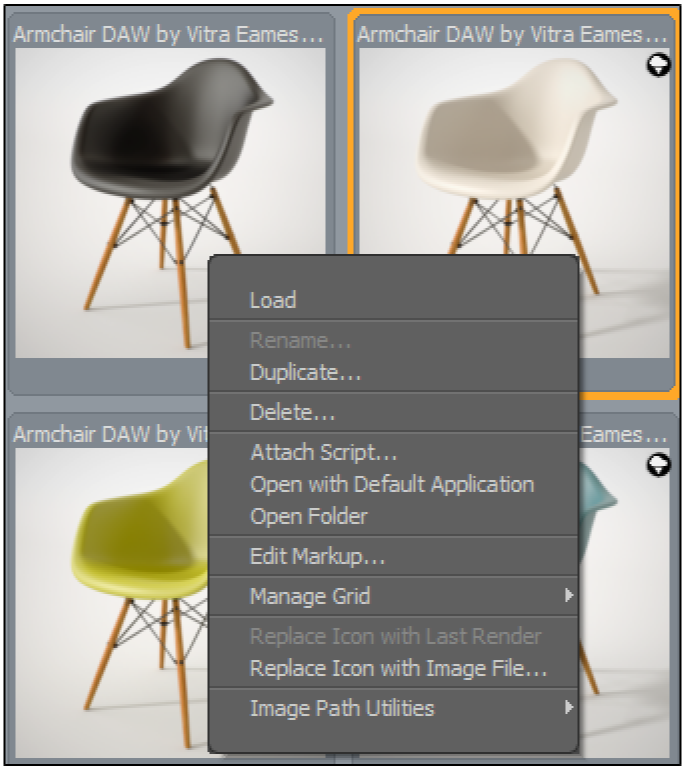
To change the search options:

- In the Search input field, click the More options ▾ button on the right of the search field, and select one of the following options:
  - **Simple Search** - Locates assets using the text entered. Ignores case.
  - **Pattern Matching** - Locates assets using the exact pattern of the text entered.
  - **Match Case** - Matches the case of the search term against the preset name.

Note: If the cursor keys don't work on Mac OS X, try **Alt**+cursor to scroll through the options. This is due to a glitch in Modo's key handlers.

Using the Contextual Menu

Right-click on a preset to open a contextual menu for managing presets, as well as customizing the icon display.


Armchair DAW by Vitra Eames...
Armchair DAW by Vitra Eames...
Armchair DAW by Vit
Eames...
Load
Rename...
Duplicate...
Delete...
Attach Script...
Open with Default Application
Open Folder
Edit Markup...
Manage Grid
Replace Icon with Last Render
Replace Icon with Image File...
Image Path Utilities

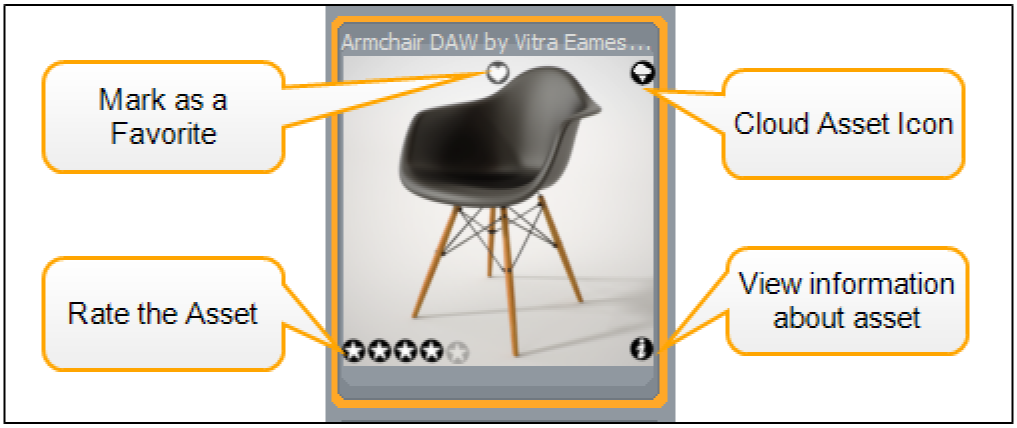
Option	Description
Load	Loads the currently-selected preset into the scene. For mesh presets, this creates a new item centered at Modo's origin. For materials, you need to select a material group in the **Shader Tree** before the material preset loads.
Rename	Applies a new label to the preset.
Duplicate	Opens a dialog requesting a new file name for a new duplicate preset file.
Delete	Removes the currently-selected preset. This also removes the file from the hard drive.
Attach Script	Opens an OS-specific dialog where you can specify a script - an **.lxm** macro file or a Python, Perl, or Lua script. This is useful to when you want an automatic setup for assets added to a scene. The target script attaches itself to the preset. The next time you add the preset, you can run the script by double-clicking it.

	Note: The Script must remain in the same disk location relative to the preset.
Open with Default Application	For non-Modo specific assets, such as image maps, this command uses the OS's default application to open the file for further editing.
Open Folder	Opens the file location in your OS's file browser.
Edit Markup	Opens the **Dir Cache Markup Editor** dialog as a pop-up for editing. For more information, see Editing Preset Metadata.
Manage Grid	When viewing the Preset Browser in Grid mode, you can easily organize the files in a specific order if the **Sorting** option is set to **Manually**. You can then use these options to add grid spacing lanes - additional lanes with which to organize and sort the presets. For more information about Grid mode, see Browser Mode.
Replace Icon with Last Render	Embeds the last **F9** render result into the currently selected preset item.
Replace Icon with Image File	Opens an OS-specific file browser to select a saved image that is embedded into the currently selected preset item.
Image Path Utilities	The Image Path Utilities offer you a way to locate and repair presets with broken image file links (such as might occur when reorganizing the assets): • **Fix Broken Image Paths** - Fixes any broken image paths in the currently selected presets. • **List Broken Image Paths (selected)** - Produces a text report listing broken image paths found in the selected presets. • **List Broken Image Paths (Folder)** - Produces a text report listing broken image paths found under a folder hierarchy. • **Image Use Report (Folder)** - Produces a text report listing all image paths found under a folder hierarchy. • **Image Use Report (selected)** - Produces a text report listing all image paths found in the selected presets.

Reviewing Cloud Asset Information

A cloud icon is displayed on cloud assets. Click this icon to download the asset. After downloading a cloud asset to your local drive, this icon is displayed in gray when you hover your cursor over the asset.

You can mark an asset as a favorite, rate the asset, and review information about the asset.



Tip: You can use the scale slider to change the size of the icons displayed in the Preset Browser. For more information, see Scaling the Thumbnails.

Editing Preset Metadata

The **Preset Browser** allows you to manually add additional markup (metadata) to presets to aid in organization and enhance filtering and searching of presets. This includes the ability to add keywords, comments, star ratings, and favorite presets.

Metadata can be added for your own personal use or to be shared. In addition, you can edit markup for a selected tree directory. All assets within that directory are updated.

The following markup options are available:

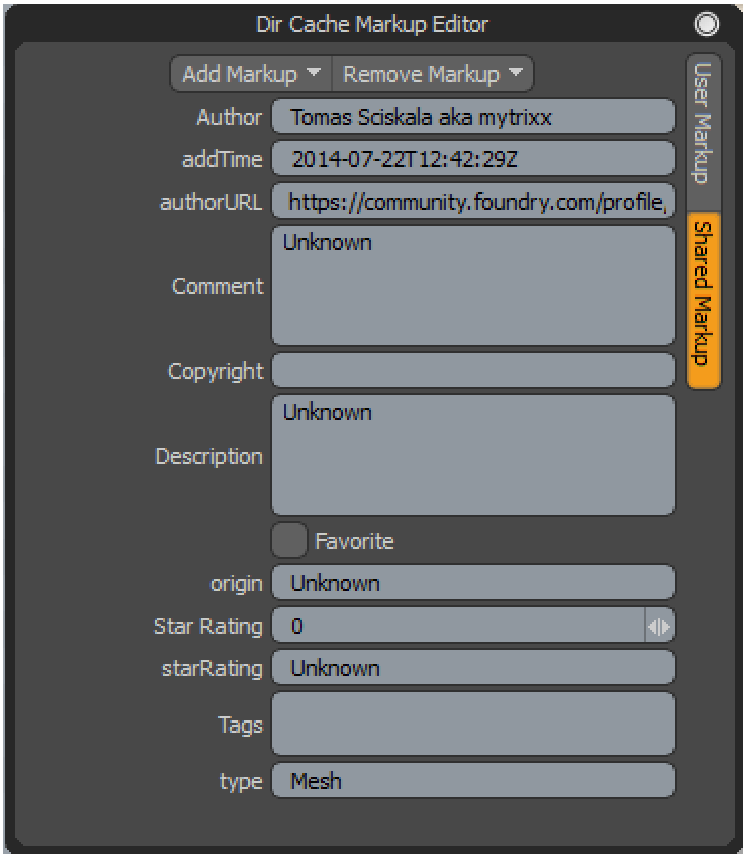
- **User Markup** - Metadata is only saved to your local assets.
- **Shared Markup** - Metadata is shared with others. Your updates are seen by clicking the Info icon for the updated Cloud Asset.

Editing Markup for an Asset

To edit markup for a selected asset:

1. In the **Preset Browser**, right-click on an asset, and select **Edit Markup**...

   The **Dir Cache Markup Editor** displays.



Author: Tomas Sciskala aka mytrixx
addTime: 2014-07-22T12:42:29Z
authorURL: https://community.foundry.com/profile,
Comment: Unknown
Copyright:
Description: Unknown
☐ Favorite
origin: Unknown
Star Rating: 0
starRating: Unknown
Tags:
type: Mesh

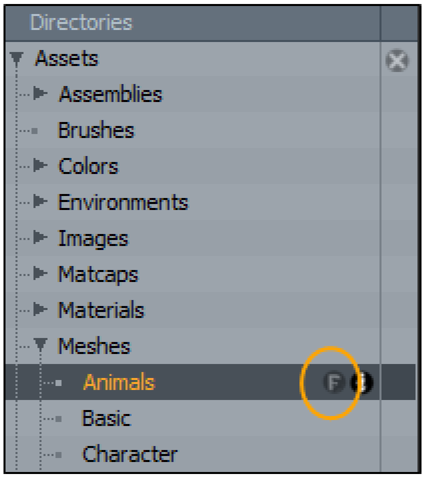
2. Open the **Shared Markup** tab.
3. Add the string, for example **Blue Chair**, into the **Comment** field.
4. Close the **Dir Cache Markup Editor** dialog if it's obscuring the preset.
5. On the updated asset, click the Info icon .

   The **Comment** field displays **Blue Chair**.

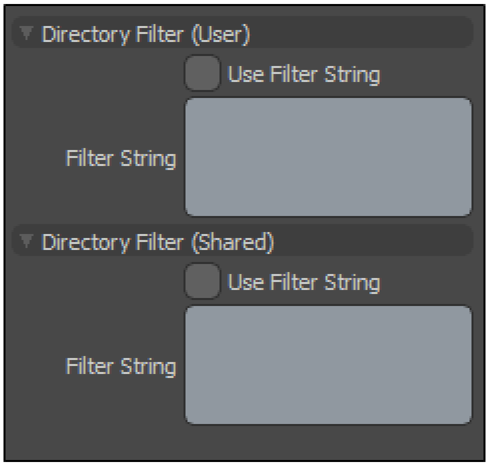
Filtering a Directory

To filter a directory:

1. On the left panel, select a directory you would like to add metadata to, and click the **Filter** icon.


Directories
Assets
Assemblies
Brushes
Colors
Environments
Images
Matcaps
Materials
Meshes
Animals
Basic
Character

The **Dir Cache Markup Editor** dialog displays.


Directory Filter (User)
Use Filter String
Filter String
Directory Filter (Shared)
Use Filter String
Filter String

2. Click **Use Filter String** under **Directory Filter (User)** and/or **Directory Filter (Shared)**.
3. Type markup data into the **Filter String** fields.

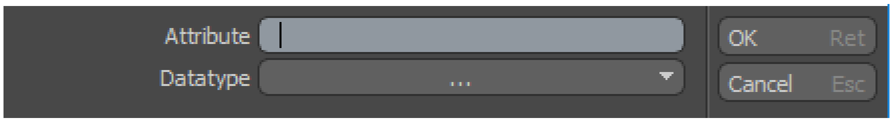
Adding a Custom Markup Attribute

To add a custom markup attribute:

1. In the Preset Browser, right-click on an asset, and select **Edit Markup...**

   The **Dir Cache Markup Editor** dialog displays.
2. At the top of the dialog, select **Add Markup** > **Add Custom Markup...**

   The **Add Custom Markup Attribute** window displays.


Attribute
Datatype
...
OK Ret
Cancel Esc

3. In the **Attribute** field, type the name of your custom markup attribute.

4. In the **Datatype** field, click the drop-down arrow, select a datatype, and click **OK**.

Downloading Presets

You can select which presets you would like to work with and download them onto your local drive.

To download presets:

- Double-click on the desired preset in the **Cloud Asset** folder.
- Drag-and-drop the asset into the 3D (OpenGL) Viewport.
- Drag-and-drop the asset onto one of the viewport tabs on the top-right panel. For more information about viewports, see Understanding Viewports.

  The asset is downloaded to your **Asset** directory. In addition, the cloud icon is displayed in gray on the asset thumbnail to give you a quick visual reference of which assets you have previously downloaded to your local drive.

> **Note:** Modo checks if any of the selected cloud assets are more recent than the presets previously downloaded on your local system. You are prompted with a warning if you have an older version before downloading.

- Modo Kits can be downloaded and installed by double-clicking on the appropriate kit in the **Cloud Assets** > **AddOns** > **Foundry** > **Kits** directory. You must then restart Modo to complete the installation.

For information on how to create your own presets, see Creating Your Own Presets.

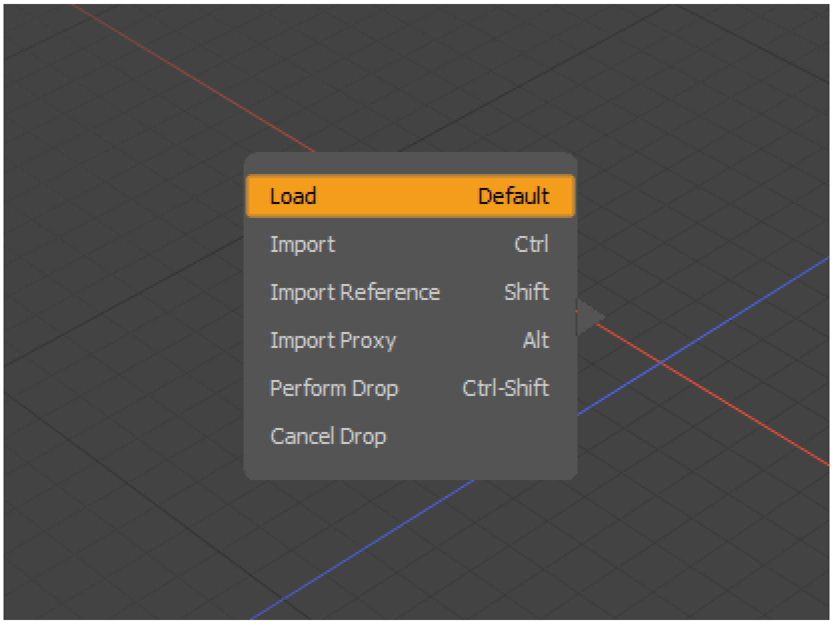
Importing Scenes

You have multiple options to bring a scene (**.lxo** file) from the Preset Browser into Modo.

When you right-click the scene file in the Preset Browser, you can choose from the following options:

- **Load** - Loads the scene independently from any other open scenes.
- **Import** - Imports the scene to the current scene.
- **Import Reference** - Imports the scene as a reference. This means that a link is created between the two scenes. If the referenced scene is modified in any way, for example, an item is added or its surfacing is changed, the referenced elements automatically update in the master scene upon re-opening. For more information on working with reference files, see Import Reference.
- **Import Proxy** - A proxy item is created in the current scene, which links to the imported scene. For more information on proxy items, see Render Proxy.

When you drag a scene to the 3D viewport and keep holding the left-mouse button, a menu appears with the same options. Select the required option and release the mouse button.

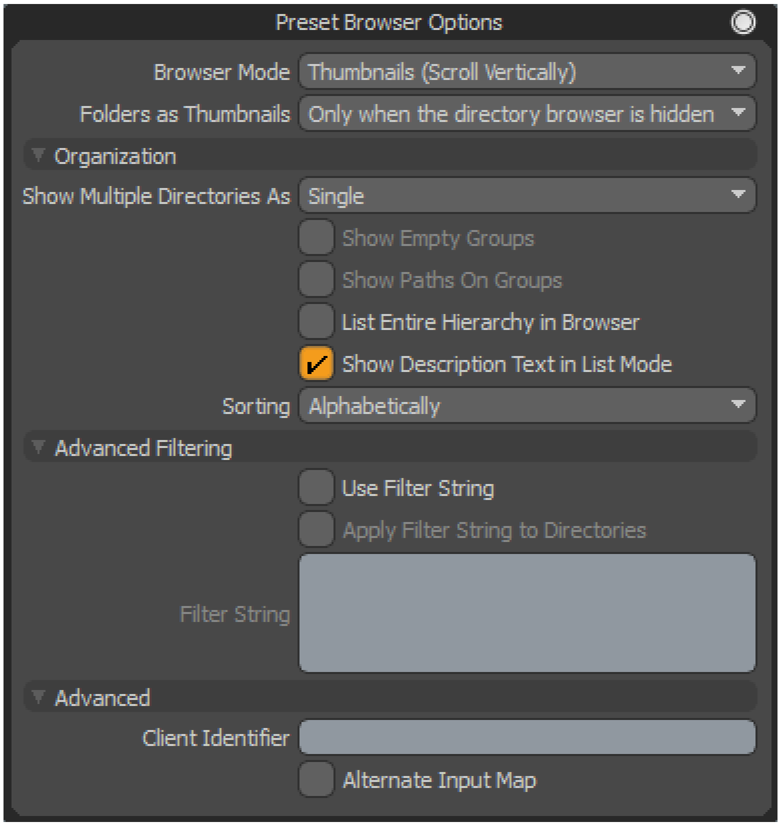

Load Default
Import Ctrl
Import Reference Shift
Import Proxy Alt
Perform Drop Ctrl-Shift
Cancel Drop

You can also use keyboard shortcuts with drag-and-drop:

- **Import** - **Ctrl**
- **Import Reference** - **Shift**
- **Import Proxy** - **Alt**

Preset Browser Options

The gear icon, in the bottom-right corner of the Preset Browser, opens the viewport options dialog, which allows you to further customize the information displayed in the viewport.



Preset Browser Options

Browser Mode: Thumbnails (Scroll Vertically)
Folders as Thumbnails: Only when the directory browser is hidden

Organization

Show Multiple Directories As: Single
☐ Show Empty Groups
☐ Show Paths On Groups
☐ List Entire Hierarchy in Browser
☑ Show Description Text in List Mode
Sorting: Alphabetically

Advanced Filtering

☐ Use Filter String
☐ Apply Filter String to Directories
Filter String:

Advanced

Client Identifier:
☐ Alternate Input Map

Browser Mode	How the presets are displayed in the browser: • **List** - Displays icons next to the name of the preset, and two columns containing a star rating and comment text. You can choose to show/hide one or both of these columns by right-clicking on the column headers. You can also rearrange columns by dragging-and-dropping the column headers. • **Thumbnails (Scroll Vertically)** - Large icon view that scrolls vertically. • **Thumbnails (Scroll Horizontally)** - Large icon view that scrolls horizontally. • **Grid** - Similar to the **Thumbnail** view, but only displays the image thumbnail without the preset name.
Folders as Thumbnails	Controls whether the folders themselves, used to organize the presets, display as thumbnails within the browser area. This is only active when either of the thumbnail options (above) are selected. Folders can be a helpful addition for navigation, more akin to regular OS file navigation.

Organization Options

Show Multiple Directories As	Controls how multiple folder selections are displayed (hold down **Shift** or **Ctrl** to select more than one folder in the **Directories** tree at a time): • **Single** - Displays the contents of single folders only (no multiple folder selections allowed). • **Multiple (Merged)** - Displays the contents of all selected folders simultaneously, as if they were a single folder. • **Multiple (Grouped)** - Same as above, but separates the contents of each folder with a named divider.
Show Empty Groups	Empty folders are displayed in the browser. Files can be dropped into empty groups when they are visible. **Note:** This option is only available when the Show Multiple Directories is set to **Multiple (Grouped)**.
Show Paths on Groups	Displays the full directory hierarchy in the group name in the preset list. This only applies if you are viewing multiple directories using the **Multiple (Grouped)** option.
List Entire Hierarchy In Browser	When this option is enabled, all child folders in the hierarchy are also displayed in the browser without being explicitly selected.
Sorting	Controls the display order of the presets in the browser. • **Alphabetically** - Displays in alphanumerical order. • **Manually** - Allows you to define the order by dragging-and-dropping within the browser (not available for multiple directories as merged). • **Star Rating** - Ordered by the number of stars assigned to a preset. • **File Extension** - Ordered by the preset's file extension, organized alphabetically. • **Modification Time** - Defines the order based on the last time the file was saved (or modified), starting with the most-recent file or directory. • **File Size** - Defines the order based on the actual size of the file as saved on the disc.

Advanced Filtering Options

The Preset Browser adds markup information to presets. Filtering procedurally refines what is visible in the browser, based on a scripting-like annotation

The filter options take into account both the user-defined markup, as well as metadata (information associated with the file, such as name and format).

For more information, see Filtering a Directory.

Use Filter String	Toggles the filtering **on** or **off** for the browser.
Apply Filter String to Directories	When this option is enabled, the filtering is applied to the directory folders as well, otherwise all directories are shown.
Filter String	Enter the data string to filter the elements within the Preset Browser. **Note:** The **Use Filter String** and/or **Apply Filter String to Directory** option must be set before typing a filter string.

Advanced Options

Client Identifier	Allows a **Preset Browser** to have its own unique selection. This is most useful when configuring a **Preset Browser** to be used with a specific tool. For example, this is how the **Polygon Bevel** tool knows which of many possible **Preset Browser** viewports to use (in the case of **Polygon Bevel**, the browser viewport is embedded in the **Tool Properties** viewport). **Note:** Each tool must be specially coded to look for a specific client identifier, so this feature is only really useful to tool developers. For more information, see our SDK documentation.
Alternate Input Map	An advanced setting for developers that controls the number of mouse clicks used to assign a preset to a tool. When enabled, they assign with a single click, and when

	disabled they assign with a double-click (the default behavior). For more information, see our SDK documentation.

Preset Browser Advanced Filtering

In the viewport options, there's a simple multi-line edit field labeled **Filter String**. This allows advanced users to limit what is displayed in the Preset Browser, by testing the markup and metadata. This filter affects the currently-visible files in the thumbnail browser or flat list, and can be set to filter only files, or both files and directories.

The basic input format for the filter is:

```
(NOT class:attribute(options)={value}..{value}) COMBINER
```

You can have as many of these as you like, space-delimited.

- **Parentheses** - you can optionally wrap multiple tests in parentheses for grouping purposes. The COMBINER determines how this test is handled relative against the next test.
- **NOT** - the optional NOT prefix inverts the following test.
- **class** - the class is one of "intrinsic" metadata, user or shared, and determines where the attribute is pulled from.
- **attribute** - the internal name of the attribute to test. Attributes are case-sensitive, and are fairly arbitrary (beyond the fact that they obey internal string naming rules).

The intrinsic class supports the following special properties:

- path: (filepath) The full local path to the file.
- name: (string) The name portion of the path, including the extension.
- ext: (string) The extension of the file, if any.
- size: (memory) The file size on disk. Directories currently return a size of 0.
- isFile: (boolean) True if this is a file; false if this is a directory.
- numFiles: (integer) The total number of recognized files in a directory.
- numDirs: (integer) The total number of sub-directories in a directory.

- numChildren: (integer) The total number of recognized files and sub-directories in a directory.
- modTime: (string) When this file or directory was last modified or created (whichever is more recent).

Some examples of common user and shared markup include:

- isFavorite: (boolean) True if this entry is flagged as a favorite.
- starRating: (integer) 0-5 indicating the star rating assigned by the user.
- tags: (string) Arbitrary strings provided by the user for tagging purposes.
- author: (string) Author of the file.
- copyright: (string) Copyright information.
- desc: (string) Description of the file.

The markup is extensible by the user, and can contain arbitrary attributes.

(options)

These are optional but, if present, must be in parentheses and follow the attribute name. The available options depend on the datatype. The list of options are space-delimited. For strings:

- substring: Search the attribute for the test value as a substring. This is the default.
- pattern: Do pattern matching.
- case: Make the comparison case-sensitive. The default is case-insensitive.
- exact: The string must exactly match the test value. This can be used with case to enforce a case-sensitive compare.

Operator

The operator is a standard mathematical test, although the behavior is a bit different depending on the datatype.

- = (Equal); the value must match exactly. When used with strings, this does pattern matching, a substring search, or an exact match depending on which options are set.
- < (Less Than), > (Greater Than), <= (Less Than or Equal), >= (Greater Than or Equal); basic logic tests.
- On numeric types, this works as you would expect.
- On float numbers, this compares against a very small threshold (epsilon), since you can't directly compare floats for technical reasons.
- On strings, this does the C function strcmp (or the case in-sensitive version, depending on if the case option is set) and tests the result. This is most useful when comparing the modTime attribute against another time to find files newer or older than that time.

If no operator is provided, then the filter matches as long as class:attribute exists on the file or directory.

If the attribute does not exist on the file or directory, and an operator is present (meaning, you're not just doing an existence test), the test is considered a failure, irrespective of the value being tested against.

Value

The value to test against. It must be wrapped in one of the following ways:

- Curly braces {...}: The value string is parsed as a "raw" format value.
- Square braces [...]: The value string is parsed as a "nice" format value.
- Less than/greater than bracing <...>: The string is in the form class:attribute path and represents another attribute to be tested against. The path is optional. If present, it is the full local path to another file or directory in the dir cache; if omitted, then the attribute is pulled from the current entry.

These braces are allowed to nest, in case you need braces inside the search string. If there is an unbalanced curly brace inside of that string, it must be escaped with ` (backtick). If you have to search for a ` (backtick), it needs to be escaped with another ` (backtick). For example, ``(two backticks side-by-side).

Testing Against Non-Existent Attributes

It is also important to note that a value test only happens if the attribute actually exists on the item. This means that if you wanted to check for everything that has the **isFavorite** attribute set to false and you did this:

```
user:isFavorite={false}
```

It would only match entries that have been previously flagged as favorites and then turned off again (since turning it off just reset the attribute to false; it didn't remove it from the markup). Thus, you would need to do this for your filter:

```
user:isFavorite={false} OR NOT user:isFavorite
```

Which would match if the **isFavorite** attribute is false, or if the **isFavorite** attribute does not exist.

Range Testing

The **..** (two periods) syntax can be used to do a range test between the value described above and a second value. The second value after the **..** follows the same formatting rules as the first value. This is only valid with the **=** (equal) operator, since it doesn't make sense with any of the others.

When comparing numbers, values from **class:attribute** must by greater than or equal to the first value and less than or equal to the second value.

When comparing strings, they are both tested against **class:attribute**'s value with **strcmp( )**. (For a case-insensitive version, if the case is set, other string options are ignored.)

For those who don't know, **strcmp()** returns if a string is "less than" (which can generally be thought of as meaning "comes before in sort order") or "greater than" another string. The **class:attribute** value string must test greater than or equal to the first value string, and less than or equal to the second test string. This is primarily useful when testing file times with the **intrinsic:modTime** attribute, allowing you to see if a file was created in between two time ranges.

Combiner

When performing multiple tests, the **AND** and **OR** combiners can be used to decide if they all must match or only some must match. **AND** is optional and is inferred, if not provided. This is most useful when combined with parentheses.

For example, this matches if a Favorite flag is set in the user markup, or if the user markup Star Rating does not exist, or it does exist and is less than 1.

```
        user:isFavorite={true} AND (NOT user:starRating OR user:starRating<{1})
```

Creating Your Own Presets

Your workflow may involve creating similar components each time you start a project, such as a ground plane, a certain environment gradient, or maybe a specific material. Creating your own presets is a great way to streamline a workflow and speed up scene creation, eliminating repetitive tasks so you can concentrate on the important stuff.

Saving Presets

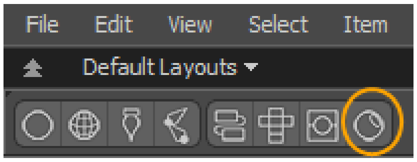
Modo provides different ways to save items in your scene as presets.

- Preset Drag-and-Drop - Drag-and-drop items from the **3D viewport** or from one of the viewport tabs on the top-right panel onto the **Preset Browser**. For more information about viewports, see Understanding Viewports.
- Preset Viewport - Within the viewport tabs, for example **Items** or **Shading**, right-click on an item and select **Presets** > **Save...** or **Presets** > **Save Assembly Preset...** . For more information about viewports, see Understanding Viewports.
- Preset File Menu - Select your items in your scene and click one of the following options: **File** > **Save Item Preset...**, **File** > **Save Assembly Preset...**, or **File** > **Save Environment Preset...**

Note: Workflows vary when creating Presets depending on the type of item you have selected. For more information, see Creating Different Preset Types.

Preset Drag-and-Drop

1. Click the **Preset Browser** palette icon found on the top left corner of the Modo interface.



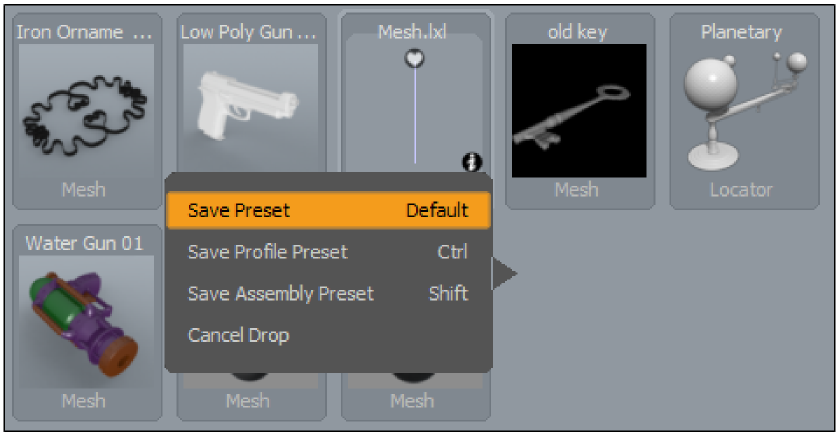
   Alternatively, press **F6** while working in any layout to open the pop-up palette. The Preset Browser is also found on the bottom panel of the **Layout** view.

2. In the **Preset Browser** viewport, expand **Assets** on the left panel and open the directory you would like to save your preset to.

Note: If the directory tree is hidden, drag the widget on the left side of the navigation panel.

3. Drag an item in the 3D viewport into the **Preset Browser**. Alternatively, drag an item from a viewport tab on the right panel into the **Preset Browser**.

Note: By holding your cursor on the Preset Browser a pop-up menu displays. Depending on type of item you have selected, the menu options will vary. The example below uses the Mesh item.


Iron Orname …
Low Poly Gun …
Mesh.lxl
old key
Planetary
Mesh
Mesh
Locator
Water Gun 01
Mesh
Mesh
Mesh
Save Preset Default
Save Profile Preset Ctrl
Save Assembly Preset Shift
Cancel Drop

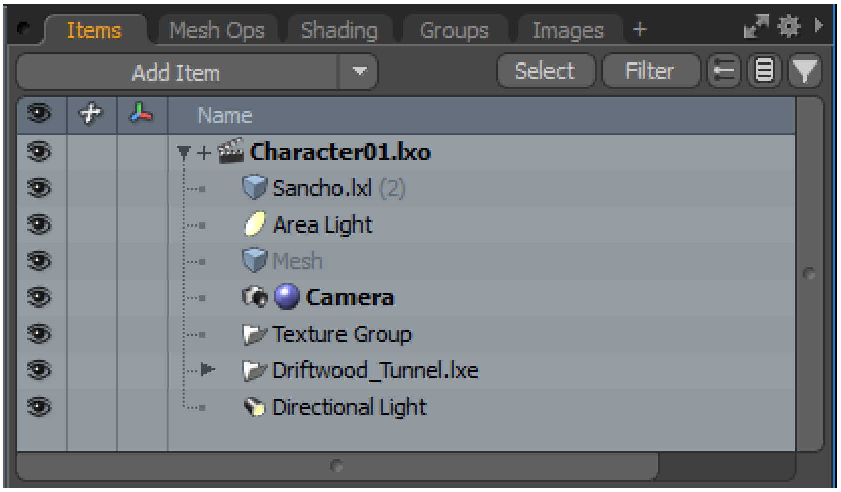
4. Select from the pop-up menu the type of preset you want to save.

The following menu options are available:

- **Save Preset** - Saves a preset made from almost any selected item type. For more information, see Item Presets
- **Save Profile Preset** - Saves a selected curve drawn in Modo as a preset. For more information, see Profile Presets.
- **Save Assembly Preset** - Saves an assembly made in Modo as an preset. For more information, see Assembly Presets.
- **Create Material Preset** - Saves selected Shading items as a preset. For more information, see Material Presets.
- **Cancel Drop** - Cancels the action.

Preset Viewport

The viewport tabs are located on the top-right panel. These viewports can be used to select and save items as presets.


Items
Mesh Ops
Shading
Groups
Images
+
Add Item
Select
Filter
Name
Character01.lxo
Sancho.lxl (2)
Area Light
Mesh
Camera
Texture Group
Driftwood_Tunnel.lxe
Directional Light

1. On the right panel, open an appropriate viewport tab. For example, open **Items**, **Shading**, **Mesh Ops** or **Groups**.
2. Use **Ctrl** + click to select the appropriate items you want to save as a preset.
3. Right-click on your selection and select one of the following options.

Note: Menus vary depending on which viewport tab you are working in. For example, **Items**, **Shading**, and **Groups**.

- **Presets** > **Save...** - Saves selected item as a mesh preset (***.lxl** file).
- **Presets** > **Save Assembly Preset...** - Saves selected item as an assembly preset (***.lxp** file).
- **Save Preset...** - Saves selected item as a preset (***.lxp** file).
- **Save Preset with Thumbnail...** - Saves a selected item as a preset and uses an image of the item for the thumbnail (***.lxp** file).

4. Navigate to the location where you'd like to save your preset and then click **Save**.

Tip: To make your new preset show up automatically in the Preset Browser, use the directory where your presets are stored and save it within that directory. This is usually under **Luxology\Content\Assets.** You can also create a new directory for saving presets. On the Directories list, click **(add path)**, navigate to an appropriate directory, and then click **Select Folder**. The Preset Browser displays the name of that folder at the bottom of the navigation tree.

Preset File Menu

1. Select the items you wish to save as a preset.
2. In the menu bar, click one of the following options:
   - **File** > **Save Item Preset...** - Allows you to save any selected item as a mesh preset (***.lxl** file).
   - **File** > **Save Environment Preset...** - Allows you to save your customized environment as an environment preset (***.lxe** file).
   - **File** > **Save Assembly Preset...**Allows you to save selected items and their hierarchy as an assembly preset (***.lxp** file).

   A file browser opens.
3. Navigate to the location where you'd like to save your preset, type a file name, and then click **Save**.

Tip: To have the preset automatically appear in the **Preset Browser**, save it to the location where your other presets are stored. This is usually under the **Luxology\Content\Assets** directory.

Creating Different Preset Types

Item Presets

Item presets can be made from almost any item type. They are saved as **.lxl** files and can be added to your scene in various ways, depending on what type of item they are. Any layer or collection of layers can act as a preset.

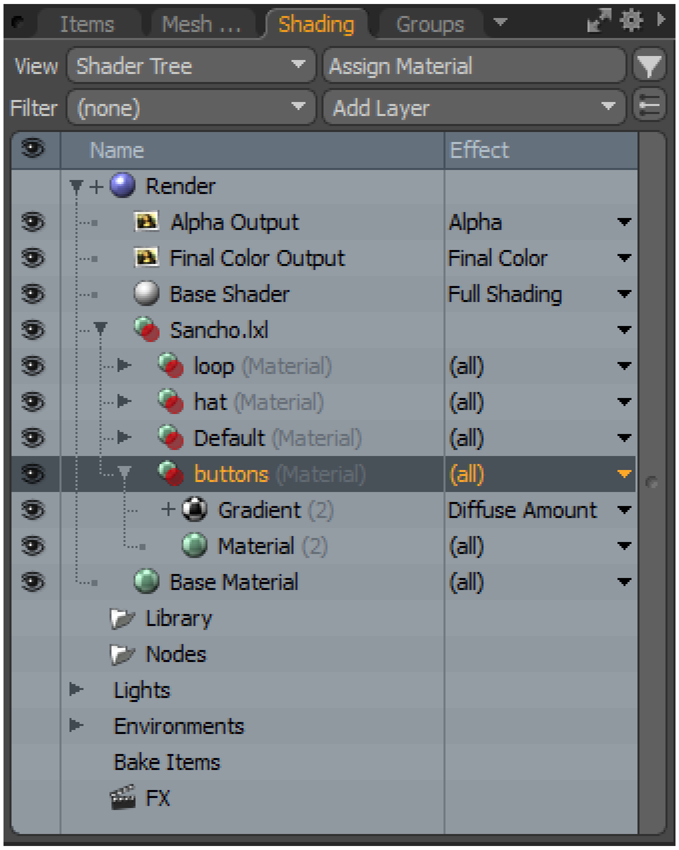
1. Open any viewport tab on the top right panel. For example, open the **Items** tab.
2. Right-click on your selection and click **Presets** > **Save...**

Material Presets

Any layer or collection of layers in a **Shading** tab on the right panel can act as a preset. The following example uses a **Base Material** item to create a preset.

Tip: You can also create a new directory for saving presets. On the Directories list, click **(add path)**, navigate to an appropriate directory, and then click **Select Folder**. The Preset Browser displays the name of that folder at the bottom of the navigation tree.

1. Click **Layout** below the menu bar to view the Preset Browser directories on the bottom of the interface.
2. In the **Preset Browser** viewport, open the **Materials** tab, and then open a directory you would like to save your preset to.
3. On the right panel, open the **Shading** tab.



View: Shader Tree
Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Sancho.lxl	
loop (Material)	(all)
hat (Material)	(all)
Default (Material)	(all)
buttons (Material)	(all)
Gradient (2)	Diffuse Amount
Material (2)	(all)
Base Material	(all)
Library	
Nodes	
Lights	
Environments	
Bake Items	
FX	

4. Click + hold and drag the **Base Material** item onto a Preset Browser directory.

   Alternatively, right-click on a layer in the **Shader Tree**, then select **Save Preset...** from the dropdown menu. Navigate to the location where you'd like to save your preset. To make your new preset show up automatically in the Preset Browser, use the directory where your presets are stored and save within that directory. This is usually under **Luxology\Content\Assets**.

5. From the pop-up menu, select **Create Material Preset**.

Note: Quite often you may have a surface made up of multiple texture layers. In that case, you might want to create the preset from the Material Group (▣), as that includes all associated texture layers within the preset.

Tip: Fur is made up of texture layers, so making hair presets is just as easy as making surface presets.

Mesh Presets

Mesh presets require a small amount of setup to work properly. Mesh presets align themselves based on their relationship to Modo's origin when they are created. Therefore, it is recommended that you create mesh presets at Modo's origin.

You'll also want them resting on the ground, so they are facing the Y up (positive) direction. For example, when you create a light fixture that you are applying to a series of walls, each one is aligned to the wall in the orientation when placed. Once your model is properly positioned, you can save it as a preset.

Note: Item transforms are not considered in presets, so make sure you make your alignment changes at the component level, or use the **Freeze** command in the item's **Properties** panel.

1. Click **Layout** under the menu bar, select the mesh item in the 3D viewport, and drag-and-drop it onto the **Preset Browser**. Alternatively, right-click on the mesh layer in the **Items** list and select **Presets** > **Save...**

Tip: To create a new directory for saving presets, click **(add path)**on the Directories list and navigate to an appropriate directory.

Render Presets

Render Presets can be made from a number of settings on a render item, which contain core rendering attributes such as Frame, AA and ray trace settings, and Global Illumination settings.

1. Right-click on the **Render** Item (the top-most item of the **Shader Tree**) and select **Save Preset...**. Alternatively, you can select the **Render** item and in the menu bar, click **File** > **Save Item Preset...**
2. To display the preset automatically in the Preset Browser, navigate to the directory where your presets are stored, and click **Save**.

   You can also create a new directory for saving presets. On the Directories list, click **(add path)**, navigate to an appropriate directory, and then click **Select Folder**. The Preset Browser displays the name of that folder at the bottom of the navigation tree.

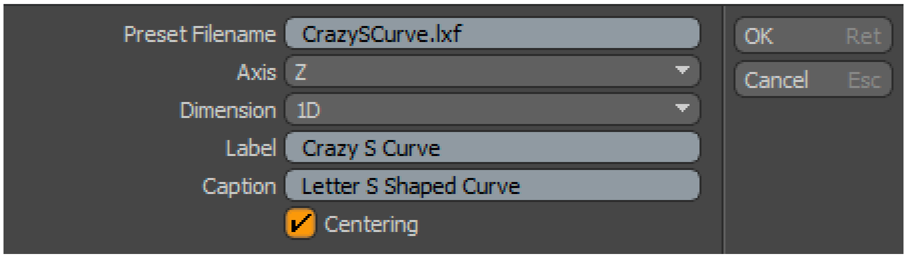
Tip: Addition Camera items can be added to a render preset. For example, automatic DOF and motion-blur cameras can be useful. When the saved Render Preset is applied to a scene, these additional camera items are added to the existing cameras used in your scene.

Profile Presets

Profile Presets can be made from curves drawn in Modo. For example use the Adding Curves, Bezier , or Sketch tools.

1. Click **Layout** below the menu bar to view the Preset Browser directories on the bottom of the interface.
2. Expand **Meshes** and open a directory you would like to save your preset to.
3. Drag the curve item in the 3D viewport onto the **Preset Browser**.
4. From the pop-up menu, click **Save Profile Preset**.
5. Click **Save**.

   The following dialog displays.



Preset Filename CrazySCurve.lxf

Axis Z

Dimension 1D

Label Crazy S Curve

Caption Letter S Shaped Curve

☑ Centering

6. Set the following options:
   - **Preset Filename** - Contains the name you specified in the previous dialog. You can edit this field.
   - **Axis** - Specifies the axis that the profile was created on: either X, Y or Z.
   - **Dimension** - Specifies whether the drawn profile is **1D** or **2D**.
   - **Label** - Labels the preset.
   - **Caption** - Provides a caption for the preset.
   - **Centering** - When enabled, automatically centers the curve (if it was created away from the origin).
7. Click **OK** to save.

Assembly Presets

Assembly Presets can contain collections of multiple items, even including full hierarchies of rigged elements. For information about how to create an assembly using the Schematic Viewport, see Creating Preset Assembly Assets in the Schematic Viewport.

1. Click **Layout** below the menu bar to view the Preset Browser directories on the bottom of the interface.

2. Open the **Meshes** tab and then open a directory you would like to save your preset to.
3. Select the root item of the setup in the **Groups** list, or all the individual items you want to save (from the various viewports such as related materials, if applicable).
4. Drag-and drop your selection onto the **Preset Browser**.
5. From the pop-up menu, select **Save Preset**.

To save a rigged assembly preset, with channel connections, modifiers, and/or constraints, you have to save the Assembly preset from the Schematic Viewport.

1. In the Schematic Viewport, create an Assembly node of all the elements you need.
2. Right-click on the header/title area of the resulting node and select the **Save Assembly Preset** option from the context menu.

Note: Any scripts used by the preset needs to be saved into the root script location so they can be found automatically.

Environment Presets

Environment presets differ from Item Presets in that you can store complete scenes in them, including lights, environments, meshes, and so on. Using Environment Presets allows you to quickly switch out entire studio setups by double-clicking them in the Preset Browser. They are saved to **.lxe** files.

1. In the menu bar, click **File** > **Save Environment Preset...** Alternatively, right-click on the environment material layer in the **Shading** tab on the right panel and select **Save Environment Preset**.
2. Navigate to the location you'd like to save your preset to, then click **Save**.

Note: To have the preset automatically appear in the Preset Browser, save it to the location where your other presets are stored. This is usually under **Luxology\Content\Assets**. Modo saves your scene as an **.lxe** Environment Preset file and it becomes accessible from the Preset Browser.

Tip: You can drag and drop an Environment Preset from the Preset Browser into the **Environment** folder on the **Shader Tree**. A contextual menu appears, allowing you to either replace or add your selected Environment.

Using a Preset Mesh as a Replicator

Mesh presets are fully compatible with replicators, though they are not automatically applied as such. To use a preset mesh as a replicator:

1. Double-click the mesh object in the **Preset Browser** viewport, so it is created as its own Mesh Item at the origin.
2. In **Item** mode, select the **Point Source** and the newly created preset item mesh.
3. Create the Replicator by clicking **Item** > **Add Replicator**.

   Please refer to Replicator topic for more information on working with Replicators.

Creating Preset Assembly Assets in the Schematic Viewport

You can create your own assembly item presets and share them with other users using the **Schematic** viewport. Your assembly alias preset can contain a selection of mesh items and their hierarchy, or comprise of a complex setup, using deformers and modifiers.

This is done by creating Assembly Presets containing a collection of multiple mesh items and their hierarchy, such as rigging elements. You can also create more complex assembly setups using mesh items, cameras, deformers, and modifiers. Once created, you can collapse these items into a first class item type, such as a Channel Modifier assembly, a Selection Operations assembly, and a Mesh Operations assembly to share as Presets. The following defines each type.

Channel Modifier

The Channel Modifier evaluates the various modifiers inside the assembly using the input values on the alias, and it passes the results to the output channels on the alias. A channel modifier is the simplest form of an assembly alias. Setting this option results in behavior that is identical to a first class channel modifier.

You can expose relationship graphs from items contained within the assembly, and any form of evaluation can be performed within the assembly, to calculate the results of the outputs from the inputs.

Mesh Operation

The Mesh Operation allows you to collapse the assembly to directly integrate it into the procedural modeling system. The assembly alias requires a single mesh operation inside the assembly to be evaluated by the procedural modeling system when the alias is evaluated. You can choose a mesh operation within the assembly, and when the alias is evaluated by the procedural system, the evaluation is redirected to the chosen item within the assembly.

Adding a mesh operation alias preset into the scene automatically attaches the alias item to the primary mesh.

Selection Operation

The Selection Operation allows a collapsed assembly to integrate directly with a mesh operation and it controls which elements it operates on. Like the mesh operation alias type, the selection operation requires an item with the assembly that can be evaluated by the procedural system, when the alias is evaluated.

Adding a selection operation alias preset into the scene automatically attaches the alias item to the selected mesh operation.

Assembly alias presets can be integrated into Modo scenes and they act like real items. No programming skills are required. If you can create an assembly, you can create an assembly alias preset. For more information about assemblies, see Assembly Presets.

This functionality allows technical users to create assemblies that are easier to use for others and it makes the scene appear tidier. The assembly and its contents are entirely hidden from the **Schematic** viewport and display as a single assembly alias item in the right-hand pane, under the **Mesh Ops** tab. These presets can be expanded later to apply modifications.

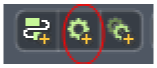
Creating an Assembly Alias Preset

To save assembly setups as assembly alias presets, create and save your presets using the Schematic Viewport.

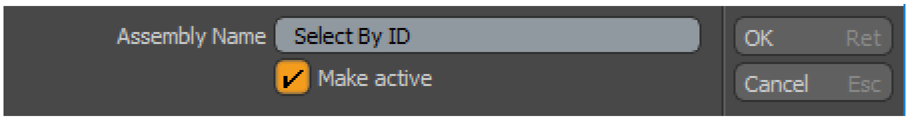
In this example, an alias Selection Operation assembly is created which selects a specific element on a mesh by an index. Once the assembly alias preset is saved, it is used in a new scene. A simple grid mesh item is added to the scene and then used to apply the assembly alias preset to. A number is then entered for the index to update the mesh.

To create an alias selection operation assembly preset:

1. Open the **Setup** layout.
2. In the **Schematic** viewport, click the **Add new root assembly** button.



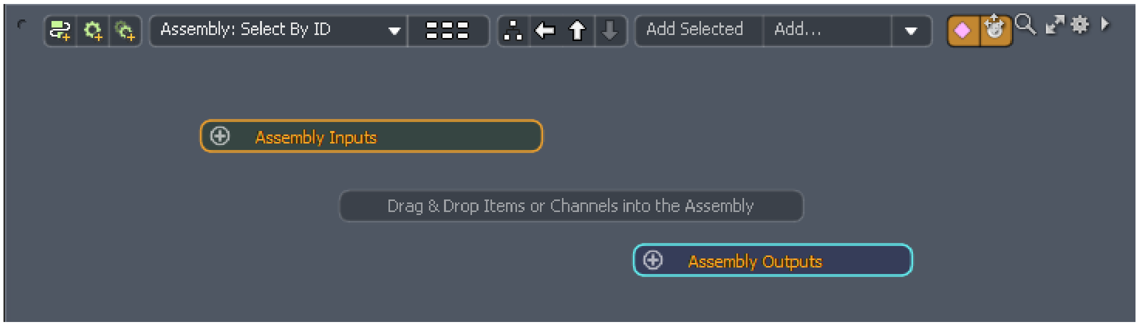
   The **New Schematic Assembly** dialog displays.
3. Type a name for your assembly, enable **Make active**, and click **OK**.



Assembly Name: Select By ID
☑ Make active

OK Ret
Cancel Esc

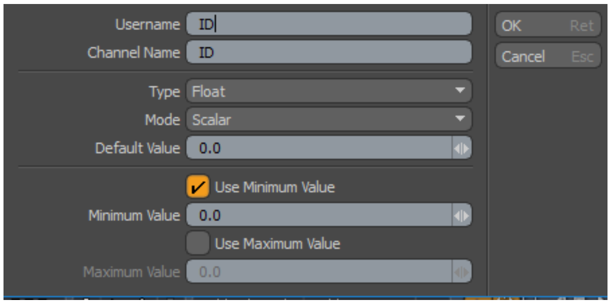
The assembly displays in the Schematic viewport.



Assembly: Select By ID
Add Selected
Add...
Assembly Inputs
Drag & Drop Items or Channels into the Assembly
Assembly Outputs

4. Click **Add...**, expand **Mesh Operations** > **Selection**, and double-click **Select By Range**.
5. In the **Properties** panel, under the **Select By Range** tab, right-click **Start** and select **Add Channel to Schematic** or drag and drop it from the **Properties** panel onto the **Select By Range** item in the **Schematic** viewport.
6. Right-click **End** and select **Add Channel to Schematic**.
7. On the **Assembly Inputs** item, click the plus button.

   The **Create User Channel** dialog displays.
8. In the **Username** field type **ID**, enable **Use Minimum Value** and then click **OK**.

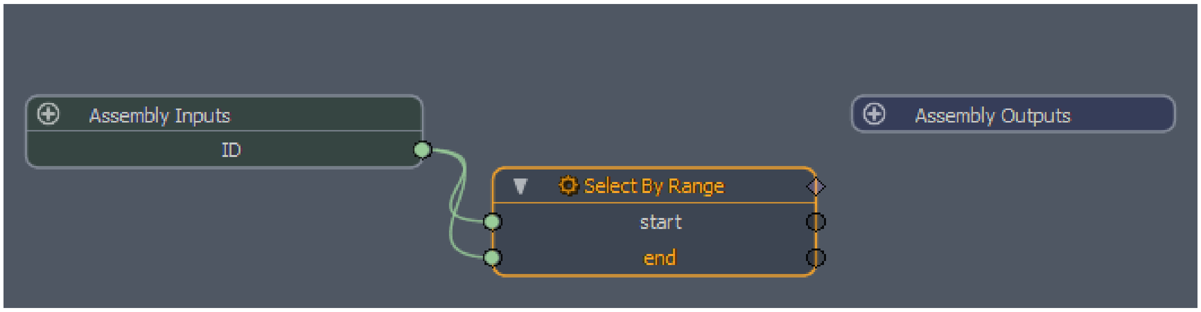


Username: ID
Channel Name: ID
Type: Float
Mode: Scalar
Default Value: 0.0
☑ Use Minimum Value
Minimum Value: 0.0
☐ Use Maximum Value
Maximum Value: 0.0

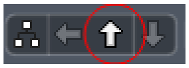
OK Ret
Cancel Esc

Note: The Channel Name is populated automatically.

9. Hook the **start** and **end** items to the **ID** channel. To do this, drag the black circle for **start**, under Select By Range, and drop it onto the **plus** button for Assembly Inputs.


Assembly Inputs
ID
Assembly Outputs
Select By Range
start
end

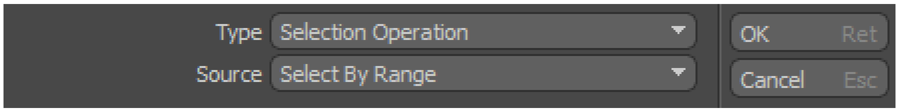
10. Click **View Parent Assembly**.



11. Right-click on **Select By ID** and select **Collapse to Item** from the dropdown menu.

    The **Assembly To Item** dialog displays.

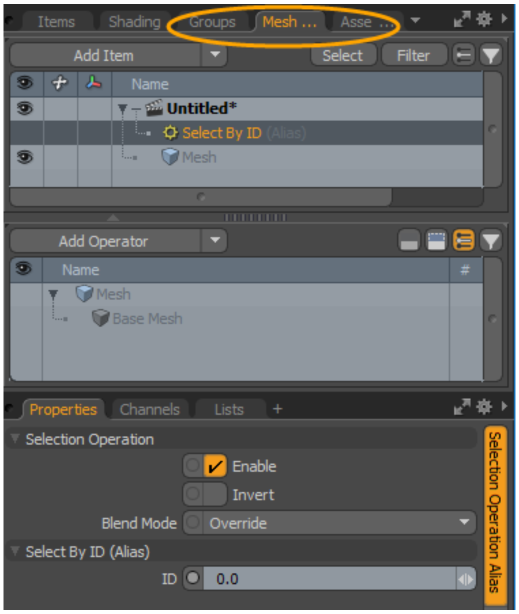
12. Set **Type** to **Selection Operation**, set **Source** to **Select by Range** and click **OK**.



Type Selection Operation
Source Select By Range

OK Ret
Cancel Esc

The assembly is hidden in the Schematic viewport and displayed in the right pane, under the Mesh Ops tab. Click the arrow beside your scene item to see the **Select By ID (Alias)**. The **Source** is evaluated by the procedural system when this assembly alias preset is evaluated.



Selection Operation
☑ Enable
☐ Invert
Blend Mode Override
Select By ID (Alias)
ID 0.0

Items
Shading
Groups
Mesh ...
Asse ...
Add Item
Select
Filter
Name
Untitled*
Select By ID (Alias)
Mesh
Add Operator
Name
Mesh
Base Mesh
Properties
Channels
Lists
Selection Operation Alias

To save the assembly alias preset:

1. With **Select By ID (Alias)** selected, click **File** > **Save Assembly Preset**.
2. In the **File Name** field, type **Select By ID** for the name of your preset and click **Save**.

Note: The file is saved to the default **Content** > **Aliases** directory. Alternatively, browse to another directory to change the location where your preset is saved to.

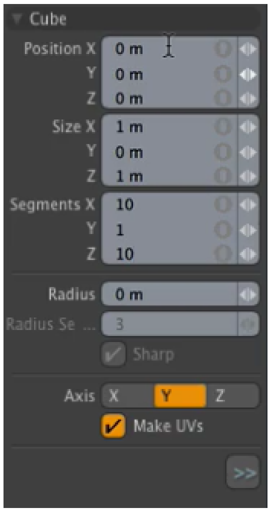
3. Click **File** > **Close Scene**.

   The **Unsaved Changes** dialog is displayed.

4. Click **Don't Save**.

To create a new scene and apply the assembly alias preset:

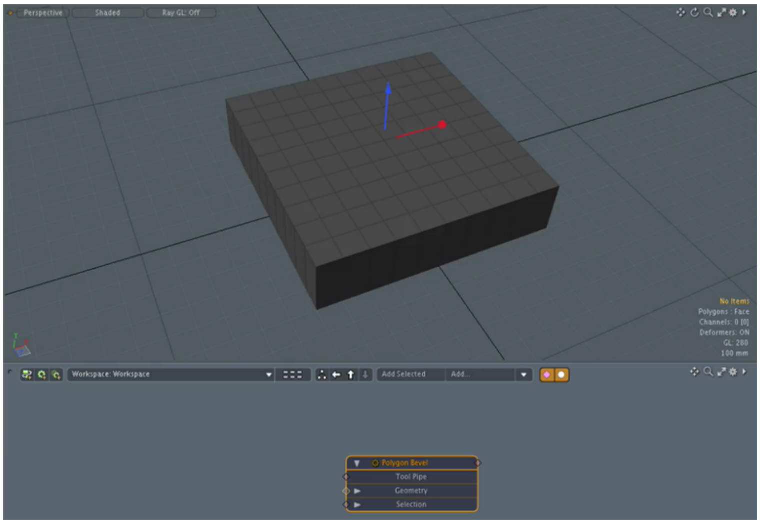
1. In the **Setup** layout, click the **Model Palette** icon on the top left panel (alternatively, press **F2**).
2. Open the Toolbox **Basic** tab, select **Cube**, set the properties for your grid, and click **Apply**. The following image shows example properties.



Cube
Position X 0 m
Y 0 m
Z 0 m
Size X 1 m
Y 0 m
Z 1 m
Segments X 10
Y 1
Z 10
Radius 0 m
Radius Se ... 3
☑ Sharp
Axis X [✓ Y] Z
☑ Make UVs

   A simple grid is added to your scene.

3. Press the **Spacebar** or press the **Esc** key, to drop your selection.
4. In the **Schematic** viewport on the bottom of the **Setup** layout, click **Add** and select **Mesh Operations** > **Polygon** and double-click **Polygon Bevel**.
5. Click and drag the blue handle to change the bevel amount.


Perspective
Shaded
Ray GL: Off
Workspace: Workspace
Add Selected
Polygon Bevel
Tool Pipe
Geometry
Selection

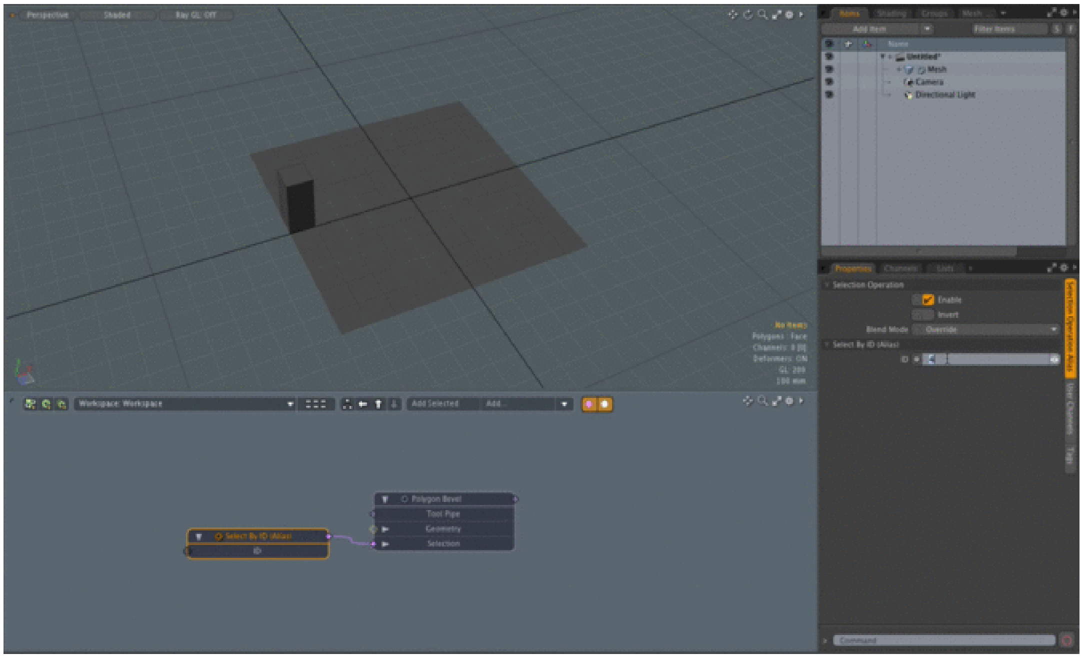
6. In the **Mesh Operations** tab, expand **Polygon Bevel** > **Selection** and click **Add Selection**.

Tip: This can also be done in the **Schematic** viewport.

7. From the **Add Operation** dialog, expand **Assembly Aliases**, and double-click **Select By ID**.

   The assembly alias preset, named **Select By ID** is automatically hooked up to the **Polygon Bevel**.
8. Click the **Select By ID (Alias)** element to display its properties.
9. In the **Selection Operations Alias** tab, change the ID value to select that element by index.

   As you change the ID value, different elements on the grid are displayed with the assembly settings while the remainder of the gird remains flat.


Perspective
Shaded
Ray GL: Off
Items
Shading
Groups
Mesh
Add Item
Filter Items
Name
Untitled*
Mesh
Camera
Directional Light
Properties
Channels
Lists
Selection Operation
Enable
Invert
Blend Mode
Override
Select By ID (Alias)
ID
Workspace: Workspace
Add Selected
Polygon Bevel
Tool Pipe
Geometry
Selection
Select By ID (Alias)
ID
Command

Creating Assets Containing Embedded URLs

Modo allows you to create and share assets containing embedded URLs. This is useful when you want to add URLs to a custom preset to direct an artist to external URLs to get additional information from the Web. Once added to the Presets Browser for others to see, users can double-click on the asset to open the URL in a web browser. You can customize these assets in Modo to change the icon image and the URL.

There are a number of ways to create assets containing embedded URLs.

- In the **Model** layout, open the **Images** tab on the right panel, right-click on an image, and select **Export as LRL**. A ***.lrl** file is created using the image you have specified as the icon image.

> **Tip:** If the **Images** tab is not displayed, click down arrow at the top of this viewport and select **Data List** > **Clips**.

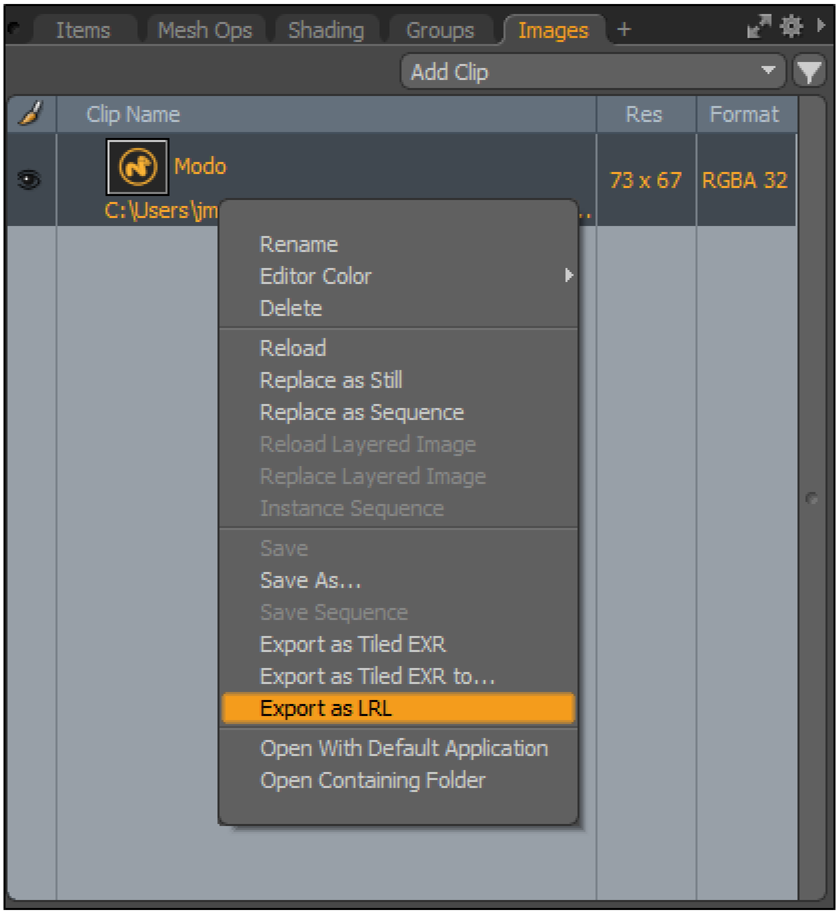
- Click **System** > **Create URL Preset.** A ***.lrl** file is created using a default icon image.
- Run the command: **image.createLRL url:**<URL> <filename> <clip>.
  - **URL**: Specify the desired URL to embed into the asset file.
  - **filename**: (Optional) Specify the absolute path and file name for the output. If not given, a standard save dialog opens for the user.
  - **clip** (Optional): Specify the clip name. If not specified, the current clip name is used.

To create a preset with an embedded URL:

1. In the **Model** layout, click **File** > **Load Image**.
2. Browser to the a directory and double-click on an image to load it into Modo.

> **Tip:** Click **All Formats** to view all of the supported Load Image formats.

3. In the right panel, open the **Images** tab.
4. Right-click on the loaded image and select **Export as LRL**.


Items
Mesh Ops
Shading
Groups
Images
Add Clip
Clip Name
Res
Format
Modo
C:\Users\jm
73 x 67
RGBA 32
Rename
Editor Color
Delete
Reload
Replace as Still
Replace as Sequence
Reload Layered Image
Replace Layered Image
Instance Sequence
Save
Save As...
Save Sequence
Export as Tiled EXR
Export as Tiled EXR to...
Export as LRL
Open With Default Application
Open Containing Folder

The **Create URL File** dialog displays.

5. Type a URL into the **URL** field and click **OK**. For example, type **https://www.foundry.com/products/modo**.

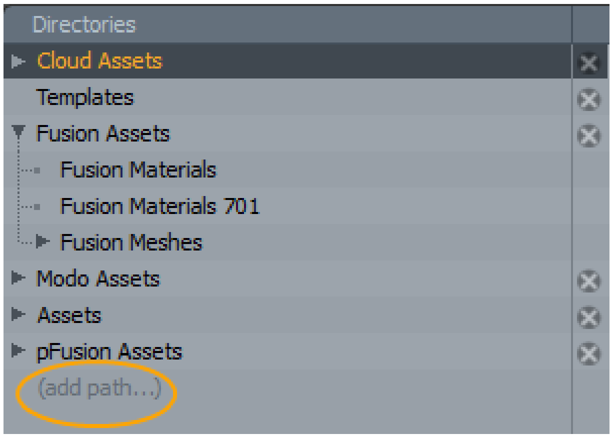
   The browser displays the directory of the original file you uploaded.

6. Type a name for the asset file and click **Save**.

   Modo automatically creates a file using the ***.lrl** extension with the URL embedded into the XML. This file uses the image you have specified as the icon image.

Sharing a *.lrl File

1. In the **Model** layout, on top of the toolbox on the left panel, click the **Preset Browser** palette icon. Alternatively, press **F6**.
2. In the left panel, click **(add path)**.


Directories
Cloud Assets
Templates
Fusion Assets
Fusion Materials
Fusion Materials 701
Fusion Meshes
Modo Assets
Assets
pFusion Assets
(add path...)

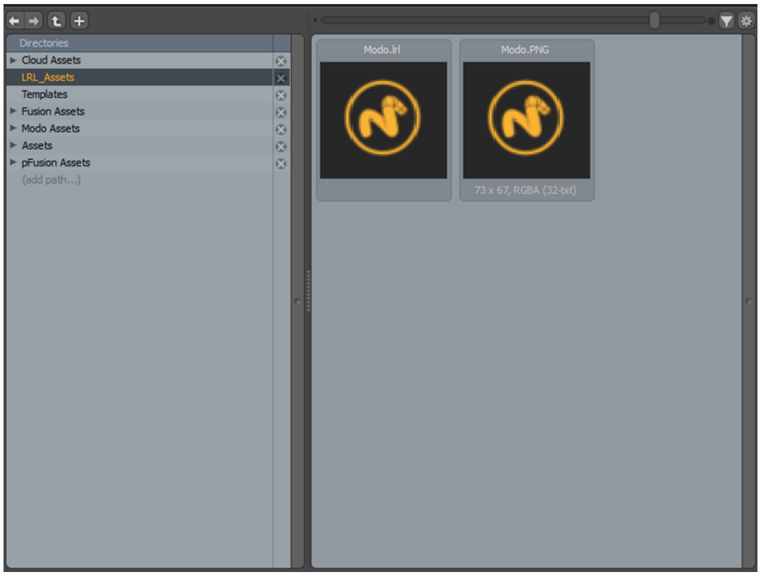
The browser windows displays.

3. Navigate to the directory you have saved your ***.lrl** file to and click **Select Folder**.

   By default, the directory specified is saved to the **Preset Browser** under the **Cloud Assets** directory.

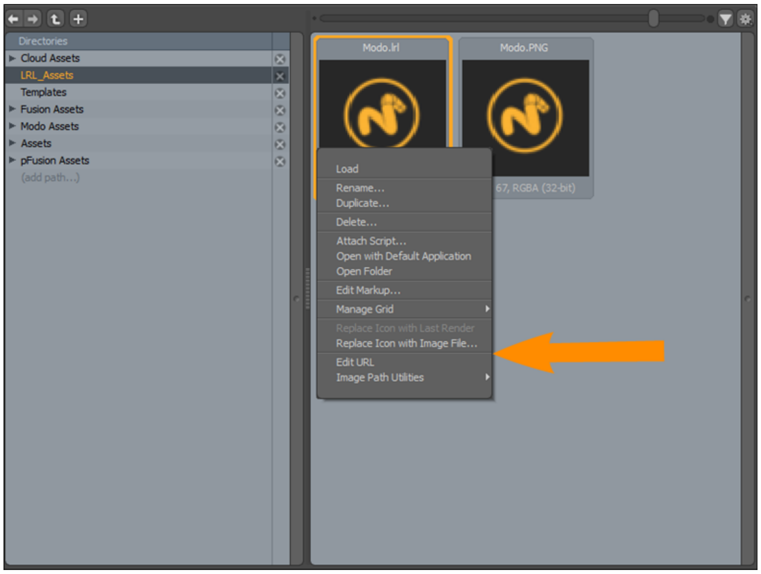
Editing a *.lrl File

1. In **Model** layout, click the **Preset Browser** palette icon, expand **Cloud Assets**, and select your directory.


Directories
Cloud Assets
LRL_Assets
Templates
Fusion Assets
Modo Assets
Assets
pFusion Assets
(add path...)
Modo.lrl
Modo.PNG
73 x 67, RGBA (32-bit)

2. Right-click on the ***.lrl** file.
3. Click any of the following menu items:

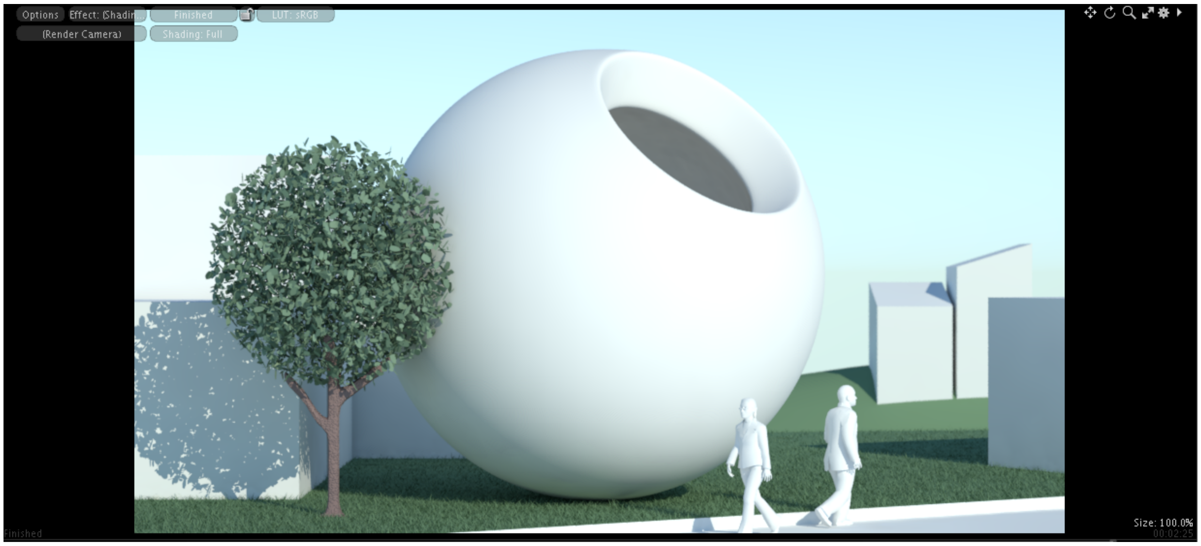
   - **Replace Icon with Image File** - Navigate to a new image to use as the icon image and click **Open**.
   - **Edit URL** - The **Edit URL** dialog displays. Enter in a new URL and click **OK**.


Directories
Cloud Assets
LRL_Assets
Templates
Fusion Assets
Modo Assets
Assets
pFusion Assets
(add path...)
Modo.lrl
Modo.PNG
67, RGBA (32-bit)
Load
Rename...
Duplicate...
Delete...
Attach Script...
Open with Default Application
Open Folder
Edit Markup...
Manage Grid
Replace Icon with Last Render
Replace Icon with Image File...
Edit URL
Image Path Utilities

Tip: Advanced Users: For information about how to limit what is displayed in the Preset Browser, see Preset Browser Advanced Filtering .

Preview Viewport

The **Preview** viewport uses the Modo render engine in a progressive refinement mode that allows you to see all changes in a scene as a fully rendered image. This viewport is thread-aware so that when it's not the current viewport, it doesn't interrupt interactivity from other views. This allows you to work on your project without performance degradation, yet anytime there are "free cycles", the **Preview** viewport begins a progressive render, starting at low quality and filling it in to higher quality as time permits. In the **Preview** viewport you can view changes to the geometry, materials, textures, cameras, lights, and even render settings, such as ray tracing and global illumination.



In traditional systems, the 3D viewport gives a proxy view of the project in real-time and a separate offline render engine (if available) is summoned periodically to generate a full quality image. This offline rendering is a modal effect that requires you to completely turn over control of the system to the render engine until the image is complete. This start-and-stop process can interrupt your workflow taking you out of your creative flow. The **Preview** window enables you to simply pause to glance at the progress of your project and then continue the process of refining your vision resulting in a much more fluid and art-conducive process.

To open the Preview Viewport:

- Select **Render** > **Open Preview**. Alternatively, press **F8**.

Tip: The **Preview Viewport** can be scripted (run) from a command line using **Modo CL**. There are now a number of commands available to render a preview to an image, image sequence, and movie. For more information, see Modo SDK.

Usage

The main advantage to using the **Preview** viewport is the fast display of the rendered scene. Navigating the camera around in the **Preview** window works just like any other 3D (OpenGL) Viewport set to **camera**, where dragging with the standard modifier keys changes the position of the camera with the additional ability to define a rectangular area by pressing **Ctrl**+**Alt**+right-click to define a box area to zoom in.

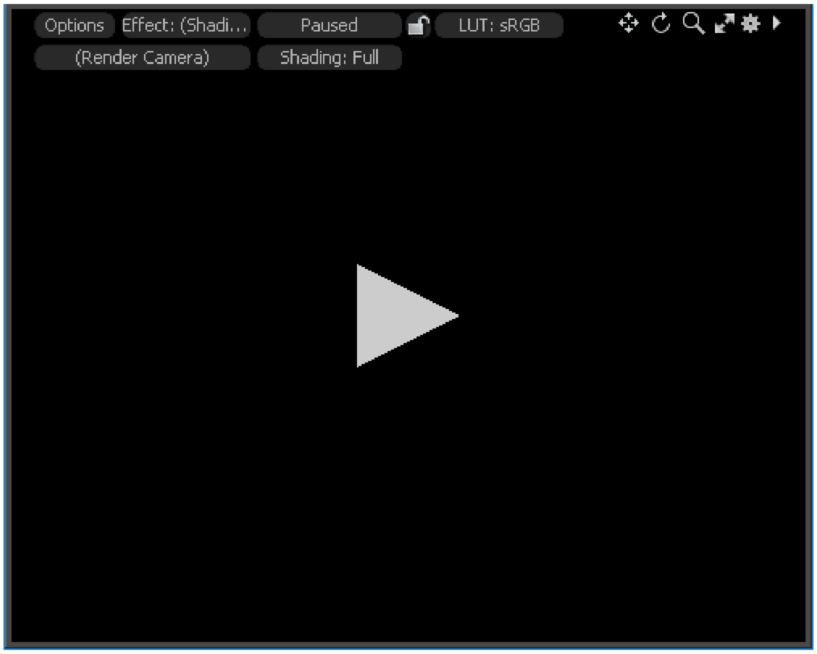
Note: Navigation only works when the **Full Resolution** option is disabled. When enabled, you are only able to pan around the rendered image itself.

Tip: Auto-Save saves the current **Preview** rendered state, allowing you to pick up a progressive render from it's last saved point when you open an auto-saved scene, rather than having to start the render again.

You can also zoom in for a closer look without affecting the camera's position with a middle scroll wheel mouse. The view snaps to 25%, 50%, 200%, 400% while scrolling your mouse button. To zoom in this way, press and hold **Shift**, then place the cursor over the center point of the zoom target and roll the mouse wheel in and out to zoom. To reset the view, use the **Reset Render** command in the **Options** menu available in the top left of the viewport.

Preview Viewport Controls

The buttons at the top of the viewport also offer additional useful functionality. You can click any of them to open dropdown menus or enable/disable specific functions.



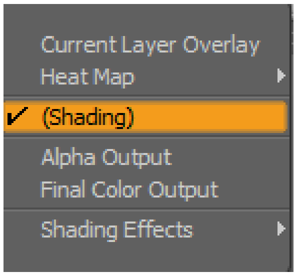
Options

The Options button opens a dropdown menu with controls to customize the **Preview** window's rendering engine for additional optimization. Similar options listed in this menu are also available in the Preview Settings.

Effect

Directly next to **Options** is the **Effect** button that lets you define a specific rendering channel for previewing, allowing for a number of diagnostic possibilities, such as previewing the amount of Fresnel reflectivity on a surface. The following menu options are available.

For details on these menu options, see Preview Settings



Tip: You can drag and drop any render output from the **Shader Tree** onto the **Preview** window automatically, switching it to view that **Effect**.

Rendering Status

The **Rendering Status** button provides useful feedback of the **Preview** window's current state. You can click this button to pause preview rendering temporarily. When clicked again, the **Preview** window refreshes the image, starting a new render. When finished, the button displays **Finished**.

Lock

The **Lock** 🔓 icon allows you to lock the preview to avoid accidentally restarting it. For more information, see Locking Preview.

LUT

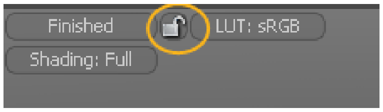
The **LUT** button allows you to choose the colorspace of the monitor. For more detailed information, see LUT.

Camera Selection

Below the **Options** button is a camera pop-up selector, allowing you to easily select the current camera for rendering, without the need to specify it in the Render item.

Locking Preview

You can lock preview to avoid accidentally restarting it. To do this, click the small lock button at the top of the screen:



Finished

LUT: sRGB

Shading: Full

There are two ways you can lock the preview:

- Clicking the Lock 🔓 button locks the current settings and allows preview to continue rendering.
- **Ctrl**+clicking the Lock 🔓 button sets Preview in a final render mode, using the **Final Render Quality** settings.

The **Final Render Quality** default settings are the following:

- **Full Resolution** is enabled
- **Render All Outputs** is enabled
- All ray tracing options are enabled (**Global Illumination**, **Shadows**, **Reflections**, and **Refractions**)

Note: If they are disabled in the Render settings, they are still enabled in the **Preview** window.

- **Fur** is enabled (**Draft Fur** is disabled)
- **Displacements** are enabled (**Draft Displacement** is disabled)
- **Volumetrics** is enabled
- **Motion Blur** is enabled
- The render level for both Subdiv and PSUBs is used
- Cage mode is disabled
- Secondary **Irradiance Cache** is enabled, unless disabled in the Render settings

When Preview is locked, any updates that can cause it to restart rendering are disabled. The UI is simplified, and only the **Options**, **Effect**, and **Pause/Rendering** buttons remain active.

Note: If you resize the **Preview** window and **Full Resolution** is not enabled, **Preview** restarts once it's unlocked.

When you unlock the preview, all your previous settings are restored.

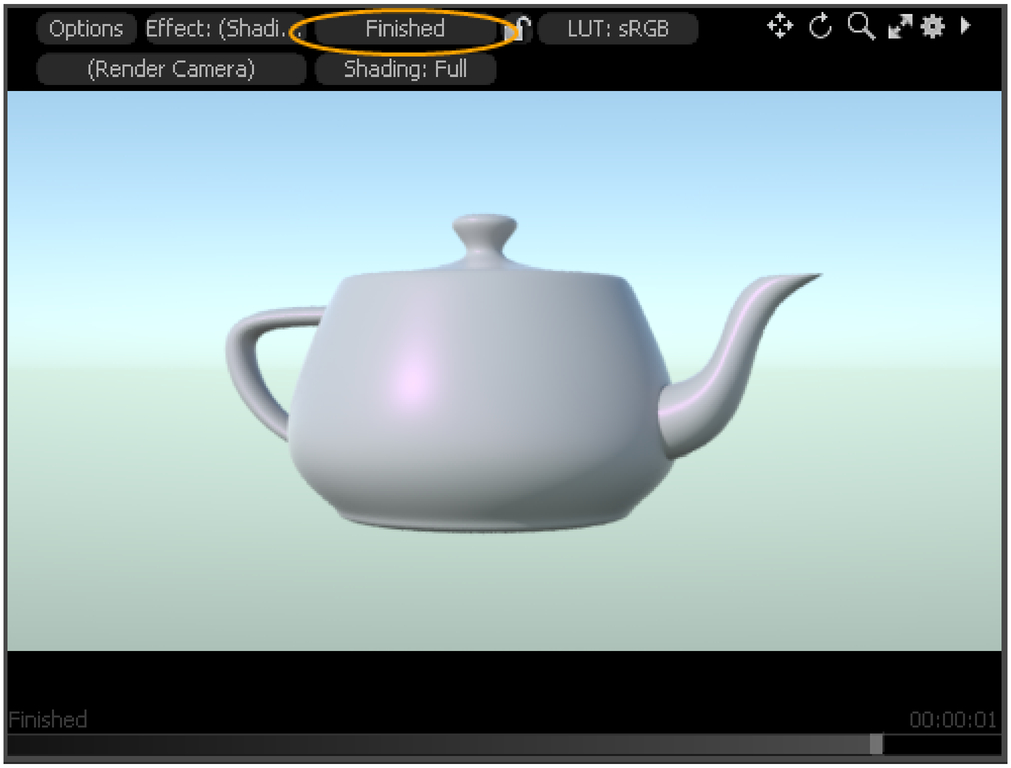
Tip: You can also use **Preview** lock when baking.

Changing Environments

You can quickly select an Environment preset from the Preset Browser and drag it onto the Preview Viewport to view your scene in a different background. This is useful when experimenting with different backgrounds using different colors, gradients, and image maps.

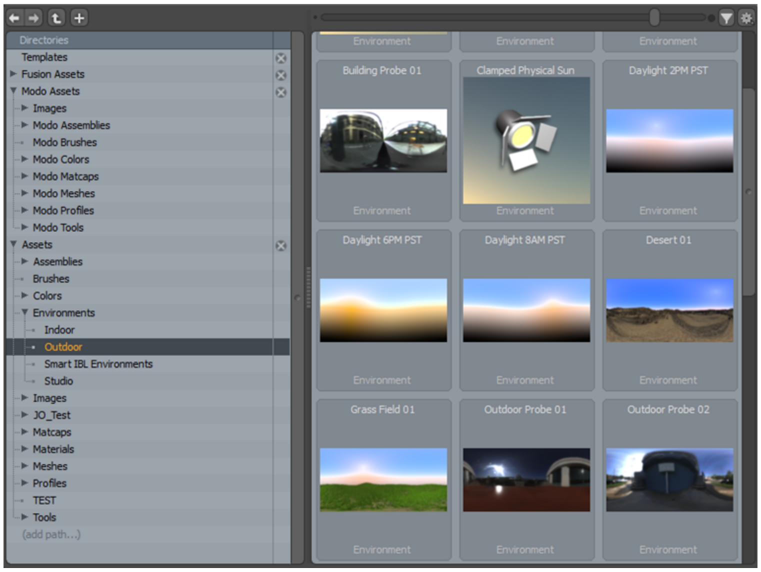
To apply a environment preset:

1. Load your mesh using the **Model** layout and click **Render** > **Open Preview**. Alternatively, press **Fn** + **F8**.
2. Click the Rendering Status button to render your scene.



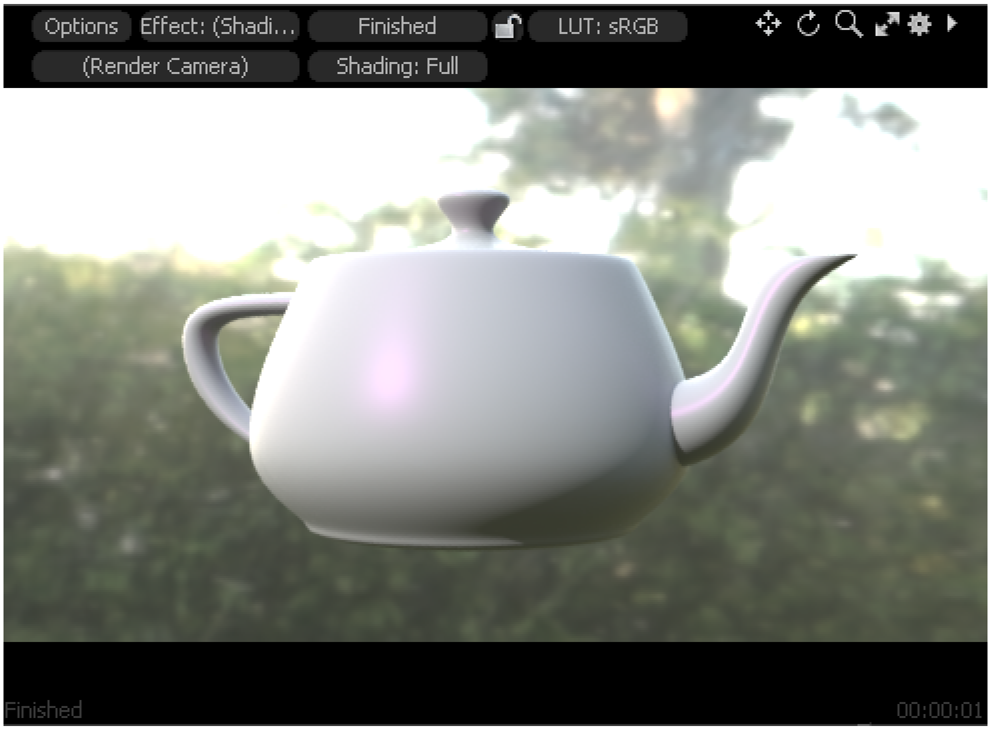
3. On the top of the Toolbar, on the left panel, click **Preset**. Alternatively, press **Fn** + **F6**.

   The **Preset Browser** opens.
4. On the left panel, expand **Assets** > **Environments**.
5. Expand any one of the sub-categories.


Directories
Templates
Fusion Assets
Modo Assets
Images
Modo Assemblies
Modo Brushes
Modo Colors
Modo Matcaps
Modo Meshes
Modo Profiles
Modo Tools
Assets
Assemblies
Brushes
Colors
Environments
Indoor
Outdoor
Smart IBL Environments
Studio
Images
JO_Test
Matcaps
Materials
Meshes
Profiles
TEST
Tools
(add path...)
Building Probe 01
Clamped Physical Sun
Daylight 2PM PST
Daylight 6PM PST
Daylight 8AM PST
Desert 01
Grass Field 01
Outdoor Probe 01
Outdoor Probe 02
Environment

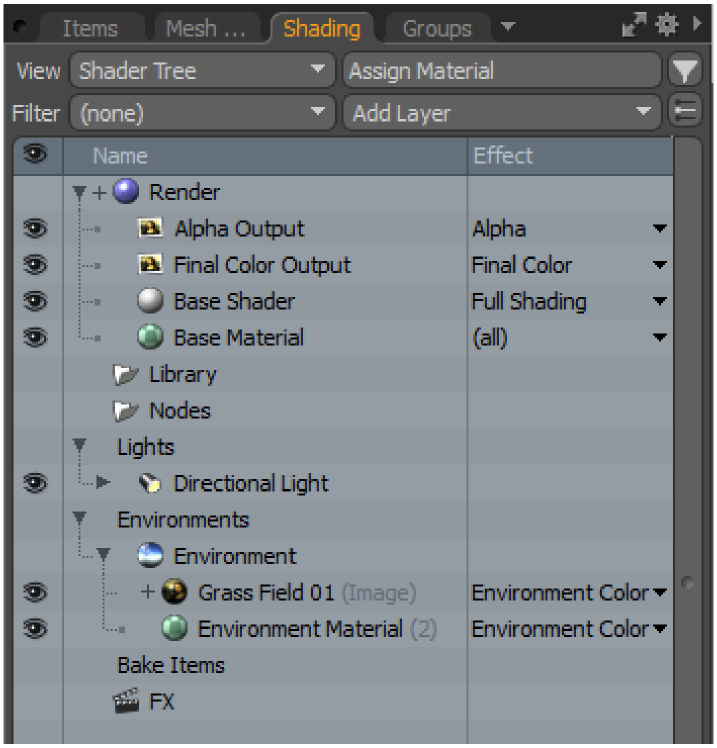
6. Click and drag one of the environments onto the **Preview viewport**.

   The **Preview viewport** displays the new environment.


Options
Effect: (Shadi...
Finished
LUT: sRGB
(Render Camera)
Shading: Full
Finished
00:00:01

7. From the **Preset Browser**, select another environment, right-click and drag it onto the **Preview viewport**.

   The **Preview viewport** displays the new environment. In addition, this environment is added to the Shader Tree, under **Environments**, along with any other environments created for your scene.



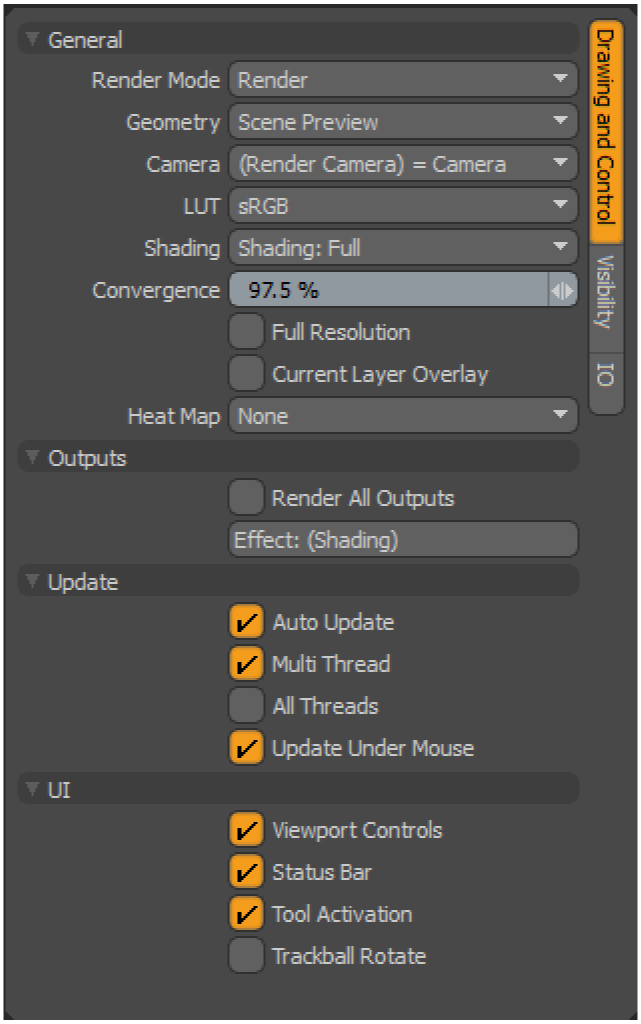
Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Base Material	(all)
Library	
Nodes	
Lights	
Directional Light	
Environments	
Environment	
Grass Field 01 (Image)	Environment Color
Environment Material (2)	Environment Color
Bake Items	
FX	

8. On the **Shader Tree**, under **Environments**, click the **eye** button to toggle an environment off and on to view different results.

Tip: You can also add and replace environments in the **Shader Tree.** For more information, see Adding and Modifying Shader Tree Item Layers.

Preview Settings

You can access a complete list of settings in the **Preview Settings** window. Click on the Cogwheel ⚙ button in the top-right of the **Preview** window.



General

Render Mode: Render

Geometry: Scene Preview

Camera: (Render Camera) = Camera

LUT: sRGB

Shading: Shading: Full

Convergence: 97.5 %

☐ Full Resolution

☐ Current Layer Overlay

Heat Map: None

Outputs

☐ Render All Outputs

Effect: (Shading)

Update

☑ Auto Update

☑ Multi Thread

☐ All Threads

☑ Update Under Mouse

UI

☑ Viewport Controls

☑ Status Bar

☑ Tool Activation

☐ Trackball Rotate

Drawing and Control Options

Option	Description
General	
Render Mode	Set the **Preview** render mode to Render or Bake. For more details on Bake mode, see Preview Bake Mode.
Geometry	Select the **Preview** display types. The following options are available: • **Scene Preview** - Renders the active scene, based on the settings of the **Shader Tree**. When used as such, it also works as an interactive view of any changes made to the scene, updating as the changes are made.

Option	Description
	- **Sphere**, **Plane**, and **Preset Sphere** - These options are useful for adjusting settings of individual materials in the **Shader Tree**, providing a way to see adjustments made interactively away from the surface the material might be applied to. The top-most **Shader Tree** layer always takes precedence in this mode. The three primitive options use the current scene environment and lighting.
Camera	Set which camera to use for creating the preview.
LUT	Set the colorspace of the monitor. **Note:** Unless there is a specific need, you should just leave this setting on the default **sRGB** setting. For more information, see Color Management.
Shading	Select a quick **Preview** optimizer to choose between full and fast shading. - **Shading: Full** - At each update, Modo evaluates the entire **Shader Tree**. - **Shading: Fast** - Modo only evaluates the associated surface that was changed in the **Shader Tree**, which in many cases can really speed up preview.
Convergence	With **Convergence** enabled, **Preview** checks that the image has converged to the minimum requirement (meaning the quality is good enough) before rendering the next frame. You can specify the convergence value here. Use this option to analyze your rendering based on the primary render output, not on whatever output you might have selected in the **Shader Tree**. This reduces the risk of you changing a render setting based on an output that is less computationally intensive than the final render output.
Full Resolution	Enable or disable full resolution for the preview. When enabled, the **Preview** window renders at the resolution defined by the Render item **Frame** settings, regardless of the window size. When disabled, the render resolution is defined by the actual window itself.

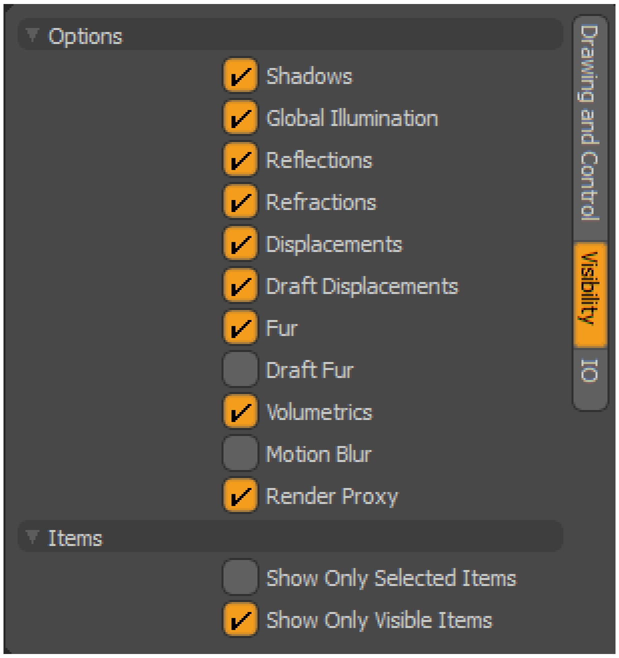
Option	Description
Current Layer Overlay	Allows you to toggle the visibility of a selected layer's contribution to a scene, visualized as a semi-transparent overlay, by enabling that option. Once enabled, it displays the current texture as an overlay on top of the render, visualizing the raw value unaffected by things like **Blending Mode**, **Opacity**, or shading. For more information, see Usage.
Heat Map	**Heat Map** displays the diagnostic values for progressive rendering. The following options are available: • **Number of Samples** - Indicates the number of samples per pixel so that you can see how much work is needed in various parts of the image to clean the noise. This is a useful information for shading in general as you can see which parts of the image are most expensive evaluation wise. • **Convergence** - Indicates the converged pixels, those that are identical to the same pixel evaluated in a previous pass. Converged pixels show as blue, other (non-converged) pixels are red when they are still very noisy and set color when they get close to the converged state. For information about Progressive Rendering Options, see Render Item: Settings.
Outputs	
Render All Outputs	When enabled, the **Preview** window renders all the visible Render Output Items from the **Shader Tree** simultaneously, allowing you to switch between the various layers (using the **Effect** menu) while the scene is rendering. Selecting between the different outputs does not trigger a preview refresh, though picking any of the other alternate **Effect** types does.
Effect	The **Effect** of the Render Output layer can be changed to a variety of options for different purposes. The following options are available: • **Shading** - Displays the evaluated results of all visible layers of the **Shader Tree**. • **Alpha Output** - Calculates the opacity of the objects based on their

Option	Description
	material settings to determine an alpha (transparency) value. With this setting, an environment does not contribute to the alpha channel, so when the final image is viewed with an application that supports alpha channels, only the rendered geometry is visible. For more information, see Basic Render Outputs. • **Final Output** - Represents all the settings of the **Shader Tree**, leveraging all visible material and texture layers as well as shaders and the render item settings. For more information, see Final Color. **Shading Effect** - Select an effect to apply to the **Preview** viewport. For more information on each of the shading effects, search the Online Help for a specified effect.
Update	
Auto Update	Set **Preview** to automatically update. Preview can be useful to see a fully-rendered view of geometry, as it's created and tweaked. Each time the cursor is released on a modeling action, such as a bevel, the geometry cache is updated to reflect the change. This causes a new, full evaluation of the scene by the **Preview** window. By disabling the **Automatic Update**, you can go into a manual mode where preview resets the geometry cache only when the **Reset Render** option is invoked, or the preview itself is paused and then re-activated.
Multi Thread	Enable multi-threading in the **Preview** viewport. Modo intelligently uses available CPU cores, interactively adjusting the number, based on the current needs of the system. This helps to keep the system interactive, while still keeping the preview fast. For information about Render Threads, see Rendering Preferences.
All Threads	Enable multi-threading in the **Preview** viewport to render the maximum number of threads.

Option	Description
	When enabled, uses all CPU cores, same as a full render would. This option maximizes preview speed but it can also result in sluggish performance when working outside of the **Preview** window. This is best to enable when the **extended refinement passes** option is selected and the system is not used for other purposes for a while.
Update Under Mouse	Set the **Preview** viewport to update under cursor movement. When enabled, Modo concentrates the **Preview** window evaluations to the area under the cursor when it's moving, allowing you to control what areas update first in any given preview. When the cursor is not moving or not over the **Preview** viewport, the standard random evaluation pattern is used, evenly spreading the CPU calculations across the image.
UI	
Viewport Controls	Display viewport controls. For more information about viewport controls, see 3D (OpenGL) Viewport.
Status Bar	Display the Preview status bar.
Tool Activation	Allows you to toggle tool visibility in the **Preview** viewport. When the option is enabled, you are able to see tool handles directly in the viewport, which can be used to transform objects interactively. When disabled, tool handles are invisible and you aren't able to modify items interactively.
Trackball Rotate	Toggle **Trackball Rotation** in the **Preview** viewport navigation. The **Trackball Rotation** option determines the default navigation rotational type. For more information, see Remapping. When the **Full Resolution** option is disabled, you can navigate in the **Preview** window like any other standard 3D viewport. Toggling the **Trackball Rotation** option enables/disables this function when navigating.

Option	Description
	Note: Any navigation of the **Preview** viewport moves the camera itself in 3D space.

Visibility Options



Options
☑ Shadows
☑ Global Illumination
☑ Reflections
☑ Refractions
☑ Displacements
☑ Draft Displacements
☑ Fur
☐ Draft Fur
☑ Volumetrics
☐ Motion Blur
☑ Render Proxy

Items
☐ Show Only Selected Items
☑ Show Only Visible Items

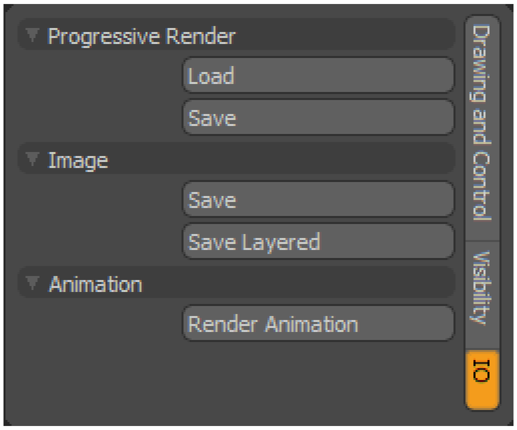
Option	Description
Options	
Shadows	Select whether to evaluate the direct light source shadows. These are lights such as **Point** and **Spot Lights**. When disabled, shadows are not evaluated in the **Preview** window, providing a speed boost dependent on the number of lights in the scene and their contribution. When enabled, all active direct light shadows are evaluated, as defined by each light's **Shadow** settings. For more information about shadow options, see Shader.
Global Illumination	Toggle **Global Illumination** to diffuse shading due to **Indirect Illumination,** therefore allowing you to view or hide all bounced light on the items in your scene.

Option	Description
	Note: Indirect illumination includes light that has either bounced from other surfaces or has come from the environment but not the light that has come directly from a light source (**Direct Illumination**). For more information about **Global Illumination**, see Render Item: Global Illumination.
Reflections	Toggle **Reflections** to view or hide all reflections between objects and the environment in your scene. **Note: Reflections** don't necessarily involve the environment. For example, the camera might be in a fully enclosed room. For information about Reflection options, see Effect - Texture Item.
Refractions	Toggle **Refractions** to view transparent surfaces as opaque in your scene. For information about Refraction Roughness, see Effect - Texture Item.
Displacements	Toggle **Displacements** to view or hide the effect of all displacement maps in the scene. For more information about Displacements, see Effect - Texture Item.
Draft Displacements	When enabled, automatically halves the associated value for **Displacements**, providing a faster preview evaluation. For more information about Displacements, see Effect - Texture Item.
Fur	Toggle **Fur** in the **Preview** viewport. For more information, see Fur Material.
Draft Fur	Enable to lower the quality of **Fur** in the **Preview** viewport. When

Option	Description
	enabled, Modo automatically halves the associated value for **Fur**, providing a faster preview evaluation. For more information, see Fur Material.
Volumetrics	Toggle **Volumetrics** to view or hide volumetric items as well as volumetric lights. For more information, see Volume Render Outputs.
Motion Blur	Toggle **Motion Blur** in the **Preview** viewport. The motion blur effect is calculated by taking time samples both forwards and backwards from the current frame and blending them together for the final blurred result. For more information about setting **Motion Vector** outputs for **Motion Blur**, see Render Outputs. **Note:** Procedural meshes, created in the **Mesh Operations** list, support Motion Blur but there is a limitation. Elements can only be blurred if a similar element exists on both shutter open and shutter close.
Render Proxy	Toggle **Render Proxy** in **Preview** to completely remove one or more selected Mesh Item layers within a scene, offloading them to an external file that isn't loaded again until render time. For more information, see Render Proxy.
Items	
Show Only Selected Items	Render only selected items in your scene in Preview. When enabled, only items that are selected in the **Item List** are evaluated in the **Preview** viewport. When disabled, all items set to render are evaluated. You can set **Render** in the item's **Properties** panel. For information about selecting items in your scene, see Items (Scenes) List.

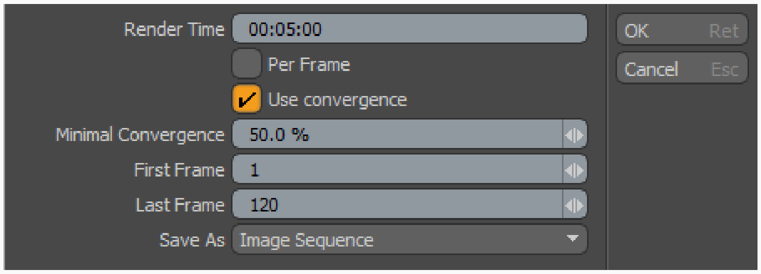
Option	Description
Show Only Visible Items	Set to render only visible items in your scene in the **Preview** viewport. You can set whether to preview items in the **Mesh Operations List**, **Items** by toggling the visibility 👁 button for the required items. For information about setting visibility of items in your scene, see Items (Scenes) List.

IO Options


Progressive Render
Load
Save
Image
Save
Save Layered
Animation
Render Animation
Drawing and Control
Visibility
IO

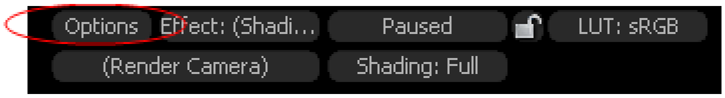
Option	Description
Progressive Render	
Load	Loads any previously-saved progressive render states, and allows the continuation of the render when loaded. **Tip:** Preview saves its full progressive render state when Auto-Save completes a save. You can load the previous render state at a later time. For information about Progressive Rendering Options, see Render Item: Settings.
Save	Saves the current state of the progressive render to disk for further evaluation at a later time.

Option	Description
	⚠ **Warning:** Files may be quite large depending on the scene. **Tip:** Preview saves its full progressive render state when Auto-Save completes a save. You can load the previous render state at a later time.
Image	
Save	Save the current state of the **Preview** window image to a variety of image formats.
Save Layered	Save all Preview layers as a layered image. When enabled, you can simultaneously save all outputs in their current state into a single layered file in either **.psd** or **.exr** format. Each layer is named after the render output it represents.
Animation	
Render Animation	Preview **Render Animation** in the **Preview** viewport. Opens the **Preview Animation Saver** that allows you to leverage the **Preview** window rendering system to render animations. **Note:** This option cannot be used to Render Passes, only the current pass is rendered. Otherwise, all the current settings of the **Preview** window are respected when rendering animations. Once started, a progressive render can only be aborted by pressing the **Abort** button in the progress bar, or by pressing **Esc**. The dialog has a few options:

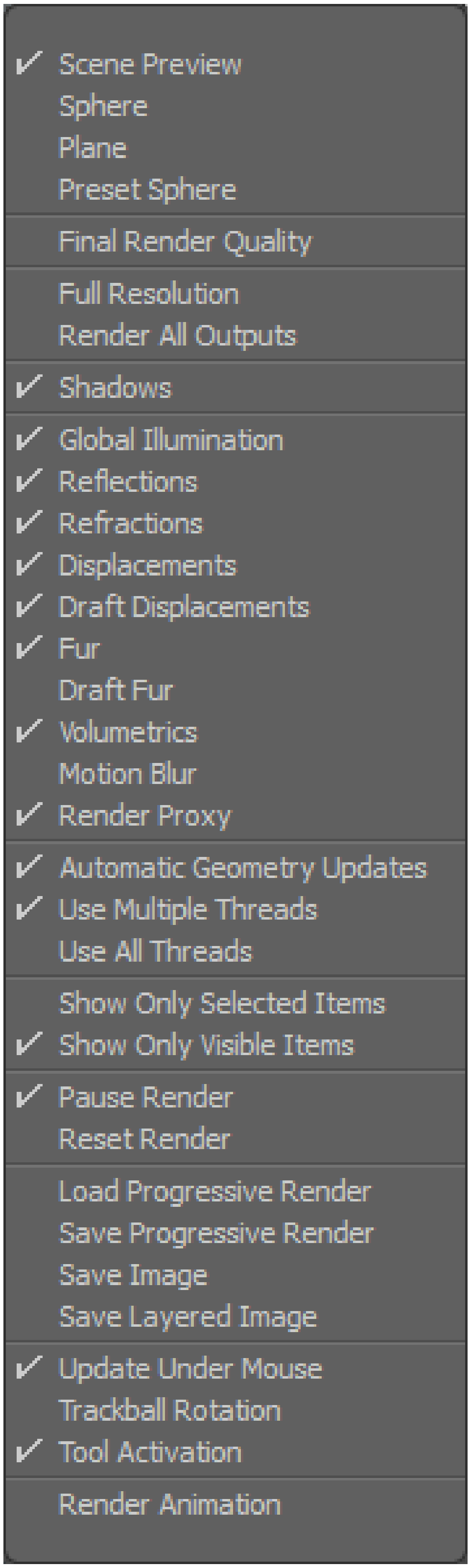
Option	Description
	 Render Time 00:05:00 ☐ Per Frame ☑ Use convergence Minimal Convergence 50.0 % First Frame 1 Last Frame 120 Save As Image Sequence OK Ret Cancel Esc

Options

The **Options** menu provides a quick method for applying Preview Settings to optimize the performance of the viewport by limiting or focusing the rendered data.


Options
Effect: (Shadi...
Paused
LUT: sRGB
(Render Camera)
Shading: Full

You can enable or disable certain functions with a mouse click. The check mark means the feature is enabled; no check mark means it's disabled.


✓ Scene Preview
Sphere
Plane
Preset Sphere
Final Render Quality
Full Resolution
Render All Outputs
✓ Shadows
✓ Global Illumination
✓ Reflections
✓ Refractions
✓ Displacements
✓ Draft Displacements
✓ Fur
Draft Fur
✓ Volumetrics
Motion Blur
✓ Render Proxy
✓ Automatic Geometry Updates
✓ Use Multiple Threads
Use All Threads
Show Only Selected Items
✓ Show Only Visible Items
✓ Pause Render
Reset Render
Load Progressive Render
Save Progressive Render
Save Image
Save Layered Image
✓ Update Under Mouse
Trackball Rotation
✓ Tool Activation
Render Animation

The options here are similar to those available in the Preview Settings. The following additional **Options** menu items are available:

- **Final Render Quality** - when enabled, the following options are enabled:
  - **Full Resolution**
  - **Render All Outputs**
  - All ray tracing options (**Global Illumination**, **Shadows**, **Reflections**, and **Refractions**).

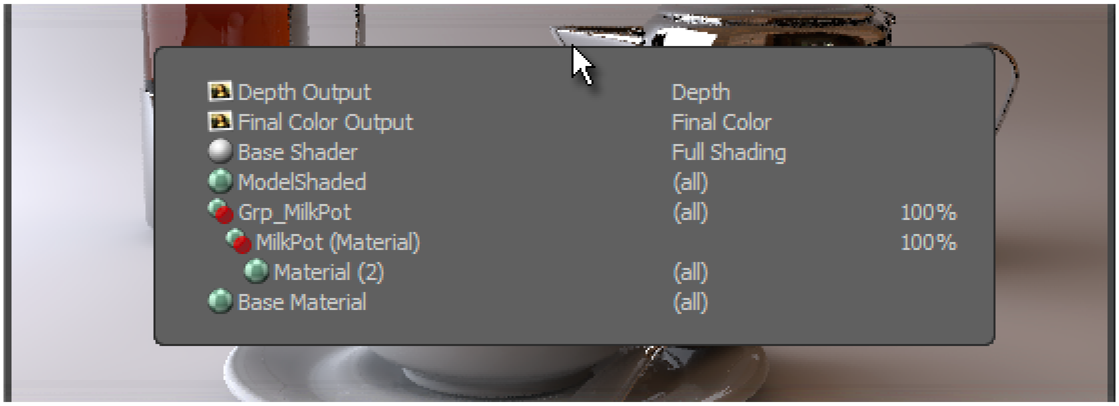
> **Note:** If any of these options are disabled in the Render settings, they are also disabled in Preview.

  - **Displacements** (**Draft Displacements** are disabled)
  - **Fur** (**Draft Fur** is disabled)
  - **Volumetrics**
  - **Motion Blur**
  - **Render Proxy**
  - The render level for both Subdiv and PSUBs is used
  - Cage mode is disabled
  - Secondary **Irradiance Cache** is enabled, unless disabled in the Render settings

  Current render options are restored when Final Render Quality is disabled.
- **Pause Render** - When enabled, the **Pause Render** option stops the preview at its current state with no further evaluations. When disabled, the preview render resets itself and begins evaluating a new image.
- **Reset Render** - Resets the **Preview** window and refreshes the geometry cache and the GI cache if necessary, to render a new image.

Visually Selecting Shader Tree Layers

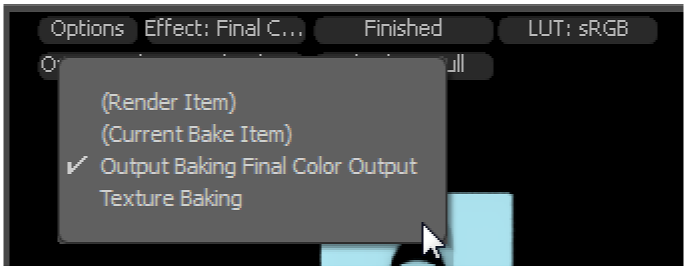
To make it easier to find a particular surface, right-click over any area of the preview image. A pop-over menu displays all **Shader Tree** layers that contributed to the pixel directly under the cursor. Clicking on any of the layers in the dialog selects them in the **Shader Tree**.



Depth Output	Depth	
Final Color Output	Final Color	
Base Shader	Full Shading	
ModelShaded	(all)	
Grp_MilkPot	(all)	100%
MilkPot (Material)		100%
Material (2)	(all)	
Base Material	(all)	

Preview Bake Mode

The Preview Settings allow you to change the **Render Mode** to **Bake** to preview bake results. In **Bake** mode, the camera selector button changes to a bake item selector.



You can choose from the following bake items:

- **(Render Item)** - Bake through the Render Items This lets you select any of the **Render Outputs** that are in the **Shader Tree**.

Note: If you select **(Render Item)**, you must also manually select the UV map to use for the bake. Select the UV map from the **Lists** viewport.

- **(Current Bake Item)** - Bake using the currently-selected bake item.
- Other scene bake items (if there are any).

To select which bake output to display in the **Preview** viewport, click the **Effect** button. Outputs can either be render outputs or texture outputs depending on the selected bake item type. The choice of outputs corresponds to the outputs used in the selected bake item.

Note: For more information, see Baking and Working with Bake Items and Baking Tools.

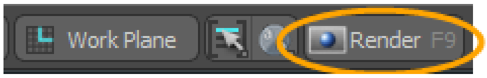
Render Display

The **Render Display** window provides a variety of useful options and feedback for working with the resulting rendered pixels for any Modo scene.

Accessing Render Display

There a number of ways to access the Render Display viewport.

- On the main menu bar, click **Render** > **Render** to open the Image Display and render pixels.
- On the main menu, click **Render** > **Open Render Window** to open the Image Display without rendering any pixels.
- With items selected in the **Items List**, invoke any of the Render Commands.
- With the **Render** layout open, click the start arrow in the Image Display area.
- With any layout open, under the layout menu, click **Render**.


Work Plane
Render F9

Once open, the window provides a view of the rendered scene, along with a number of useful statistics, comparing tools and image adjustment controls for fine tuning an image. The window is broken into four main areas that are all inter-related: Image Display, Control Panel, Render Browser, and the Network View (minimized by default).

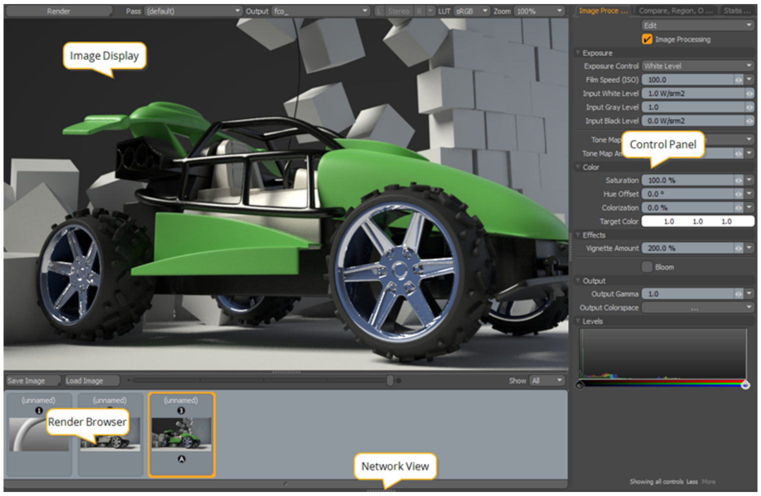


Image Display
Control Panel
Render Browser
Network View
Render
Pass
Output
Stereo
LUT
Zoom
Image Proce ...
Compare, Region, O ...
Statis ...
Save Image
Load Image
Show
(unnamed)
Showing all controls Less More

Pass (default)
LUT sRGB
Zoom 100%
Edit
☑ Image Processing
Exposure
Exposure Control White Level
Film Speed (ISO) 100.0
Input White Level 1.0 W/srm2
Input Gray Level 1.0
Input Black Level 0.0 W/srm2
Color
Saturation 100.0 %
Hue Offset 0.0 °
Colorization 0.0 %
Target Color 1.0 1.0 1.0
Effects
Vignette Amount 200.0 %
☐ Bloom
Output
Output Gamma 1.0
Output Colorspace ...
Levels
Show All

Image Display

This is the main area of the **Render Display**. This is both a final image display and a source of useful visual feedback to the progression of the rendering frame. Directly above this window are the controls that determine what is actually displayed:

- **Render Type** - Sets what is rendered in the **Render Display**, such as **Render All**, **Render Passes**, and **Bake Texture**. Click and hold the **Render Type** button to display the available options.
- **Render Pass** - Sets which pass or passes are rendered from the specified **Render Type**.
- **Render Output** - Sets the output type rendered from the selected **Render Type**, such as Full Color Output (**fco_**).
- **Stereo Options** - Sets which individual view to display from stereo renders, or how to display both views simultaneously, such as anaglyph or side-by-side.
- **LUT** - The LUT currently selected for display purposes on your monitor (not to be confused with the output colorspace).
- **Zoom** - Controls the size of the rendered frame. You can also zoom in the window using **Ctrl**/**Cmd**+**Alt**+click and drag. Once zoomed in, click and drag to navigate across the image. Pressing **Ctrl**/**Cmd**+**A** fits the entire image to the viewable area.

Control Panel

The **Control Panel** is organized into three tabs, each with a dedicated function:

- **Image Processing** - Provides non-destructive controls to adjust, correct, and tone the rendered image.
- **Compare, Region Options** - Provides tools for rendering limited regions and comparing subsequent and prior renders.
- **Statistics** - Provides dynamic information regarding the frame during rendering, such as memory usage, elapsed time, and the final numbers of polygons.

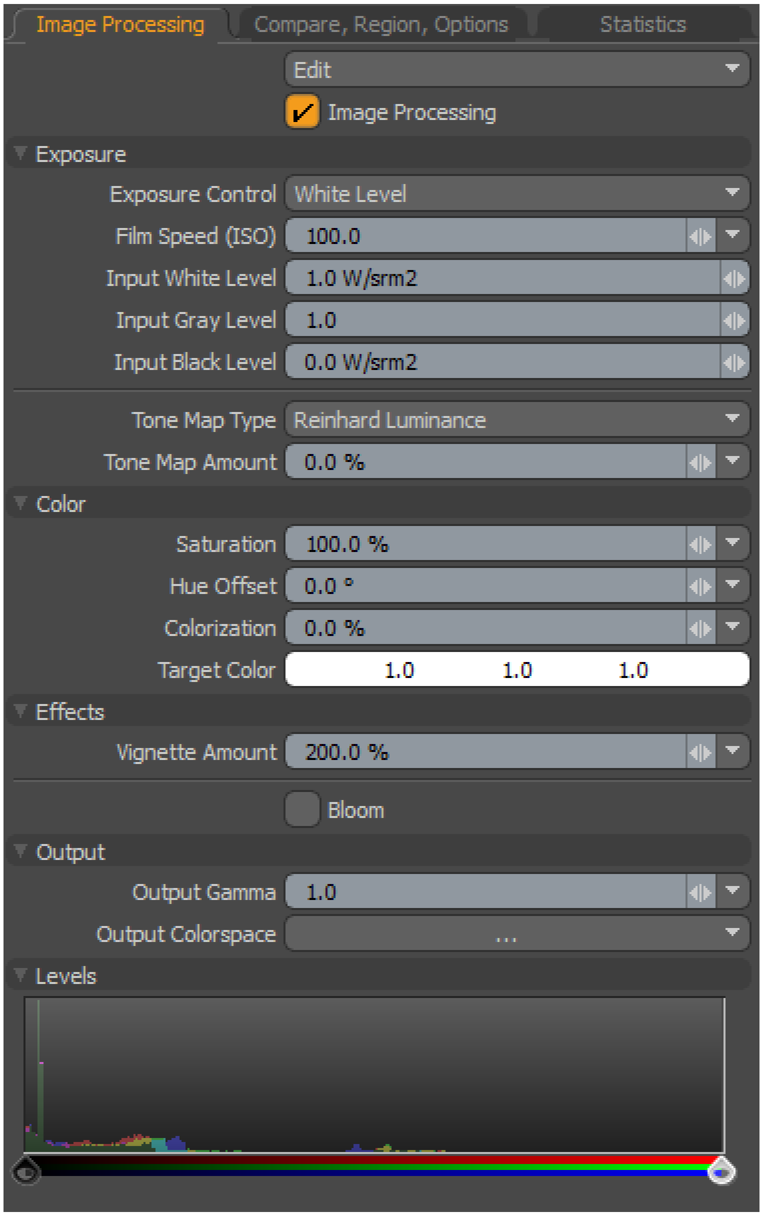


Image Processing
Compare, Region, Options
Statistics
Edit
Exposure
Color
Effects
Output
Levels

☑ Image Processing

Exposure Control: White Level
Film Speed (ISO): 100.0
Input White Level: 1.0 W/srm2
Input Gray Level: 1.0
Input Black Level: 0.0 W/srm2
Tone Map Type: Reinhard Luminance
Tone Map Amount: 0.0 %
Saturation: 100.0 %
Hue Offset: 0.0 °
Colorization: 0.0 %
Target Color: 1.0 1.0 1.0
Vignette Amount: 200.0 %
☐ Bloom
Output Gamma: 1.0
Output Colorspace: ...

Note: The **Levels** histogram is hidden by default. Click the **Showing Standard Controls More** button at the bottom of the panel to display **Levels**.

Image Processing

The **Image Processing** section provides controls to adjust the look of the final rendered image. It can replace an external image editor, but best of all, it's non-destructive. Adjusting the values in the panel updates the **Image Display**, providing real-time feedback to the changes, making it easy to experiment and try different combinations.

Option	Description
Edit	Provides options for moving image processing settings between the render output and the **Render Display** window. • **Copy From Scene** - If adjustments were made to the render output options after the rendered image completes, the **Copy from Scene** option grabs those settings, updating the options of the **Render Display** window to match. • **Paste to Scene** - Any adjustments made to the rendered image within the **Render Display** window do not apply to subsequent renders, as the **Render Display** window reverts to the settings of the **Render Item**. The **Paste to Scene** option copies the settings to the render output back to the scene, applying it to all subsequent renders. • **Reset** - Returns all image adjustment options to their default values.
Image Processing	Toggles the effect of the **Image Processing** controls on and off, allowing you to quickly switch between the previous image and adjusted image.
Exposure	
Exposure Control	Sets the exposure preset to apply to the image: • **White Level** - Defines the amount of luminance (illumination brightness) that corresponds to pure white in the rendered image, adjusted with the **Input White Level** value. • **Photographic** - Produces similar results to **White Level**, but uses the **Film Speed (ISO)** setting more familiar to photographers.
Film Speed (ISO)	Sets the sensitivity of the film or sensor when **Exposure Control** is set to **Photographic**. Higher values simulate increased exposure time, but without any added noise.
Input White Level	Specifies the radiance or luminance level that corresponds to pure white in the final rendered image (a pixel color value of 1.0), when **Exposure Control** is set to **White Level**. Higher values tend to darken the image, requiring a much brighter area to represent the whitest white, but can also be used to bring out detail in overblown areas.

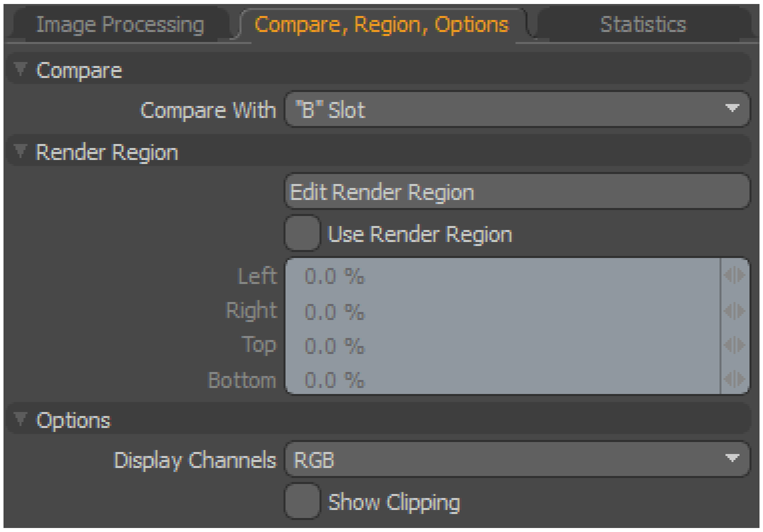
Option	Description
	Note: To use the input level controls effectively, **Clamp Colors** must be disabled in the **Shader Tree** > **Render Output** sub-tab.
Input Gray Level	A non-linear luminance adjustment that applies a curve-like function to the pixels, modifying the mid-range the most while attenuating the adjustment amount for values moving toward the defined **Black** and **White** points. Values above 1.0 lighten the mid-range, while values below 1.0 darken the mid-range.
Input Black Level	Specifies the radiance or luminance level that corresponds to pure black in the final rendered image (a pixel color value of 0.0). **Note:** To use the input level controls effectively, **Clamp Colors** must be disabled in the **Shader Tree** > **Render Output** sub-tab.
Tone Map Type	Sets the mapping type used to control tonal balance. Select the required type and then use the **Tone Map Amount** to control the range.
Tone Map Amount	Controls the overall tonal balance of the rendered image, compressing the rendered dynamic range into something viewable, where shadows and highlights can reveal previously obscured details. Often, it takes a combination of settings for white level, tone mapping, and gamma to achieve the best results. For further examples, see Tone Mapping. Dynamic range is measured as the difference between the darkest shadow to the brightest highlight in an image. Most image formats' dynamic range pale in comparison to that of the world around us. In the same way that a photographer struggles to capture all of the range our eyes can see, you may find your renders have shadow areas that are too dark, and highlight areas that are too bright. Luckily, Modo renders in full floating point accuracy, providing dynamic range well beyond what any monitor is capable of displaying. **Note:** To use the input level controls effectively, **Clamp Colors** must be disabled in the **Shader Tree** > **Render Output** sub-tab.

Option	Description
Color	
Saturation	Controls the concentration or amount of color in an image independent of the luminosity of brightness. At 100%, colors are fully saturated as defined in the item's material settings. Reducing the value reduces the overall color saturation to a value of 0%, producing an image only with gray shades. **Tip:** Modo renders in full 32-bit floating point accuracy, so values above 100% increase saturation without introducing the color banding and artifacts generally associated with over-saturating images.
Hue Offset	Adjusts the color values of the rendered image independently of the luminosity or brightness values, shifting them across the entire spectrum in a sequential fashion. Hue values are arranged on a standard colorwheel, where a rotation of 180° inverts all the color values. For example, shifting a red color in a positive direction adjusts reds toward an orange hue, then yellow, and so on.
Colorization	Introduces an overall color tint into an image. This can be purely for artistic reasons, but can also be helpful in reversing color casts introduced by image-based lighting among other things. Define a tint color using the **Target Color** control and then adjust the **Colorization** amount to determine the strength of the color in the image. A value of 0% produces no colorization, attenuating toward 100% where the image is replaced fully by the **Target Color**.
Target Color	Specifies the tint color when the **Colorization** control is set above 0%.
Effects	
Vignette Amount	Setting **Vignette Amount** to 100% creates a physically accurate vignette simulation, a natural reduction of the brightness of an image toward the outer edges of the frame, based on the camera's focal length and film size. Values above 100% darken the resulting vignette, and values below 100% reduce its visibility.

Option	Description
Bloom	When enabled, two additional options display, **Threshold** and **Radius**. Adjusting these controls provides near real-time feedback in the **Render Display** window. In photography, when a very bright part of an image neighbors a very dark part, the bright part appears to glow. This phenomenon is known as **Bloom**. Modo's **Bloom** simulates this effect by removing excess energy from a pixel and distributes it around to its neighbors.
Bloom Threshold	Sets the lower threshold to which pixels are effected, when **Bloom** is enabled. When set to 100%, only the hottest specular highlights bloom, whereas at 0%, every pixel in the scene receives the effect. For examples, see Basic Render Outputs. **Note:** This control is only available when **Bloom** in enabled.
Bloom Radius	Sets the distance that energy is distributed, in effect controlling the size of the glow. Higher values can impart an ethereal look to the rendered image. For examples, see Basic Render Outputs. **Note:** This control is only available when **Bloom** in enabled.
Output	
Output Gamma	Controls the amount of gamma applied to the rendered image, for both display and when the image is saved. It can be used as an image adjustment control to lighten or darken the image without the need to externally adjust it in an external bitmap editor. The gamma adjustment is non-linear and affects the mid-tones of the image more than the shadows and highlights. For best results, leave the gamma value at the default 1.0.
Output Colorspace	Defines the colorspace of the rendered frame. When set to **default**, the colorspace definition is derived from the Scene Item settings. Changing the value here results in an override that can be set per output.

Option	Description
	The default settings generally produce the correct results, but you can experiment as required due to the non-destructive nature of the Render Display. For more information on color management, see Color Management.
Levels	
Levels	Slides all color values up or down the value scale, shifting them identical amounts in the rendered image. For example, if an image has a 0% black pixel and a 50% gray pixel, offsetting the values 50% results in the black pixel changing to 50% gray and the gray pixel changing to 100% white with all other values changing with the corresponding amounts. **Note:** The **Levels** histogram is hidden by default. Click the **Showing Standard Controls More** button at the bottom of the panel to display **Levels**.

Compare, Region, Options



Compare With: "B" Slot

☐ Use Render Region

Left 0.0 %
Right 0.0 %
Top 0.0 %
Bottom 0.0 %

Display Channels: RGB

☐ Show Clipping

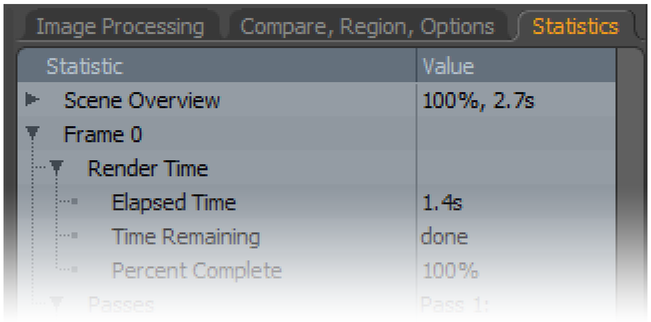
Option	Description
Compare - These options allow you to examine two images in a variety of ways, making it easier to contrast any differences between them.	

Option	Description
Compare with	Determines which opposing images are compared: • **None** - No image comparison • **"B" Slot** - Two separate images are compared. **Slot A** is determined by the current image selected in the browser and **Slot B** is determined by holding the **Ctrl/Cmd** key and clicking on a second image slot in the browser. A small **A** and **B** icon appears beneath each image, signifying which is which. • **"A" Slot Output** - Compares passes or outputs associated with a single image, selected using the **Pass** and **Output** controls.
Pass	When the **"A" Slot Output** option is selected, determines which **Pass** to compare against the default pass.
Output	When the **"A" Slot Output** option is selected, determine which **Output** to compare against the default output.
Apply Image Processing	Toggles the effect of the **Image Processing** controls on and off, allowing you to quickly switch between the original image and adjusted image.
Compare Mode	Several modes define how the two images are compared, each with additional options. • **Wipe** - The images are overlaid and the top image is wiped away, revealing the lower image. Use the **Wipe Method**, **Position**, and **Opacity** Controls to set the wipe behavior. • **Spotlight** - Reveals the **B** images as a circle around the pointer (move over the image to see the results). The size of the circle is determined by the **Spotlight Radius** control. • **Side by Side** - Displays two half-width images next to each other.
Wipe Method	When **Compare Mode** is set to **Wipe**, allows you to select **Horizontal**, **Vertical**, or **Dissolve** wiping.
Position	When **Wipe Method** is set to **Horizontal** or **Vertical**, sets the position of the wipe widget in the viewport.
Opacity	When **Wipe Method** is set to **Dissolve**, sets opacity of the top image to reveal the image underneath.

Option	Description
Spotlight Radius	When **Compare Mode** is set to **Spotlight**, allows you to control the radius of the spotlight.
Stack Vertically	When **Compare Mode** is set to **Side by Side**, enable this control to stack the images vertically instead of horizontally.
Difference	Calculates the pixel variance between the two images and displays it as a grayscale image. Black indicates no difference between the two pixels, attenuating toward white, which indicates the maximum difference. The amount of difference displayed can be scaled using the **Difference Scale**. Each of the available options displays the difference between the two images for the chosen aspect, such as the difference between the **Hue** of both images, or the difference of the **Saturation** amount. Utilizing the **Difference** option can be especially helpful in pinpointing subtle differences between two similar images. **Note:** This is mostly for high dynamic range images (HDR), but in some cases it provides a better indication of dramatic or subtle differences. The difference calculation is the same math used to determine the difference between two numbers (subtracting one value from the other).
Difference Scale	Scales the amount of difference for the selected **Difference** option. **Note:** This control is only available when **Difference** is set to something other than **none**.
Render Region - Allows you to selectively render a smaller segment of the entire image, making the iterative process of creating a final rendered image faster and more fluid. For convenience, these settings duplicate the functionality of the render item's **Render Region** functionality. For more details on the **Render Region**, see Render Item: Frame	
Edit Render Region	When enabled, you can drag directly over the rendered image to define the limited region area interactively. When the initial area is defined, edges and corners of the region are highlighted, allowing you to further refine the region. Click and drag the highlighted border to adjust.

Option	Description
	If edits are made to the same scene, the area outside the defined region is used as a background to successive renders until disabled. When switching between scenes or image resolutions, the background area outside the region renders black.
Use Render Region	Temporarily enables and disables the **Edited Render Region** without losing any of the settings.
Left/Right/Top/Bottom	Allows you to specify the total rendered region width and height manually.
Options	
Display Channels	All resulting rendered images are generated as standard RGB bitmap format images. For diagnostic purposes, it may be useful to see the individual **Red**, **Green** or **Blue** color channels of an image.
Show Clipping	When enabled, a diagonal black and white stripe pattern (called a zebra pattern) displays over overexposed or blown out areas of the image, providing useful feedback for adjusting image white point and tone mapping an image to compress as much dynamic range information as possible.

Statistics

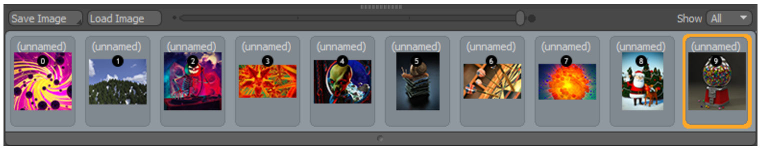


Statistic	Value
Scene Overview	100%, 2.7s
Frame 0	
Render Time	
Elapsed Time	1.4s
Time Remaining	done
Percent Complete	100%
Passes	Pass 1:

The **Statistics** panel provides a huge amount of statistical information related to rendering the scene. Included in this information is the elapsed time, as well as an approximate remaining time for in-progress frames. For finished frames, the total rendering calculation time is displayed, along with total memory usage, and total number of polygons generated.

Navigation of the list works the same as other viewports. You can open or close a specific section by clicking the preceding arrow (▸). To open or close all sub-sections of a specific topic, press **Shift** and click preceding arrow (▸) to view all the information related to a single frame.

Render Browser



The **Render Browser** allows for an unlimited number of rendered frames. The default is set to show the past 10 rendered frames, but you can change this in the **Options** panel. The browser itself serves a couple of purposes, the most important being that you can select the current image to display in the display area by clicking on any of the icons. The thumbnail of the selected image is highlighted in orange; only the selected image can be modified using the image adjustment controls. The render order of images is also displayed using the small number icons above each image. You can use the keyboard numeric keys to select a specific render slot as well as easily swap between different renders, simply by pressing its corresponding numeric key.

When you hover over an icon, two additional controls appear: an **X** that removes the image from the browser (deleting it from disk) and a small lock icon you can click to stop items being deleted accidentally. Click the lock icon again to unlock. You can select multiple images by holding the **Ctrl**/**Cmd** key, allowing you to remove, lock, or unlock multiple images at once.

The **Render Browser** is also used to load and save images. Images in the **Render Browser** are held in the render buffer until you save them, but you can still view and modify them in the **Render Display**. Using the **Save Image** button directly above the browser allows you to define a format and location for the image. For more information concerning images and formats, see Saving Images.

You can also load images into the browser area, which is very helpful when you want to apply image adjustments, or use previous renders to compare a current render, such as when the render buffer no longer exists. To load an image:

1. Click the **Load Image** button above the browser area.

   An OS-specific browser displays.
2. Navigate to the stored image location and select the target file.
3. Click **Open** to load the file into the browser.

Note: If any modifications are applied to the file, it must be saved again to retain the changes.

Network View

Below the **Render Browser** area is a minimized **Network View** panel. Open the panel by clicking on the small group of tiny squares and dragging it upward. This browser indicates the number of cores available on each machine, machine name, user name, mode, status and message. You can also tell if a machine is connected to your active host by the color of the text:

- **Green** - Indicates that the machine is connected and ready for rendering.
- **Black** - Indicates systems that are connected, but not enabled as workers.

Any other color indicates a problem with the connection or availability.

General View Shortcuts

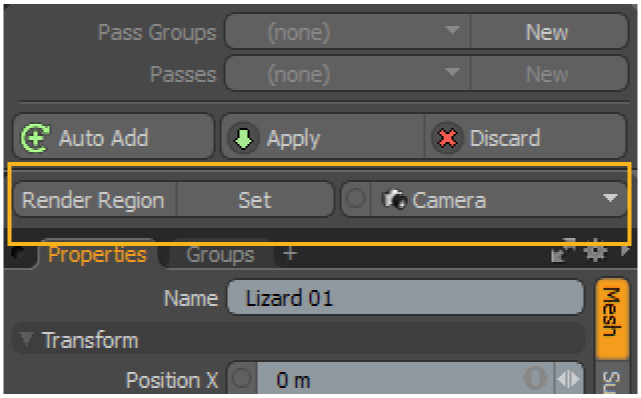
The **Render Display** provides standard Modo controls to adjust size and minimize frames, all the while automatically adjusting the interface, giving you control over what can be seen. Dragging the splitters between the panels scales the contents of each. There are also some additional controls that provide an easy way to quickly make certain panels within the display visible.

- **Spacebar** - Minimizes everything except the image pane.
- **T** - Toggles thumbnail panel visibility.
- **N** - Toggles the network view.
- **I**/**C**/**S** - Toggles the right-hand panel tabs (**Image Processing**/**Compare**/**Statistics** respectively).

Render Region

The 3D process is one of constant iteration and refinement. Rendering entire images repeatedly is a significant waste of time and a disruption to the creative workflow. To alleviate some of this pain, Modo provides the Render Region rendering functionality. You can use the **Left**/**Right**/**Top**/**Bottom** functions to specify the region manually, or use the Render Region tool to specify an area by dragging in any 3D viewport.

You can find the tool in the left panel of the **Render** layout, in the Render Region toolbar. Using the toolbar, you can activate and set a render region in preview, and change the camera being used to render it.



Pass Groups (none)

Passes (none)

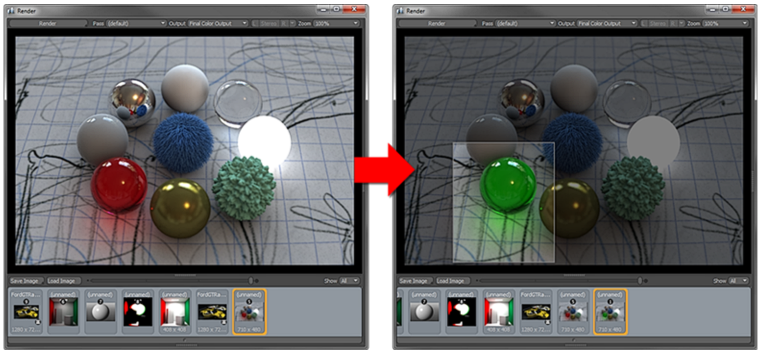
Name Lizard 01

Position X 0 m

To use the tool:

1. Click **Render Region** to activate the tool.
2. To set a region, click **Set**, then click and drag in the 3D viewport to specify area.

   When a render command is invoked (such as by pressing **F9**), only the area within the region specified is rendered.



To change the render camera, click **Camera**, and select the camera you need from the dropdown.

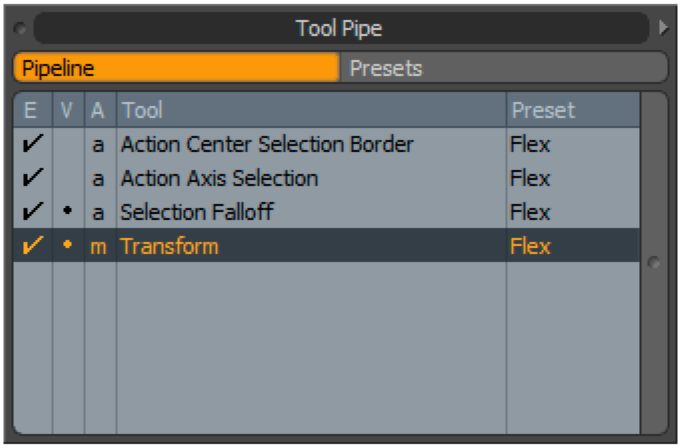
Additionally, the function can be activated through a button of the **Render Display** window, under the **Options** tab.

Tool Pipe

In any 3D modeling toolset, each tool needs to infer, or be informed about, several key variables in order to work properly. It needs to know its central position, meaning the location in 3D space, from where the desired effect is applied. It also needs to know its operational axis or orientation. And, finally, it needs to know which portion of the scene is going to be affected and by how much. In Modo, each of these variables has been extracted to its own operational layer. Modo offers each as a building block individually, such as Specifying Action Centers and Falloffs , which combines a center location and an axis orientation, andUsing

Falloffs that provide a means to automatically attenuate or fade the strength of a tools' action across a distance.

You can assemble these blocks to make custom tools. Taking a basic Rotate tool and adding a linear falloff makes it a twist tool. Taking a Move tool, and adding an element action center and radial falloff, makes it a Soft Move tool. However, having to manually define each of these variables each time would get very tedious, so Modo has a system of defining collections of these functions as presets. As a matter of fact, this is how all the tools in Modo are built. They are essentially pre-built combinations of a variety of functions packaged into a single tool. All of these presets are defined with the **Tool Pipe** viewport.



E	V	A	Tool	Preset
✓		a	Action Center Selection Border	Flex
✓		a	Action Axis Selection	Flex
✓	•	a	Selection Falloff	Flex
✓	•	m	Transform	Flex

The **Tool Pipe** is a seemingly simple but very powerful viewport that allows the combination of the tools and the tool modifiers, activating the proper functions automatically. It is also useful as a simple view to the currently-active tool combination. There are many settings available through the **Tool Pipe** interface using context menus on the specific **Tool Pipe** components. The **Tool Pipe** also acts as a list, which shows all the tools in the current preset combination, with the action tool at the bottom.

In the default Modo setup, the **Tool Pipe** is found in the **Model** interface layout. It is located on the right-hand side of the screen, under the **Lists** tab (next to the **Properties** viewport tab). The viewport itself is divided into several columns, where tools can be made visible or invisible, and enabled or disabled. They can also be removed, renamed, or reset. The columns are labeled **E**, **V**, **A**, **Tool**, and **Preset**.

E column

The **E** column indicates the state of the enable/disable toggle. When the check mark is visible in this column, the tool or modifier is enabled. This can be very useful when working with modifiers such as falloffs or guides, and it allows you to quickly disable the effect of the modifier without removing it from the **Tool Pipe** completely. It also allows the modifier to be re-enabled with its settings intact.

V column

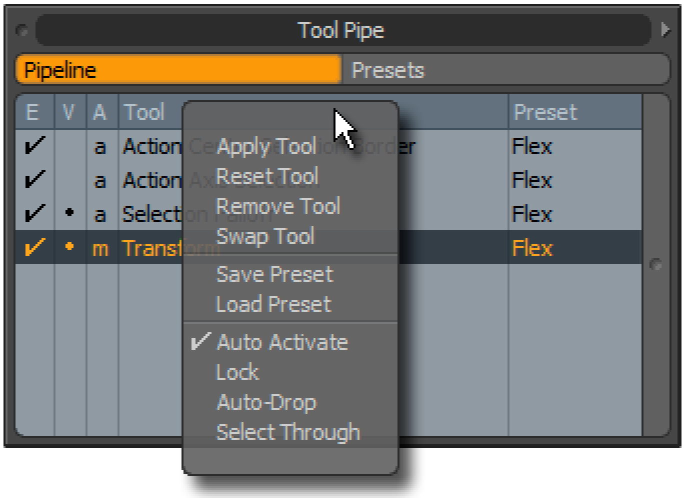
The **V** column indicates the handle visibility setting for the current tool or modifier. There are several visibility states available for most tools. Right-clicking on this column opens the contextual menu. The default setting (in most cases), **Preference**, uses the tool handle drawing setting in the **Preferences** file. This allows you to globally set the tool handles to a specific style. The **Advanced** setting, in many case, adds extra handles, or in-viewport display elements that add additional controls for the tool or modifier. The **Basic** setting uses the core set of primary tool handles available, and the **Invisible** setting hides the handles altogether, if you prefer the hauling behavior over handle manipulation. (Hauling is off-handle tool manipulation achieved by clicking and dragging in the viewport when the tool is active.) The icons in the column indicate the current state. A dot designates **Preference**, a **-** (dash) for **Basic**, a **+** (plus) for **Advanced**, and nothing for **Invisible**. You can quickly toggle between the **Invisible** and **Preference** settings by clicking in this column over the target element.

A column

The **A** column indicates the tool or modifier's current state. This columns lists three possible states: **m** for manual, **a** for automatic, and blank for inactive. When a tool or modifier is set to **Manual**, it is then defined by user input; when set to **Automatic**, it is derived automatically by the system.

Tool column

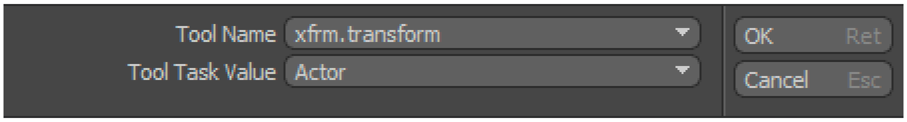
The **Tool** column shows a list of the tools and modifiers that currently occupy the tool pipe. Right-clicking on this column opens up yet another contextual menu that allows the user to **Apply**, **Reset**, **Remove**, or **Swap** a tool from the **Tool Pipe**, as well as **Save** and **Load** new presets and set the activation, locking, and selection style states.


Tool Pipe
Pipeline
Presets
E
V
A
Tool
Preset
Action Ce
Action Ax
Selection
Transform
Flex
Flex
Flex
Flex
Apply Tool
Reset Tool
Remove Tool
Swap Tool
Save Preset
Load Preset
Auto Activate
Lock
Auto-Drop
Select Through

- **Apply Tool** - applies the tool's settings using the attributes defined in the **Properties** panel. This does not result in the tool being activated. It is the same as the more convenient **Apply** button found in the **Tool Properties** panel.

  Many of the transform tools are set to **Auto Activate**, which makes the **Apply** option irrelevant. However, in some cases when working with an enormous number of polygons where editing feedback would be diminished, it would be beneficial to disable the **Auto Activate** feature and simply define a value for the tool, then **Apply** those settings. This allows any of Modo's tools to be used non-interactively.
- **Reset Tool** - resets the properties of the currently-selected tool or modifier to the pre-defined default state. This is useful to return a tool to its initial state when a user has arrived at an unsatisfactory result, where it would be easier to simply start over.
- **Remove Tool** - removes a tool or modifier from the **Tool Pipe** stack without affecting the other components of the **Tool Pipe**.
- **Swap Tool** - opens a dialog that allows the user to swap out the target in the **Tool Pipe** stack for any other tool or modifier.

  The resulting dialog organizes each tool or modifier by its functionality and then by the individual command used to activate that option. The ones of particular interest in the case of modeling tools are the **Falloffs**, **Visual Aids**, **Action Centers**, **Snap**, **Constraint**, **Action Axis**, and **Actor** (**Actor** being all the basic tools).


Tool Name xfrm.transform
Tool Task Value Actor

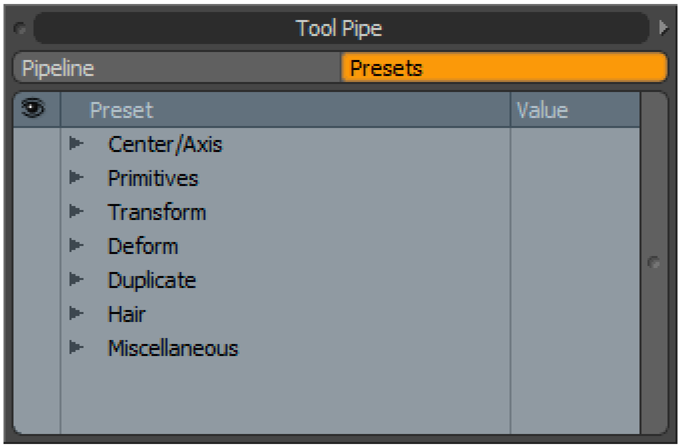
- **Save Preset** - allows the entire contents of the **Tool Pipe** to be captured as a preset that can be recalled at a later time using the **Load Preset** option or using the command toolset.
- **Load Preset** - inserts the contents of a preset configuration into the existing **Tool Pipe**. If there are components of the preset that collide with any elements present when it is loaded, the components being loaded take precedence. Any contents that don't overlap remain in place.
- **Auto Activate** - this is a useful toggle that can help reduce a click when working with certain tools. When active, a tool becomes activated immediately upon clicking its button or pressing the hotkey. All of the transform tools in Modo are set to **Auto Activate** and this is why their handles appear immediately in the 3D viewport. Geometry creation tools, such as the Cube or Bevel tools, are not auto activated, so they require an initial click in the 3D viewport. This allows you the opportunity to set values on the **Property** viewport prior to activating, or simply applying the tool.
- **Lock** - keeps the current state of the tool's properties as the default setting each time the tool is enabled. For instance, if you use the Cube tool to drag out a 1 meter cube and then activate the **Lock** flag, whenever you activate the Cube tool, the numeric properties are set up for a 1 meter cube. Without **Lock** on, each time you complete the Cube tool, the resulting values become the values for the next activation.
- **Auto-Drop** - most useful for painting style tools, such as Mesh Paint or 3D painting operators. With most tools, when you release the mouse, the tool is still active and allows you to edit the current values. For

instance, if you activate Bevel, you can click and drag repeatedly in the 3D view to edit the same Bevel settings. If you **Shift**+click you start a new bevel. If the same were true of Mesh Paint, each time you clicked the previous mesh, placement would jump to your mouse rather than creating a new mesh. With **Auto-Drop** active, each new stroke is a new Mesh Paint. If you set the Bevel tool to **Auto-Drop**, you could quickly click and drag multiple times to create complex bevels.

- **Select Through** - a highly useful option for a specific workflow, where you may not want to drop tools to change their current selection. In the default mode, Modo uses any click away from a tool's handle to haul the primary values. If you set the tool to **Select Through** you no longer have hauling behavior but get selection events instead.

Preset Column

The **Preset** column displays the saved preset name to which the function belongs to, or the tools command. The **Tool Pipe** presets view displays all the available presets organized by tool task.

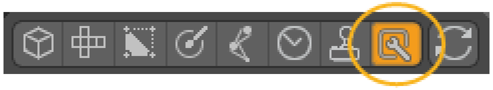


Form Workbenches

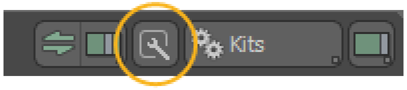
Workbenches are forms that you can edit in the UI. They allow you to group and organize tools, commands, channels, scripts, macros, and so on.

You can open a workbench in two different places.

- To open a vertical workbench, in the **Model** layout click the Workbench button at the top of the left panel.

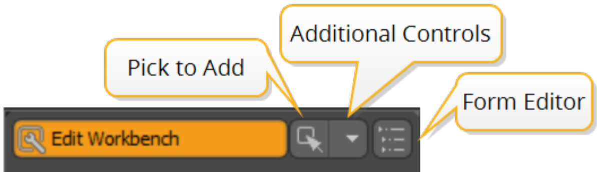


- To open a horizontal workbench, click the Workbench button in the top-right area of the UI.



Editing Your Workbench

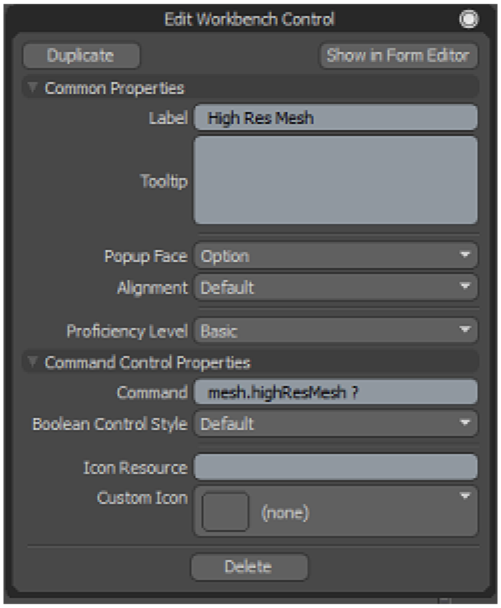
To start customizing your workbench, click the Edit Workbench button. This reveals three additional buttons. In the horizontal version, the same functions can be found by clicking the Additional Controls button.



Click the Pick to Add button. This allows you to click on an element in the UI and add it to the workbench. You can pick buttons, properties, presets, and even open the **Command History** and click any of the commands used previously to add it to the workbench. As you hover over elements with Pick to Add active, they are highlighted in green.

Note: Make sure you have the required menus open before activating Pick to Add.

Once you've added an item to your workbench, while in Edit mode, you can right-click to further customize it.



Common Properties

Label: High Res Mesh

Tooltip:

Popup Face: Option

Alignment: Default

Proficiency Level: Basic

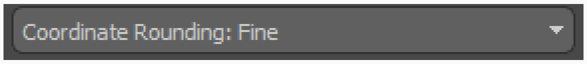
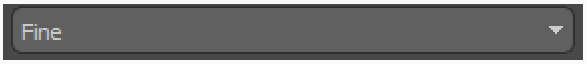
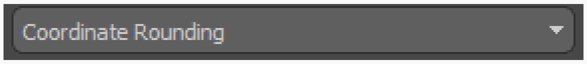
Command Control Properties

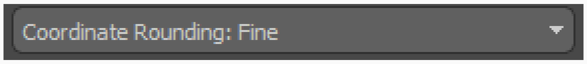
Command: mesh.highResMesh ?

Boolean Control Style: Default

Icon Resource:

Custom Icon: (none)

Common Properties	
Label	The text on the button.
Tooltip	The text of the tooltip, which displays when you hover the cursor over the button.
Popup Face	Determines what appears on the face of a popup control. The following options are available: • **Option** - Display the selected option if available, otherwise display **(none)**.  Coordinate Rounding: Fine • **Option or Label** - Display the selected option if available, otherwise display the label.  Fine • **Label** - Display the label only.  Coordinate Rounding • **Label: Option** - Display a combination of the label and the option in the format 'Label: Option'. Display **(none)** if no option is selected.

	 Coordinate Rounding: Fine • **Label: Option or Label** - Display a combination of the label and the option in the format 'Label: Option'. Display label only if no option is selected.
Alignment	Determines how the controls in a form are aligned. **Wide** makes the control wider to show more of the body, and **Full** makes the control the full width of the form.
Proficiency Level	The skill level/utility of the command, which can be used to filter what controls are shown in the form. For more information, see **Base Form Proficiency Level** in Defaults Preferences. The following options are available: • **Basic** - Core functionality that should always be visible. • **Standard** - Functionality that you can use to get the most out of a feature. • **Advanced** - Functions used by advanced users for special behaviors.
Command Control Properties	
Command	Allows you to edit the command directly.
Boolean Control Style	The style of the control. You can use the **Default** option, which is the style defined by the command, a **Checkmark**, or a **Toggle Button**.
Icon Resource/Custom Icon	Allows you to specify a custom icon from disk.

In addition to the options above, this panel allows you to **Duplicate** or **Delete** the control, or to **Show in Form Editor**.

Tool Presets, Macros, and Scripts

In addition to adding UI items, you can add tool presets, macros, and scripts to your workbench. You can find these options by clicking the Additional Controls ▾ button.

Click **Add Current Tool Pipe and Save Preset** to save the currently created tool preset and add it to the workbench.

Tip: For more information on creating tool presets, see Tool Pipe.

To add a macro, record it, then either click **Add Recorded Macro and Save to File** to save it as an **.lxm** file, or **Add Recorded Macro and Store in Config** to store it in the user config. In both cases, a button is added for the macro to the workbench.

Note: For more information on recording macros, see Macro Recorder.

You can add a script in two ways. To add one from the **Script Editor**:

1. Open the **Script Editor** containing your script. You can find it in the menu bar, under **Layout** > **Palettes** > **Scripting**.
2. In the workbench, click the Additional Controls ▾ button, then click **Add Script from Script Editor**.

   This opens a file browser.
3. Name your script and navigate to the folder to which you want to save it, then click **Save**.

   Your script is saved on disk and a button for it is added to your workbench.

To add a script from disk:

1. Click the Additional Controls ▾ button, then click **Add Script File**.

   This opens a file browser where you can select the script to use.
2. Select the script and click **Open**.

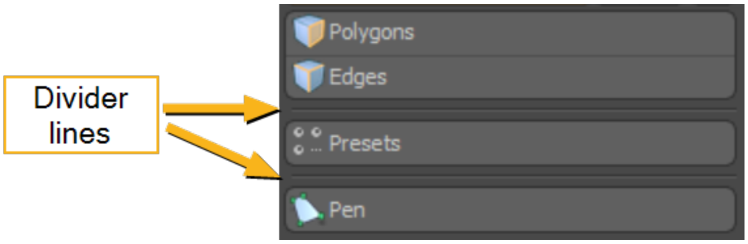
   The script is added to your workbench.

Modo allows you to open a macro or script directly from the workbench. Click **Edit Workbench**, then right-click the macro's or script's button. In the context menu, under **Script** you have the following options:

- **Reveal in File Viewer** - Open the file's location in a file browser.
- **Open in External Editor** - Open the file in a text editor.
- **Open Python in Script Editor** (Scripts only) - Open the file in Modo's **Script Editor**.

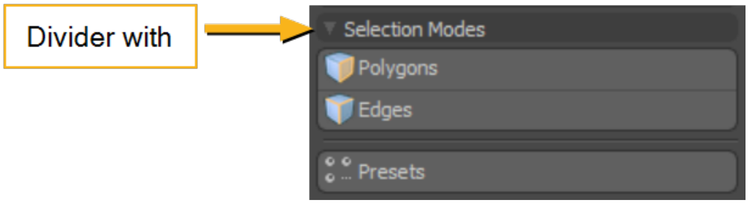
Adding Dividers

You can add horizontal divider lines between the buttons in your workbench. To do this, click the Additional Controls ▾ button, then click **Add Divider**. A divider line is added to your workbench and you can drag it into place.



To add a divider with a label, click **Add Divider with Label**. This adds a divider with a header, and makes the section collapsible.

> **Note:** You can't collapse a section when **Edit Workbench** is enabled.

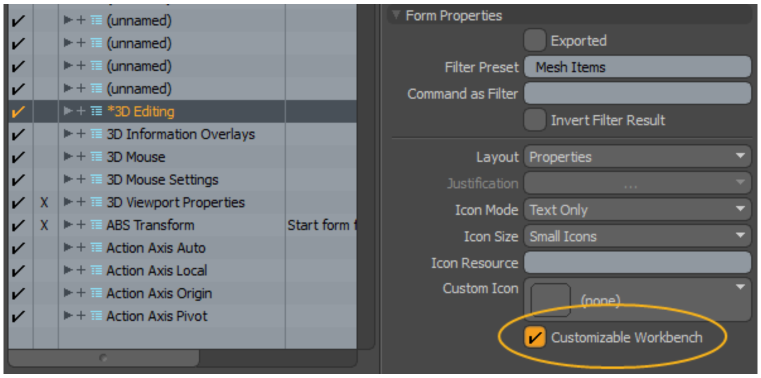

Divider with
Selection Modes
Polygons
Edges
Presets

Customizing Existing Forms

Using the **Form Editor**, you can turn any form in Modo into a workbench.

1. Open the Form Editor from the menu bar by clicking **System** > **Form Editor**.
2. In the **Form Editor**, click a form on the left.

   This opens the form's properties on the right.
3. Under **Form Properties** at the bottom, enable **Customizable Workbench**.


(unnamed)
(unnamed)
(unnamed)
(unnamed)
*3D Editing
3D Information Overlays
3D Mouse
3D Mouse Settings
X
3D Viewport Properties
X
ABS Transform
Start form
Action Axis Auto
Action Axis Local
Action Axis Origin
Action Axis Pivot
Form Properties

☐ Exported
Filter Preset: Mesh Items
Command as Filter:
☐ Invert Filter Result
Layout: Properties
Justification: ...
Icon Mode: Text Only
Icon Size: Small Icons
Icon Resource:
Custom Icon: (none)
☑ Customizable Workbench

This adds an **Edit Workbench** button to that form, and you can customize it as described above.

Using the Input Editor

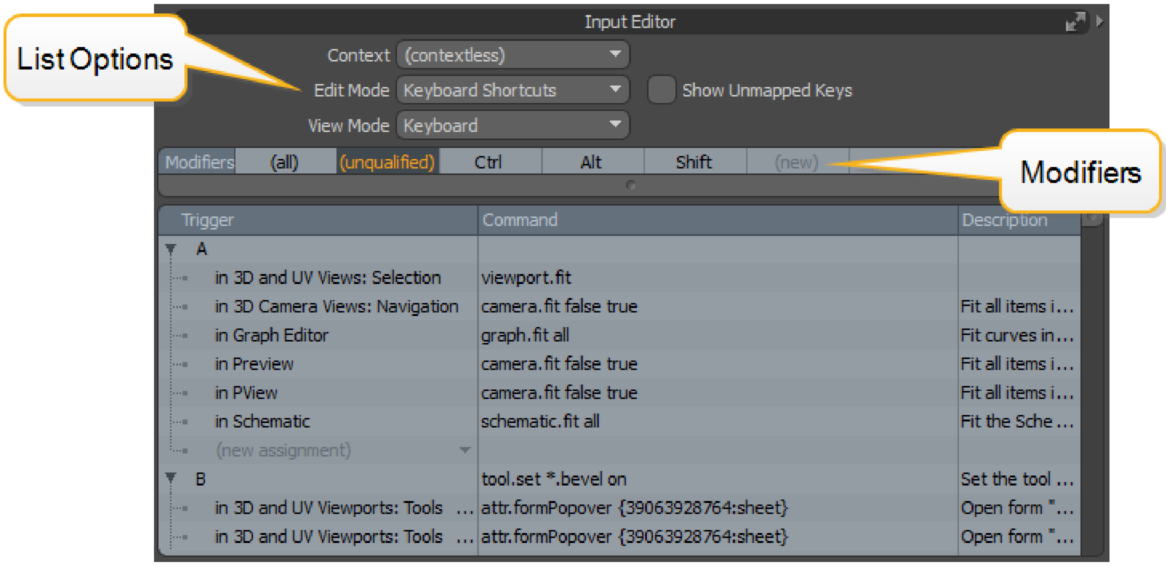
The **Input Editor** allows you to create custom keyboard shortcuts for Modo. Using keyboard shortcuts decreases the number of clicks and can greatly simplify and speed up your workflow.

To open the **Input Editor**, in the menu bar, click **System** > **Input Editor**.

By default, the **Input Editor** displays a list of all keyboard shortcuts in Modo, including every context and viewport. You can change what information is shown using the list options at the top.

You can filter the list of inputs by modifier key by clicking the required modifier in the **Modifiers** section above the list. You can select multiple modifiers by holding down **Ctrl**/**Cmd** while clicking, for instance to display keyboard shortcuts starting with **Ctrl**+**Alt**.

To display or hide keys with no command or action assigned, toggle the **Show Unmapped Keys** option next to the modes.



List Options

Modifiers

Context: (contextless)

Edit Mode: Keyboard Shortcuts

View Mode: Keyboard

☐ Show Unmapped Keys

Trigger	Command	Description
A		
in 3D and UV Views: Selection	viewport.fit	
in 3D Camera Views: Navigation	camera.fit false true	Fit all items i…
in Graph Editor	graph.fit all	Fit curves in…
in Preview	camera.fit false true	Fit all items i…
in PView	camera.fit false true	Fit all items i…
in Schematic	schematic.fit all	Fit the Sche…
(new assignment)		
B	tool.set *.bevel on	Set the tool …
in 3D and UV Viewports: Tools …	attr.formPopover {39063928764:sheet}	Open form "…
in 3D and UV Viewports: Tools …	attr.formPopover {39063928764:sheet}	Open form "…

List Options

You can adjust the information displayed in the list using the options at the top.

Context

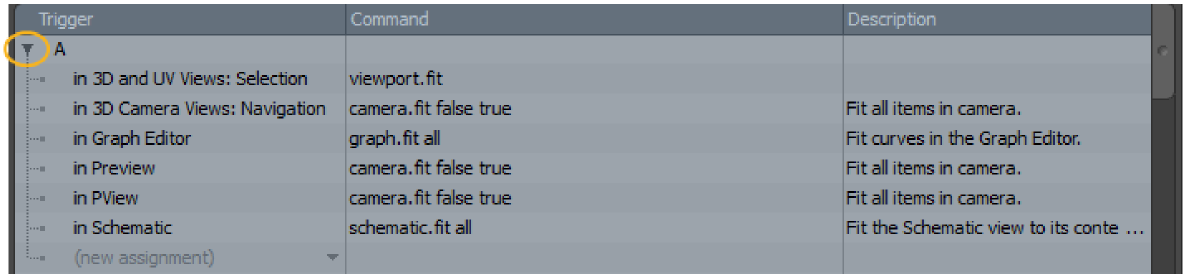
This option allows you to specify what Modo context the shortcut is accessible in, for example Item or Component selection mode, or a specific layout. By default, it's set to **(contextless)**, which means that the shortcut can be used in any mode or layout.

Note: If the input has another command set to it with a context applied, it takes priority over the contextless command.

Edit Mode

This option allows you to filter the list of inputs by keyboard shortcut, viewport, or tool.

- **Keyboard Shortcuts** - lists all the current shortcuts. Click the arrow in front of a keyboard shortcut to see a list of specific viewport parameters the shortcut is set in.



Trigger	Command	Description
▼ A		
in 3D and UV Views: Selection	viewport.fit	
in 3D Camera Views: Navigation	camera.fit false true	Fit all items in camera.
in Graph Editor	graph.fit all	Fit curves in the Graph Editor.
in Preview	camera.fit false true	Fit all items in camera.
in PView	camera.fit false true	Fit all items in camera.
in Schematic	schematic.fit all	Fit the Schematic view to its conte ...
(new assignment)		

- **Viewports** - lists all current shortcuts specific to a certain viewport, such as the Schematic or the Graph Editor. When **Edit Mode** is set to **Viewports**, an additional **Viewport** field appears where you can select a viewport from a dropdown.
- **Tools** - lists shortcuts organized according to modeling, painting, and sculpting tools, such as Airbrush or Edge Bevel. When **Edit Mode** is set to **Tools**, an additional **Tool** field appears where you can select a tool from a dropdown.

When **Edit Mode** is set to **Viewport** or **Tools**, an additional **Show Global Keys** option is displayed. Toggle it to display or hide keyboard shortcuts with no context.

View Mode

This option allows you to set the type of result displayed in the editor. You have the following options:

- **Mouse and Keyboard** - lists mouse and key inputs. This option is called **Keyboard** when **Edit Mode** is set to **Keyboard Shortcuts**.
- **Commands** - lists inputs by grouping them into relevant sections, such as **Animation** or **Transforms**. When **View Mode** is set to **Commands**, an additional **Commands by Group** field appears, where you can select a grouping, such as **Scripts** or **Viewport Presets**.
- **Actions** - lists the current actions set up, specific to the **Tool** or **Viewport** defined in **Edit Mode**. **Actions** is not available for the **Keyboard Shortcuts Edit Mode**.

Note: Currently, not all the **Commands by Group** options generate results. We recommend using the **Keyboard Shortcuts Edit Mode** and the **Keyboard View Mode**.

Adding a Custom Keyboard Shortcut

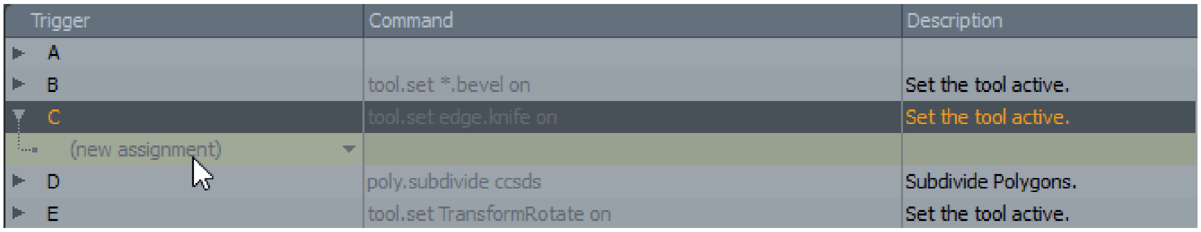
There are multiple ways of assigning keyboard shortcuts in the **Input Editor**.

Note: The easiest way to specify a new keyboard shortcut is to use the **Command History** viewport. To learn how to do this, see Creating Keyboard Shortcuts.

New Command to Shortcut

To add a new command to an existing keyboard shortcut:

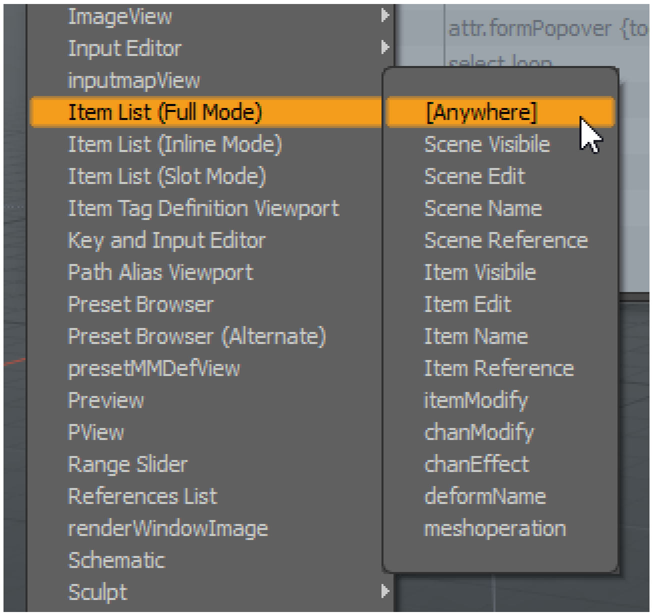
1. Expand the required input in the list and click **(new assignment)**. For example, to add a command to the **C** keyboard shortcut, you need to find **C** in the list.



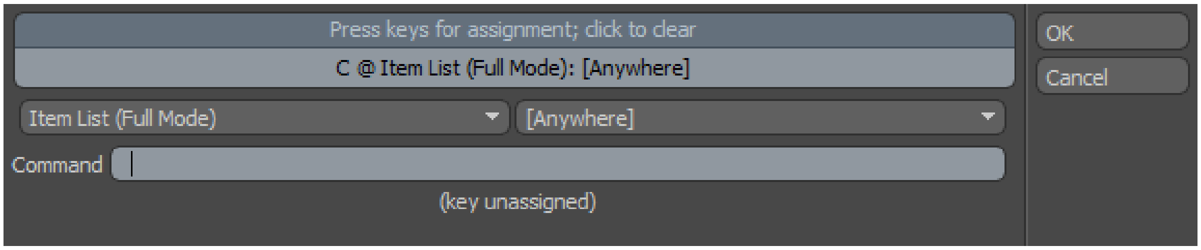
Trigger	Command	Description
▶ A		
▶ B	tool.set *.bevel on	Set the tool active.
▼ C	tool.set edge.knife on	Set the tool active.
(new assignment)		
▶ D	poly.subdivide ccsds	Subdivide Polygons.
▶ E	tool.set TransformRotate on	Set the tool active.

   This opens a dropdown listing the viewports.

2. Select a viewport, for instance **Item List (Full Mode)**, and an area within the viewport, or click **[Anywhere]** to apply the shortcut to the entire viewport.


ImageView
Input Editor
inputmapView
Item List (Full Mode)
Item List (Inline Mode)
Item List (Slot Mode)
Item Tag Definition Viewport
Key and Input Editor
Path Alias Viewport
Preset Browser
Preset Browser (Alternate)
presetMMDefView
Preview
PView
Range Slider
References List
renderWindowImage
Schematic
Sculpt
[Anywhere]
Scene Visibile
Scene Edit
Scene Name
Scene Reference
Item Visibile
Item Edit
Item Name
Item Reference
itemModify
chanModify
chanEffect
deformName
meshoperation
attr.formPopover {to
select.loop

This opens a dialog where you can specify the command.



Press keys for assignment; click to clear

C @ Item List (Full Mode): [Anywhere]

Item List (Full Mode) [Anywhere]

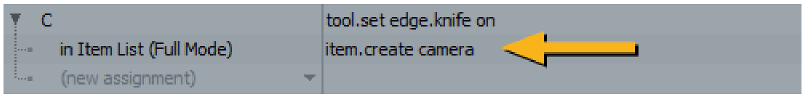
Command

(key unassigned)

3. Check that you have the correct shortcut and viewport selected, enter a command in the **Command** field, then click **OK**. For example, to create a shortcut in the **Item List** to create a camera by pressing **C**, enter 'item.create camera'.

Note: You can find out what the exact command is using the Command History viewport. Perform the action in Modo, then check for the command in the **Command History** viewport.

The keyboard shortcut is saved. You can test it in Modo, and check that it appears in the **Input Editor** list.



C tool.set edge.knife on

in Item List (Full Mode) item.create camera

(new assignment)

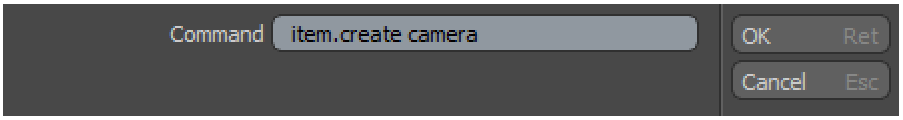
Note: If the command you've added is invalid, it is displayed in red. Right-click and click **Edit Command** to change it.

Editing an Existing Command

To edit an existing command assigned to a specific keyboard shortcut:

1. Right-click the command in the list, and click **Edit Command**.

   This opens the **Change Key Command** dialog.



Command item.create camera

2. Edit the **Command** field, then click **OK**.

   The keyboard shortcut is updated with the new command.

Removing a Command From a Key

To remove an assignment from a keyboard shortcut, right-click it and click **Clear Mapping**, then click **OK** in the dialog.

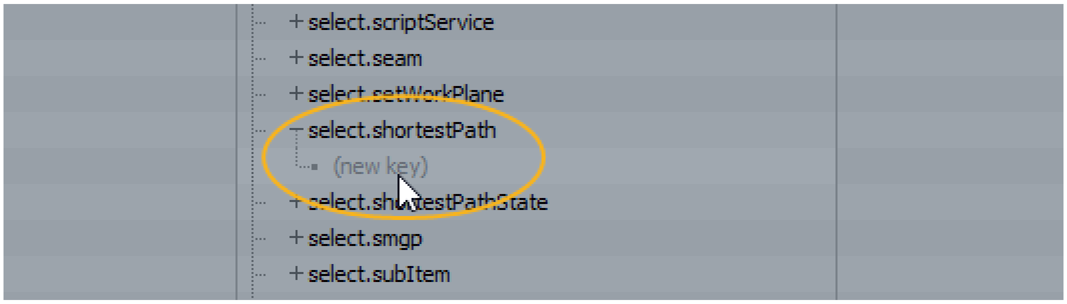
New Shortcut to Command

To add a new keyboard shortcut to an existing command:

1. To see a list of commands, set **Edit Mode** to **Keyboard Shortcuts**, and **View Mode** to **Commands** and **Commands by Group** to the relevant group.

   In this example, we want to add a keyboard shortcut for Shortest Path Selection, so we set **Commands by Group** to **Categories**.

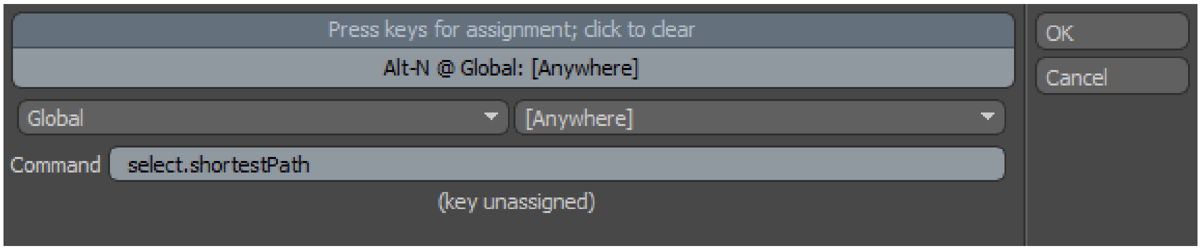
2. Expand the **Selection** category, and click the **+** button in front of the command you need to expand it, in this case, **select.shortestPath**, then right-click **(new key)** and click **Add Key**.


+ select.scriptService
+ select.seam
+ select.setWorkPlane
- select.shortestPath
(new key)
+ select.shortestPathState
+ select.smgp
+ select.subItem

   This displays a dialog to specify the key.

3. To specify the keyboard shortcut, press the key(s) on the keyboard. This example uses **Alt**+**N** for the Shortest Path Selection command.

   By default, the context is set to **Global [Anywhere]**, but you can choose a specific viewport or UI component for the shortcut to work in using the dropdowns.


Press keys for assignment; click to clear
OK
Cancel

Alt-N @ Global: [Anywhere]

Global [Anywhere]

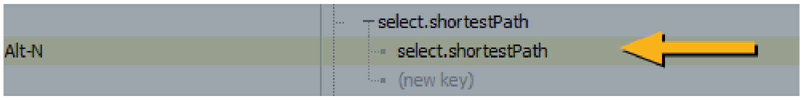
Command select.shortestPath

(key unassigned)

Note: If the intended combination is already in use by another command, Modo displays the associated command at the bottom of the dialog.

4. Click **OK**.

   The shortcut is saved. You can test it in Modo, and check that it appears in the **Input Editor** list.



```
                          select.shortestPath
Alt-N                       select.shortestPath
                            (new key)
```

Removing a Key from a Command

To remove a key from a keyboard shortcut, right-click it and click **Clear Key**.

Article: For more information on the **Input Editor**, see Foundry Knowledge Base Articles Q100123: Setting custom keyboard shortcuts in MODO and Q100307: Understanding the views in the Input Editor.

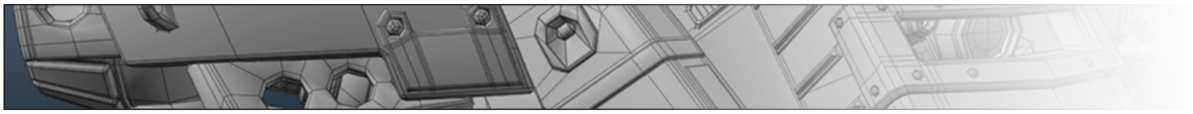
Modeling

The sheer number of modeling tools and options available may seem overwhelming at first, but to help understand them, they are grouped by function. Once you can grasp the overall utility of each group, it's easier to appreciate the subtleties of the individual tools. The first few topics provide an overview of working with Modo's toolset, while the following topics detail the operation and application of the tools and modeling functions individually.

Modeling Overview and Techniques

For all intents and purposes, modeling-wise, Modo is a tool for working with polygons. There are other ways provided for generating forms, somewhat indirectly, such as spline and bezier curve patching, or working with different types of subdivision surfaces, or even fur, but the ultimate goal of those operations is to eventually create polygons for rendering and baking, or to export to their ultimate destination.

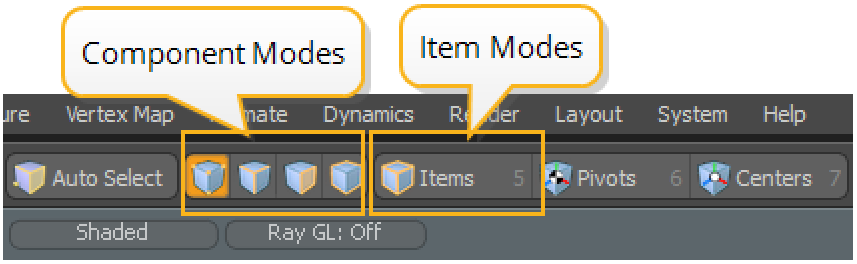
Modo provides a variety of ways to manipulate and shape those polygons into the desired shape. Some tools manipulate polygons directly, such as the transform and deform tools, others provide means to manipulate polygons indirectly, such as the sculpting of multi-resolution meshes. There are as many methodologies for reaching a finished result as there are users; everyone has their own unique approach. However, of the various methodologies, there are some basics to understand for Modo's way of working that eases the overall learning curve, allowing you to reach your goal and find your own individual style.



Modeling in Modo

In any 3D modeling toolset, each tool needs to infer, or be informed about, several key variables in order to work properly. It needs to know its central position, meaning the location in 3D space, from where the desired effect is applied. It also needs to know its operational axis or orientation. And, finally, it needs to know which portion of the scene is going to be affected and by how much. In Modo, each of these variables has been extracted to its own operational layer. Modo offers each as a building block individually, such as Specifying Action Centers and Falloffs and Using Falloffs. All the tools in Modo are made up of these building blocks and understanding how each part contributes to produce the final result is important for moving forward. You can also assemble these blocks to make your own custom tools using the Tool Pipe, providing a flexible and really powerful way to work.

Components Versus Items



One of the areas that is often confusing to new users of Modo are the various selection modes. When modeling, you can work in one of three component modes: **Vertices**, **Edges**, and **Polygons**. This is where the actual geometry is created and shaped. There is also an **Items** mode, which is useful for positioning modeled objects in a scene, and is a necessity of animation in Modo proper. When in any individual selection mode, only selections of that type can be made. The important distinction is that when editing an item, you are editing the layer itself (think of it like a container), while editing a component is editing the contents of the layer (what's inside the container). Modifications made in the **Vertex**, **Edge**, and **Polygon** modes happen around the center point, while transforms applied while in **Items** mode moves the entire item, including its center point.

To learn more about selection mode types, see Selecting Items.

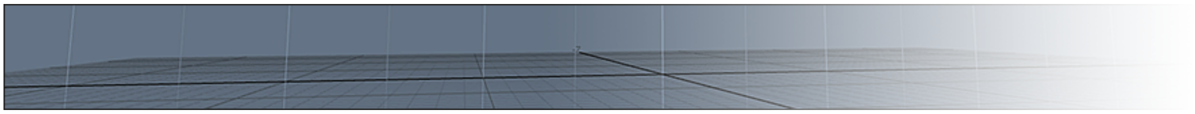
Interacting with Modo

Interacting with Modo itself is done mostly through the manipulation of the mouse, taking advantage of all three buttons, and the scroll wheel (some may prefer a trackball or tablet -both typically have functions that can be remapped to mimic the options available on a mouse). Mouse button activity is supplemented by keyboard commands, which are, at times, used in unison with the mouse. This can take some getting used to (see Mouse Standards for more information). Given that Modo is a 3D application that you interact with in a 2D fashion, this may feel awkward at first, but it produces the most predictable results.

It's recommended you read through Understanding the Interface to familiarize yourself with the many viewports, tabs, and panels. Main functions are organized as tabs across the top, such as **Modeling** or **Sculpting**. When in any of these areas, only the tools necessary for the task at hand are provided, streamlining the process. This way of working also makes it easy to jump in and out of tasks easily, providing a comfortable non-linear workflow. You can model, UV map, sculpt, and then model some more with very little worry that one process has a negative affect on the others. In fact, a non-linear workflow is encouraged in Modo.

Using Tools

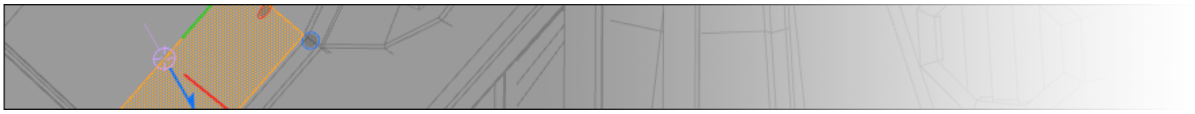
To use any tool to perform an action, you must first click the tool button or the menu command. For one-shot commands, such as **Delete**, **Fit**, or **Hide**, that is all that is necessary to produce a result. For interactive tools, such as **Move**, **Rotate**, and **Bevel**, this makes the tool itself active, but it isn't yet acting on anything in the scene. At this point you can modify the default values of the tool in the **Tool Properties** panel, and then press **Apply** to act upon the target, or click in the 3D viewport to set the tool in interactive mode. This action draws the appropriate tool handle within the viewport. Now, the property values can be interactively adjusted by clicking and dragging on any of the available tool handles (handles change to a yellow color to show they can be adjusted), constraining the action to specific axes, or by **Hauling**, which is the act of clicking and dragging the mouse pointer left/right or up/down, away from any tool handle. This allows you to modify multiple axes at once. When the desired result for a tool is achieved, you can simply drop the tool by pressing **Q** to finalize the application of it. Once a tool is dropped, interactive editing of tool attributes is no longer possible and the action of the tool is fixed into the target geometry.



Work Plane

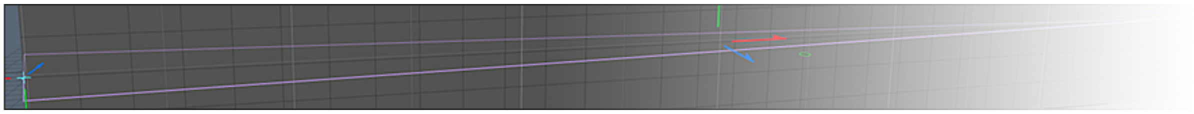
You should understand that actions don't arbitrarily happen in 3D space within a viewport. For example, when creating points, the individual vertices are created at the intersection of the **Work Plane** and where you click in the viewport, providing predictable and repeatable results. Understanding the **Work Plane** and its functions is important to understanding how Modo works. The **Work Plane** is the light gray grid that is visible in any viewport.

In a perspective view, you may notice that the grid changes position. This is the **Work Plane** orienting itself towards you, trying to stay as close as possible to parallel with the viewport window, while remaining aligned to the nearest axis. You can also modify the **Work Plane** to a fixed arbitrary position, based on a component selection. This temporarily orients the entire Modo universe to align with that selection, so all functions now work in relation to the **Work Plane**, making it easier to perform tasks that would be very difficult otherwise. For more information, refer to Using the Work Plane.



Action Centers

When performing any action on a selection, the position of the center of origin and the axis orientation of the tool handle itself has a dramatic effect over the result when applied. You can orient the **Work Plane** to suit many situations, but constantly modifying the **Work Plane** would get tedious, so Modo has what are called **Action Centers** - automatic combinations of center positions and axis directions for tools. There are a variety of options, based on different criteria; some are oriented to selections, some to a viewport's orientation, while others are based on fixed positions, such as the item's center (0,0,0). **Action Centers** are often packaged with tools to provide specific functions. Take for instance the **Flex** tool - this combines a rotation tool with an action center that orients itself automatically to the selection border, making it easy to pose characters. Understanding **Action Centers** are necessary to a fluid experience with Modo. See Specifying Action Centers and Falloffs for more information on working with them.



Falloffs

Ordinarily, when performing an action on a selection, the application is uniform across the selection area. Falloffs provide a means to vary the strength of the action across a given area. Once activated, all of Modo's tools respect the falloffs and apply themselves appropriately, opening up a variety of options that would be incredibly difficulty to reproduce otherwise. When enabled, a falloff snaps to a selection, if a tool is currently active, scaling to fit within the selection's bounding box. This makes them incredibly easy to apply and use. Falloffs can also be packaged along with tools, and often are. They are the basis of the many deformation tools - **Rotate** combined with a linear falloff produces the **Twist** tool. **Move** combined with the same falloff produces the **Shear** effect. Falloffs may appear simple, but they wield an incredible amount of power when working in Modo. For more information on using and applying falloffs, please refer to Using Falloffs.

Modeling Techniques

Modo offers a rich feature set for building scenes and animating meshes. Modo's workflow, as with most other 3D modeling applications, is largely built around direct modeling as it has proved to be a very effective and intuitive approach with universal application.

As Modo has grown, different approaches to modeling have been developed that offer an alternative to regular, Direct modeling. Procedural and Schematic modeling approaches have been designed to offer a non-destructive and reconfigurable system of modeling. The MeshFusion modeling approach has been designed to offer an advanced sophisticated set of boolean operations for creating new geometry.

The following modeling techniques are used in Modo.

Direct Modeling

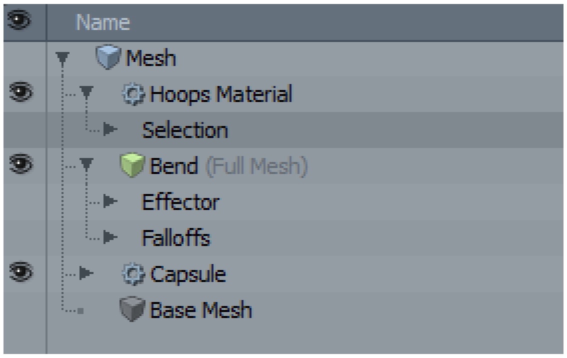
The most common approach to this is to use the many tools to manipulate the geometry directly in the 3D viewport. For example, you can add a cube, bevel it, move edges, duplicate polygons, deform, and so on, all by interacting with the geometry. This technique is called direct modeling, as the original mesh is directly manipulated using the mouse and tool widgets.

Note: Direct modeling is also known as destructive modeling as the mesh is changed by each operation. Once changed, you can't revisit previous operations and edit their effect, while preserving later operations, hence the term 'destructive'.

Procedural Modeling

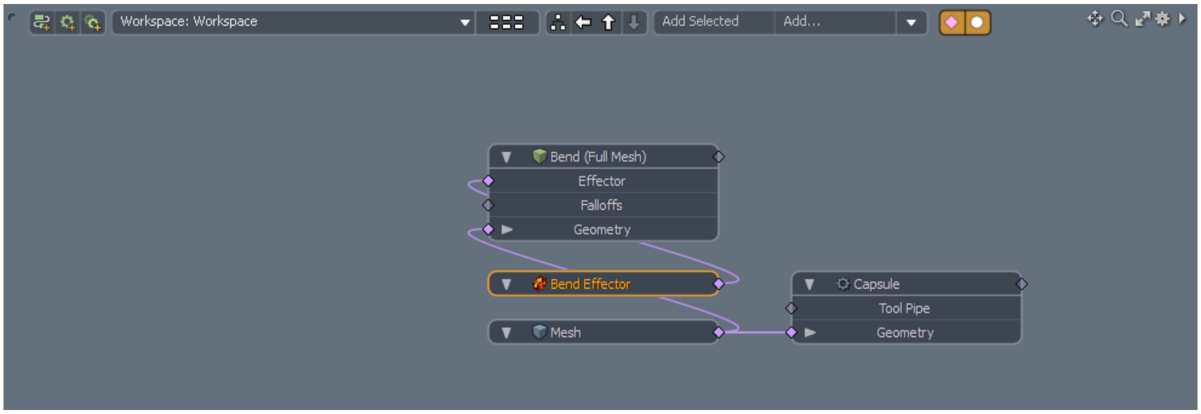
Procedural modeling lets you stack geometry and operations in the Mesh Operations List. Items are layered so that the items above affect the items below. This provides a simple structure for non-destructive modeling, as you can revisit a layer in the stack and change its settings. The image below shows a sample **Mesh Operations** list.

For more information, see Procedural Modeling with MeshOps.


Name
Mesh
Hoops Material
Selection
Bend (Full Mesh)
Effector
Falloffs
Capsule
Base Mesh

Schematic Modeling

Modo has a **Schematic** viewport, which is a 2D environment where functional nodes are visually linked, providing a way to define, and redefine relationships between items and operations. You can build visual networks of items which can be combined to produce the desired result.


Workspace: Workspace
Add Selected
Add...
Bend (Full Mesh)
Effector
Falloffs
Geometry
Bend Effector
Mesh
Capsule
Tool Pipe
Geometry

For more information, see Schematic Viewport.

MeshFusion Modeling

MeshFusion offers a sophisticated set of boolean operations for creating new geometry. With MeshFusion, you can create meshes through the non-destructive union, subtraction, and intersection of objects. Although similar in concept to existing Boolean operations, MeshFusion is significantly more advanced, supporting real-time update of operations and a high level of control over blending between the resulting surfaces.

Note: For more information, see MeshFusion.

Modeling Examples

To give you a taste of how each of these approaches work, we are using a model of a simple striped cane.

Modeling the cane requires three main steps:

1. Add a tall thin Capsule primitive.
2. Bend it using the Bend Effector.
3. Add the colored hoops by assigning a red material to rings of polygons.

Capsule Settings

Learn how to perform these steps using each modeling technique.

In each approach, a capsule primitive is added. Use the following settings to define the size and structure of the capsule:

- **Position X**, **Y**, and **Z** are **0**
- **Radius X**: **100 mm**
- **Radius Y**: **1 m**
- **Radius Z**: **100 mm**
- **Sides**: **24**
- **Segments**: **9**
- **End Segments**: **5**
- **End Size**: **110 mm**
- **Axis**: **Y**

Direct Modeling Example

Learn how to make the cane using the direct modeling workflow, starting with an empty workspace in **Model** layout.

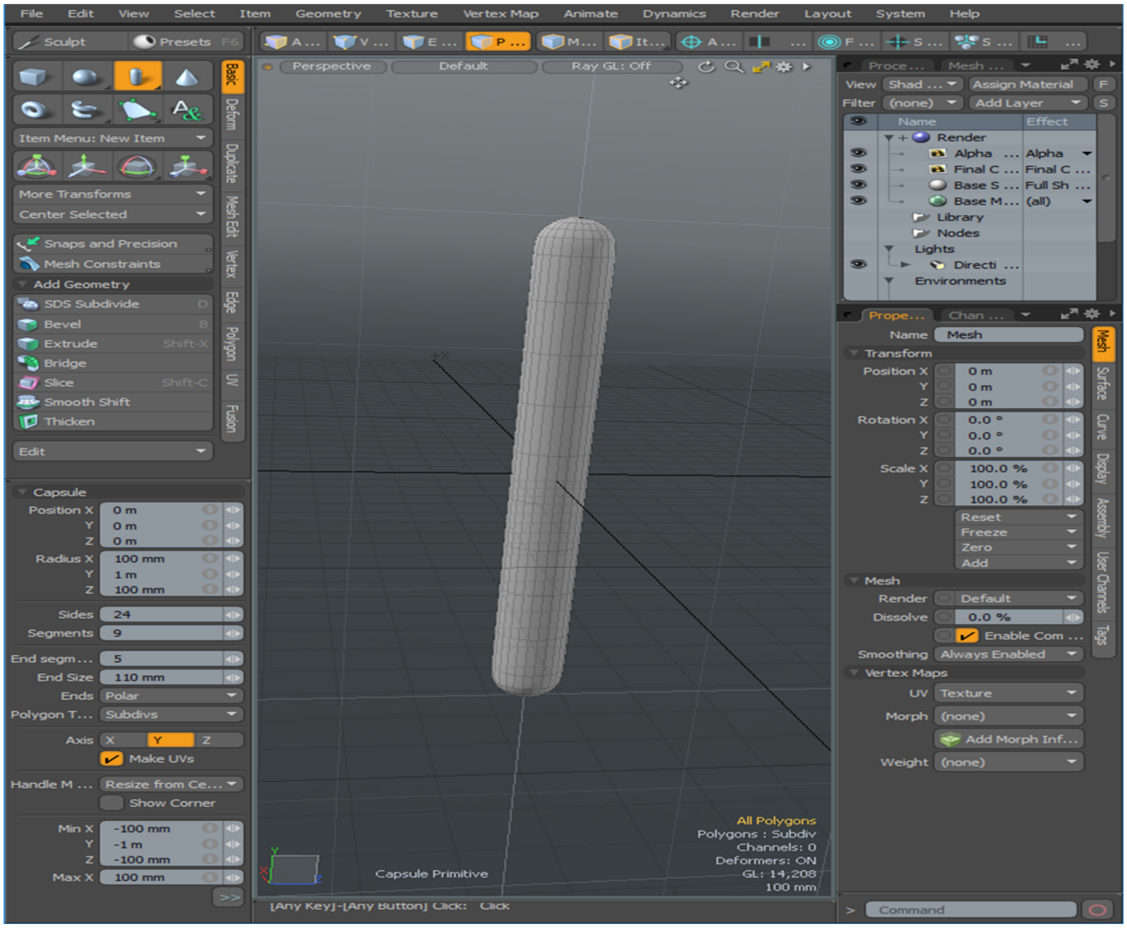
Adding a Capsule

Add a capsule primitive to the scene and apply the Capsule Settings

1. Click **Layout** > **Layouts** > **Model**.
2. On the left panel, click the **Basic** tab to view the modeling tools.
3. Click and hold on the **Cylinder** primitive button and select **Capsule**.

Tip: Alternatively, right-click on the **Cylinder** primitive button and select **Capsule**.

4. In the **Tool Properties** panel, adjust the capsule settings as listed in the Capsule Settings and click **Apply**.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Sculpt
Presets
Basic
Deform
Duplicate
Mesh Edit
Vertex
Edge
Polygon
UV
Fusion
Item Menu: New Item
More Transforms
Center Selected
Snaps and Precision
Mesh Constraints
Add Geometry
SDS Subdivide
Bevel
Extrude
Bridge
Slice
Smooth Shift
Thicken
Edit
Perspective
Default
Ray GL: Off
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 14,208
100 mm
Capsule Primitive
[Any Key]-[Any Button] Click: Click
Render
Alpha
Final C …
Base S …
Base M …
Library
Nodes
Lights
Directi …
Environments
Mesh
Transform
Reset
Freeze
Zero
Add
Command

Capsule
Position X 0 m
Y 0 m
Z 0 m
Radius X 100 mm
Y 1 m
Z 100 mm
Sides 24
Segments 9
End segm… 5
End Size 110 mm
Ends Polar
Polygon T… Subdivs
Axis X **Y** Z
☑ Make UVs
Handle M… Resize from Ce…
☐ Show Corner
Min X -100 mm
Y -1 m
Z -100 mm
Max X 100 mm

Bending the Capsule

Bend the capsule using the Bend Effector tool.

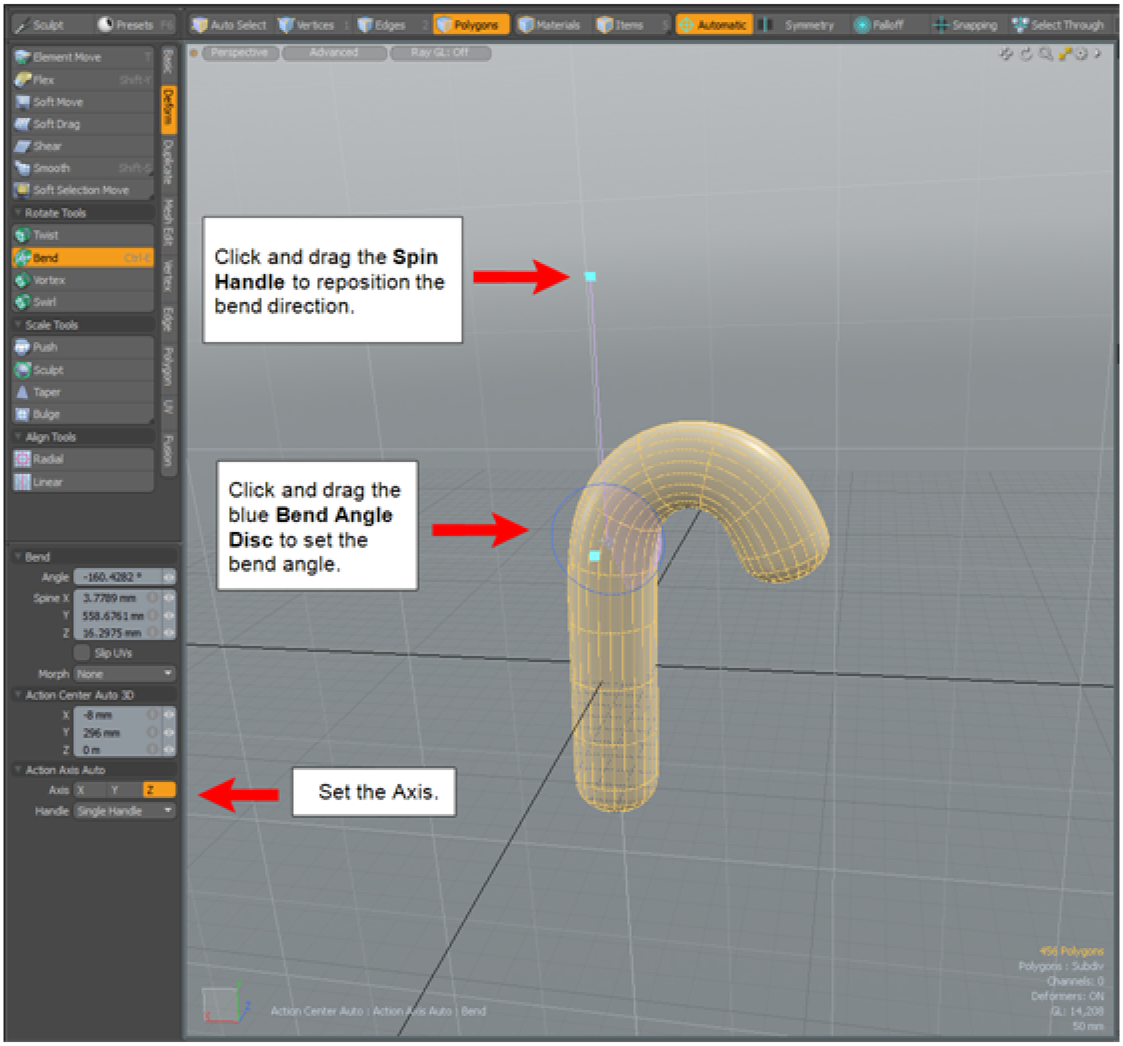
1. Click the **Polygons** selection mode tab and double-click on the capsule to select the entire item.

> **Note:** You must use either the **Vertices** selection mode or the **Polygons** selection mode when using the Bend tool to bend a specified area on the capsule.

> **Tip:** Use **Ctrl**+**Z** to undo, if you are unhappy with the results.

2. In the Toolbox, located on the left panel, open the **Deform** tab, and click **Bend**.
3. To initiate the interactive editing mode, in the 3D viewport, click at the base location of the capsule where the bend should start from.

   The Bend tool is activated.
4. In the **Bend** properties, listed in the left panel, set the **Axis**. In this example, it is set to the **Z** axis.
5. To set the bend angle, in the viewport, click and drag the blue **Bend Angle Disc**.
6. To reposition the bend direction, move the **Spin Handle** to the top of the cane item.



Click and drag the **Spin Handle** to reposition the bend direction.

Click and drag the blue **Bend Angle Disc** to set the bend angle.

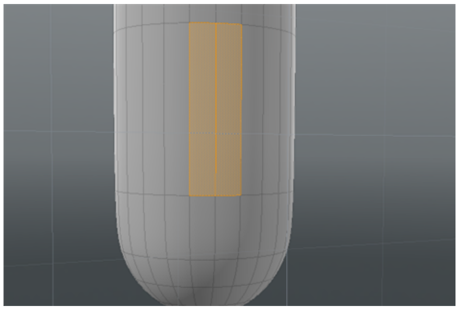
Set the Axis.

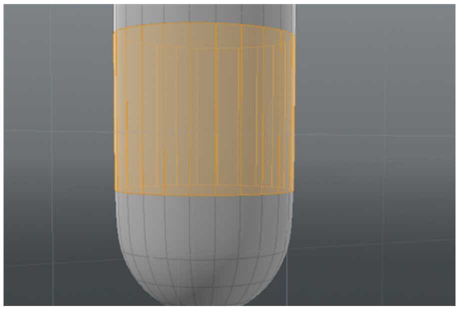
Note: For more detailed information about the Bend tool, see Bend.

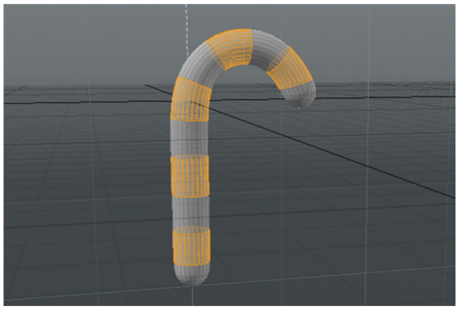
Adding Colored Hoops

Add colored hoops to the cane by assigning a red material to rings of polygons.

1. In the **Model** layout, select the **Polygons** selection mode.
2. To select a single hoop, select a polygon, and then press **L** to loop the selection. Repeat this step for every loop you want to color red.







3. Press **M** to assign a material to the hoops. In the **Polygon Set Material** dialog, name the material **Hoops**, click the **Color** value and select a red color. Click **OK** to close each dialog.
4. Press the space bar to drop your selection.

   A colored candy cane, using direct modeling, is created. You can no longer change the properties of this model.
5. Press **F8** to preview the render output.

Procedural Modeling Example

To model the cane procedurally, use procedural operations in the **Mesh Operations** list. All of the items added to the **Mesh Operations** list can later be modified.

Tip: Many of these steps involve adding procedural operations to the Mesh Operations list by navigating the directories in the Preset Browser. You can also locate an operation in the Preset Browser using the find functionality. To enable this, click the **F** button in the top right of the browser.

Adding a Capsule

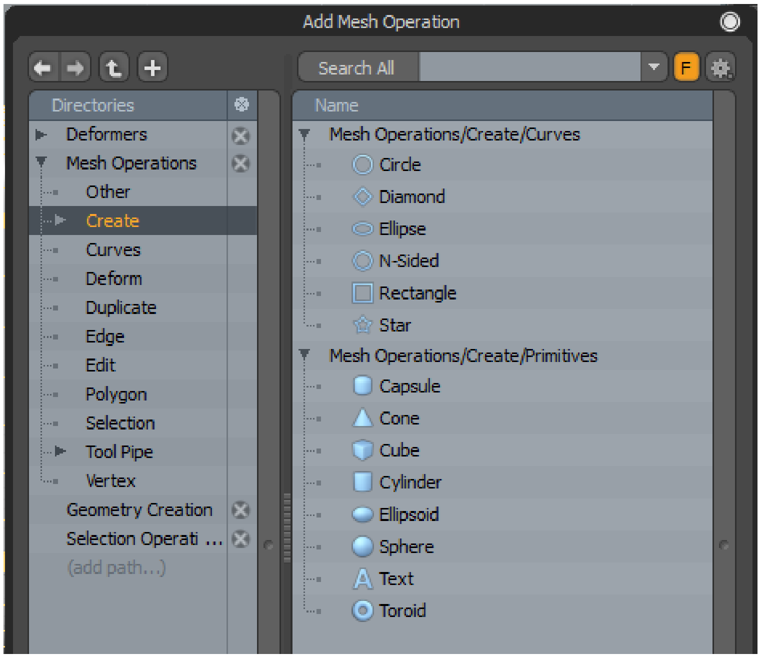
Add a procedural capsule primitive to the **Mesh Operations** list and apply the Capsule Settings

1. Click **Layout** > **Layouts** > **Model**.
2. On the top right panel, open the **Mesh Operations** list.

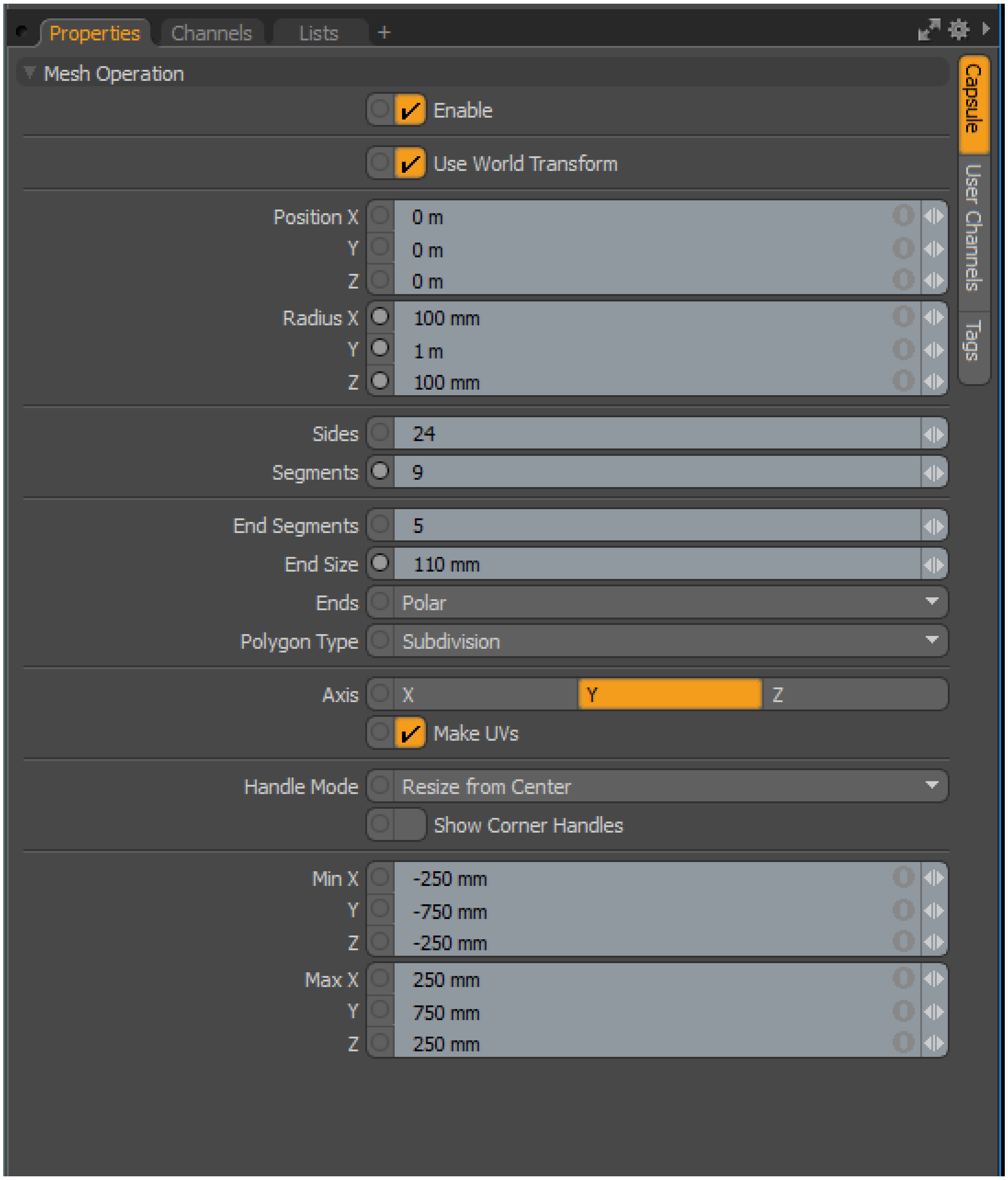
3. From the **Mesh Operations** list, click **Add Operator**.

   The Preset Browser is displayed.

4. Click **Mesh Operations** > **Create** > **Mesh Operations/Create/Primitives** and double-click on **Capsule**.


Add Mesh Operation
Search All
F
Directories
Deformers
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Edit
Polygon
Selection
Tool Pipe
Vertex
Geometry Creation
Selection Operati ...
(add path...)
Name
Mesh Operations/Create/Curves
Circle
Diamond
Ellipse
N-Sided
Rectangle
Star
Mesh Operations/Create/Primitives
Capsule
Cone
Cube
Cylinder
Ellipsoid
Sphere
Text
Toroid

5. In the **Item Properties** panel, adjust the settings as list in the Capsule Settings.



Mesh Operation

☑ Enable

☑ Use World Transform

Position X 0 m
Y 0 m
Z 0 m
Radius X 100 mm
Y 1 m
Z 100 mm

Sides 24
Segments 9

End Segments 5
End Size 110 mm
Ends Polar
Polygon Type Subdivision

Axis X Y Z (Y selected)
☑ Make UVs

Handle Mode Resize from Center
☐ Show Corner Handles

Min X -250 mm
Y -750 mm
Z -250 mm
Max X 250 mm
Y 750 mm
Z 250 mm

Bending the Capsule

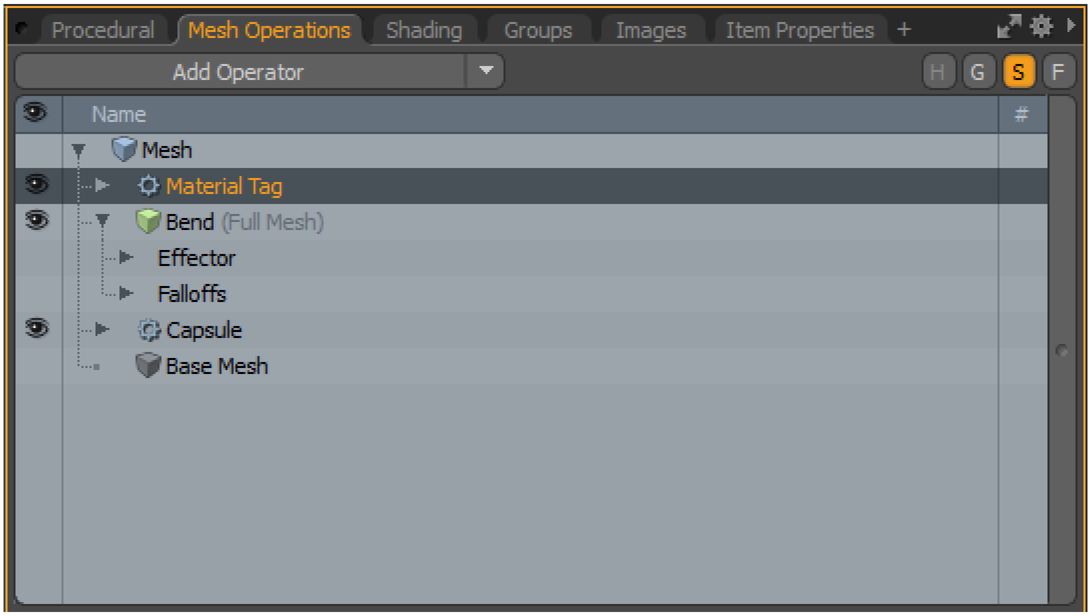
Bend the capsule using the Bend Effector tool.

1. In the **Mesh Operations** list, click **Add Item**, select **Deformers** > **Effectors**, and double-click **Bend Effector**.
2. In the **Item Properties**, select the **Bend Effector** tab, and set **Angle** to **190°**.

Adding Colored Hoops

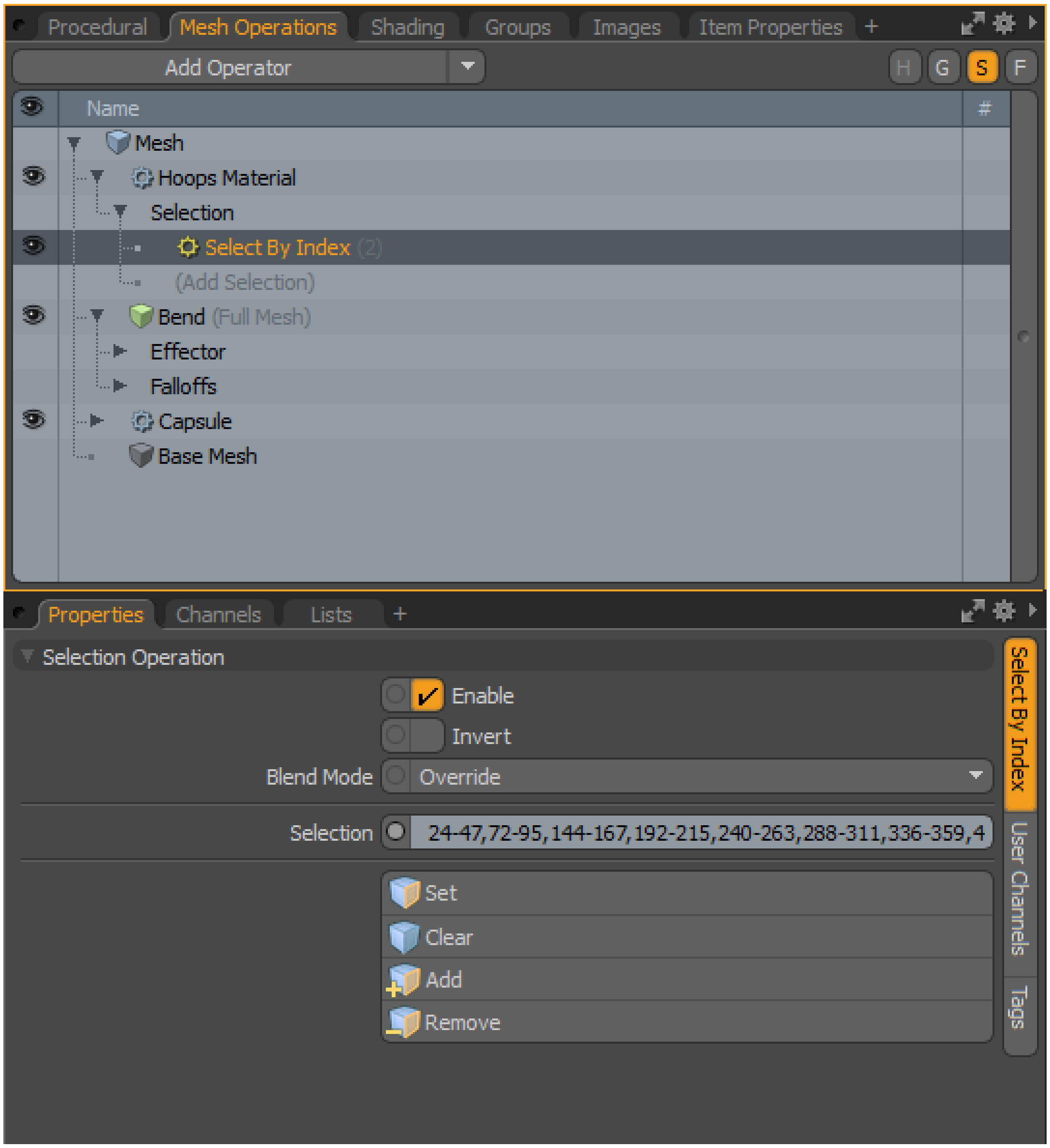
Add colored hoops to the cane by assigning a material tag to the **Mesh Operations** list.

1. In the **Mesh Operations** list, click **Add Operator**, select **Mesh Operations** > **Polygon**, and double-click **Material Tag**.


Procedural
Mesh Operations
Shading
Groups
Images
Item Properties
+
Add Operator
H
G
S
F
Name
#
Mesh
Material Tag
Bend (Full Mesh)
Effector
Falloffs
Capsule
Base Mesh

2. In the **Item Properties**, select the **Material Tag** tab, and change **Material Name** to **Hoops Material**.
3. Activate the **Polygons** selection mode, select a polygon, and press **L** to complete the loop. Repeat this step for every loop you want to color red.
4. To add the selection to the material, in the **Mesh Operations** list, expand **Hoops Material**, and expand **Selection**. Click **(Add Selection)** and double-click **Select By Index**.

   The selection is automatically populated with the polygons.


Procedural
Mesh Operations
Shading
Groups
Images
Item Properties
+
Add Operator
H
G
S
F
Name
#
Mesh
Hoops Material
Selection
Select By Index (2)
(Add Selection)
Bend (Full Mesh)
Effector
Falloffs
Capsule
Base Mesh
Properties
Channels
Lists
+
Selection Operation
Set
Clear
Add
Remove
Select By Index
User Channels
Tags

☑ Enable

☐ Invert

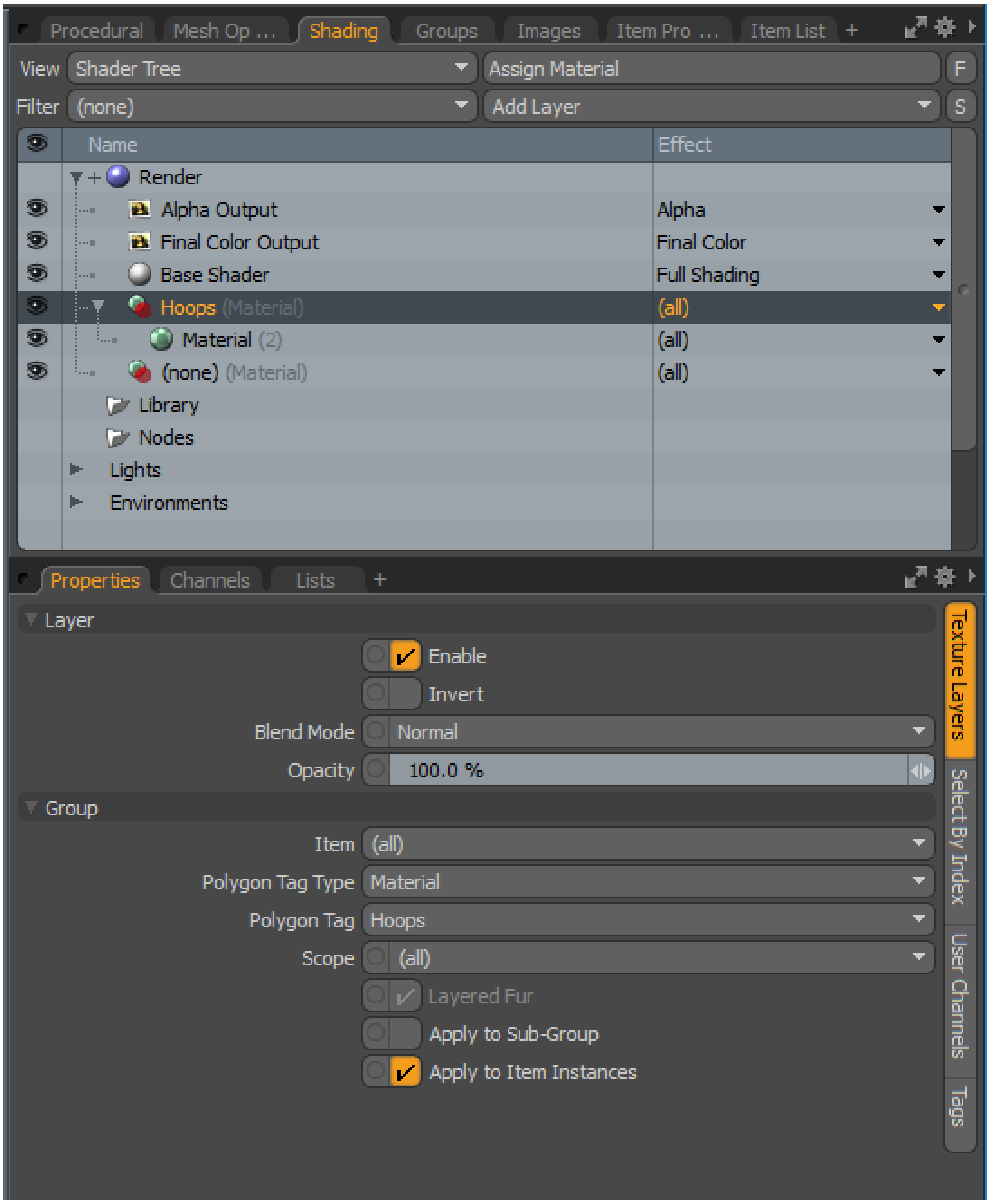
Blend Mode: Override

Selection: 24-47,72-95,144-167,192-215,240-263,288-311,336-359,4

5. To add a material group to the **Shader Tree**, click the **Shading** tab, and select **Add Layer** > **Group**.
6. In the **Shader Tree**, with the material group highlighted, select **Add Layer** > **Material**.

Tip: If the Shader Tree tab is not displayed, click the plus sign to open a new tab and then select **Data Lists** > **Shader Tree**.

7. Select the **Material Ref** properties and change the **Diffuse Color** to red.
8. To match the group to the Material Tag in the Shading properties, select the **Texture Layers** tab, ensure the **Polygon Tag Type** is set to **Material**, and set **Polygon Tag** to **Hoops**.


Procedural
Mesh Op ...
Shading
Groups
Images
Item Pro ...
Item List
+
Properties
Channels
Lists
+
Texture Layers
Select By Index
User Channels
Tags

View: Shader Tree
Filter: (none)
Assign Material
Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Hoops (Material)	(all)
Material (2)	(all)
(none) (Material)	(all)
Library	
Nodes	
Lights	
Environments	

Layer
☑ Enable
☐ Invert
Blend Mode: Normal
Opacity: 100.0 %

Group
Item: (all)
Polygon Tag Type: Material
Polygon Tag: Hoops
Scope: (all)
☑ Layered Fur
☐ Apply to Sub-Group
☑ Apply to Item Instances

A colored candy cane, using procedural modeling, is created. You can change the properties of this model at any point.

9. Press **F8** to preview the render output.

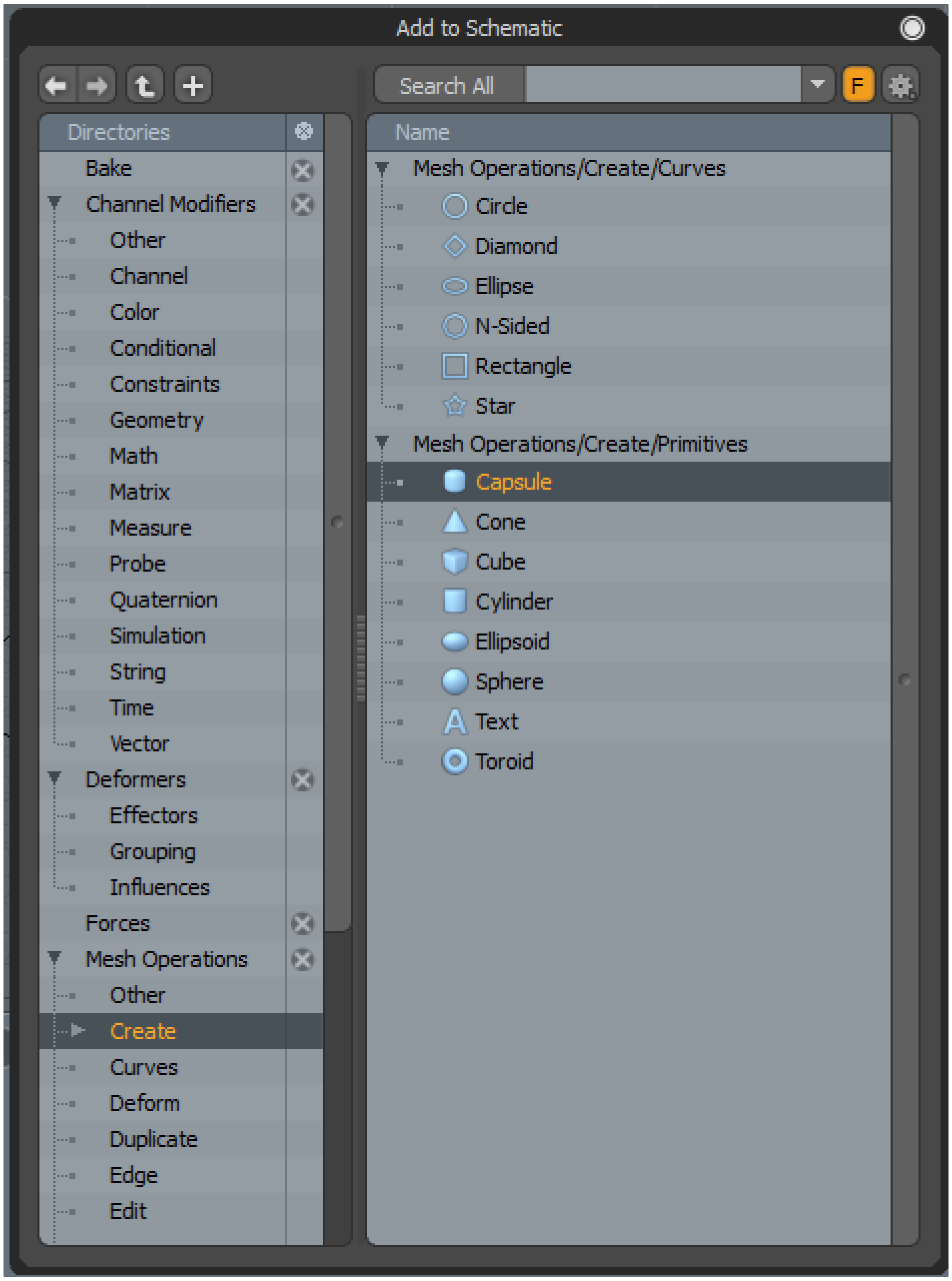
Schematic Modeling Example

To model the cane schematically, use the schematic nodes in the **Schematic** viewport. Modeling schematically is normally performed in the **Setup** layout.

Adding a Capsule

Add a capsule node to the scene and apply the Capsule Settings

1. Click **Layout** > **Layouts** > **Setup**.
2. In the **Schematic** viewport, click **Add...** to open the **Preset Browser**.
3. Select **Mesh Operations** > **Create** > **Mesh Operations/Create/Primitives** and double-click **Capsule**.


Add to Schematic
Search All
F
Directories
Bake
Channel Modifiers
Other
Channel
Color
Conditional
Constraints
Geometry
Math
Matrix
Measure
Probe
Quaternion
Simulation
String
Time
Vector
Deformers
Effectors
Grouping
Influences
Forces
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Edit
Name
Mesh Operations/Create/Curves
Circle
Diamond
Ellipse
N-Sided
Rectangle
Star
Mesh Operations/Create/Primitives
Capsule
Cone
Cube
Cylinder
Ellipsoid
Sphere
Text
Toroid

4. In the **Item Properties** panel, adjust the settings for the capsule as listed in the Capsule Settings.

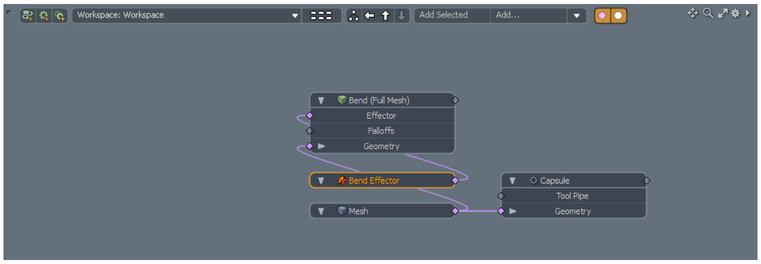
Bending the Capsule

Bend the capsule using the Bend Effector tool.

1. In the **Schematic** viewport, click **Add...**, select **Deformers** > **Effector**, and double-click **Bend Effector**.

   A **Bend Effector** node is added and automatically wired to the capsule.

2. To show the hidden connection, double-click the diamond connector on the **Geometry** input of the **Bend** node.


Workspace: Workspace
Add Selected
Add...
Bend (Full Mesh)
Effector
Falloffs
Geometry
Bend Effector
Mesh
Capsule
Tool Pipe
Geometry

3. Click the **Bend Effector** node.

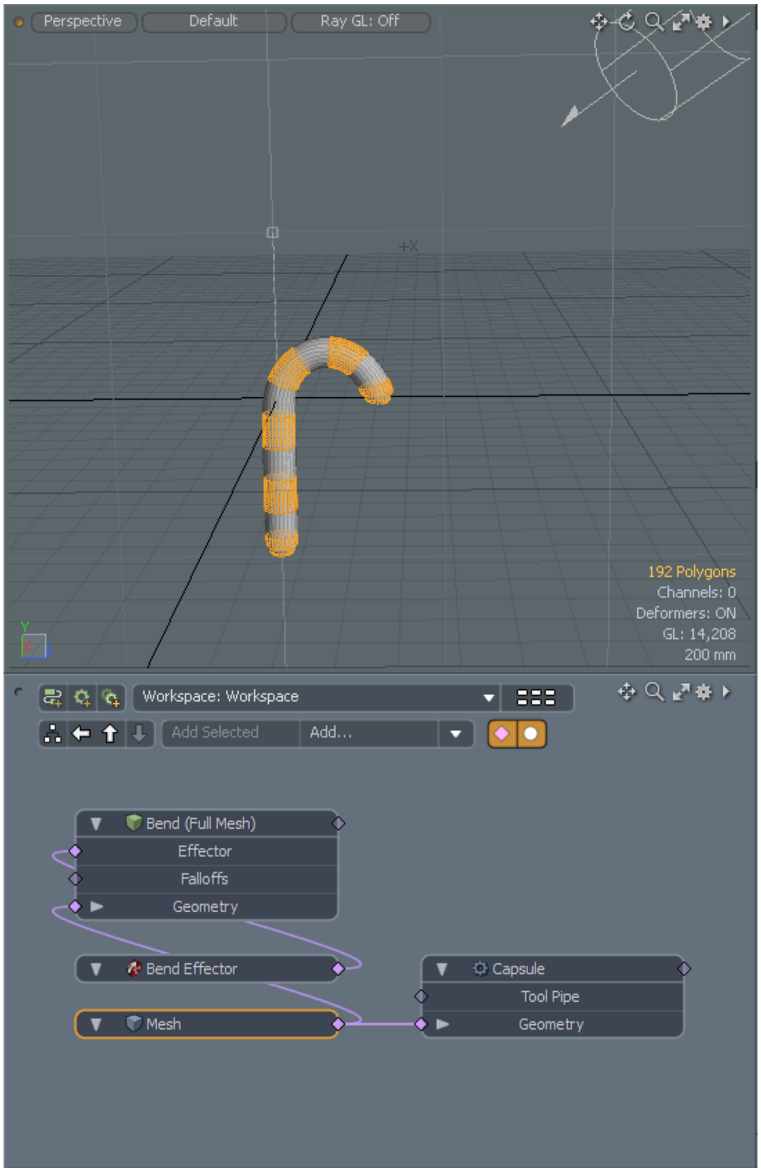
   The node is highlighted in yellow.

4. In the **Item Properties** panel, select the **Bend Effector** tab, and set **Angle** to **190°**.

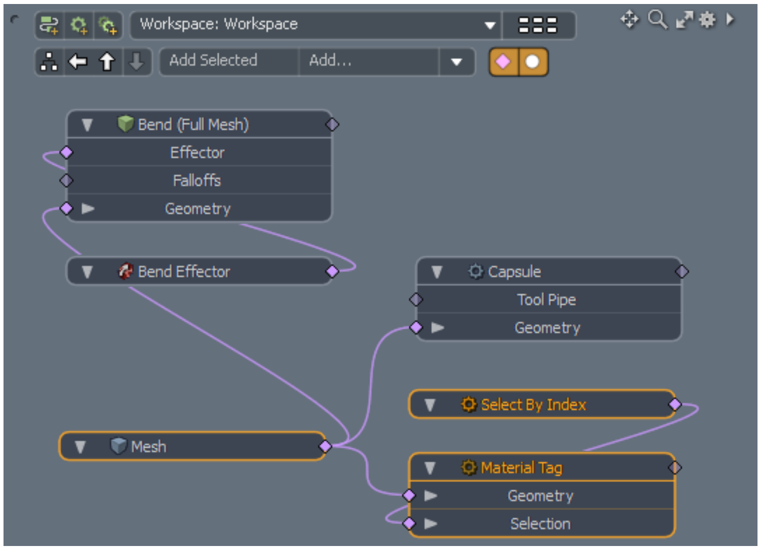
Adding Colored Hoops

Add colored hoops to the cane by assigning a material tag to the **Mesh Operations** list.

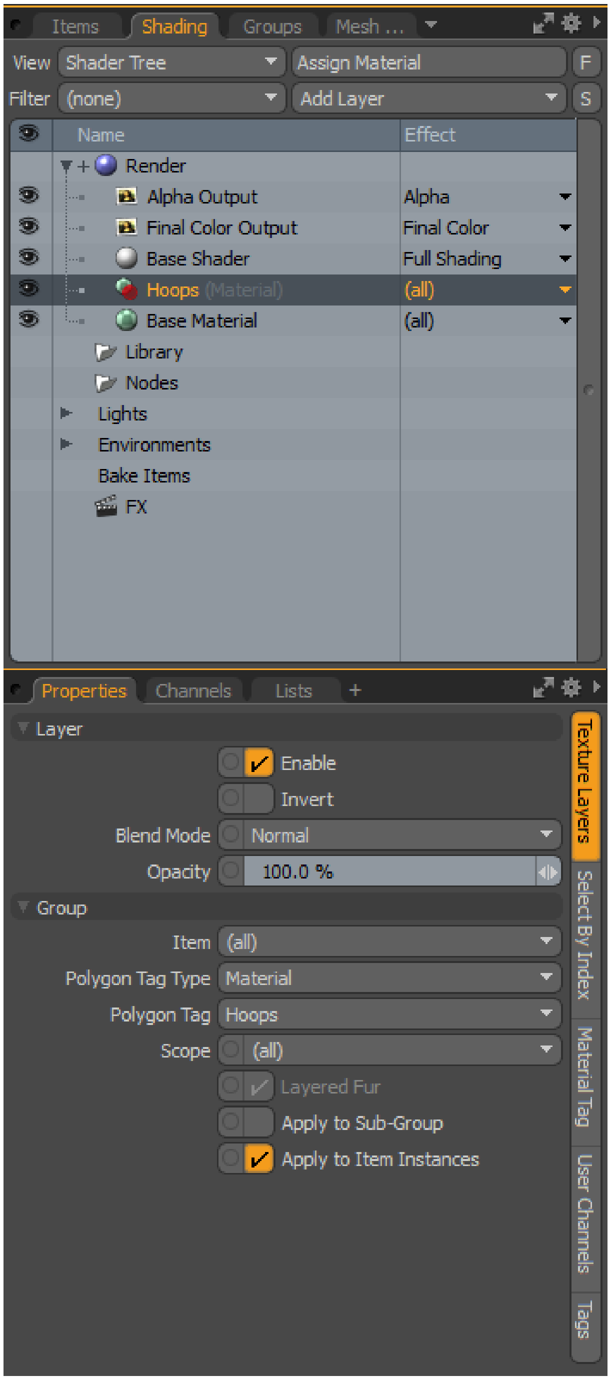
1. In the **Schematic** viewport, change the **View Type** to **Default** to see the polygons clearly.
2. Activate **Polygons** selection mode, click on a polygon, and press **L** to complete the loop. Repeat this step for every loop you want to color red.


Perspective
Default
Ray GL: Off
+X
192 Polygons
Channels: 0
Deformers: ON
GL: 14,208
200 mm
Workspace: Workspace
Add Selected
Add...
Bend (Full Mesh)
Effector
Falloffs
Geometry
Bend Effector
Mesh
Capsule
Tool Pipe
Geometry

3. To add a Material Tag, click **Add...**, select **Mesh Operations** > **Polygon**, and double-click **Material Tag**. Notice that **Select By Index** is automatically added to restrict the material to the polygons that were selected.


Workspace: Workspace
Add Selected
Add...
Bend (Full Mesh)
Effector
Falloffs
Geometry
Bend Effector
Capsule
Tool Pipe
Geometry
Select By Index
Mesh
Material Tag
Geometry
Selection

4. To assign a name to the **Material Tag**, select the **Material Tag** node, and in the **Item Properties** panel, select the **Material Tag** tab. Set the **Material Name** to **Hoops**.
5. To add a material group to the **Shader Tree**, click the **Shading** tab, and select **Add Layer**>**Group** to hold the material.
6. To match the group to the material in the **Schematic**, select the **Texture Layers** properties, ensure the **Polygon Tag Type** is set to **Material**, and set **Polygon Tag** to **Hoops**.


Items
Shading
Groups
Mesh ...
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Hoops (Material)
(all)
Base Material
(all)
Library
Nodes
Lights
Environments
Bake Items
FX
Properties
Channels
Lists
+
Texture Layers
Select By Index
Material Tag
User Channels
Tags

Layer

☑ Enable
☐ Invert
Blend Mode: Normal
Opacity: 100.0 %

Group

Item: (all)
Polygon Tag Type: Material
Polygon Tag: Hoops
Scope: (all)
☑ Layered Fur
☐ Apply to Sub-Group
☑ Apply to Item Instances

7. To add the material to hold the color, in the **Shader Tree**, with the material group highlighted, select **Add Layer**>**Material**.
8. Select the **Material Ref** properties and change the **Diffuse Color** to red.
9. Press **F8** to preview the render output.

Adding to the Model

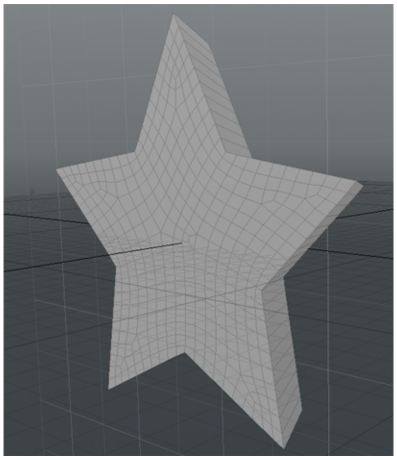
Now that you've tried all three approaches, test your knowledge further by coloring the non-red sections to something other than the red color. When using procedural and schematic modeling, you can also revisit the capsule and edit the settings such as the size or position. Try experimenting to see what happens. Also, try adjusting the Bend Effector settings.

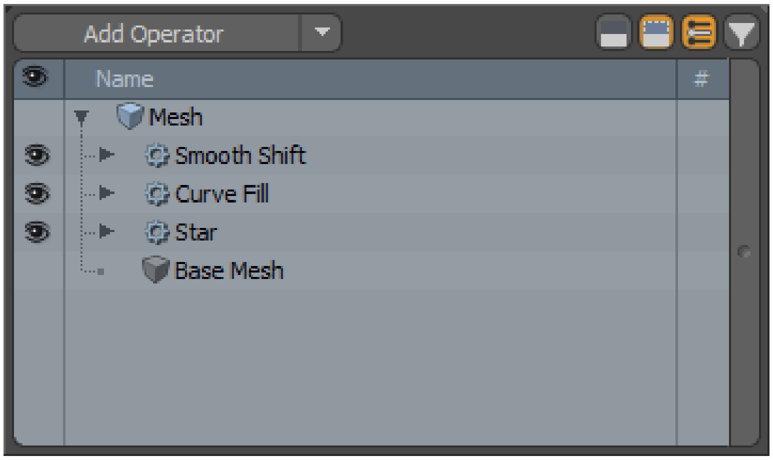
Procedural Modeling with MeshOps

Procedural modeling allows you to edit meshes in a non-desctructive way. You can perform topological edits to a mesh, which can be modified, rigged, and even animated. You can easily revert or modify operations, keeping the rest of the operations intact.

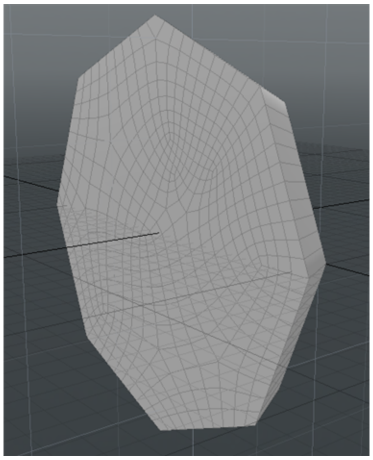
The procedural modeling system works similarly to the deformation system. A mesh layer contains the base mesh, which is procedurally modified. This means that when you edit the mesh, operations are layered on top of the base layer. If you modify the base layer, the operations layered on top are re-evaluated to generate the result.

The image below shows the **Mesh Operations** list with the Base Mesh at the bottom, containing a Star curve primitive with a Curve Fill operation applied to it, followed by a Smooth Shift operation. You can see the resulting mesh on the right.





With procedural modeling, you can change any of the mesh operations without touching the rest of the list. For example, if you swap the Star curve primitive in the example above to an N-Sided curve primitive, you get the following result:



As you can see in the image above, the Curve Fill and Smooth Shift still get applied to the N-Sided curve primitive, and there's no need to re-create the mesh from scratch.

When an operation in the stack changes, only the operations on top of that layer are re-evaluated. This means that applying a procedural operation on the top layer of the stack is as fast as applying a single operation to a base mesh.

You can access the procedural modeling tools in the **Mesh Ops** tab in the **Model** and **Setup** layouts.

For more information on how to use the panel, see Using the Mesh Ops Tab.

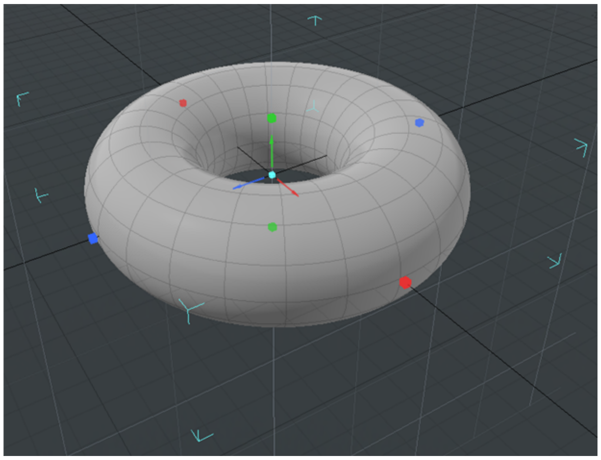
The procedural modeling system is made up of two main components: mesh operations and deformers. Mesh operations in turn have three types: tool operations, sub-tools, and selection operations.

You can find details on each of these categories below.

Mesh Operations

Mesh operations are the key components to the procedural modeling system. They are very similar to deformers, but unlike deformers, they are allowed to change topology, rather than just modifying the position of existing elements. They are integrated into the evaluation system, and are applied as a layer on top of the **Mesh Operations** list. When mesh operations are evaluated, you're provided with an editable mesh, and you can perform any edit on it.

Procedural primitives and curves are examples of mesh operations. The images below show the Toroid tool in the 3D viewport and in the **Mesh Operations** list.



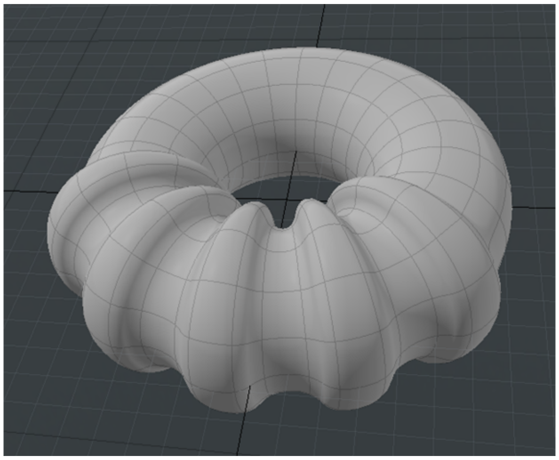
There are different types of mesh operations: tool operations, sub-tools and selection operations.

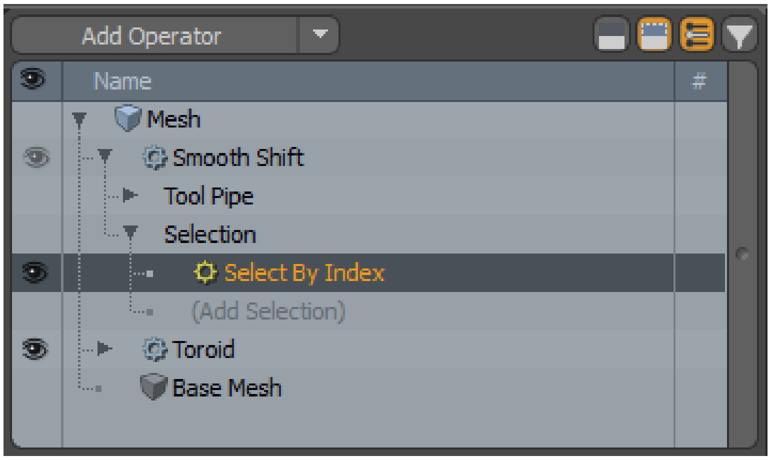
Tool Operations

Modo has a diverse and feature-rich modeling toolset. The procedural modeling system takes advantage of this set by providing each tool's functionality as a procedural operation. The result of this is a tool operation.

Most of the existing modeling tools have been converted into procedural operations, and their functionality automatically wrapped by a procedural mesh operation. Unlike a standard mesh operation, procedural tools have access to the procedural Tool Pipe, and various sub-tools to modify their behaviors. Procedural tools also provide tool handles for user interaction. Apart from these two additions, they work identically to a mesh operation.

The images below show a Smooth Shift operation applied to a Toroid procedurally.




Add Operator
Name
#
Mesh
Smooth Shift
Tool Pipe
Selection
Select By Index
(Add Selection)
Toroid
Base Mesh

Sub-Tools

Similar to the existing Modo toolset, you can modify the properties and evaluation of the procedural tools using a Tool Pipe. For example, a symmetry sub-tool can be combined with a transform operation to perform a symmetrical transform. In the procedural modeling system, sub-tools can be associated with a mesh editing tool in a similar way, to modify how it operates. For instance, you can add a Path Generator as a sub-tool to a Text mesh operation to have the text walk along a curve.

The procedural modeling system can also evaluate other item types as sub-tools. For example, falloff items can be connected to a procedural tool, and evaluated exactly like a falloff tool. Combining multiple sub-tools together can create new and unique operations.

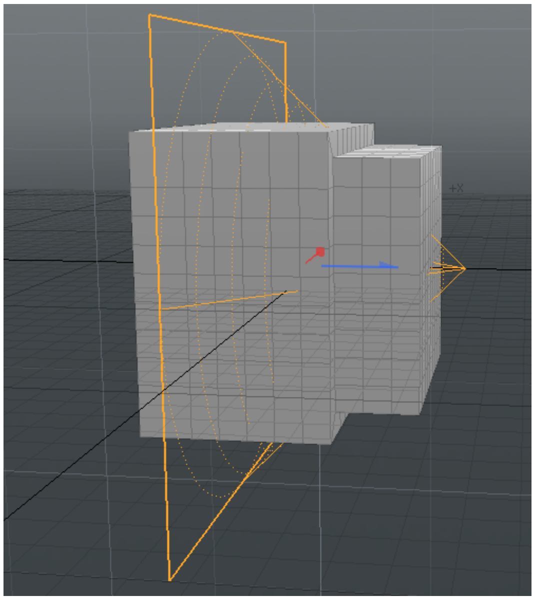
Selection Operations

Procedural modeling works by maintaining operations and their settings in the **Mesh Operations** list. Any selections must also be stored with the tool operations to allow revision of the selection in the procedural stack.

Selection operations provide this functionality. A selection operation is a special item type that can be associated with mesh operations and procedural tools. As the procedural operation is evaluated, the selection operation is also evaluated, and provides the target elements for any mesh edits. The basic selection operation uses mesh element indices to define which elements are modified. You can define these indices by painting a selection in the 3D viewport.

Additional selection operations can define a procedural selection, allowing the properties of the mesh to define the target for the edit. For example, a procedural selection operation can automatically select all elements of the mesh affected by a falloff. As the base mesh is edited, and the affected area changes, any operation using that selection is modified to operate on different elements.

The image below shows the effect of a Polygon Bevel on a cube, using a Linear Falloff to define what elements are affected.



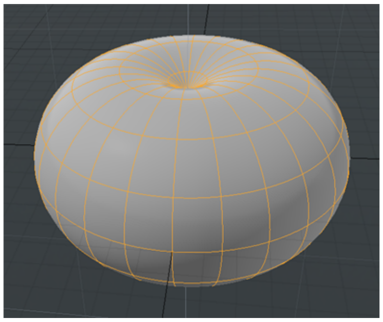
Note: For details on selection operations, see Procedural Selection.

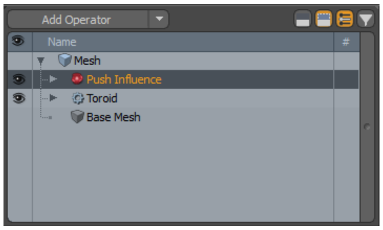
Tip: To quickly select and edit a base mesh in your **Items Lists**, right-click on a mesh and select **Edit Base Mesh** from the pop-up contextual menu.

Deformers

Deformers are tightly integrated into the procedural modeling workflow. They are managed in the same stack as mesh operations, and can be mixed together to apply deformers before or after mesh operations and vice versa. For more information on deformers, see Applying Deformers.

The images below show the Push Influence applied to a procedural Toroid mesh.





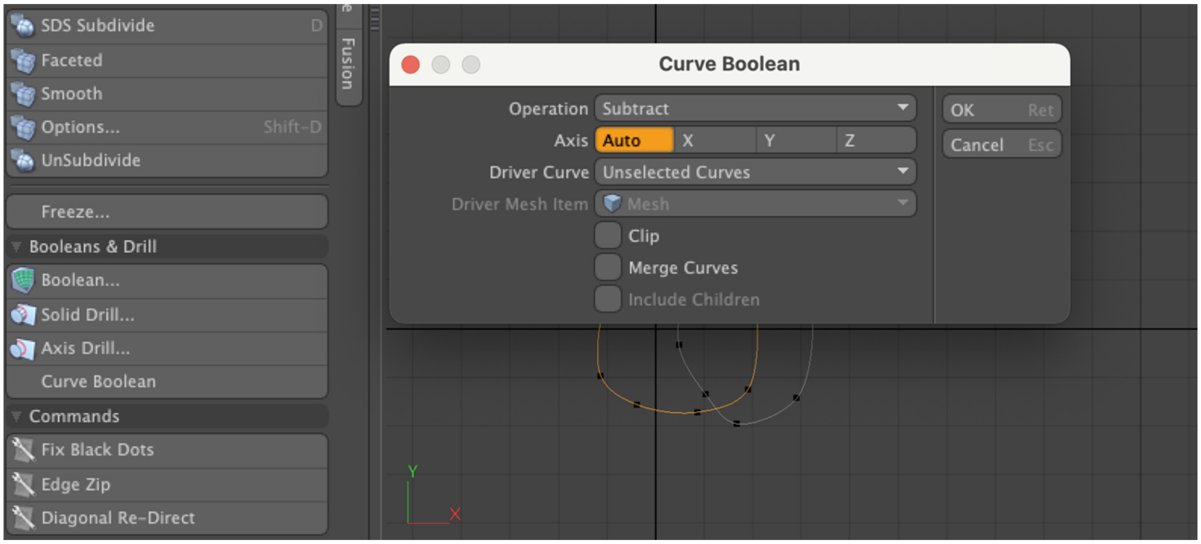
Create Complex Procedural Meshes With Curves and Curve Booleans

Curve booleans allow you to create complex 2D shapes using curve tools and then create a system of booleans to convert your 2D curves into 3D models. This tool can be used in both direct modeling and procedural.

Direct Modeling Command

You can run the **curve.boolean** command to apply the curve boolean tool to selected curve polygons in an active mesh.

1. Click on the **Edit** tab on the left hand side.
2. Under **Booleans & Drill**, click the **Curve Boolean** button.

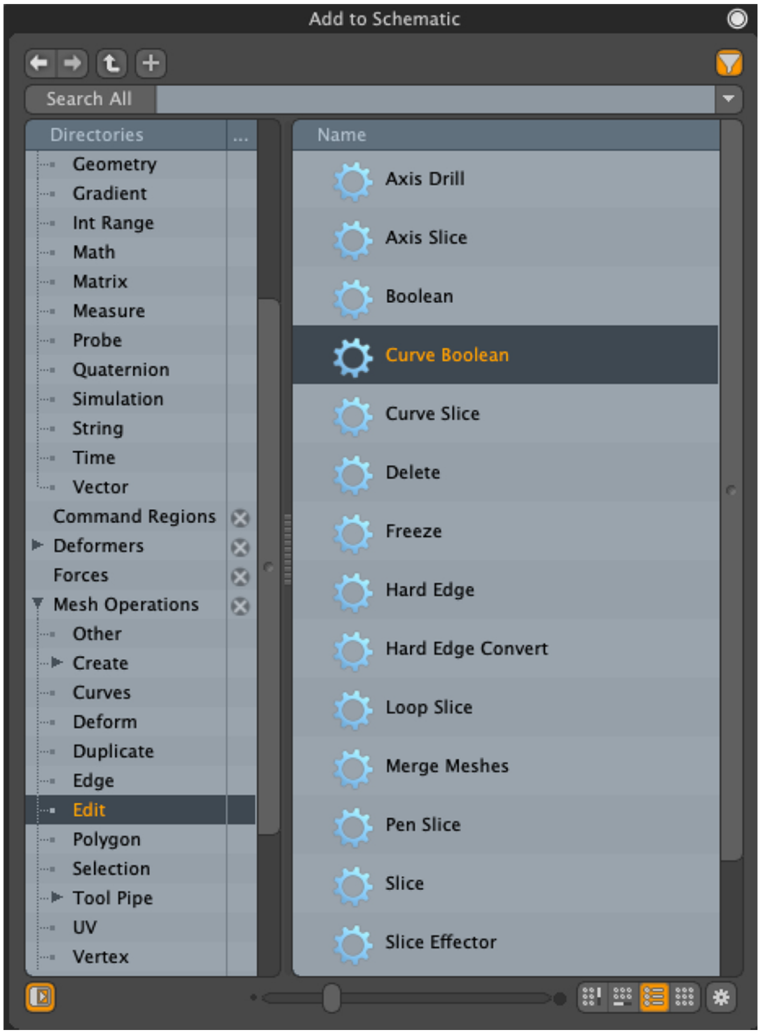


SDS Subdivide D
Faceted
Smooth
Options... Shift-D
UnSubdivide
Freeze...
Booleans & Drill
Boolean...
Solid Drill...
Axis Drill...
Curve Boolean
Commands
Fix Black Dots
Edge Zip
Diagonal Re-Direct
Fusion
Curve Boolean
OK Ret
Cancel Esc
Y
X

Operation: Subtract
Axis: Auto | X | Y | Z
Driver Curve: Unselected Curves
Driver Mesh Item: Mesh
☐ Clip
☐ Merge Curves
☐ Include Children

Procedural Modeling Mesh Operator

The **Curve Boolean** mesh operator is available in the **Edit** category of the **Mesh Operations**.


Add to Schematic
Search All
Directories
Geometry
Gradient
Int Range
Math
Matrix
Measure
Probe
Quaternion
Simulation
String
Time
Vector
Command Regions
Deformers
Forces
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Edit
Polygon
Selection
Tool Pipe
UV
Vertex
Name
Axis Drill
Axis Slice
Boolean
Curve Boolean
Curve Slice
Delete
Freeze
Hard Edge
Hard Edge Convert
Loop Slice
Merge Meshes
Pen Slice
Slice
Slice Effector

Options

Just like with regular booleans, there are four types of curved boolean you can create: **Slice**, **Union**, **Intersect** and **Subtract**.

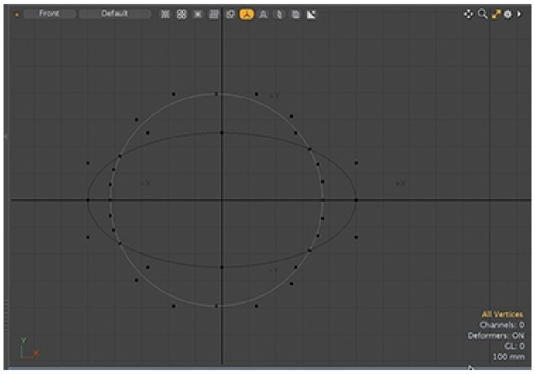
Operation

Slice - allows you to overlap multiple curves and create new control points where the curves intersect.

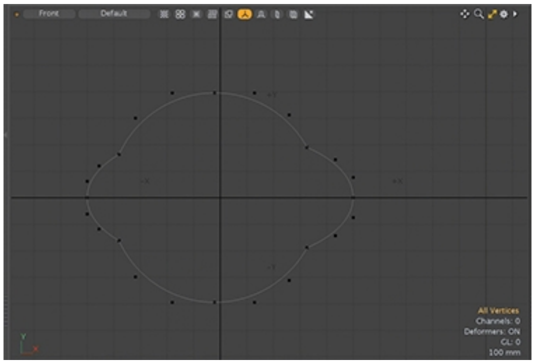
Union - allows you to merge multiple overlapping curves into one shape and remove any overlapping curve segments.

Intersect - allows you to overlap two curves and then delete anything that exists outside of the overlap.

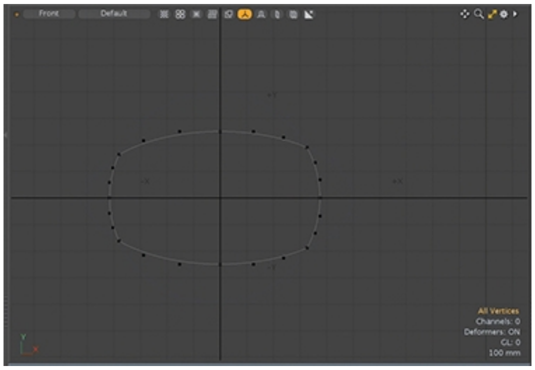
Subtract - allows you to use a curve to remove segments of another curve and can be used to cut out shapes from a base mesh.


Front
Default
All Vertices
Channels: 0
Deformers: ON
GL: 0
100 mm

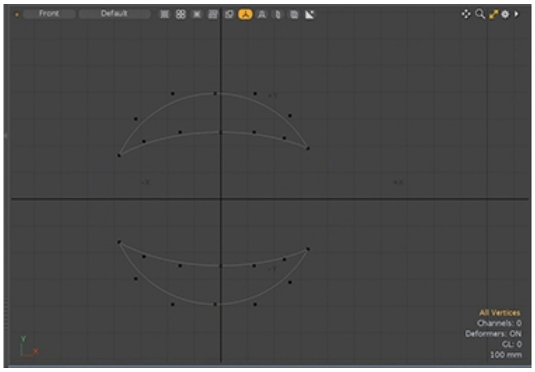
Slice


Front
Default
All Vertices
Channels: 0
Deformers: ON
GL: 0
100 mm

Union


Front
Default
All Vertices
Channels: 0
Deformers: ON
GL: 0
100 mm

Intersect


Front
Default
All Vertices
Channels: 0
Deformers: ON
GL: 0
100 mm

Subtract

Axis

Here you can specify the axis which defines the 2D plane for the boolean operation. Auto is the default mode, which computes the fitting plane from point positions of active curves.

Driver Curve

Unselected Curve operates selected curves by unselected curves in same mesh item.

When mode is **Union** and all curves are selected, curve boolean makes union curves with all selected curves.

When **Driver Mesh** is set, curve boolean operates selected curves in active mesh by curves in specific separated mesh item.

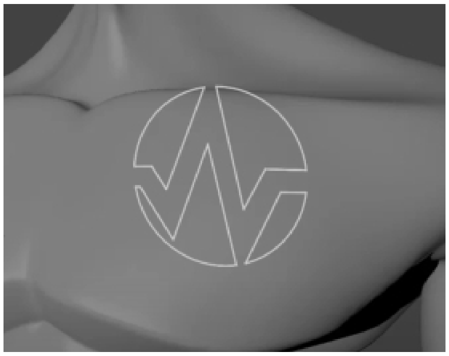
Driver Mesh Item

A driver mesh is a mesh that is used to create, or drive, the boolean.

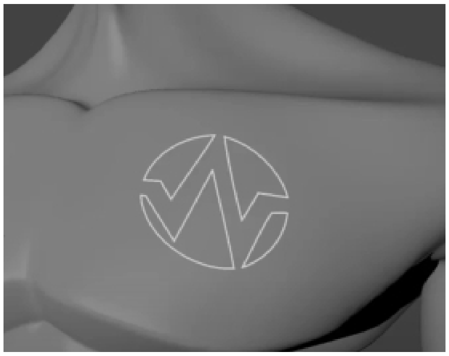
Choose a mesh item with driver curves from this list when **Driver Curve** is **Driver Mesh**. If you're going to be adding multiple drivers it is recommended that you use the schematic viewport, because it keeps organization a bit easier and means you don't have to keep jumping between mesh items.

Constraint Target

The **Constraint Target** option is useful for applying boolean operations to 3D curves which are then constrained to a target 3D mesh surface. When a mesh item is assigned a **Constraint Target** the curves are given new vertex positions on the constraint target mesh surface.

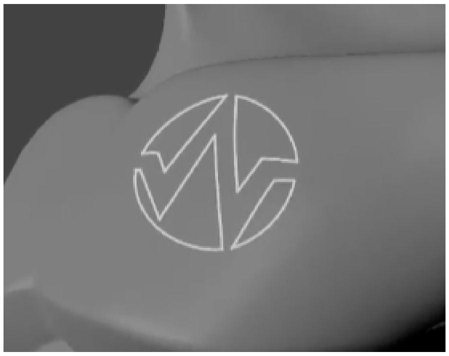


Curve before **Constraint Target** assigned

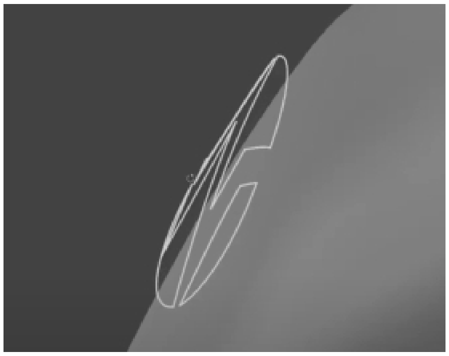


Curve after **Constraint Target** assigned

The **Offset from Target** property defines how far along the normal vector from the constraint target the curve is positioned. Higher values move the curves away from the target mesh.



Offset from Target 0

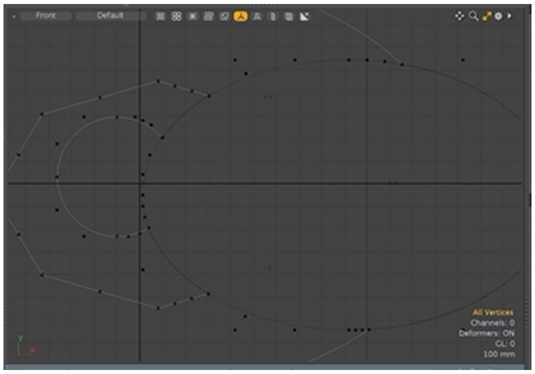


Offset from Target 10 mm

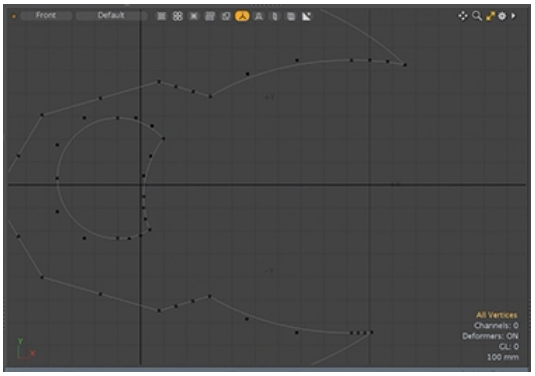
Once the curve is constrained to the target mesh, you can use the translate handles to position the curve by moving it along the constrain surface.

Clip

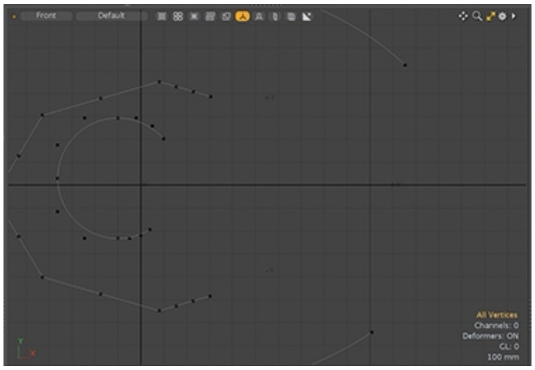
When **Clip** is enabled, the curve boolean cuts the active curves. It does not append curve segments from driver curves.



Curve applied



Clip disabled



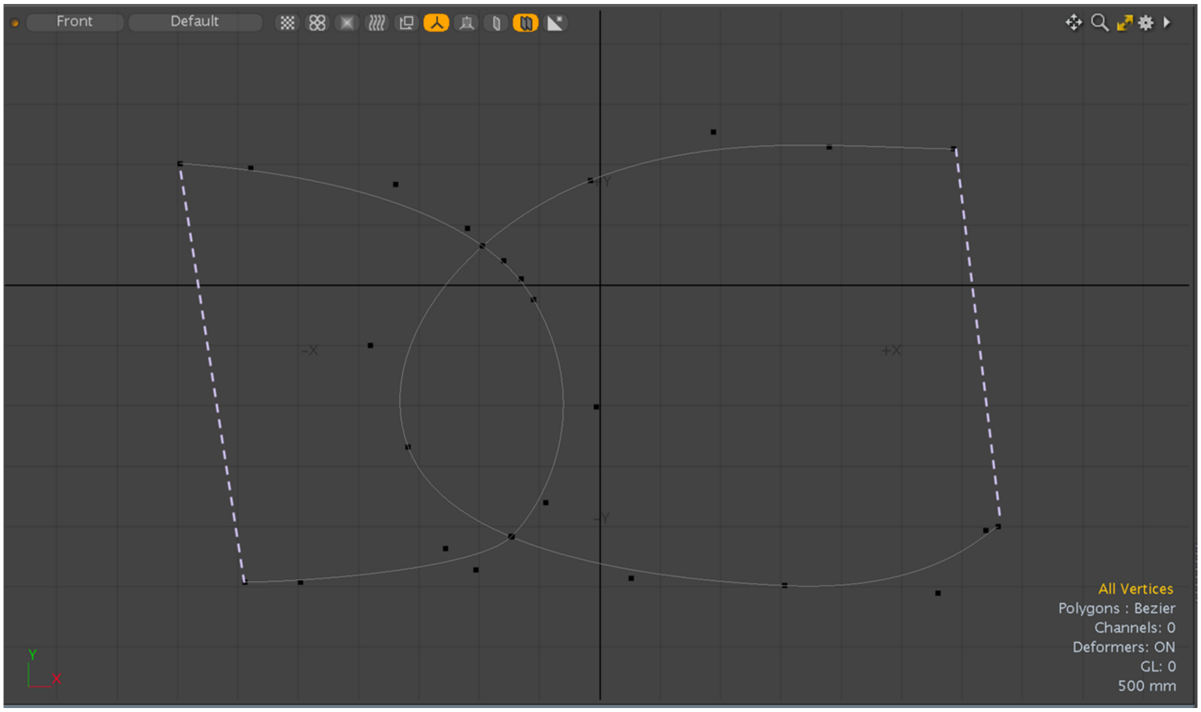
Clip enabled

Merge Curves

When **Merge Curves** is enabled, the curve boolean joins split curve segments into a new curve polygon. This applies to Polyline and Bezier polygon types. Curve and B-Spline types are not merged so as to maintain sharp corners.

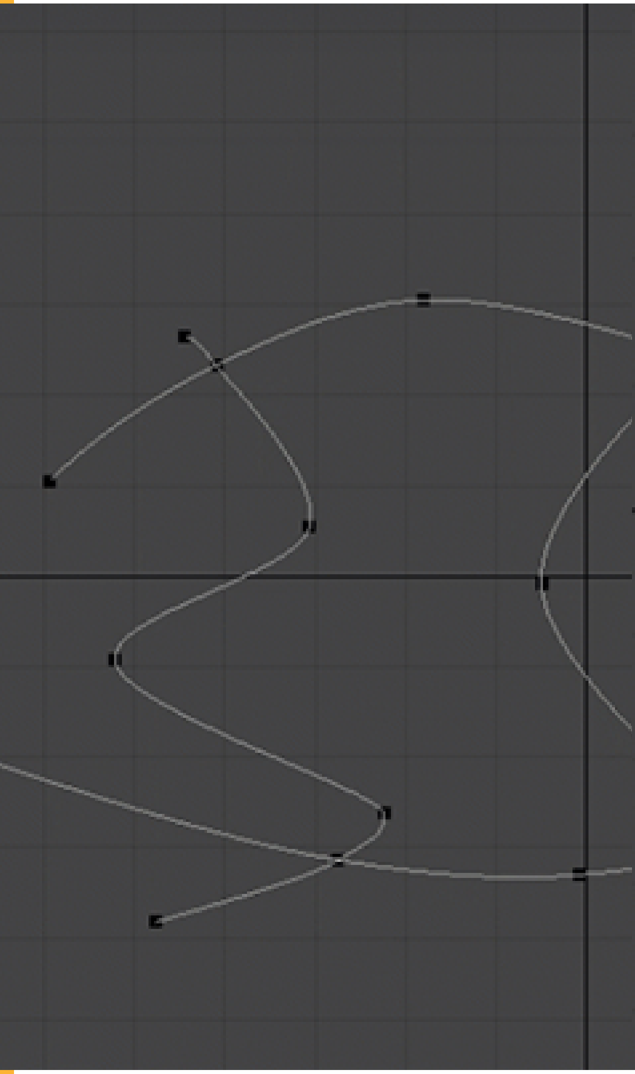
Close Open Curves

When **Close Open Curves** is enabled, the curve boolean treats opened curves as a closed loop by connecting the 1st and last points. This is useful for applying Subtract or Intersect to opened curves.

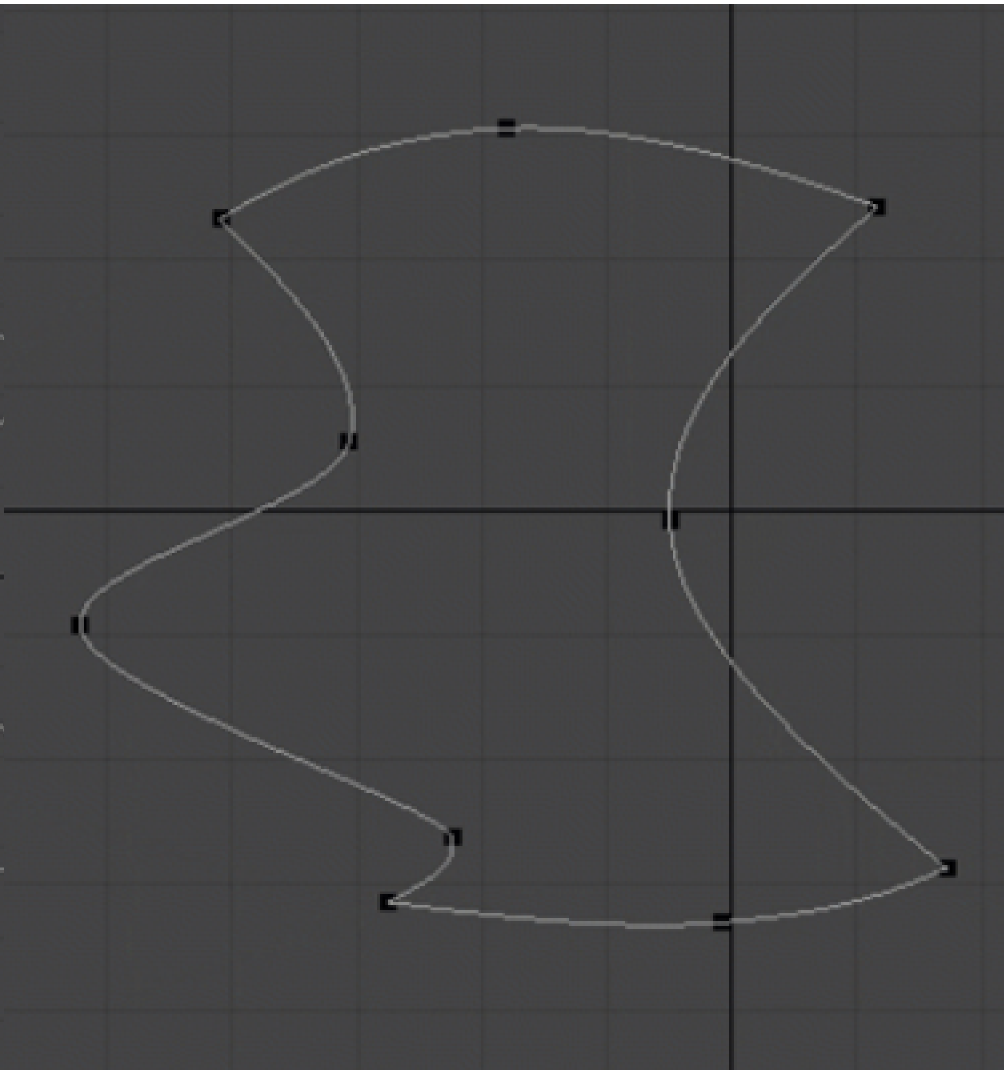

Front
Default
+Y
-X
+X
-Y
Y
X
All Vertices
Polygons : Bezier
Channels: 0
Deformers: ON
GL: 0
500 mm

Trim Open Curves

When **Trim Open Curves** is enabled, the Curve Boolean trims opened curves at the point of intersection.



Overlapping curves with **Trim Open Curves** disabled.

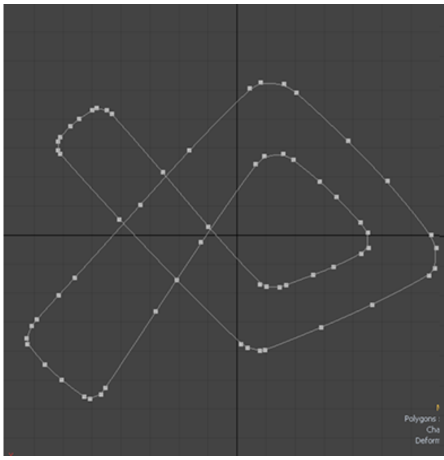


Trim Open Curves is enabled. The overlapping curves are merged into a single shape, and excess parts of the curve are removed.

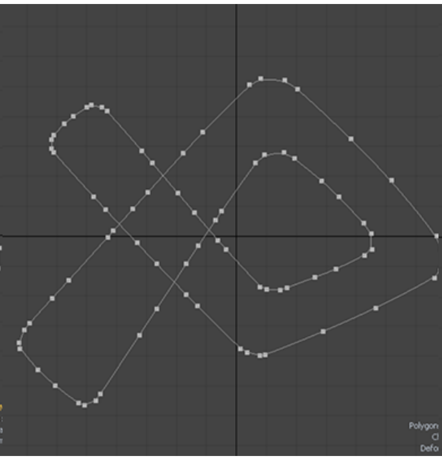
Self Intersection

When **Self Intersection** is enabled, Curve Boolean makes intersections by segments in the same curve loop.

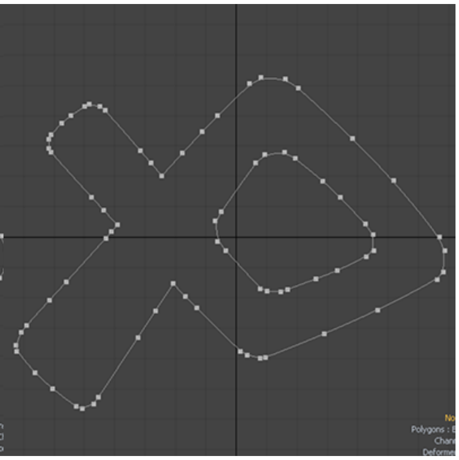
- **Slice** mode adds the intersection points on the interfering segments.
- **Union** mode removes the sliced segments inside the curve loop.



Self Intersection disabled.



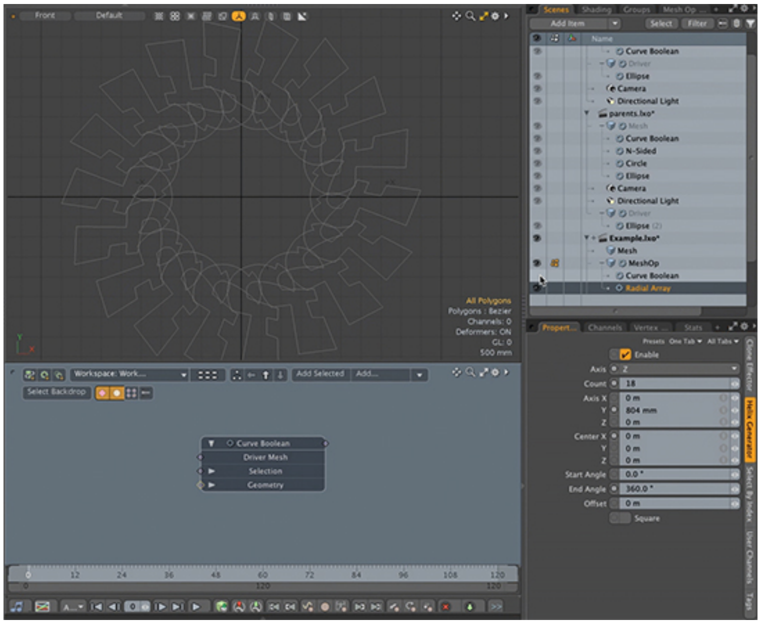
Self Intersection enabled with **Slice** mode selected.



Self Intersection enabled with **Union** mode selected.

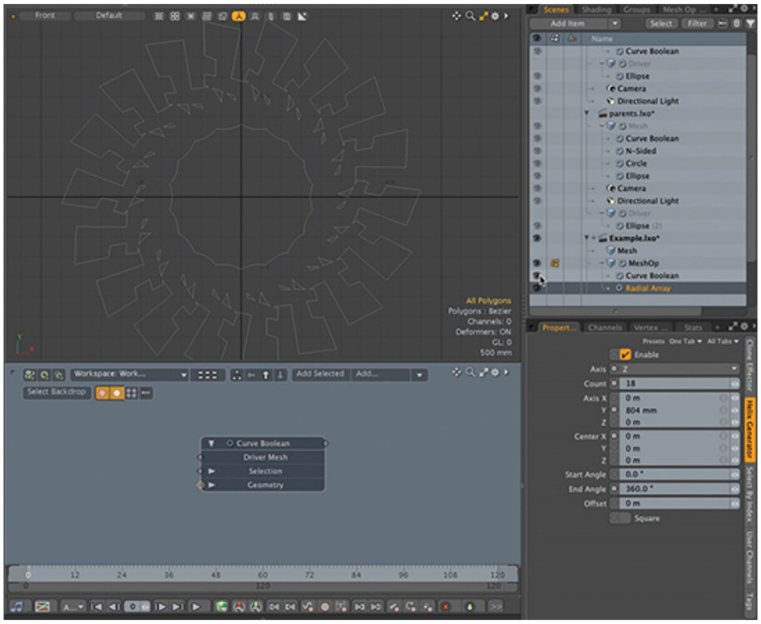
Operation Examples

Union Example


Front
Default
All Polygons
Polygons : Bezier
Channels: 0
Deformers: ON
GL: 0
500 mm
Scenes
Shading
Groups
Mesh Op ...
Add Item
Select
Filter
Name
Curve Boolean
Driver
Ellipse
Camera
Directional Light
parents.lxo*
Mesh
Curve Boolean
N-Sided
Circle
Ellipse
Camera
Directional Light
Driver
Ellipse (2)
Example.lxo*
Mesh
MeshOp
Curve Boolean
Radial Array
Workspace: Work...
Add Selected
Add...
Select Backdrop
Curve Boolean
Driver Mesh
Selection
Geometry
Propert...
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Clone Effector
Helix Generator
Select By Index
User Channels
Tags

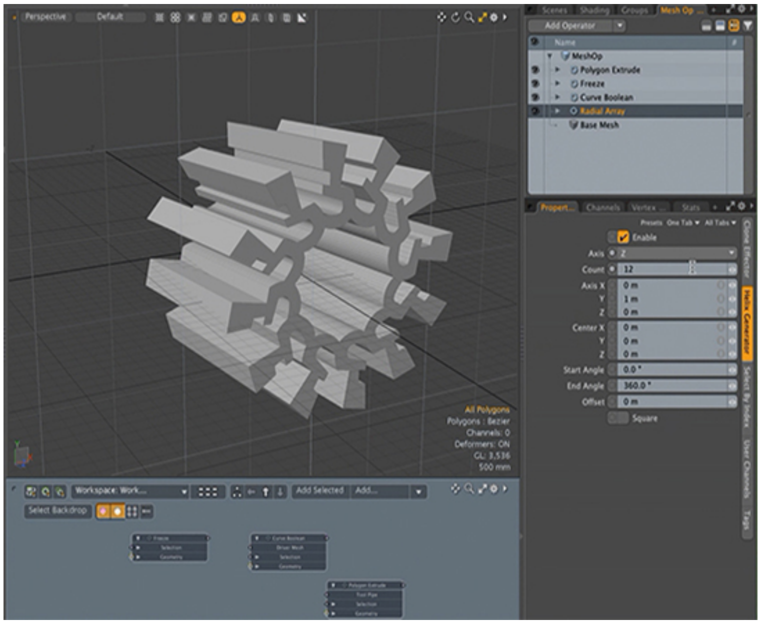
☑ Enable
Axis Z
Count 18
Axis X 0 m
Y 804 mm
Z 0 m
Center X 0 m
Y 0 m
Z 0 m
Start Angle 0.0 °
End Angle 360.0 °
Offset 0 m
☐ Square

Union off


☑ Enable
Axis: Z
Count: 18
Axis X: 0 m
Y: 804 mm
Z: 0 m
Center X: 0 m
Y: 0 m
Z: 0 m
Start Angle: 0.0 °
End Angle: 360.0 °
Offset: 0 m
☐ Square

Curve Boolean
Radial Array
MeshOp
Helix Generator

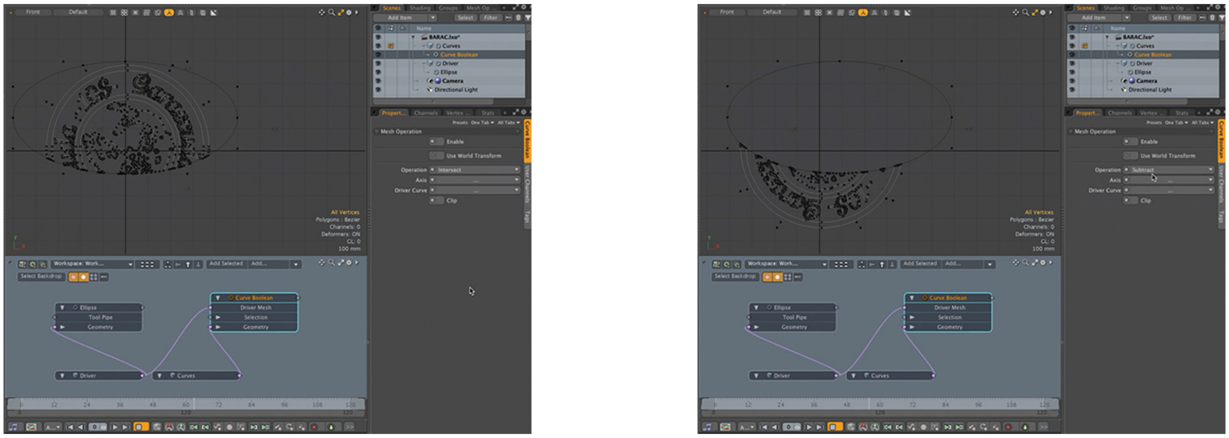
Union on


☑ Enable
Axis: Z
Count: 12
Axis X: 0 m
Y: 1 m
Z: 0 m
Center X: 0 m
Y: 0 m
Z: 0 m
Start Angle: 0.0 °
End Angle: 360.0 °
Offset: 0 m
☐ Square

MeshOp
Polygon Extrude
Freeze
Curve Boolean
Radial Array
Base Mesh
Helix Generator

Freeze and Polygon Extrude applied

Intersect and Subtract Example



Intersect

Subtract

Create Flat 2D Geometry With the Particle Plane Tool

The **Particle Plane** operator allows you to create a flat 2D plane generated by points, and is driven by the **Primitive Generator** operation. You are also able to create planes from primitive shapes by selecting a preset within the **Content Preset** operation found in the **Particle Plane** operation's Tool Pipe.

When working with the **Decals...** button or decals workflows, if a source decal is not set, the **Particle Plane** operator is automatically made active, and a plane to carry a decal is created.

Note: For more information on decal workflows in Modo, see Projecting Decals with Planar Decals.

To use the **Particle Plane** operator:

1. Create a new Mesh Item by pressing **N**, or by clicking **Add Item** in the Items List and selecting **Mesh** from the window that opens.

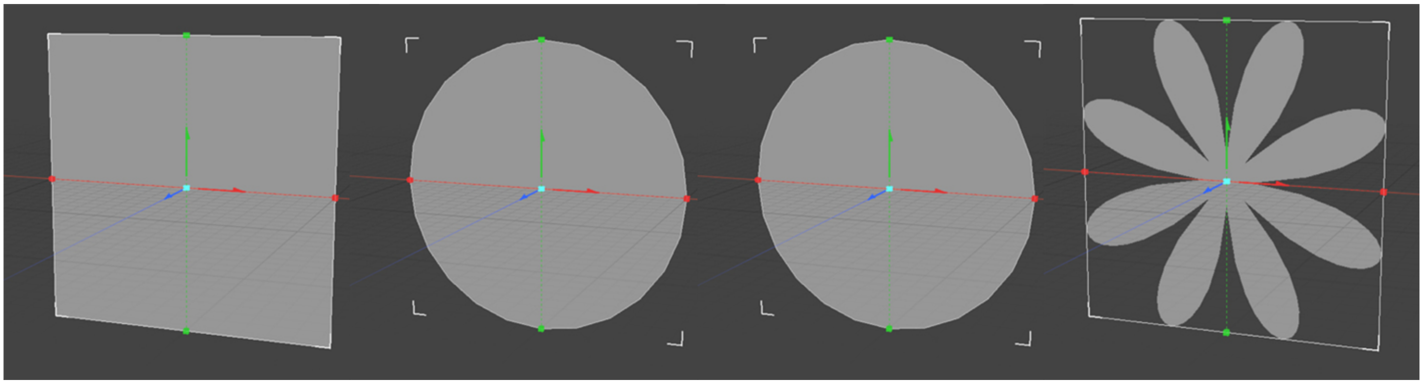
   This Mesh Item holds the **Particle Plane** operator, so it is important to leave the Mesh Item empty.
2. In the Mesh Operations Viewport, click the **Add Operator** button.

   If working in the Modo layout, open the Mesh Operations Viewport by clicking the Mesh Operations Viewport button above your 3D Viewport.
3. In the window that appears, type in **Particle Plane**, and select it from the list to place it in the Mesh Item.

A flat, square plane is created by default. The shape created by the **Particle Plane** operator is adjusted by changing the **Type** in the Properties menu for the **Primitive Generator** available in the Tool Pipe of the **Particle Plane**. The options available are:

- Rectangle
- Circle
- Ellipse
- Preset

Preset allows you assign a custom shape created with curves to the **Particle Plane** operator, allowing you to create complex shapes quickly and easily.



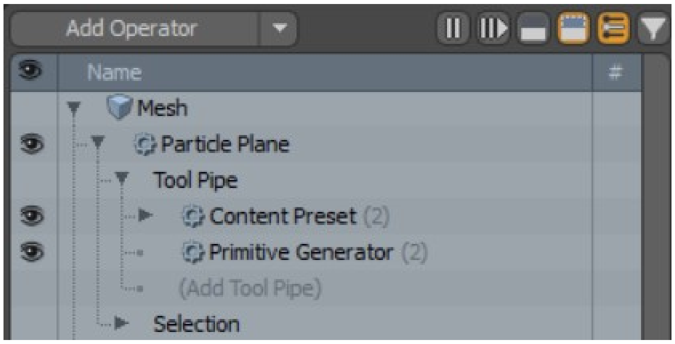
Type set to Rectangle **Type** set to Circle **Type** set to Ellipse **Type** set to Preset

Create Custom Shapes with Presets

Within the Tool Pipe of the **Particle Plane** operator, is a **Content Preset** operator. The **Content Preset** generator allows you to select preset shapes, profiles that ship with Modo, or any custom presets you may have created.

To tell the **Particle Plane** operator to use a preset shape:

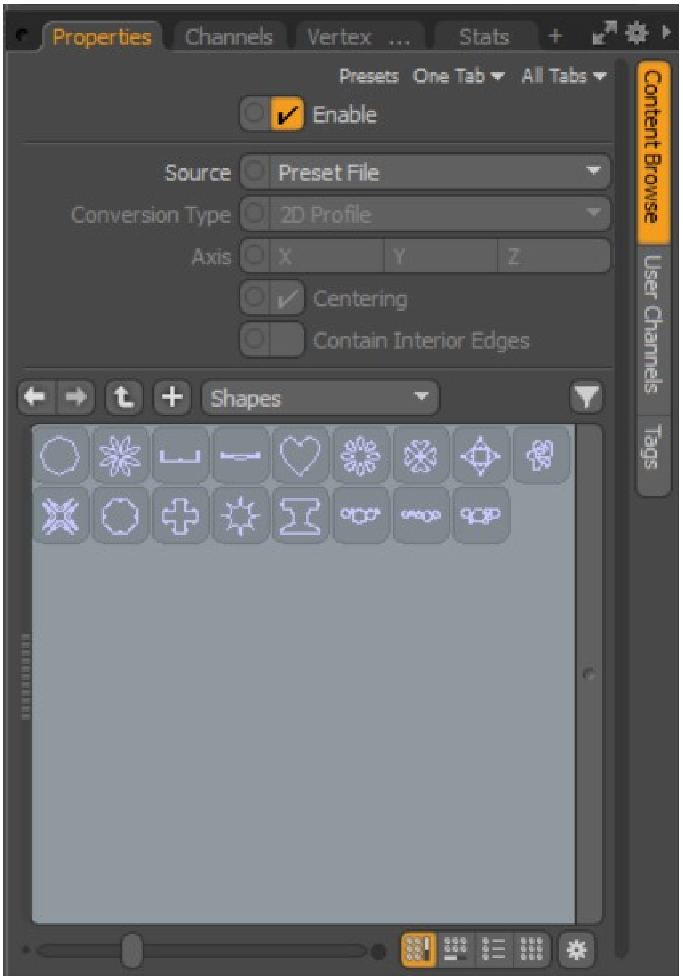
1. In the Mesh Operations Viewport, expand the **Particle Plane** operator to the Tool Pipe level.



2. Click the **Primitive Generator** operation to access its Properties menu.
3. In the Properties menu, change the **Type** to **Preset**.

   Once changed, your plane may disappear from the viewport. This is because a preset for the plane to be created from needs to be set.
4. Click the **Content Generator** operation to access its Properties menu.

5. In the Properties menu, select the preset that you want to use from the **Content Browse** tab.


Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Content Browse
User Channels
Tags
Shapes

☑ Enable

Source: Preset File

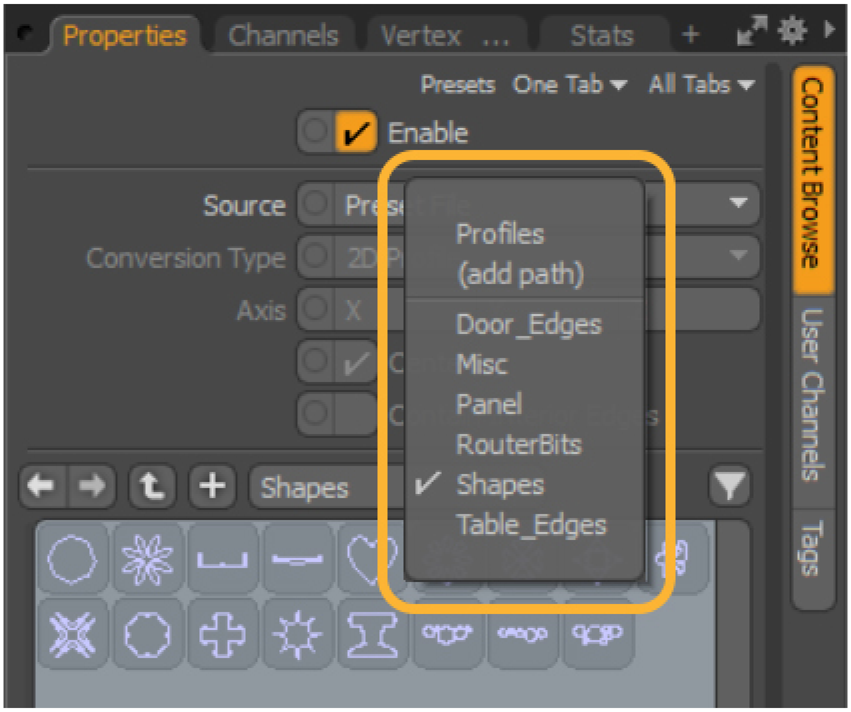
Conversion Type: 2D Profile

Axis: X Y Z

☑ Centering

☐ Contain Interior Edges

Additional presets are accessed by clicking the **Shapes** button and then selecting a new preset type from the dropdown.

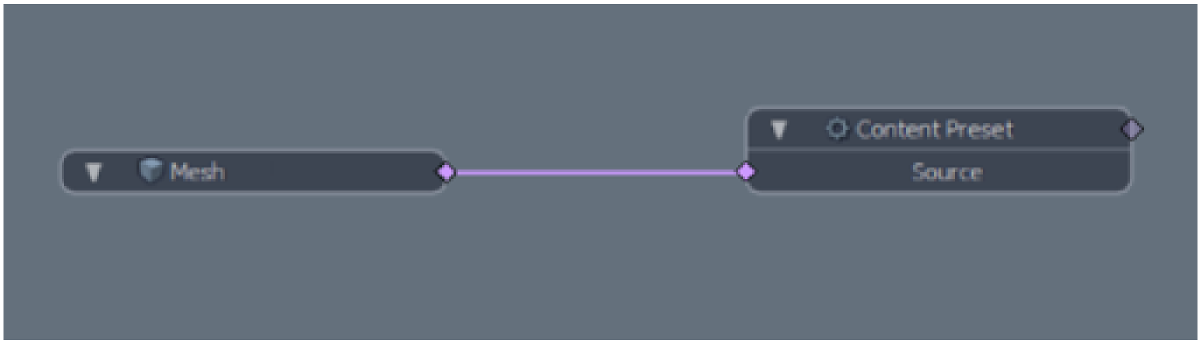

Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Enable
Source
Conversion Type
Axis
Profiles
(add path)
Door_Edges
Misc
Panel
RouterBits
✓ Shapes
Table_Edges
Shapes
Content Browse
User Channels
Tags

Creating Interactive Custom Profile Presets

Profile presets are 2D curves that, when used in conjunction with operations like bridge or bevel, allow you to change the profile or silhouette of your geometry. Modo comes with several presets out of the box, but you can also create your own using MeshOps. Profile Presets can also be used with the Slice Effector and Primitive Slice tools.

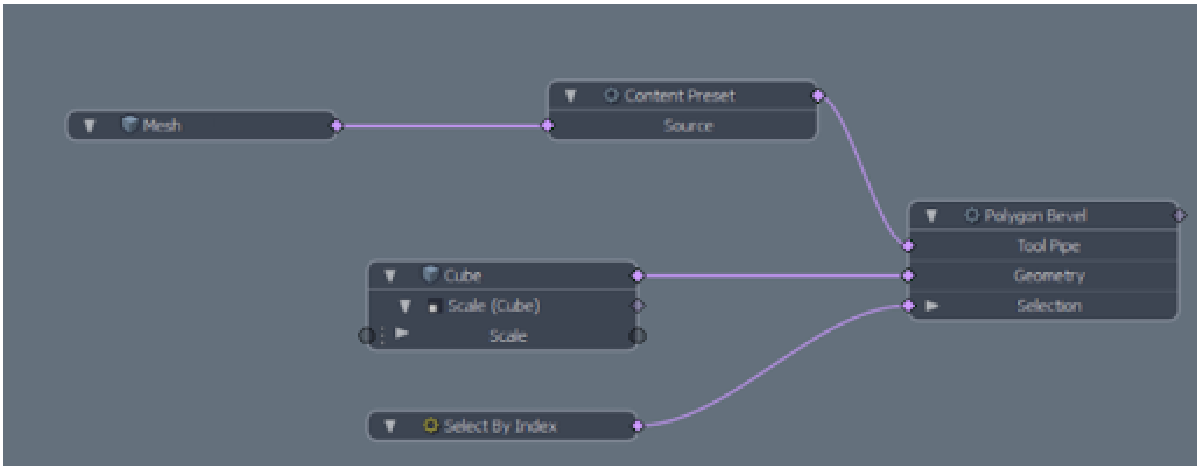
When created procedurally, the curve remains active and can be edited even while applied to a bevel or primitive slice. To create an Interactive custom Profile Preset:

1. In an empty mesh item, create a curve. The curve can be open or closed.
2. Add the curve to the Schematic viewport by dragging it from the Items List into the Schematic viewport or by selecting it in the Items List and clicking the **Add Selected** button in the Schematic viewport.
3. In the Schematic viewport, create a Content Preset node and make a connection between your mesh item and the content preset node.

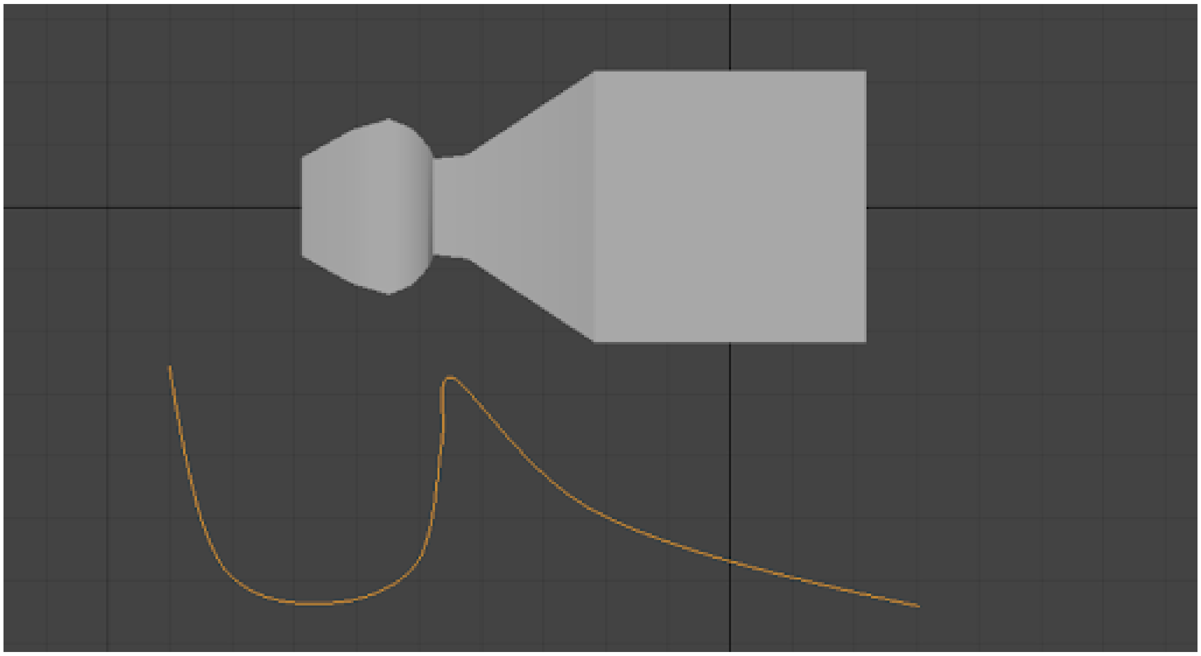

Mesh
Content Preset
Source

4. In the **Properties** tab for the Content Preset node, change the **Source** to **Scene Item**. Depending on what operation you're using the Profile Preset for, you may also need to change the **Conversion Type**. 2D Profile is used for Primitive Slice workflows, while 1D Profile is used for operations such as the Polygon Bevel or Sweep Effector tools.

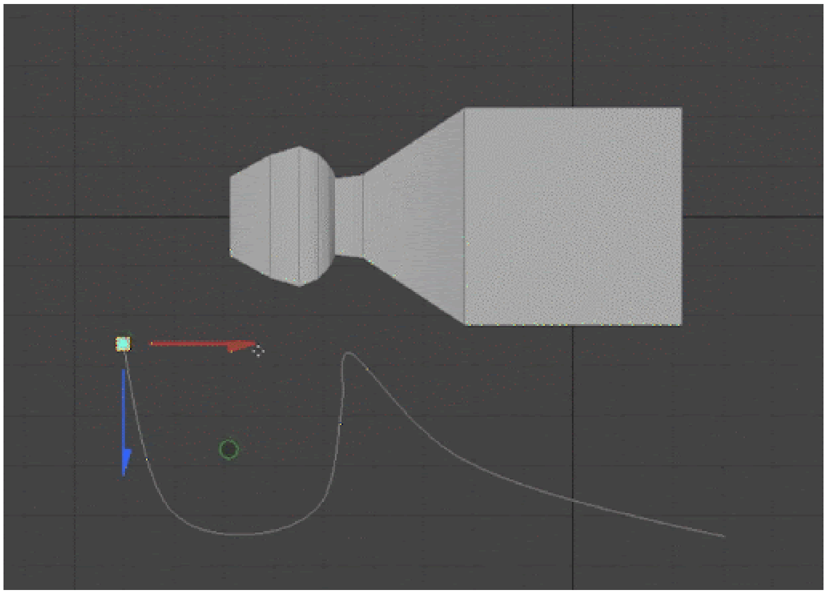
5. Connect the Content Preset node to the **Tool Pipe** port of the operation you want the preset to be applied to. For this example, we are using a Poly Bevel.


Mesh
Content Preset
Source
Polygon Bevel
Tool Pipe
Geometry
Selection
Cube
Scale (Cube)
Scale
Select By Index

The preset is applied.

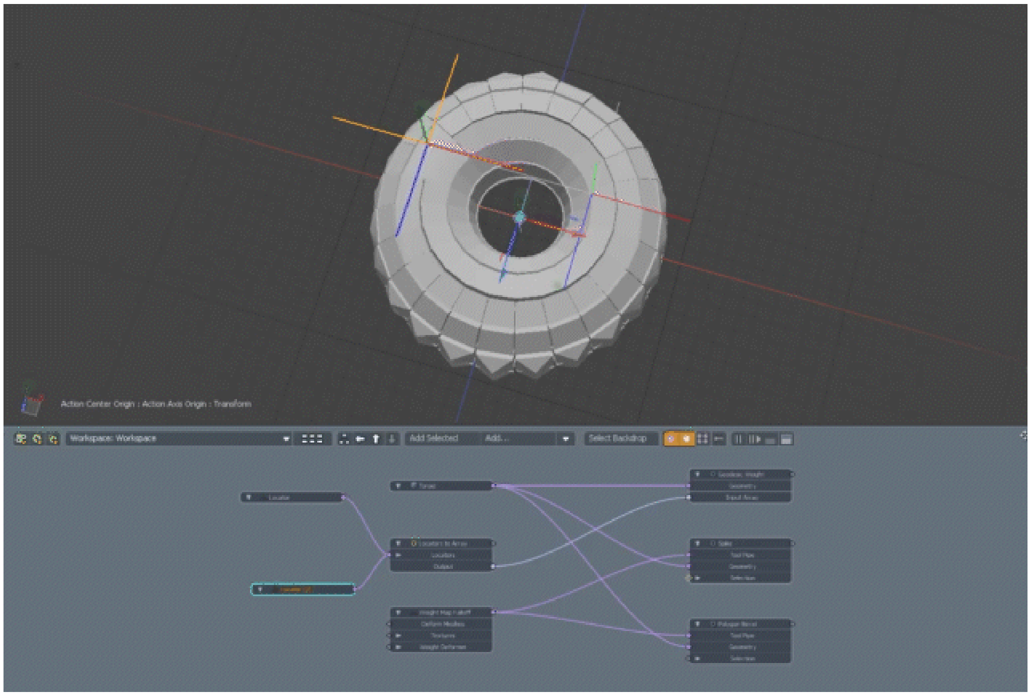


You can continue to edit procedural custom presets even after they are applied. Simply select points or edges on the curve driving the preset and move them. The operation that uses the preset updates as the curve is edited.



Create Intricate Patterns using Geodesic Distance Modifier

The **Geodesic Weight** MeshOp allows you to create a vertex map based on a mesh and a set of input points defined in an array. Through the use of locators and the Geodesic map, you can adjust the intensity of operations applied to a mesh based on the size, distance, and position of the locators.


Action Center Origin : Action Axis Origin : Transform
Workspace: Workspace
Add Selected
Add...
Select Backdrop
Locator
Locators to Array
Locators
Output
Weight Map Falloff
Deform Meshes
Textures
Weight Deformer
Geodesic Weight
Geometry
Input Array
Spike
Tool Pipe
Geometry
Selection
Polygon Bevel
Tool Pipe
Geometry
Selection

The easiest way to use the Geodesic Weight MeshOp is within the Schematic Viewport. To set up the Geodesic Weight operation:

1. Open the Schematic viewport by pressing the Schematic Viewport button at the bottom of the viewport
2. Bring the Geo you want the modifier applied to into the schematic viewport.

   This can be done by adding geo to the schematic viewport directly via the **Add...** button in the schematic viewport, or by dragging your mesh item from the items list into the schematic viewport.

3. Using the **Add...** button, create a **Geodesic Weight** node. Plug your mesh node into the Geometry input in the **Geodesic Weight** node.
4. Create a locator by clicking **Add Item** in the scenes menu and navigating to **Locators>Locators**. Once created, drag the locator item from the items list into the schematic viewport, and then resize and place it as needed within the viewport.
5. Create a **Locator to Array** node in the schematic viewport, and connect your locator and the **Locators** input in the **Locator to Array node**.
6. Connect the **Output** of the **Locator Array** node to the **Input Array** parameter in the **Geodesic Weight node**.

   Locators must be contained within an array for them to be used. However, this does mean that multiple locators can be used to drive the geodesic weight map within one array.

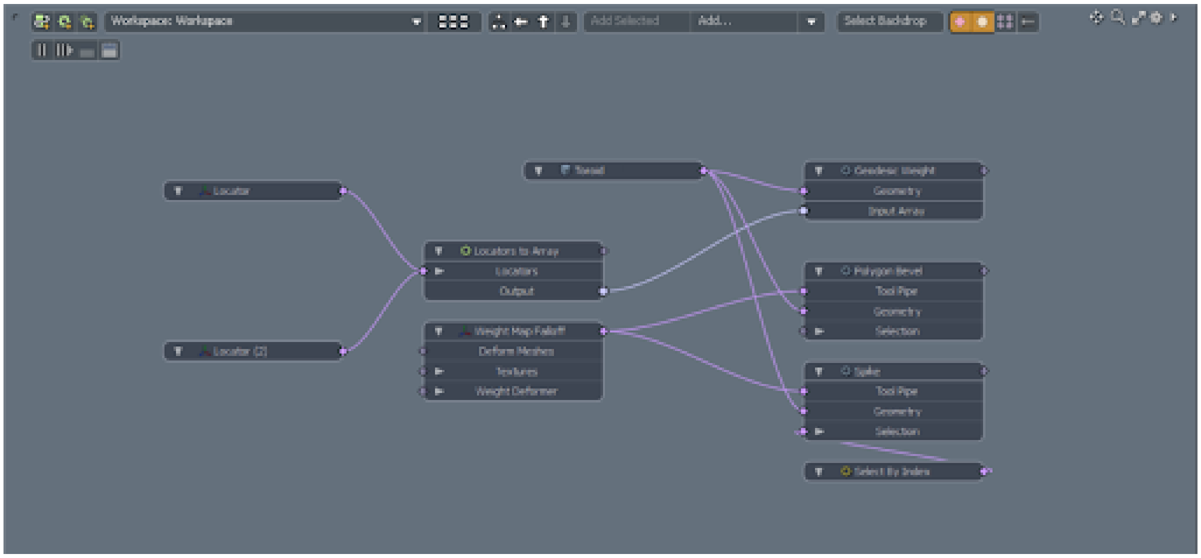
7. From here, you can apply whatever deformation operations you want to your mesh item; for example a poly bevel.
8. Create a **Weight Map Falloff** once the deformations have been applied, and then plug it into the **Tool Pipe** inputs of your deformer nodes.

If multiple deformations have been applied to your mesh, you will need to plug the **Weight Map Falloff** into the tool pipe input of each deformer node

As the **Geodesic Weight** operation creates a weight map, for it to be useable it needs to be assigned to the Mesh Item via the **Weight Map Falloff**.

9. In the properties window of your **Weight Map Falloff**, change the weight map parameter to **Geodesic**

 Your geodesic weight map will be named 'Geodesic' by default. If you have manually changed the name of your Geodesic weight map, this name change will be reflected in the list of assignable weight maps within the weight map falloff.


Workspace: Workspace
Add Selected
Add...
Select Backdrop
Locator
Locator (2)
Torus
Locators to Array
Locators
Output
Weight Map Falloff
Deform Meshes
Textures
Weight Deformer
Geodesic Weight
Geometry
Input Array
Polygon Bevel
Tool Pipe
Geometry
Selection
Spike
Tool Pipe
Geometry
Selection
Select By Index

Example of a **Geodesic Weight** node setup in the Schematic Viewport.

There are also a few options available to you within the **Geodesic Weight** MeshOp properties:

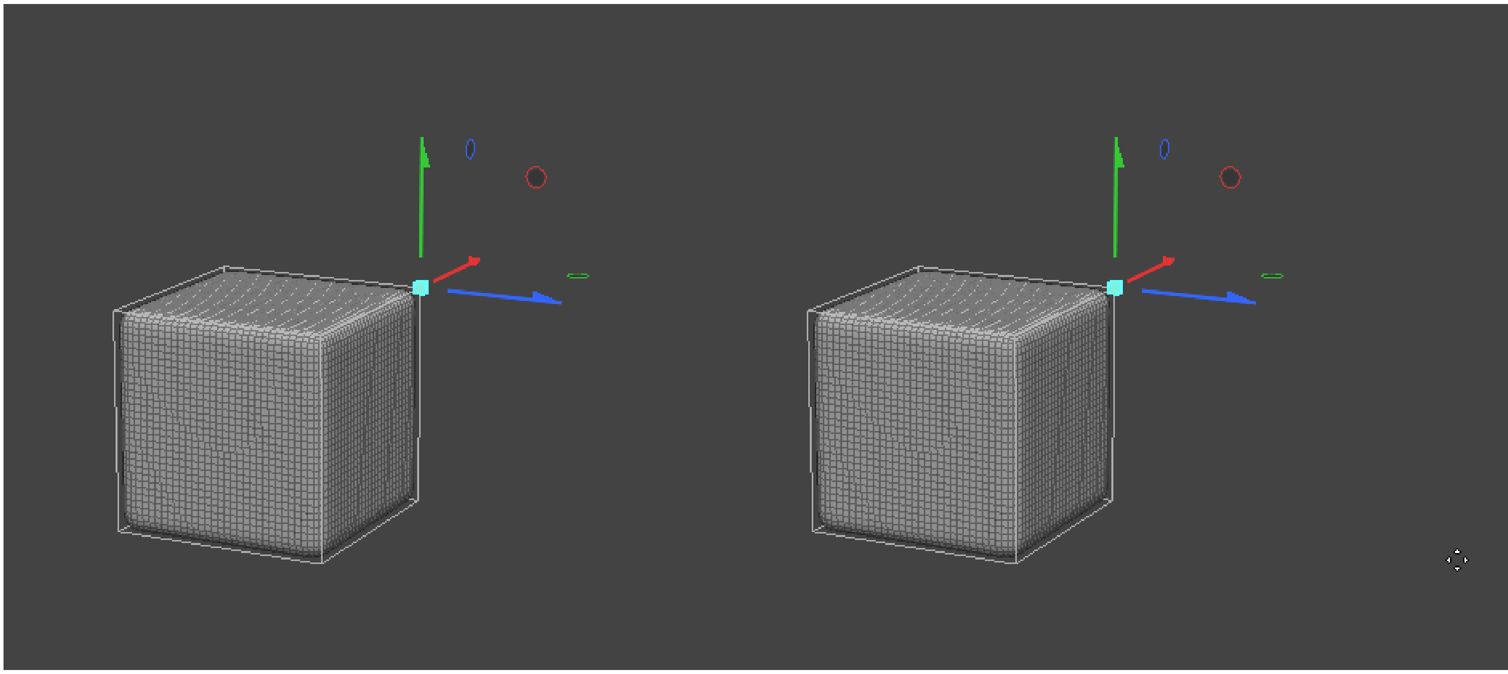
Geodesic Weight Properties	
Map Name	Sets the name of the Geodesic Map to use for output
Select Geodesic Map	Set the generated Geodesic Map as the currently selected Vertex Map
Normalize	Normalizes the values in the vertex map between the input point (0.0) and the furthest point along the surface from the input point (1.0)
Triangulate Mesh	Converts your geometry into triangles when calculated, so that the distance between points can be calculated. This does not impact the topology of your mesh and is used for calculations only. If your

Geodesic Weight Properties	
	mesh is guaranteed to be triangulated, this can be turned off
Invert	Inverts the distance values of your mesh. Your furthest point will gain a value of 1.0, while your input gains a value of 0.0

Improve Mesh Interaction Speeds with the Wrap Effector

The wrap effector is a MeshOp that allows you to assign a low poly version of your mesh to its high poly equivalent so that broad edits can be made without having to compromise on mesh density.

The wrap effector is different from the wrap influence in that it binds target points relative to positions on the wrap mesh, and then blends the binding regions together giving a smoother result.



Wrap Influence

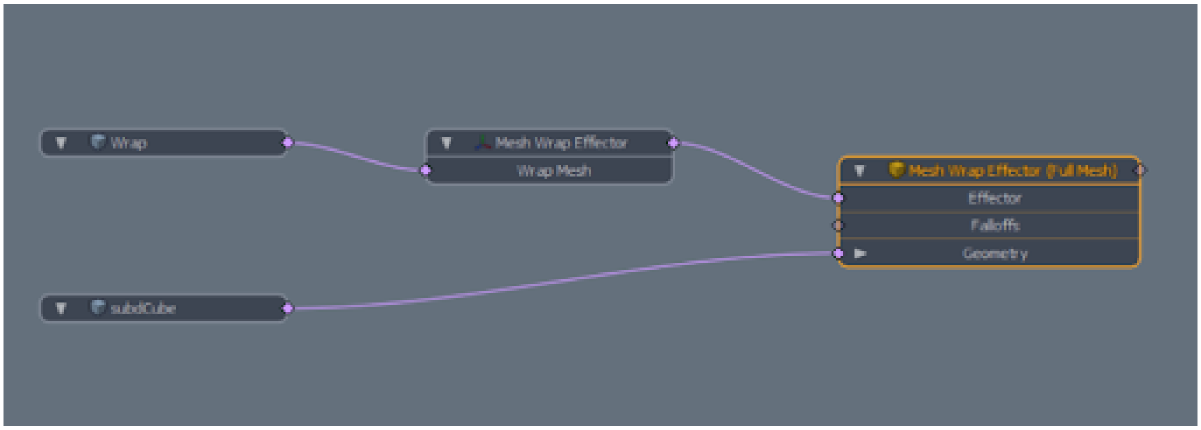
Wrap Effector

The Wrap Effector is generally best used in conjunction with the schematic viewport. To set up the Wrap Effector:

1. Create a mesh and then duplicate it. One of these meshes will be our high poly mesh, while the other will act as our wrap mesh.
2. Subdivide one of the meshes, but leave the other at its original poly count.
3. Bring both meshes into the schematic viewport by selecting both in the Items List and then dragging and dropping them into the Schematic viewport.

4. Within the schematic viewport, create a **Mesh Wrap Effector** node. This can be done by clicking **Add...** and then navigating to **Deformers/Effectors>Mesh Wrap Effector**, or by typing the name of the node into the search bar.

   Once created, a **Mesh Wrap Effector** locator node is added to the schematic viewport. This will be needed in later steps, so leave it as is.
5. From the **Mesh Wrap Effector**, draw a connection between your high poly mesh and the geometry input in the **Mesh Wrap Effector** node.
6. Draw a connection between your lowpoly mesh and the **Wrap Mesh** input in the **Mesh Wrap Effector** locator node.



Example of a **Mesh Wrap Effector** node setup in the Schematic Viewport.

Once set up, you can select points on your low poly or wrap mesh and move them as needed. As points on the wrap mesh are deformed, the high poly mesh deforms to match the low poly mesh. As long as the number of points is not changed, the wrap mesh can be continually edited and deformed. If the wrap mesh is disconnected, the deformations present on the target mesh are reset until a new wrap mesh is connected.

There are also a number of parameters that can be adjusted within the Item Properties of the **Mesh Wrap Effector** node.

Wrap Effector Controls	
Wrap Mesh	Allows you to set the wrap mesh used to deform the target mesh
Strength	Determines how strong the deform effect is.
Sharpness	Determines how strongly the target mesh adheres to the wrap mesh
Set Bind Positions	Freezes the position of the points and polygons in the chosen wrap mesh as the default position. Any additional deformations will be in relation to the frozen points and positions of the wrap mesh.

Wrap Effector Controls	
Create Wrap Mesh	Allows you to generate a latticed mesh around the deforming geometry
Mode	Allows you to set whether influence regions are created for every vertex in the wrap mesh, or for every polygon
Shear Compensate	When enabled, any distortions applied to the wrap mesh attempt to be accounted for in the target mesh. **Note:** Ddepending on the complexity of the wrap mesh, having this enabled may cause the tool to run slower or cause unwanted deformations.
Use Wrap Transforms	When enabled, Item Level transforms on the wrap mesh will deform the target mesh. If disabled, Item Level transforms will be ignored.
Use All Points	When enabled, the range of influence needed will automatically be set to the furthest distance needed to encompass every point. When enabled, all other settings related to setting influence will be disabled.
Draw Ranges	Controls if the influence regions are drawn or not
Radius	If Use All Points is disabled, this acts as the base radius for the influence region of each vertex or polygon on the wrap mesh
Remap Weights	By default, the weight of all points on the wrap mesh are normalized and influence is equal actross all points. However, these weights can be remapped and further manipulated with a gradient

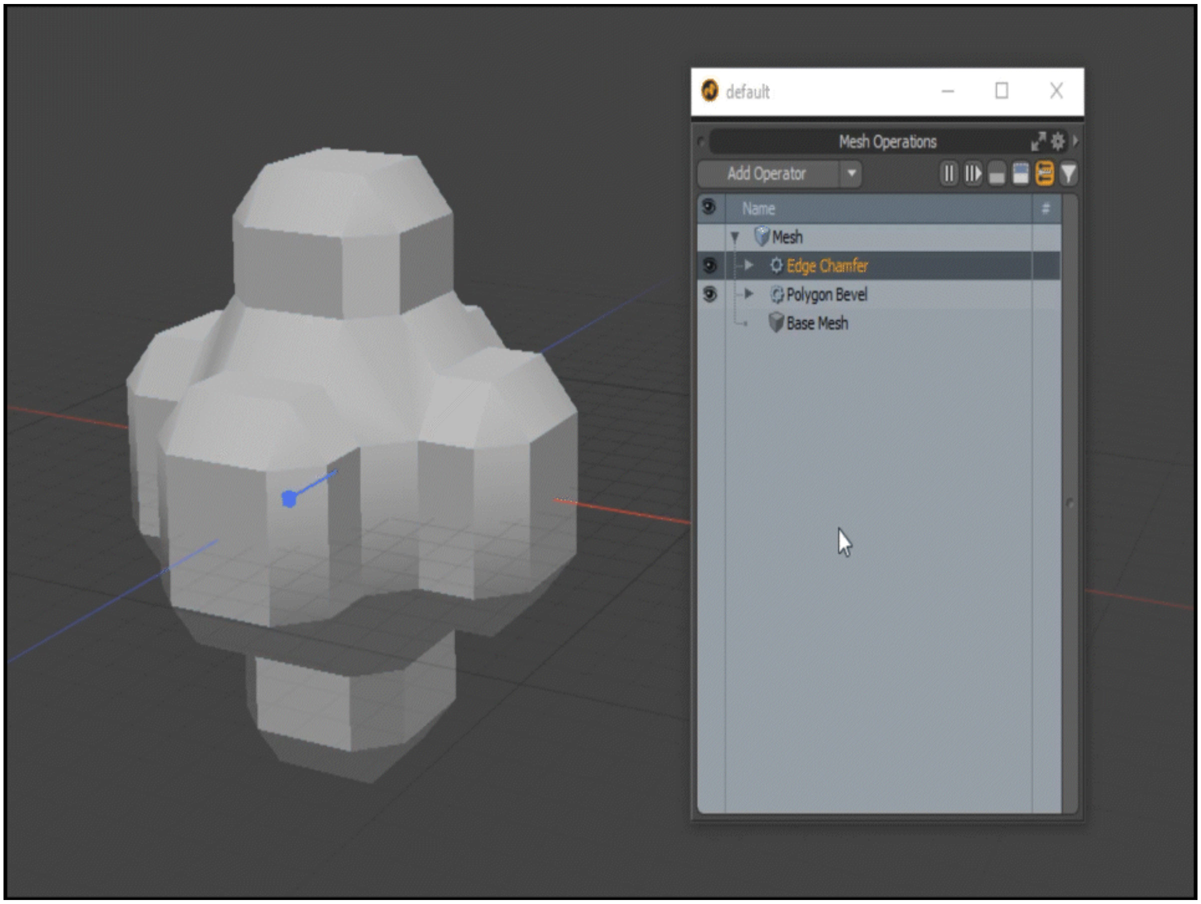
MeshOp Stack

The procedural system in Modo is built around an order of operations that evaluates from bottom to top in a list. When working in the **Schematic Viewport**, it can be difficult to establish that order of operations an manipulate operations to the fullest extent. The **MeshOp Stack** node makes it easy to visualize the order in which operations deform your mesh and allows you to fully manipulate multiple deformers from the **Schematic Viewport** without needing to jump back and forth between the **Schematic Viewport** and the **Mesh operation Viewport**.

Order of Operations

In Modo, the way in which operations are ordered within a stack or list can greatly impact how your mesh is deformed. Modo evaluates operations in a bottom-up way: the last operation in your list will be the first one to be evaluated, while the first one in your list will be the last to be evaluated.

Changing the order of your operations impacts how your mesh is evaluated. For example, if a mesh has a bevel applied followed by a chamfer, the bevel is evaluated first and the chamfer will be applied on top of the bevel. If the order of these two operations are changed so that the chamfer is applied before the bevel, the mesh item being edited will produce a different result.



Changing the order of your operations impacts how your mesh is edited.

Being aware of how the order of your operations impacts how your mesh is evaluated is important when procedural modeling

Stacks in the Schematic Viewport

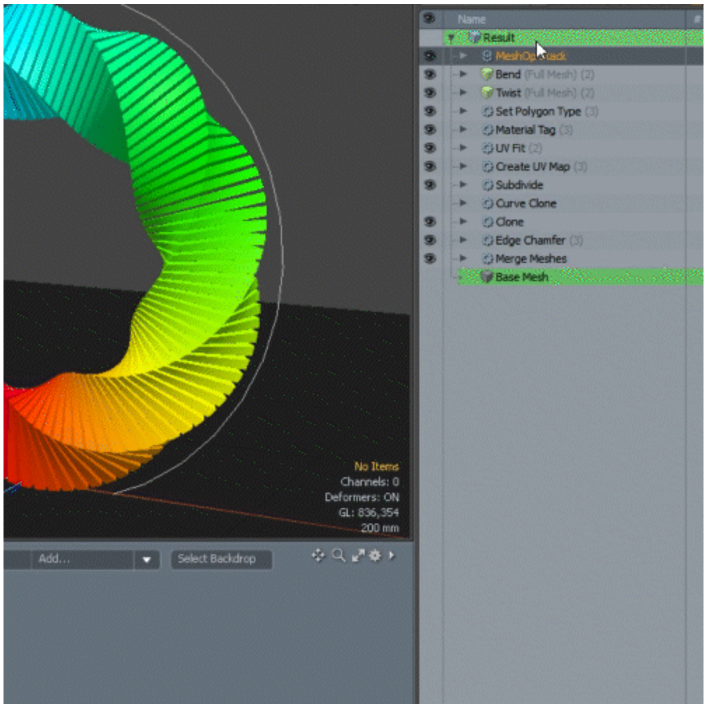
Because Modo evaluates operations starting at the bottom and then ending at the top of a list, using the **Schematic Viewport** with this node is recommended as it allows you to visualize the order operations of you procedural model.

There are two ways to create a **MeshOp Stack** node in the **Schematic Viewport**:

- In the **Schematic Viewport**, click **Add** and create a **MeshOp Stack**Node

You can find this operator either by typing **MeshOp Stack** in the **Search All**bar, or under **Deformers > Grouping** in the **Add to Schematic**window.

- Create a **MeshOp Stack** operation in the **Mesh Operations Viewport** and then click and drag the operation into the **Schematic Viewport** to make it visible.

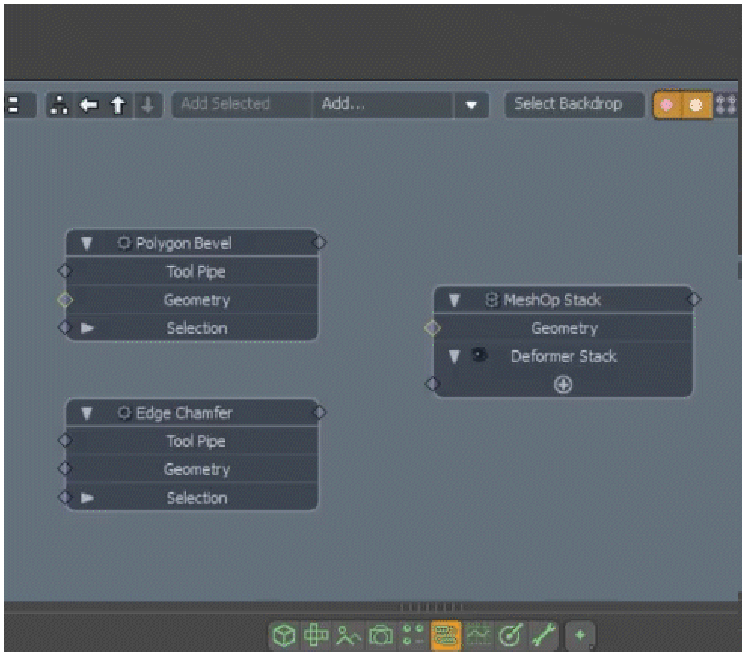

Name
Result
MeshOp Stack
Bend (Full Mesh) (2)
Twist (Full Mesh) (2)
Set Polygon Type (3)
Material Tag (3)
UV Fit (2)
Create UV Map (3)
Subdivide
Curve Clone
Clone
Edge Chamfer (3)
Merge Meshes
Base Mesh
No Items
Channels: 0
Deformers: ON
GL: 836,354
200 mm
Add...
Select Backdrop

Any operations dragged into the **Schematic Viewport** will remain present in your list of operations, while any new operations added to the **Schematic Viewport**will also be listed in the **Mesh Operations Viewport**

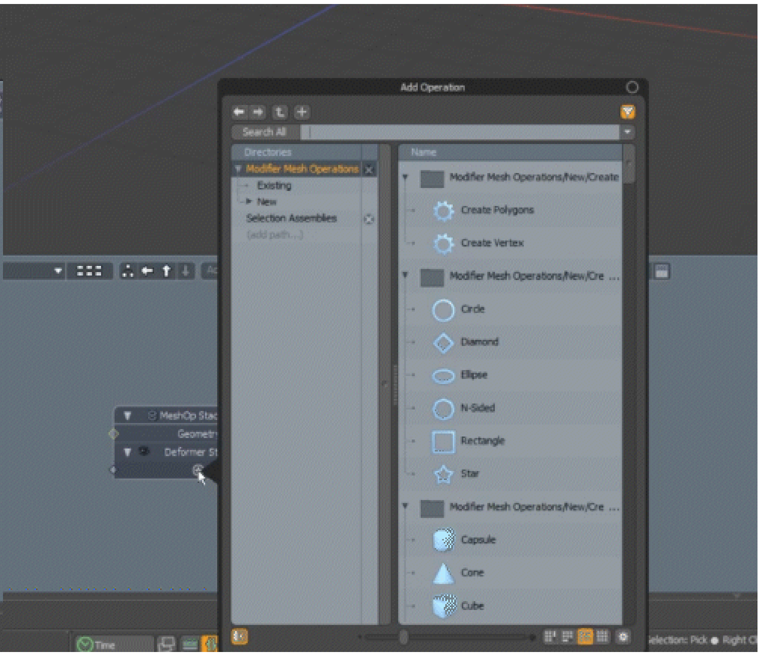
Within the node, there are two inputs: **Geometry** and **Deformer Stack**.

By default, the **Geometry** input is linked to the **Mesh item** that the **MeshOp Stack** was created within. Additional geometry can be added to this input by dragging mesh items into the **Schematic Viewport** and dragging a link into the **Geometry** input.

The **Deformer Stack** input holds all the deformers and operations within your procedural model. Operations and deformers are added to the stack by dragging a link from the operation or deformer into the **Deformer Stack** input, or by clicking the plus button found under the deformer tab of the MeshOp stack node. When pressed, the **Add Operation** window opens and from there you are able to add MeshOps to your stack.


Add Selected
Add...
Select Backdrop
Polygon Bevel
Tool Pipe
Geometry
Selection
Edge Chamfer
Tool Pipe
Geometry
Selection
MeshOp Stack
Geometry
Deformer Stack

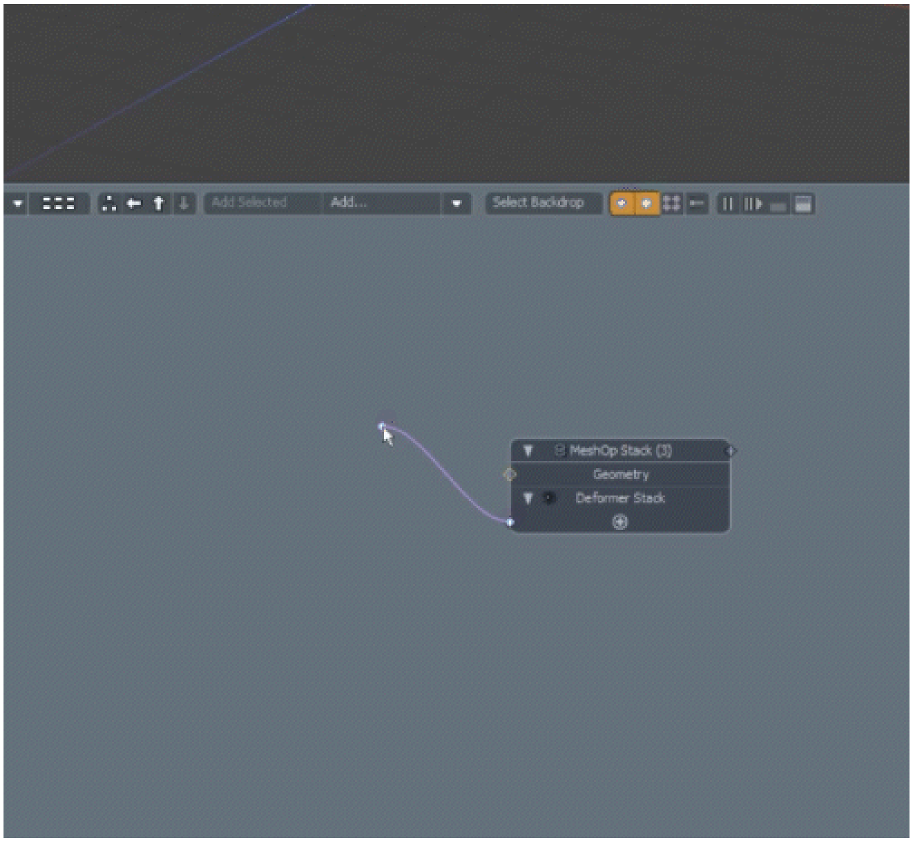
Drawing links into the MeshOp Stack


Add Operation
Search All
Directories
Modifier Mesh Operations
Existing
New
Selection Assemblies
(add path...)
Name
Modifier Mesh Operations/New/Create
Create Polygons
Create Vertex
Modifier Mesh Operations/New/Cre ...
Circle
Diamond
Ellipse
N-Sided
Rectangle
Star
Modifier Mesh Operations/New/Cre ...
Capsule
Cone
Cube
MeshOp Stac
Geometry
Deformer St

Adding deformers within the MeshOp Stack

Whenever a new link is dragged into the **MeshOp Stack**, an orange line is drawn to indicate the insertion location.

Operators can also be added by dragging from the input connector and releasing on an empty area of the Schematic. The Preset Browser will open and after adding an operator it will be positioned where the Preset Browser was opened. Links can also be dragged out from existing operators in the Stack and the chosen operator will be inserted above the existing operator.

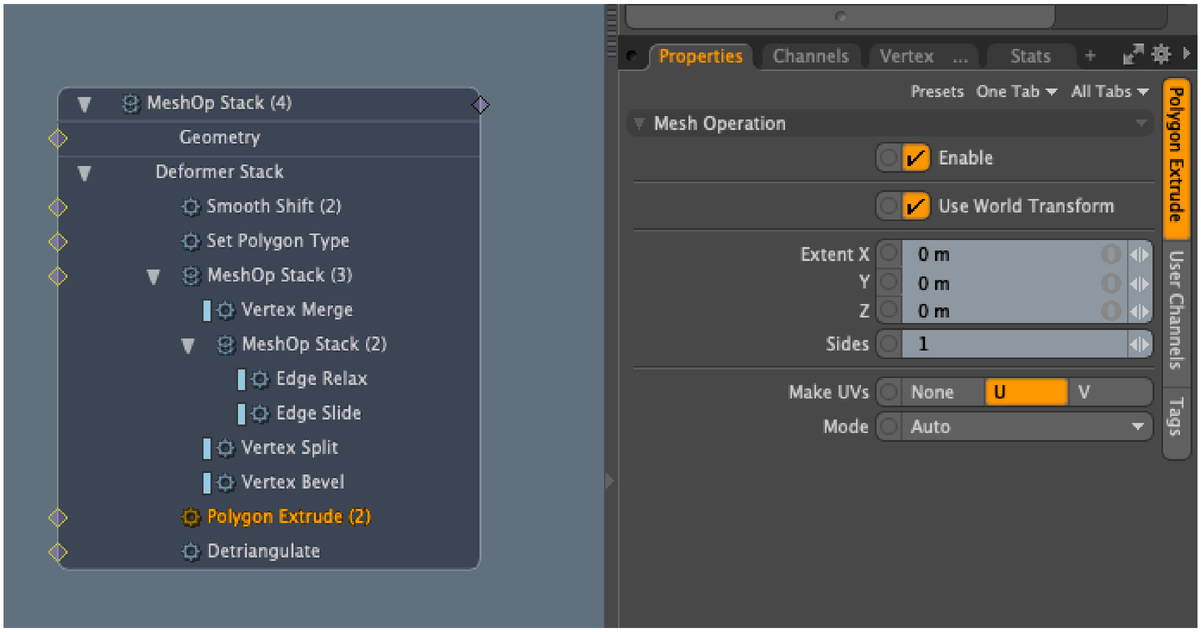


Drawing a link to empty space to assign a new operation

The **MeshOp Stack** node evaluates operates in the same way that the **Mesh operations Viewport** does, so it's important to pay attention to the order in which you create links between your deformers and the **Deformer Stack**.

Operations can be reordered in the stack by left-clicking and dragging the operation up or down within the stack. As above, an orange line will appear to indicate the insertion location of where the operation will be placed

MeshOps, Deformers and nested**MeshOp Stacks** can be selected in the **MeshOp Stack** node with the left mouse button, allowing for properties to be edited without their node being present in the **Schematic Viewport**. This allows you to have full control over you deformers and MeshOps within the **Schematic Viewport** wihtout having multiple nodes present in the viewport, or having to select deformers or operations from the **Mesh Operations Viewport**.


MeshOp Stack (4)
Geometry
Deformer Stack
Smooth Shift (2)
Set Polygon Type
MeshOp Stack (3)
Vertex Merge
MeshOp Stack (2)
Edge Relax
Edge Slide
Vertex Split
Vertex Bevel
Polygon Extrude (2)
Detriangulate
Properties
Channels
Vertex
...
Stats
+
Presets
One Tab
All Tabs
Mesh Operation
Polygon Extrude
User Channels
Tags

☑ Enable

☑ Use World Transform

Extent X 0 m

Y 0 m

Z 0 m

Sides 1

Make UVs None U V

Mode Auto

Multiple operations can be added to the properties window at once by holding down **Shift** and clicking on the operations you want to change, while holding down the **Ctrl/Cmd** key and clicking on an operation removes it from the properties window.

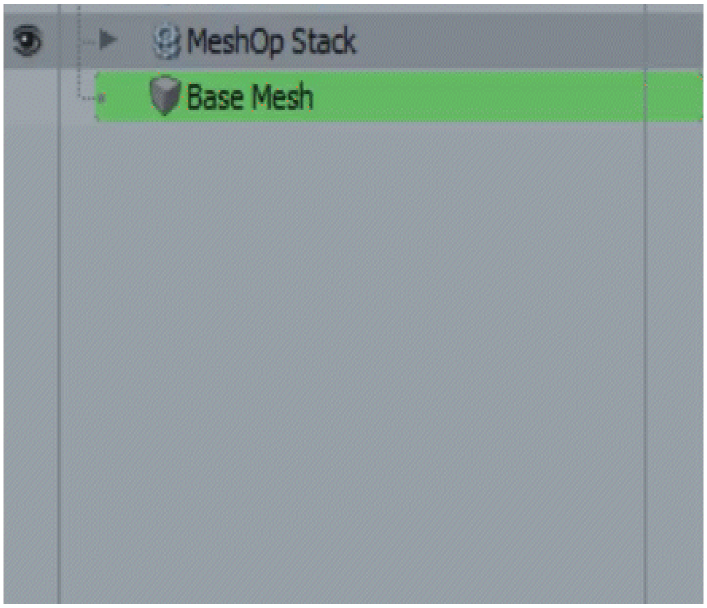
Similarly to the MeshOp viewport, deformers within the stack can be hidden or unhidden by clicking the eye icon next to the name of the deformer within the stack.

Stacks in the Mesh Operations Viewport

MeshOp Stacks can also be used in the **Mesh Operations Viewport**, but we recommend using the **Schematic Viewport** with stacks, as the **Schematic Viewport** allows for a finer level of control without the need to jump between operations and deformers to adjust their parameters.

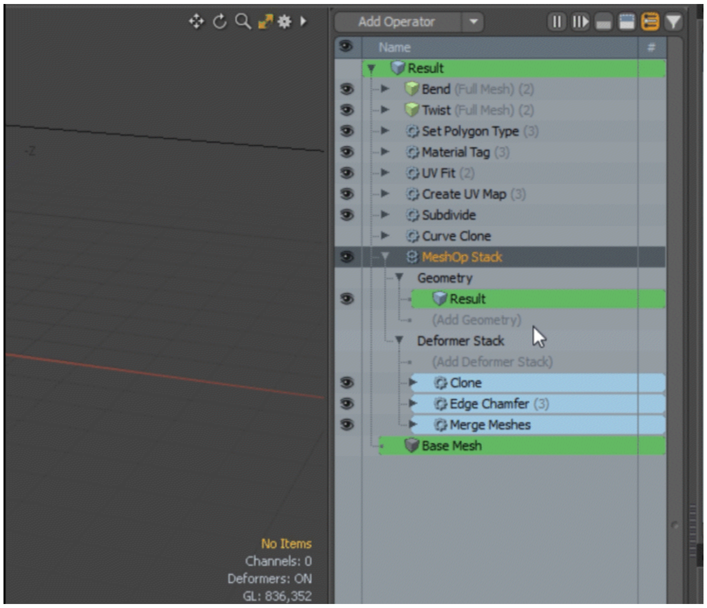
To create a **MeshOp Stack** in the **Mesh operations Viewport**, click **Add Operator** and create a **MeshOp Stack**. The **MeshOp Stack** operation can be found under the **Deformers > Grouping** or by typing **MeshOp Stack** in the **Search All** bar.

The contents of the **MeshOp Stack** can be viewed in the **Mesh Operations Viewport** by clicking the arrow icon next to the mesh item. Once expanded, both the **Geometry** and the **Deformer Stack** channels are displayed.


MeshOp Stack
Base Mesh

Operations that exist within a stack are highlight by default in light blue. This color can be changed in the **Preferences** window.

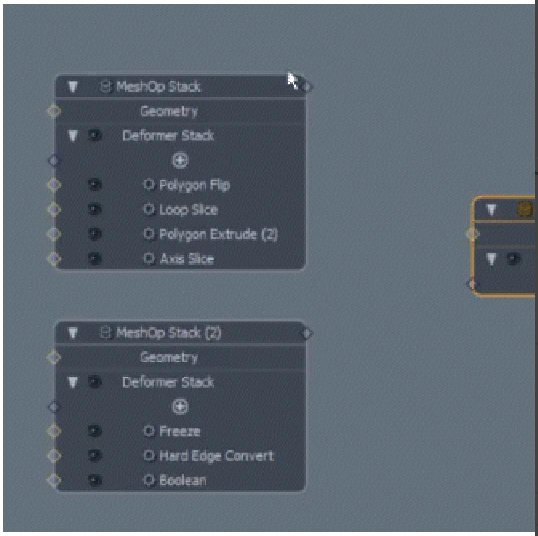
Operations and **Deformers** can be added to stacks within the **Mesh operations Viewport** by clicking **Add Deformer Stack** under the expanded **Deformer Stack** channel, and then selecting an existing operation or deformer from the **Add Operation Viewport**.

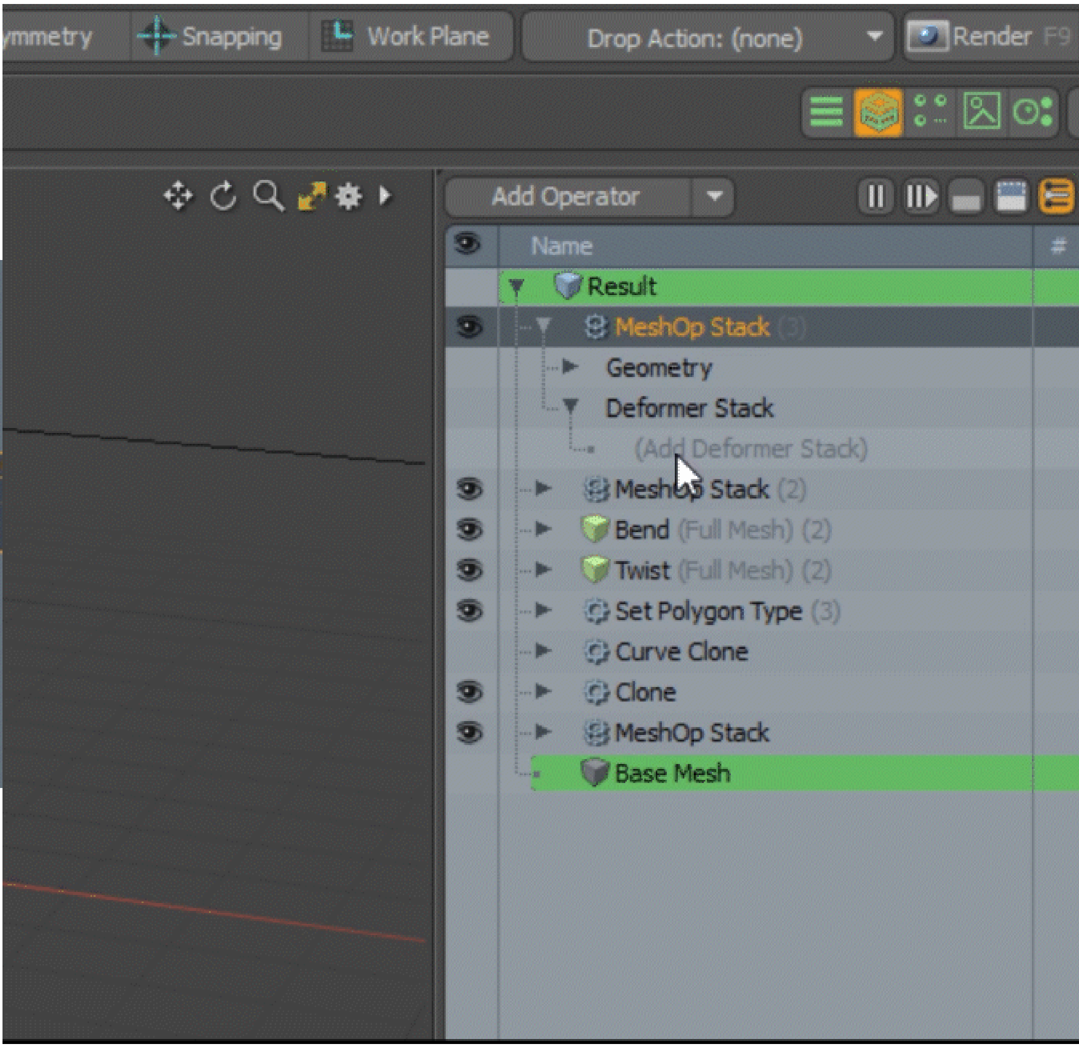

Add Operator
Name
Result
Bend (Full Mesh) (2)
Twist (Full Mesh) (2)
Set Polygon Type (3)
Material Tag (3)
UV Fit (2)
Create UV Map (3)
Subdivide
Curve Clone
MeshOp Stack
Geometry
Result
(Add Geometry)
Deformer Stack
(Add Deformer Stack)
Clone
Edge Chamfer (3)
Merge Meshes
Base Mesh
No Items
Channels: 0
Deformers: ON
GL: 836,352

Operations and deformers can also be reordered within the stack by left-clicking and dragging the operation up or down within the operations list.

Nesting Stacks

MeshOp Stacks can also be nested inside of other stacks. This can be done by drawing a link one stack into the **Deformer Stack** input og another stack within the **Schematic Viewport**, or by adding the stack from the **Add Operation** window by clicking **Add Defomer Stack** within the **Mesh Operation Viewport**.


MeshOp Stack
Geometry
Deformer Stack
Polygon Flip
Loop Slice
Polygon Extrude (2)
Axis Slice
MeshOp Stack (2)
Geometry
Deformer Stack
Freeze
Hard Edge Convert
Boolean


ymmetry
Snapping
Work Plane
Drop Action: (none)
Render F9
Add Operator
Name
#
Result
MeshOp Stack (3)
Geometry
Deformer Stack
(Add Deformer Stack)
MeshOp Stack (2)
Bend (Full Mesh) (2)
Twist (Full Mesh) (2)
Set Polygon Type (3)
Curve Clone
Clone
MeshOp Stack
Base Mesh

Nesting a stack within the **Schematic Viewport**.

Nesting a stack within the **Mesh operations Viewport**

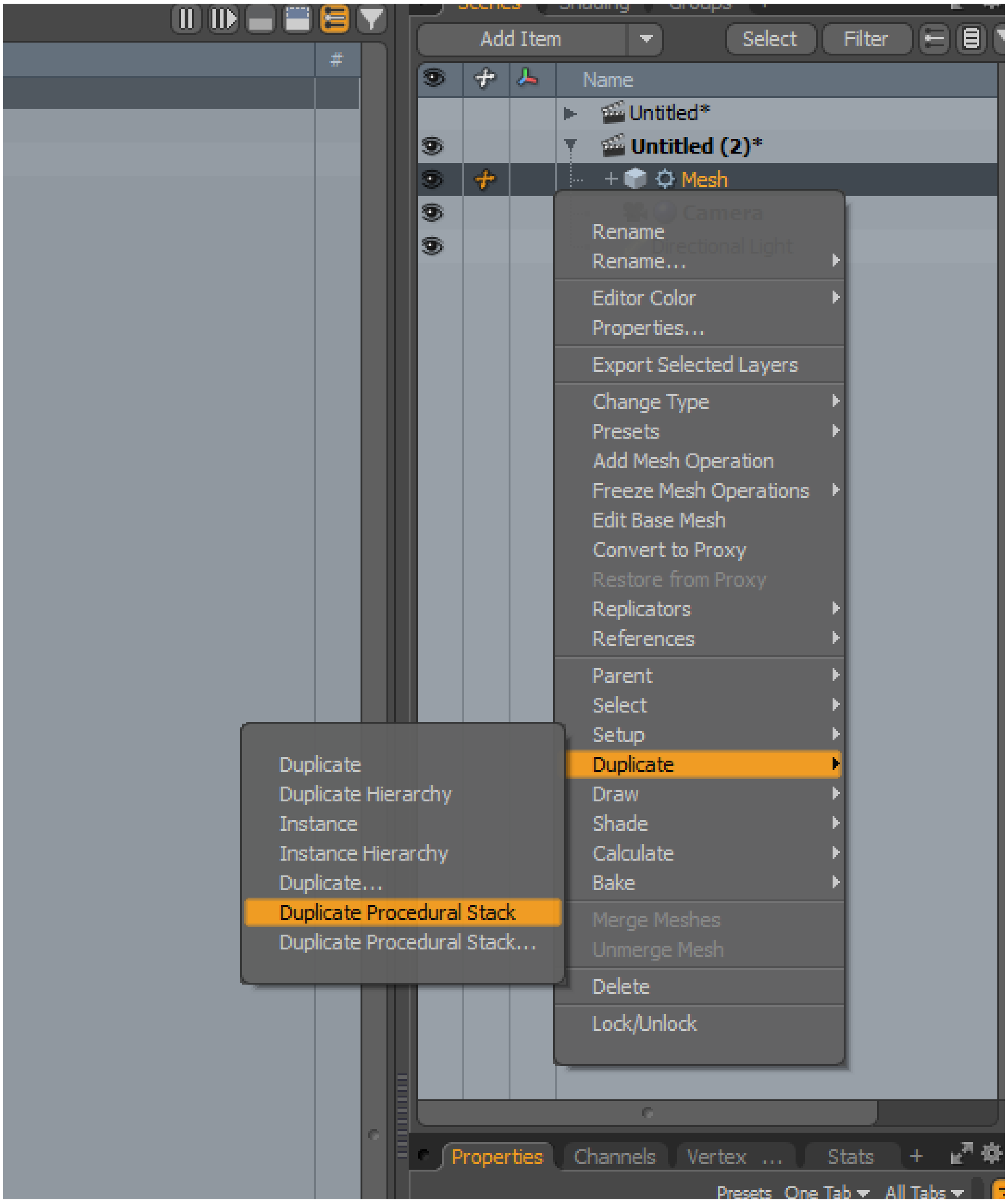
Using Stack Aliases

MeshOp Stacks can also be used with **Aliases**. This allows you to collapse a stack with no geometry inputs into an **Alias Item** so that a series of operations can be easily reused and controlled on multiple meshes.

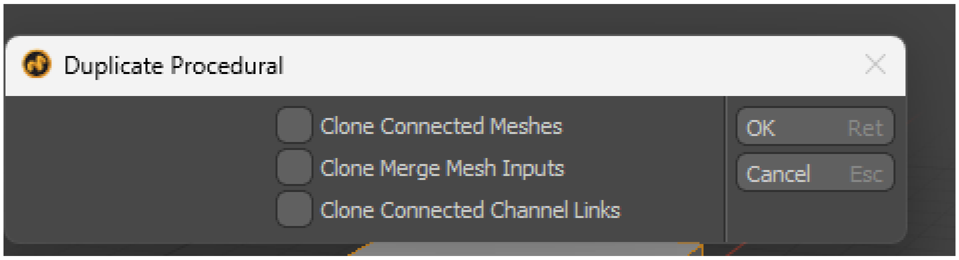
Duplicating Procedural Meshes

The **Duplicate Procedural Stack** command duplicates meshes containing mesh operations, or individual mesh operations. The command clones the selected item, and duplicates the connected items, so tool pipes or selection nodes contained within mesh operations are also duplicated.

3. Select a procedural mesh or procedural operation in the item list or mesh operations view
4. Press **ctrl**+**d**, or right-click on the mesh in the item list or operation in the mesh operations view
5. Select **Duplicate Procedural Stack**


Add Item
Select
Filter
Name
Untitled*
Untitled (2)*
Mesh
Camera
Directional Light
Rename
Rename...
Editor Color
Properties...
Export Selected Layers
Change Type
Presets
Add Mesh Operation
Freeze Mesh Operations
Edit Base Mesh
Convert to Proxy
Restore from Proxy
Replicators
References
Parent
Select
Setup
Duplicate
Draw
Shade
Calculate
Bake
Merge Meshes
Unmerge Mesh
Delete
Lock/Unlock
Duplicate
Duplicate Hierarchy
Instance
Instance Hierarchy
Duplicate...
Duplicate Procedural Stack
Duplicate Procedural Stack...
Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs

The Duplicate Procedural options window opens.



Duplicate Procedural

☐ Clone Connected Meshes

☐ Clone Merge Mesh Inputs

☐ Clone Connected Channel Links

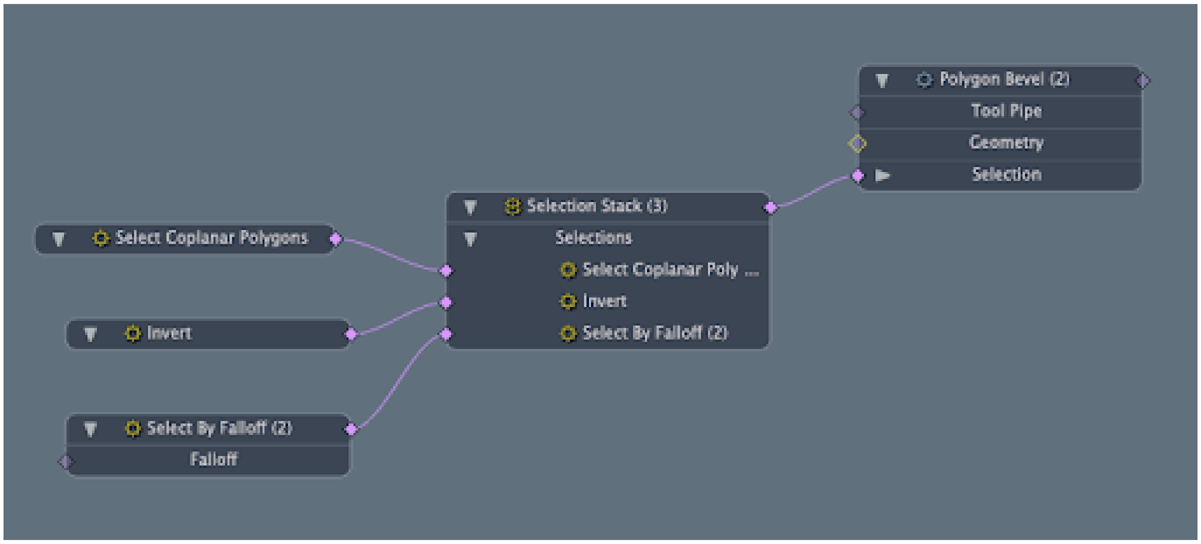
OK Ret
Cancel Esc

The following options are off by default:

- **Clone Connected Meshes** - If the item is linked to any meshes, this option clones those meshes.
- **Clone Merge Mesh Inputs** - Clones the input sources on the merge meshes node.
- **Clone Connected Channel Links** - Scans the channel link graph and clones those connected items. Use this option if you want to clone, for example, math nodes driving a procedural. Leave this option unchecked if your channel links are from a driving item and you want to control both rigs.

Organize Your Schematic Viewport using the Selection Stack

Similarly to the MeshOp stack node, the Selection Stack allows you to collate multiple selection operation nodes into one stack, making organization in the Schematic Viewport easier.


Polygon Bevel (2)
Tool Pipe
Geometry
Selection
Selection Stack (3)
Selections
Select Coplanar Poly ...
Invert
Select By Falloff (2)
Select Coplanar Polygons
Invert
Select By Falloff (2)
Falloff

As with the MeshOp stack node, evaluations are performed in a bottom to top order.

Note: For more information on the Order of Operations in Modo, see Order of Operations

Working With Selection Stacks in the Schematic Viewport

Selection stacks can be added to the schematic viewport in two ways:

1. In the **Schematic Viewport**, click **Add** and create a **Selection Stack** node. You can find this operator either by typing **Selection Stack** in the Search All bar or under **MeshOperations > Selection**

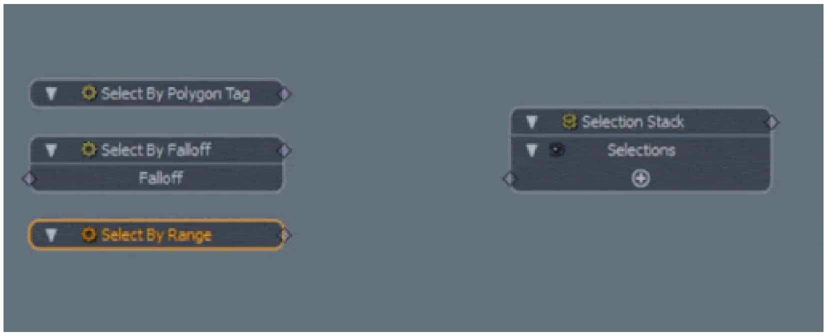
   or

2. Create a **Selection Stack** operation in the **Mesh Operations** viewport and then click and drag the operation into the **Schematic Viewport** to make it visible.

While in the schematic viewport, there are two ways you can add operations to the stack by:

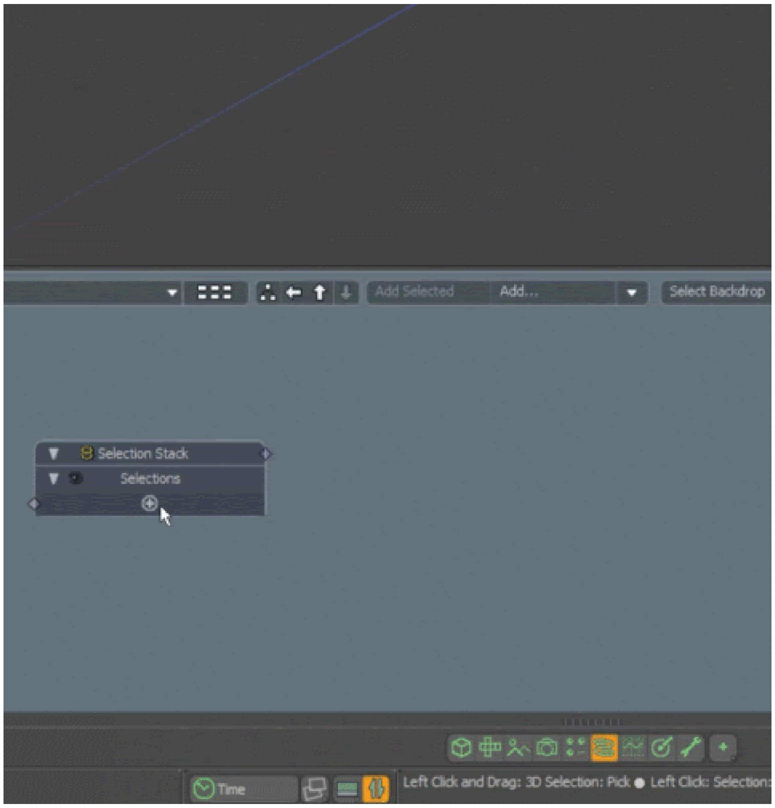
1. Dragging a connection between the selection operation and the Selection Stack.

   Whenever a link is dragged into the stack, an orange line is drawn to indicate the insertion location. The insertion line appears to make it visually clear where your operation exists within the order of operations


Select By Polygon Tag
Select By Falloff
Falloff
Select By Range
Selection Stack
Selections

   Or

2. Pressing the Add Operation button found under the Selections tab within the Selection Stack node.

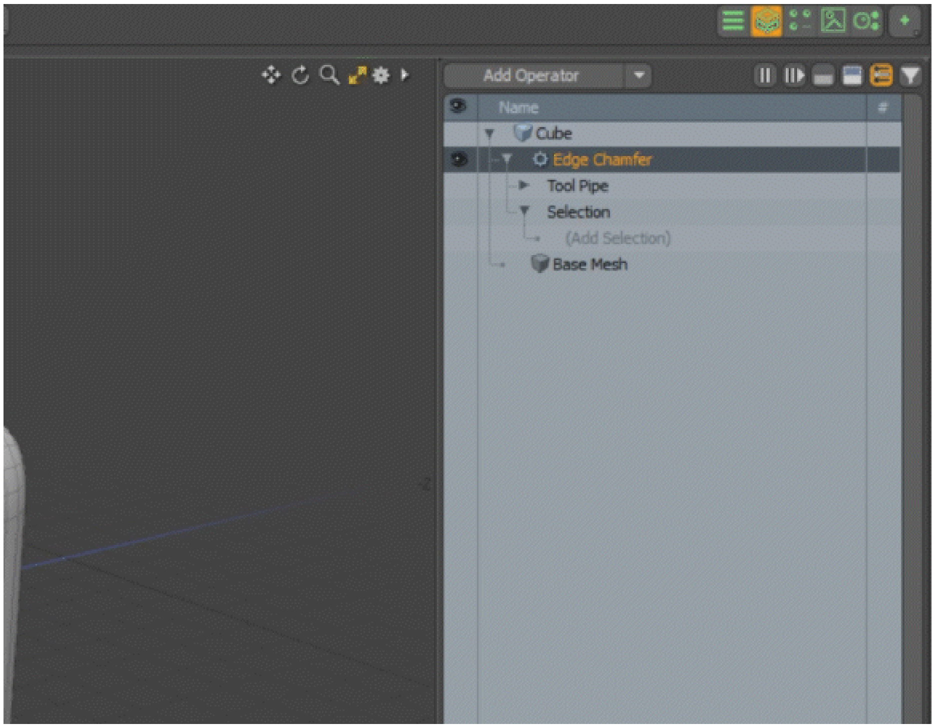

Add Selected
Add...
Select Backdrop
Selection Stack
Selections
Time
Left Click and Drag: 3D Selection: Pick ● Left Click: Selection:

Operations can be reordered in the stack by left-clicking and dragging the operation up or down within the stack. As above, an orange line indicates the insertion location of where the operation will be placed.

Working with Selection Stacks in the MeshOperations Viewport

Selection Stacks can also be used in the **Mesh Operations** Viewport, but we recommend using the **Schematic Viewport** with stacks, as the **Schematic Viewport** allows for a finer level of control without the need to jump between operations and deformers to adjust their parameters.

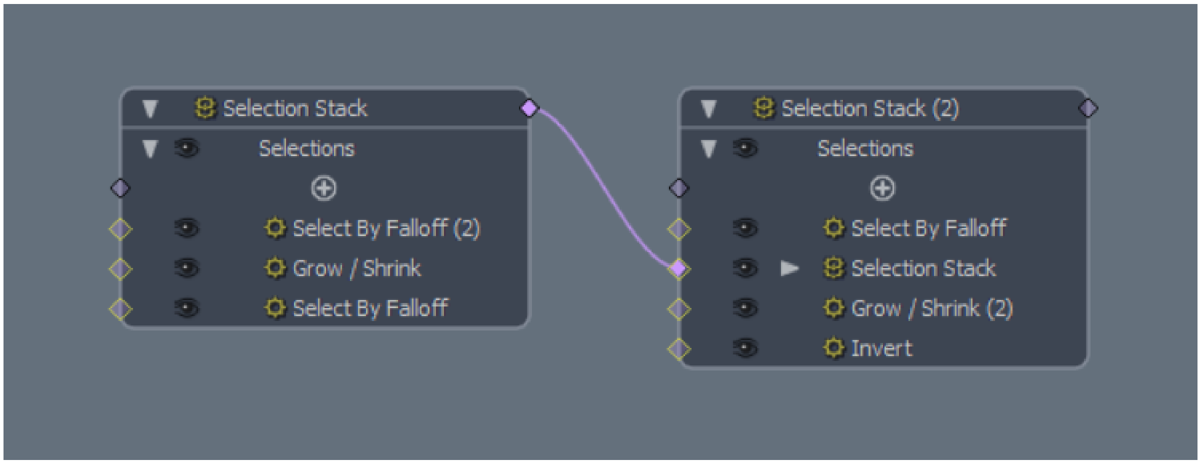
Selection Stacks can be added to the **Mesh Operations** Viewport by expanding the operation you want to add a selection stack to, and then pressing **Add Selection** in the expanded selection list. Once added, you can continue to use the **Add Selection** option to add additional selection operations to the stack.


Add Operator
Name
Cube
Edge Chamfer
Tool Pipe
Selection
(Add Selection)
Base Mesh

Selection operations that exist outside of the stack can be added by dragging and dropping them into the stack. Selection operations within the stack can be reordered in the same way.

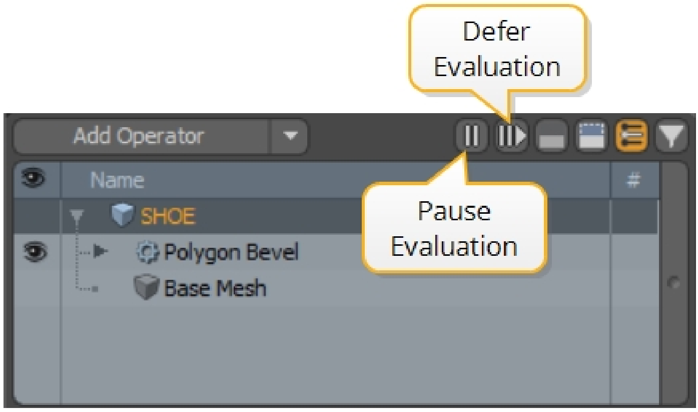
Nesting Selection Stacks Within Other Selection Stacks

Selection Stacks can also be nested inside of other selection stacks, similarly to MeshOp stacks. This can be done by drawing a connection from one stack to another in the schematic viewport, or by clicking **Add Selection** within the **Mesh Operations** Viewport while the selection list of your MeshOp is expanded and selecting the stack you want to nest.

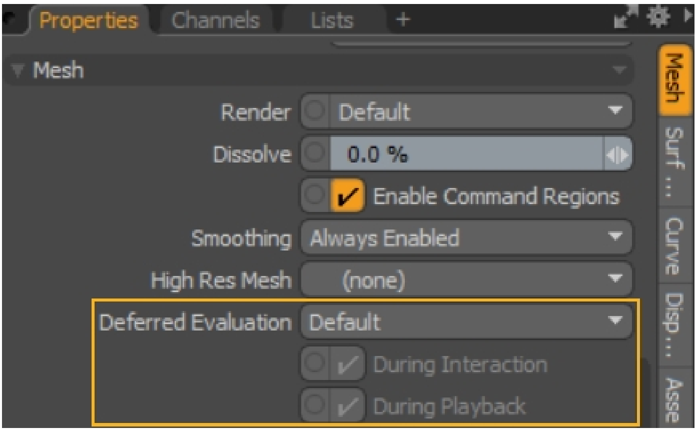

Selection Stack
Selections
Select By Falloff (2)
Grow / Shrink
Select By Falloff
Selection Stack (2)
Selections
Select By Falloff
Selection Stack
Grow / Shrink (2)
Invert

Deferring and Pausing Evaluation

The evaluation of a mesh's deformers and MeshOps can be deferred or paused to improve UI interaction. This can be controlled globally, or on a per-mesh item basis. By default, all meshes behave according to the global settings available in the MeshOps list, but any non-default settings in the mesh item's properties override the global options.



Global Defer and Pause Mode



Render: Default
Dissolve: 0.0 %
☑ Enable Command Regions
Smoothing: Always Enabled
High Res Mesh: (none)
Deferred Evaluation: Default
☑ During Interaction
☑ During Playback

Custom Defer and Pause Mode

Deferred Evaluation

Deferring evaluation of a mesh stack means delaying mesh updates until the mouse button is released. Deferring evaluation enhances UI interaction, speeding up the response of actions in the viewport, particularly on meshes with a high polygon count. For example, moving a Polygon Bevel with deferred evaluation active does not apply the adjustment until you release the mouse button.

Note: When ghosting is active, the effects of the currently selected MeshOp or deformer are still displayed interactively, but no deformers or MeshOps are computed until the mouse button is released. See Ghosted View for more information.

Paused Evaluation

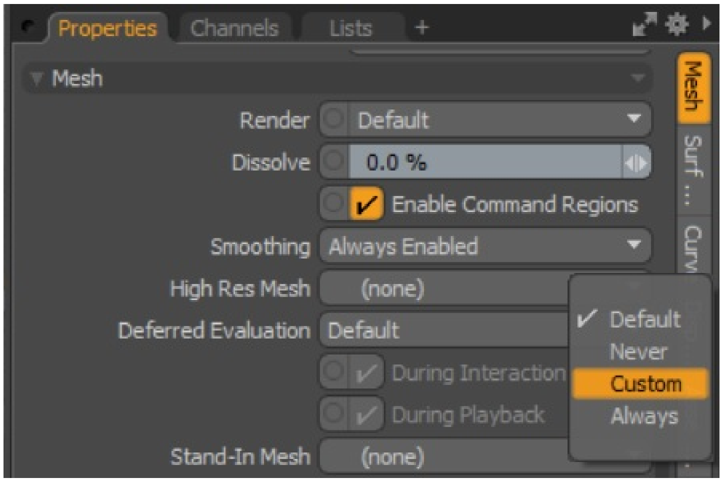
Pausing evaluation temporarily disables the effect of deformers and MeshOps so that no updates are made to your meshes. Any changes that require the deformers or MeshOps to recompute are ignored until paused mode is turned off.

Custom Mesh Item Controls

The **Pause** and **Defer** buttons in the MeshOps viewport are global controls, meaning all meshes in the scene are affected by them. Enabling **Pause** in the MeshOps viewport pauses the effect of all deformers and MeshOps on all meshes in the scene.

However, you may want the mesh you're working on to be interactive at all times, so Modo allows you to customize pause and defer on per-mesh basis in the **Mesh** properties.

1. Select the required mesh in the **MeshOps** viewport to display its **Properties**.
2. Select the **Deferred Evaluation** mode using the dropdown:



Mesh
Render Default
Dissolve 0.0 %
☑ Enable Command Regions
Smoothing Always Enabled
High Res Mesh (none)
Deferred Evaluation Default
☑ During Interaction
☑ During Playback
Stand-In Mesh (none)

- **Default** - the global controls determine whether evaluation of this mesh is deferred or paused.
- **Never** - evaluation of this mesh is never deferred or paused, even if the global options are enabled.
- **Custom** - defers evaluation of this mesh according to the state of the **During Interaction** and **During Playback** checkboxes.
- **Always** - evaluation of this mesh is deferred or paused until the **Deferred Evaluation** option is changed.

Stand In Meshes

A lower polycount stand in mesh can be used in place of a deferred mesh to improve interaction or playback performance. Specifying a stand in mesh draws the stand in mesh instead of the primary mesh, which is hidden when evaluation is deferred or paused.

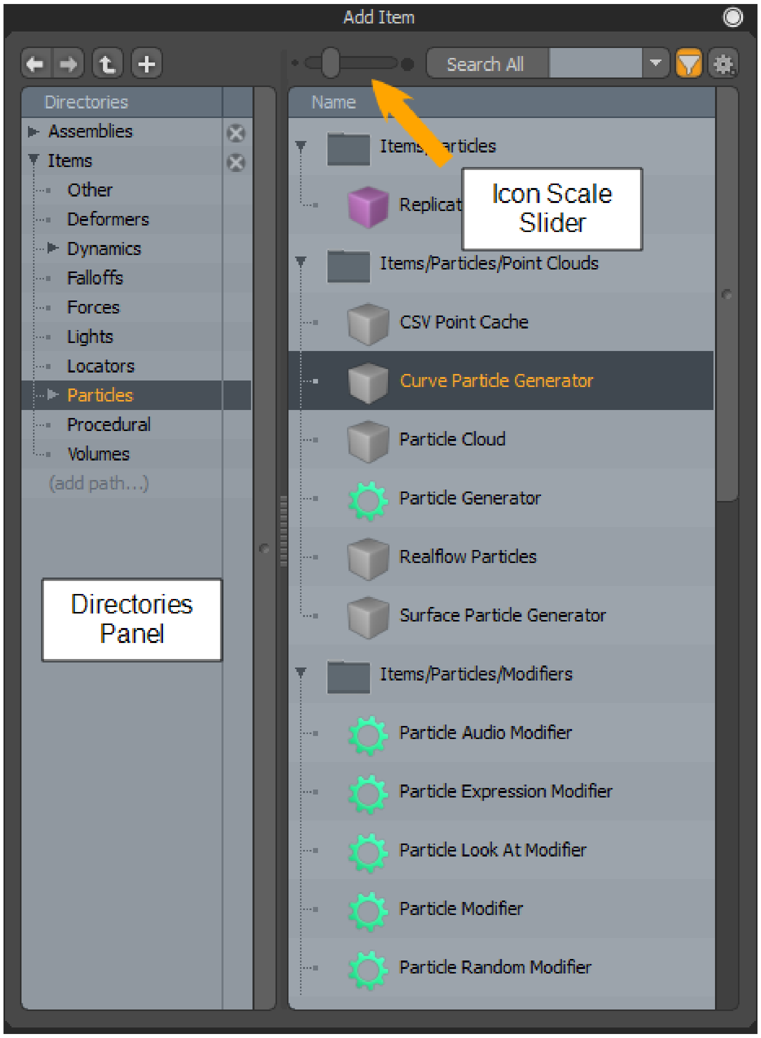
To specify a stand in mesh:

1. Import or create the stand in mesh in your scene. You can navigate to **File** > **Import** to select a mesh or create one in Modo using the **Create** tools.
2. Select the primary mesh in the **MeshOps** viewport to display its **Properties**.
3. Set the **Deferred Evaluation** dropdown to **Custom** and the **During Interaction** and **During Playback** checkboxes as required.
4. Click the **Stand-In Mesh** dropdown and select the required mesh from the list.

   The stand in mesh is now drawn in place of the primary mesh when defer or pause is enabled, improving UI interaction.

Procedural Preset Browser

Modo's procedural tools are listed in a dedicated Preset Browser that is opened by clicking the **Add Operator** button in the Mesh Operations List.


Add Item
Search All
Directories
Assemblies
Items
Other
Deformers
Dynamics
Falloffs
Forces
Lights
Locators
Particles
Procedural
Volumes
(add path...)
Directories Panel
Name
Items/Particles
Replicat
Icon Scale Slider
Items/Particles/Point Clouds
CSV Point Cache
Curve Particle Generator
Particle Cloud
Particle Generator
Realflow Particles
Surface Particle Generator
Items/Particles/Modifiers
Particle Audio Modifier
Particle Expression Modifier
Particle Look At Modifier
Particle Modifier
Particle Random Modifier

Tool Types

The Preset Browser is your palette for procedural modeling, and all tools that have a procedural equivalent can be found here. The tools are grouped by directory according to type. You can navigate the directories using the **Directories** panel on the left. You can use the scale slider to change the size of the icons. The category headers have folder icons and the tools have their individual icons.

Deformers

The Deformers allow you to manipulate the position of geometry procedurally. For example, you can twist the body of a mesh by moving the vertices using the Twist Effector. Simply add it to the **Mesh Operations** list above the mesh and edit the twist properties.

The list includes a simple **Locator** (see Locator Item) to transform a mesh. This means that you can translate, rotate, and scale procedurally, and revisit the transformation to make changes if required.

For more information on using Deformers, see Applying Deformers.

Procedural

Modo offers an extensive list of procedural operations to create meshes and edit their topology. Most of the commonly used modeling tools are listed here. Procedural operations include the following:

- **Mesh Operations and Tools** - Mesh operations are similar to deformers, however they can change mesh topology and not just modify the position of existing elements. Examples include Curve Fill, UV Transform, and Smooth. Many existing Modo modeling tools have been converted into procedural operations. Example tools include Polygon Bevel and Smooth Shift. Each tool has access to its own **Tool Pipe** to add falloffs and sub-tools, which modify the effect of the tool.
- **Sub-tools** - These are found under the **Tool Pipe** directory and can be associated with a tool to modify how it operates. For example, a Symmetry Tool can be combined with a transform operation to perform a symmetrical transform. The sub-tools act through the parent tool's **Tool Pipe**, which operates in a similar way to the Tool Pipe in direct modeling.

  The procedural modeling system can also evaluate other item types as sub-tools. For example, falloff items can be connected to a procedural tool, and evaluated exactly like a falloff tool. Combining multiple sub-tools together can create new and unique operations.

  To add a sub-tool, click **(Add Tool Pipe)** in the **Tool Pipe** branch of the tool in the **Mesh Operations** list.
- **Selection Operations** - Selection Operations provide a means to procedurally create persistent selections when working in the **Mesh Operations** list. As a procedural operation is evaluated, the selection operation is also evaluated, which provides the target elements for any mesh edits.

  The selection elements can be defined by selecting them in the 3D viewport. For example, you could select a number of polygons in **Polygon** mode and add them to a **Select by Index**. You can also select elements procedurally based on certain criteria. For example, you could automatically select all edges with an angle greater than 45°. Any edits to the angles on the base mesh will then update the selection.

Note: For more information on each of these types, see Procedural Modeling with MeshOps.

Assemblies

An assembly is a ready-made collection of items that help when building scenes. They avoid having to reconstruct common scene elements that consist of a number of items. For example, you could have an assembly for a specific lighting setup or particle emitter.

Article: In Modo 11, Adding an Assembly preset doesn't retain the order of mesh operations in the Mesh Operations list. See How to retain the order of a procedural layer stack when creating an Assembly preset in Modo 11 to learn how to avoid this.

Finding Tools

Instead of navigating the tool directories, you can very quickly locate a tool if you know part of its name. Simply activate the find functionality by clicking the **F** button at the top of the browser. You can then type a search string into the box at the top of the browser.

The search mode button can be seen on the left of the search field. To search the full directory hierarchy, set the search mode button to **Search All**. Alternatively, to search only the selected directory, set the button to **Search View**.

Note: If you prefer to rely solely on the find functionality, you can close the directory list by dragging the panel divider over to the left.

Adding and Editing Tools

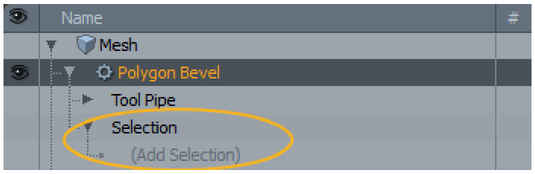
In the Preset Browser, double-click the name of an item to add it to the **Mesh Operations** list. Once added, you can highlight the item in the list and the properties appear in the **Item Properties** panel, which is on the lower-right of the UI by default. Edit the properties to change the item, and cascade subsequent changes upwards through the list.

Procedural Selection

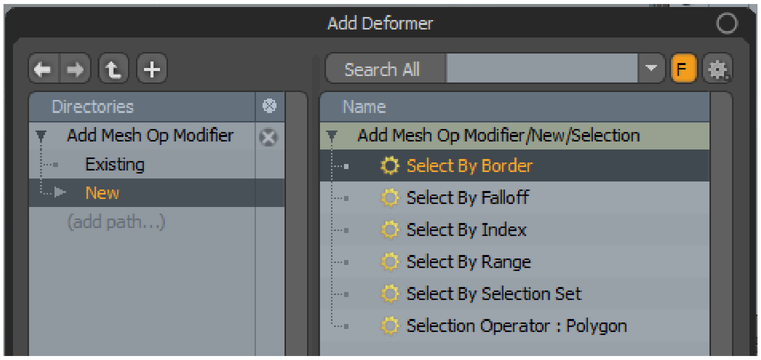
In procedural modeling, you can create selections through the **Mesh Operations** list. In the **Model** layout, this is located in the lower part of the **Mesh Ops** tab, on the right side of the interface.

First, add a mesh operation that requires a selection, for example, a Polygon Bevel.

1. You can click the small arrow in front of the operation in the list to reveal its **Selection** input:


Name
#
Mesh
Polygon Bevel
Tool Pipe
Selection
(Add Selection)

2. Under **New**, double-click the selection operation you want to add.


Add Deformer
Search All
F
Directories
Add Mesh Op Modifier
Existing
New
(add path...)
Name
Add Mesh Op Modifier/New/Selection
Select By Border
Select By Falloff
Select By Index
Select By Range
Select By Selection Set
Selection Operator : Polygon

The selection operation is added to the **Mesh Operations** list.

Note: The selection type should be compatible with the mesh operation. For example, if you are trying to add an edge selection to a Polygon Bevel operation, due to incompatibility, Modo doesn't create a selection type node in the **Mesh Operations** list.

Defining Selection Sets

Modo offers a number of selection set operations. The following options are available:

- **Assign Selection Set** - Use this mesh operation to define selection sets procedurally. It adds mesh elements to a named selection set. This selection set is then stored on the mesh and maintained throughout subsequent mesh edits.

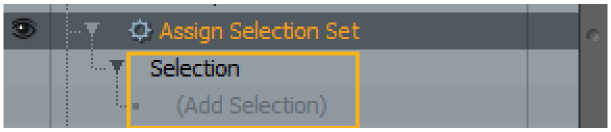
- **Select By Selection Set** - Use this mesh operation to specify which elements should be edited by a subsequent procedural mesh edit.

Assigning a Selection Set

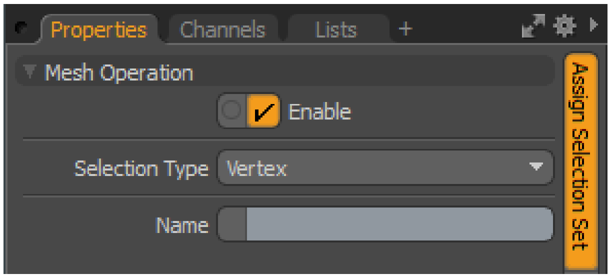
To add a selection set:

1. Click **Add Operator** in the **Mesh Operations** list.
2. Under **Mesh Operations** > **Selection**, double-click **Assign Selection Set**.

   The operation is added to the list.
3. Click the small arrow in front of the operation in the list to reveal its **Selection** input, then click **(Add Selection)**.



4. Under **New**, add the selection operation you need.
5. In the **Assign Selection Set** operation's **Properties** tab, select the **Selection Type** and name the selection set by entering typing into the **Name** field.



Selection Type Vertex

Name

6. Specify the selection. For example, if you've chosen **Select By Range** as the selection operation, define the range in the selection operation's properties.

Using Selection Sets

To use the selection set you've created, first add a mesh operation that requires a selection. Then add the Procedural Selection operation to the mesh operation's **Selection** input, and specify the name of the selection set you want to use.

Selection Operations

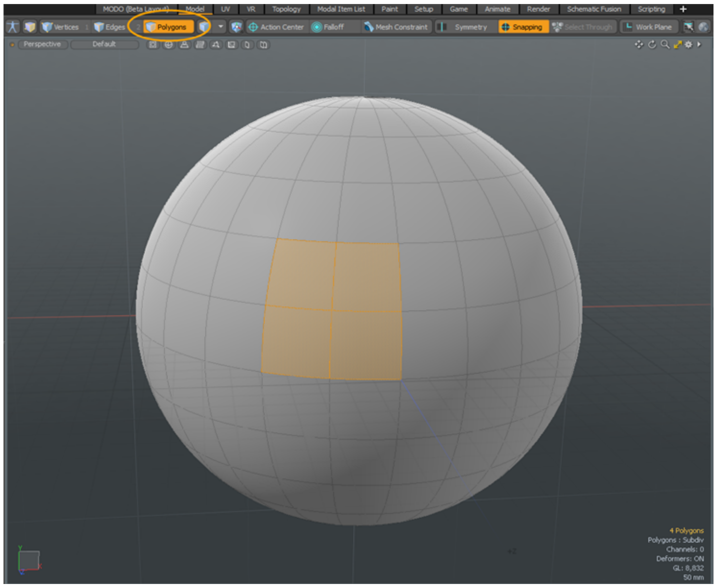
The following selection types are available in the **Mesh Operations** list.

Convert

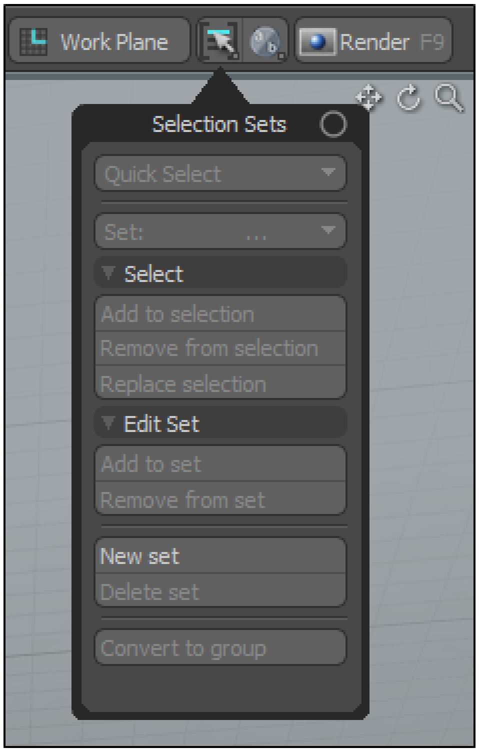
This selection operation converts a component selection from the given type to another component type, polygon, edge, or vertex, which is used by a procedural mesh operator in down stream in the **Mesh Operations** list (stack). The original source selection is retained and new selections are added.

Example

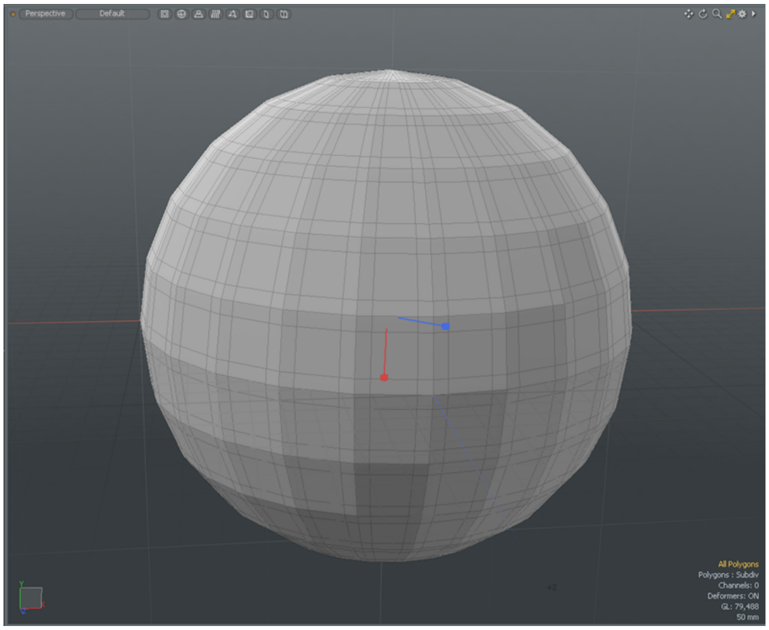
1. Add a Sphere to your scene.
2. Under the layout menu, click **Polygons**, and select four polygons on the Sphere.


MODO (Beta Layout)
Model
UV
VR
Topology
Modal Item List
Paint
Setup
Game
Animate
Render
Schematic Fusion
Scripting
Vertices
Edges
Polygons
Action Center
Falloff
Mesh Constraint
Symmetry
Snapping
Select Through
Work Plane
Perspective
Default
4 Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
50 mm

3. On the top left of the menu, beside the Work Plane button, click the Selection Sets icon and then click **New set**.


Work Plane
Render F9
Selection Sets
Quick Select
Set: ...
Select
Add to selection
Remove from selection
Replace selection
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group

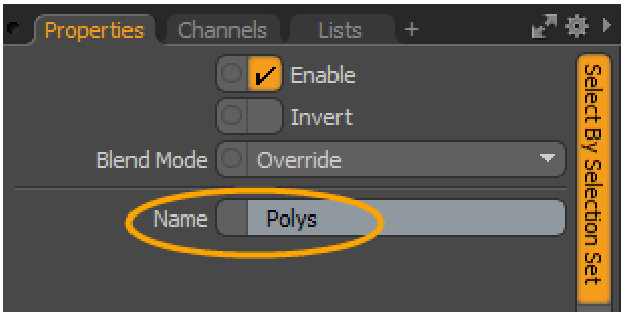
4. Name this selection set **Polys** and click **OK**.
5. On the right panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Edge** > **Edge Bevel**.
6. Drag the handles to apply a bevel on the edges.


Perspective
Default
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 79,488
50 mm

7. On the Mesh Ops tab, expand **Edge Bevel** > **Selection**, and click **(Add Selection)**.
8. Double-click **Mesh Operations** > **Selection** > **Select By Selection Set**.

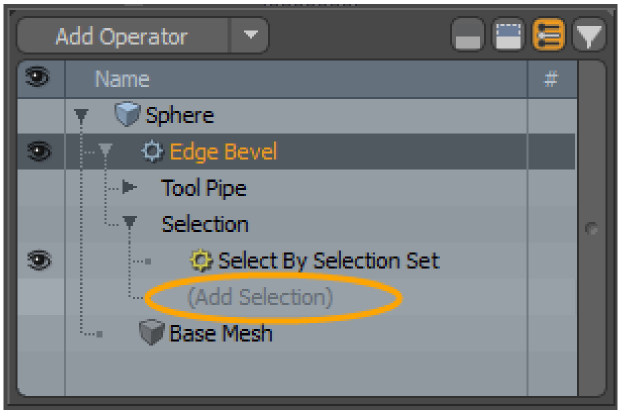
The beveled edges are no longer displayed because the Select By Selection Set uses a polygon selection but the Edge Bevel needs an edge selection to work.

9. In the **Name** field, enter **Polys**.


Properties
Channels
Lists
+
Select By Selection Set

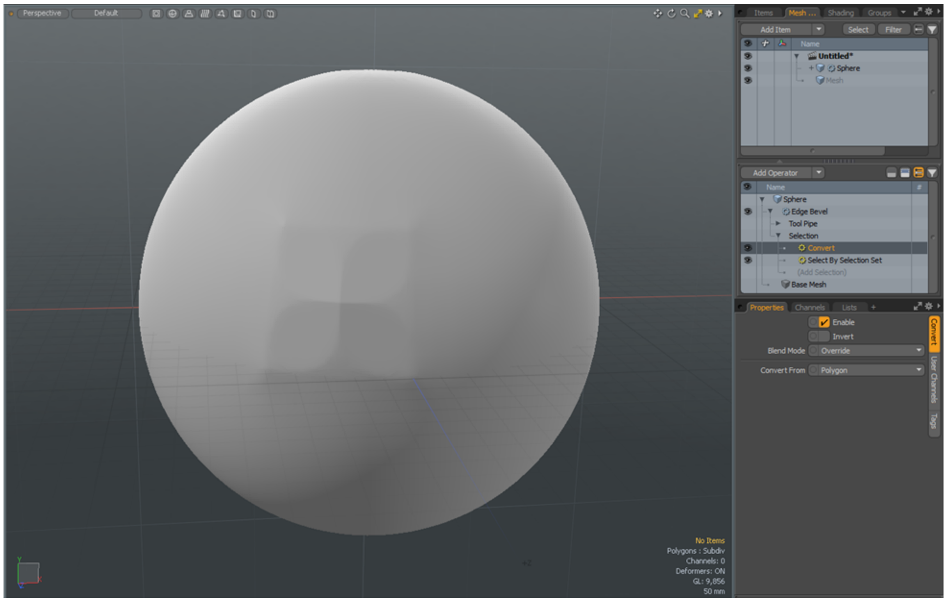
☑ Enable
☐ Invert
Blend Mode: Override
Name: Polys

10. Under **Edge Bevel** > **Selection**, click **(Add Selection)**.


Add Operator
Name
#
Sphere
Edge Bevel
Tool Pipe
Selection
Select By Selection Set
(Add Selection)
Base Mesh

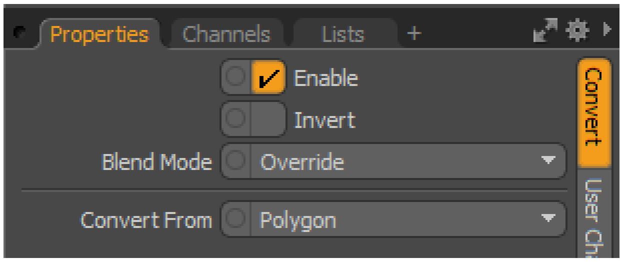
11. Double-click **Mesh Operation** > **Selection** > **Convert** and ensure the **Convert From** option is set to **Polygon**.

The Edge Bevel is now applied to only the Polys selection set.


Perspective
Default
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Sphere
Mesh
Add Operator
Name
Sphere
Edge Bevel
Tool Pipe
Selection
Convert
Select By Selection Set
(Add Selection)
Base Mesh
Properties
Channels
Lists
Convert
User Channels
Tags
No Items
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 9,856
50 mm

☑ Enable
☐ Invert
Blend Mode: Override
Convert From: Polygon

Convert Properties



☑ Enable

☐ Invert

Blend Mode: Override

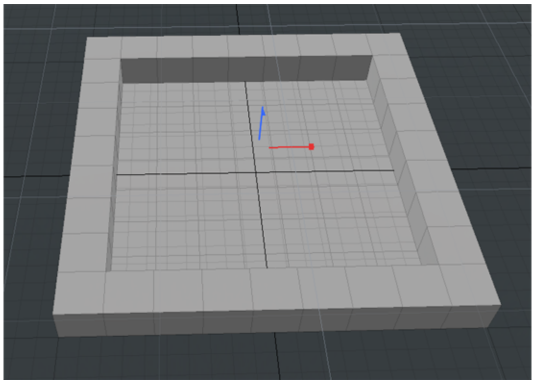
Convert From: Polygon

Enable	Enables or disables the **Select by Previous Operation** function.
Invert	Inverts a single selection operation or all selected operations in the **Mesh Operations** list.
Blend Mode	Specifies the type of blend to apply. • **Override** - The default behavior. The current selection overrides all selection operations below it in the **Mesh Operations** list. Previous selections are discarded and the new selection is used instead. • **Add** - Adds a new selection to the previous selection. Elements are treated as de-selected if they are de-selected by all selection operations in the **Mesh Operations** list. • **Subtract** - Removes any elements that are selected by the selection operation from the previous selection. • **Multiply** - Only selects elements from the new selection operation that are also selected by the first, filtering the previous selection.
Convert From	Specifies the component selection mode: **Polygons**, **Edges**, or **Vertices**.

Select by Border

This selection operation marks elements around the boundary of a geometry for edits.

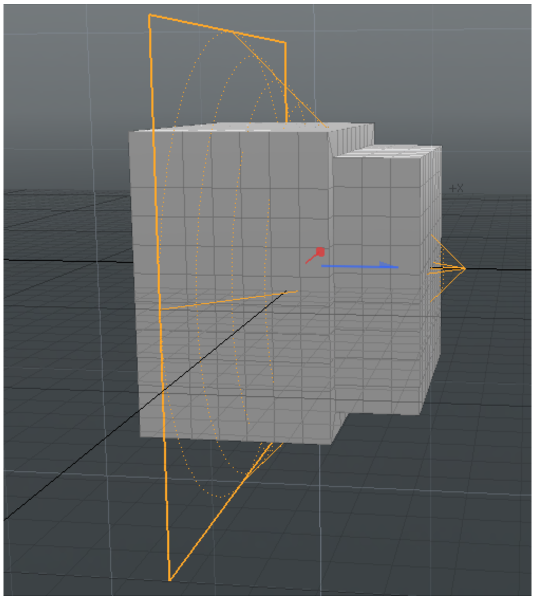
The image below shows a Polygon Bevel operation applied only to the border polygons of a plane, using the Select By Border selection operation:



Select by Falloff

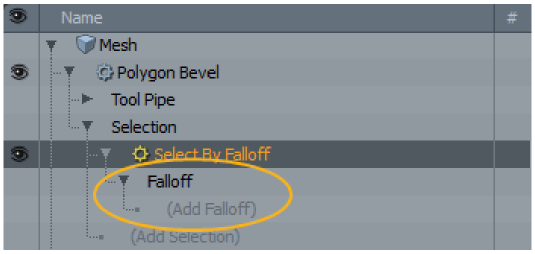
When using this selection operation, selection is defined by a falloff item, which you can also add through the **Mesh Operations** list. The added falloff item defines which parts of the mesh get selected. You can edit the properties of the falloff item to modify the selection.

The falloff is evaluated to calculate a weight for each element, based on its world space position. If the weight is above a certain threshold, the corresponding element gets selected. The image below shows the Polygon Bevel operation applied to a region of a cube affected by Linear Falloff:



To specify a falloff:

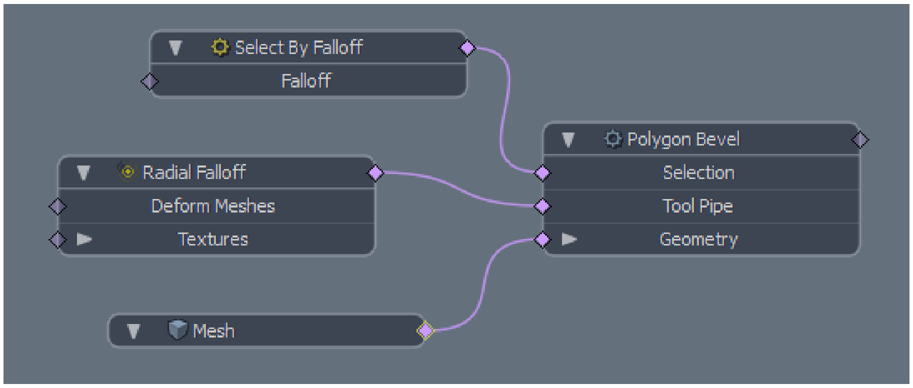
1. After adding the **Select By Falloff** operation, expand it in the **Mesh Operations** list to reveal its **Falloff** input.


Name
#
Mesh
Polygon Bevel
Tool Pipe
Selection
Select By Falloff
Falloff
(Add Falloff)
(Add Selection)

2. Click **(Add Falloff)** to specify a falloff type for the selection operation.
3. In the **Preset Browser**, under **New**, double-click a falloff type.

   The falloff appears in the list, and you can customize it in its **Properties** panel below the **Mesh Operations** list.

In the **Schematic** view, the selection operation item has a single graph input, allowing falloff items to be connected:


Select By Falloff
Falloff
Polygon Bevel
Selection
Tool Pipe
Geometry
Radial Falloff
Deform Meshes
Textures
Mesh

Select by Falloff Properties

The following options are available for this selection operation:

- **Threshold** - Determines whether a specific weight is treated as selected or not. Weights above the specified threshold get selected. For example, if the threshold is **50%**, and the evaluated weight for an element is **40%**, the element doesn't get selected. However, if the evaluated weight is **60%**, it is selected.

Select by Index

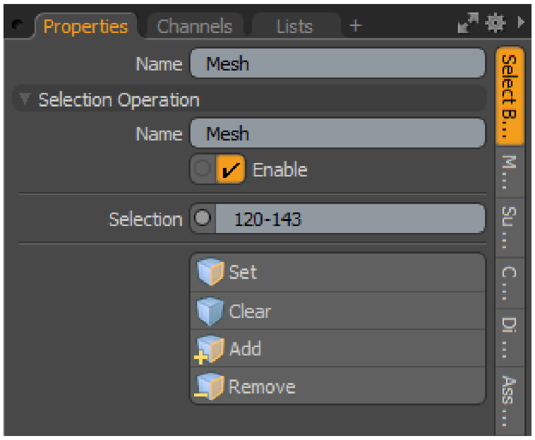
This selection node allows you to define a selection using the element index in a list of indices. Each mesh element is given an index between zero and the total number of elements in the mesh. If a matching number is found in the index list, the element is selected. The list of indices are automatically populated with the current element selection when the selection operation is created.

Note: As the input mesh changes, the index mapping may change, and the selected elements may be exchanged for other elements.

To define a selection using this operation, select elements of a mesh directly in the 3D viewport, and then add a compatible mesh operation. For example, select some polygons on a mesh, then add a Polygon Bevel tool in the **Mesh Operations** list. The **Select By Index** operation is automatically added to the operation's **Selection** input. The mesh operation then uses the selection you defined in the 3D viewport.

Select by Index Properties

You can modify or clear your selection by clicking the **Select By Index** node in the **Mesh Operations** list. This opens the operation's properties in the **Properties** panel. Here you have the following options:



Name: Mesh

Selection Operation

Name: Mesh

☑ Enable

Selection: 120-143

- **Selection** - Allows ranges of elements to be defined. For example, if the range is **0-9**, the first ten elements are selected. If the range is **0,1,2**, the first three elements get selected. You can also combine selection, for example, you can enter **0-9,12,14,16,18-30**.
- **Set** - Clears the existing selection and uses the newly-selected polygons instead.
- **Clear** - Removes all selections.
- **Add** - Adds the newly-selected elements to the existing selection.
- **Remove** - Removes the newly-added selection.

Select By Polygon Tag

The Mesh Operation, **Select By Polygon Tag**, allows you to procedurally select polygons using a polygon tag to quickly locate any polygons that have a matching value. This is useful when you want to apply procedural operations to your polygon selection sets. For example, if you have modeled a table and you want to assign different colors to the legs and to the table top. The **Select by Polygon Tag** mesh operation allows you to quickly do that. You can specify your selection operation to be a **Material**, **Part**, or **Selection**

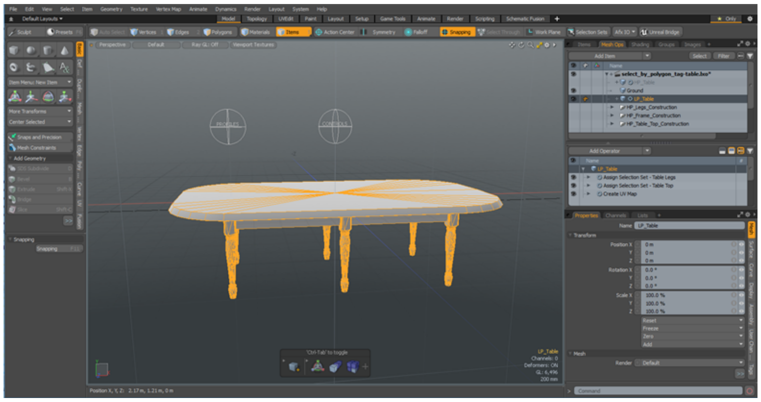
Set. You can also enter any four characters in the **Value** field to select by arbitrary polygon tag type previously set up.

For more information about how to set up Material, Part, or Selection Set options, see Defining Shaders and Using Selection Sets.

In the following example, we have previously modeled a table. This model has one mesh item and is procedural generated with Using Selection Sets created for the top and legs of the table.

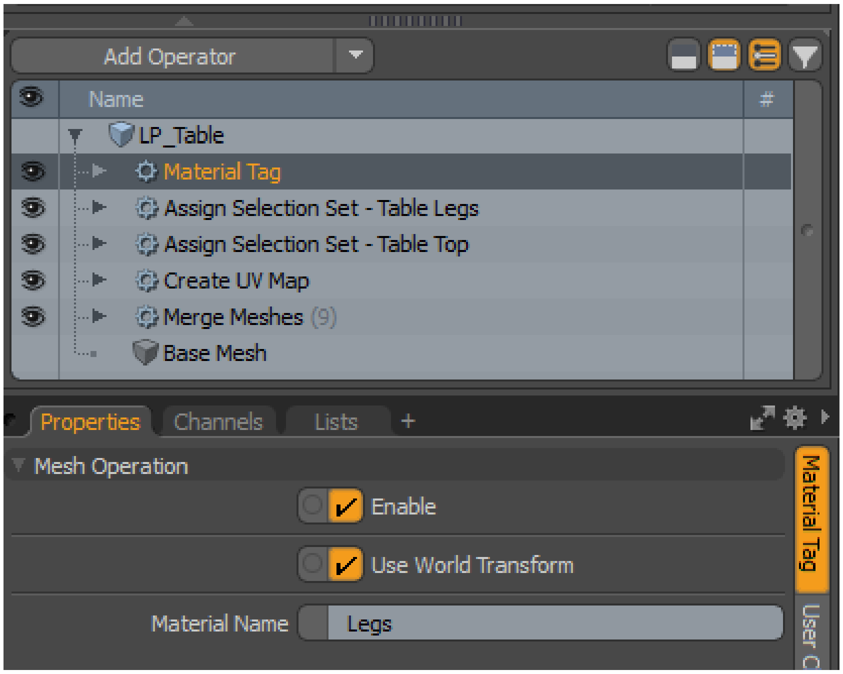
To apply the Select by Polygon Tag:

1. Download our example scenes and open **select_by_polygon_tag-table.lxo**. Alternatively, create your own model and assign selection sets. For more information, see Assigning a Selection Set.



2. From the **Mesh Ops** tab, select an **Assign Selection Set** to apply a mesh operation. In this example, select **Assign Selection Set - Table Legs**.
3. Click **Add Operator**, expand **Mesh Operations** > **Polygon** and double-click **Material Tag**.

   The Material Tag **Properties** displays in the right panel. For more information, see Assigning Material Tags.


Add Operator
Name
#
LP_Table
Material Tag
Assign Selection Set - Table Legs
Assign Selection Set - Table Top
Create UV Map
Merge Meshes (9)
Base Mesh
Properties
Channels
Lists
+
Mesh Operation
Material Tag
User C

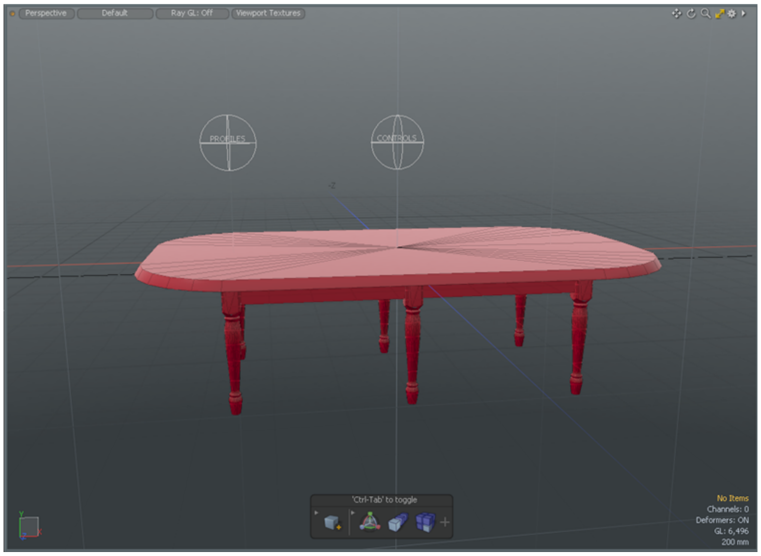
☑ Enable

☑ Use World Transform

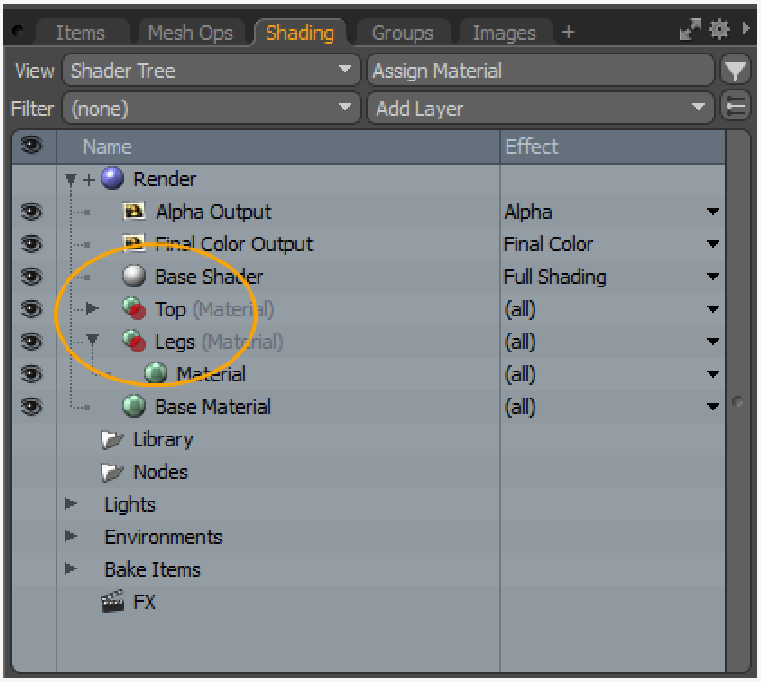
Material Name: Legs

4. In the **Material Name** field, type **Legs**, and then press **Return**.

   The entire table displays in red.


Perspective
Default
Ray GL: Off
Viewport Textures
PROFILES
CONTROLS
'Ctrl-Tab' to toggle
No Items
Channels: 0
Deformers: ON
GL: 6,496
200 mm

Note: Use the same capitalization for the created shader. To verify, open the **Shading** tab.

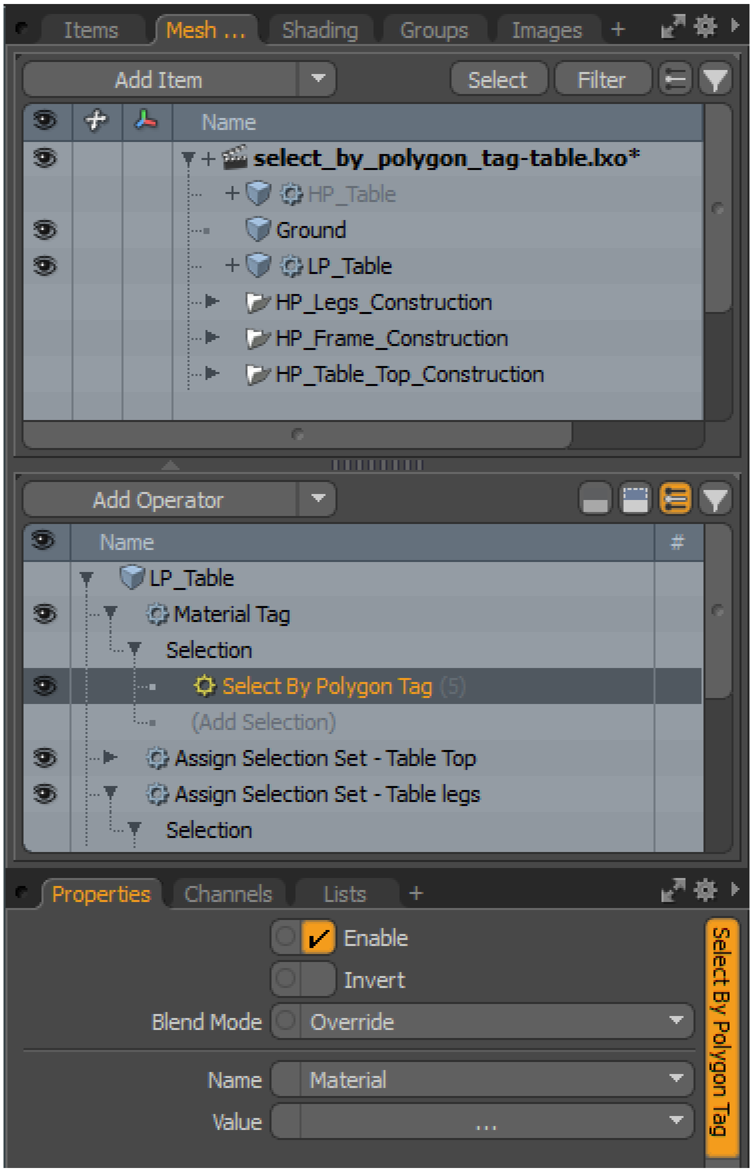


Items
Mesh Ops
Shading
Groups
Images
+
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Top (Material)
(all)
Legs (Material)
(all)
Material
(all)
Base Material
(all)
Library
Nodes
Lights
Environments
Bake Items
FX

5. In the **Mesh Ops** tab, expand **Material Tag** > **Selection**, and click **Add Selection**.

   The **Add Operation** dialog displays.

6. Expand **Mesh Operations** > **Selection** and double-click **Select By Polygon Tag**.


Items
Mesh ...
Shading
Groups
Images
+
Add Item
Select
Filter
Name
select_by_polygon_tag-table.lxo*
HP_Table
Ground
LP_Table
HP_Legs_Construction
HP_Frame_Construction
HP_Table_Top_Construction
Add Operator
Name
#
LP_Table
Material Tag
Selection
Select By Polygon Tag (5)
(Add Selection)
Assign Selection Set - Table Top
Assign Selection Set - Table legs
Selection
Properties
Channels
Lists
+
Select By Polygon Tag

☑ Enable

☐ Invert

Blend Mode: Override

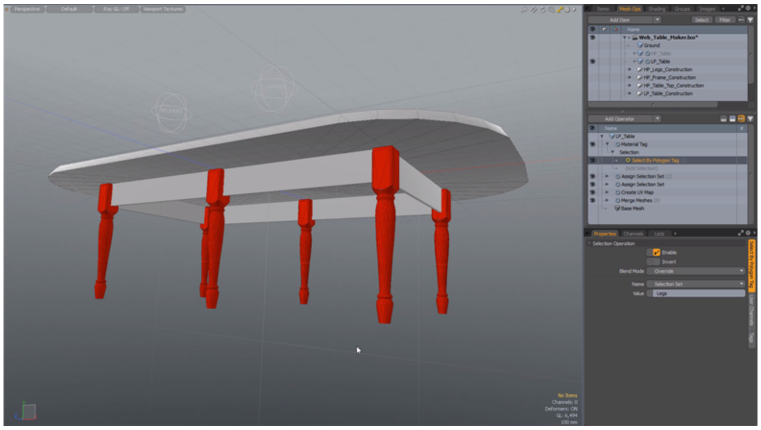
Name: Material

Value: ...

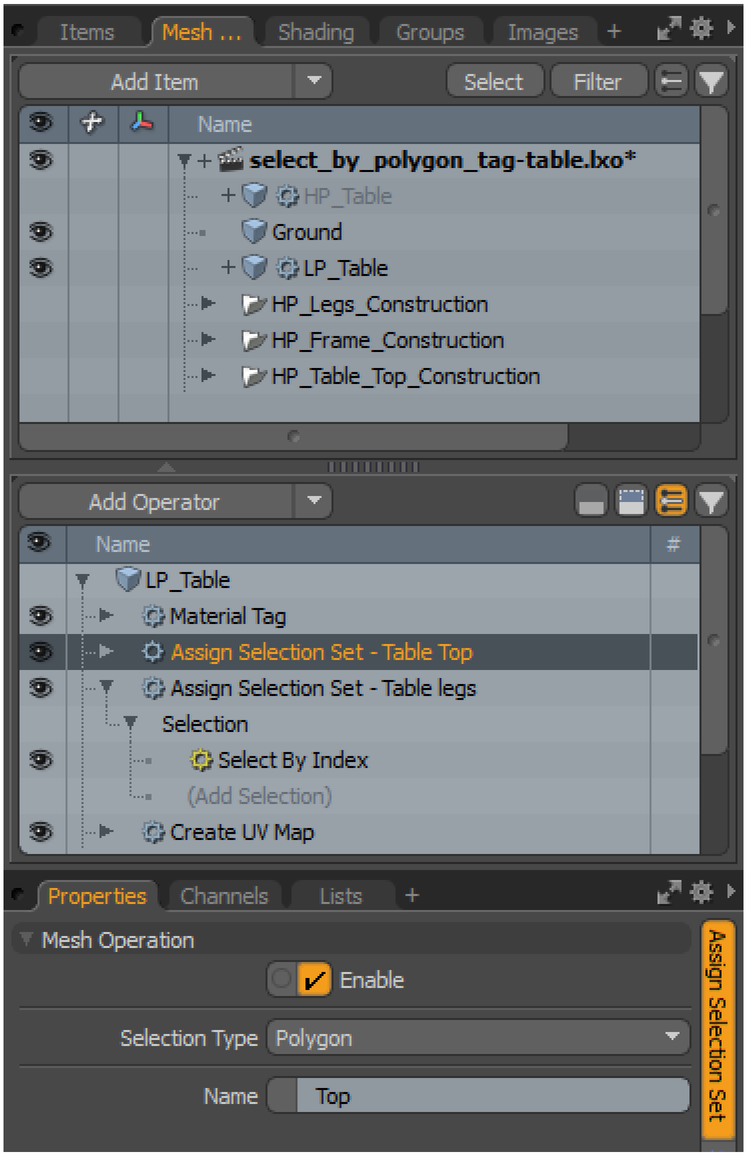
7. Choose the polygon tag you want to use. In this example, select **Selection Set** from the **Name** dropdown menu.
8. Select **Legs** from the **Value** dropdown menu.

Tip: Alternatively, you can type in the **Value** field. This can be four characters used by the name of the polygon tag. For example, if you had four selection sets that started with the word *Leg*, all polygons from all four selection sets are selected in your scene. Once done, a material is assigned to all four legs.

9. Click **Return** to view an update in the 3D viewport. Alternatively, click in the 3D viewport and press the **space bar** to drop your selection.



10. From the **Mesh Ops** tab, select an **Assign Selection Set** to apply a mesh operation. In this example, select **Assign Selection Set - Table Top**.


Items
Mesh ...
Shading
Groups
Images
Add Item
Select
Filter
Name
select_by_polygon_tag-table.lxo*
HP_Table
Ground
LP_Table
HP_Legs_Construction
HP_Frame_Construction
HP_Table_Top_Construction
Add Operator
Name
#
LP_Table
Material Tag
Assign Selection Set - Table Top
Assign Selection Set - Table legs
Selection
Select By Index
(Add Selection)
Create UV Map
Properties
Channels
Lists
Mesh Operation
Assign Selection Set

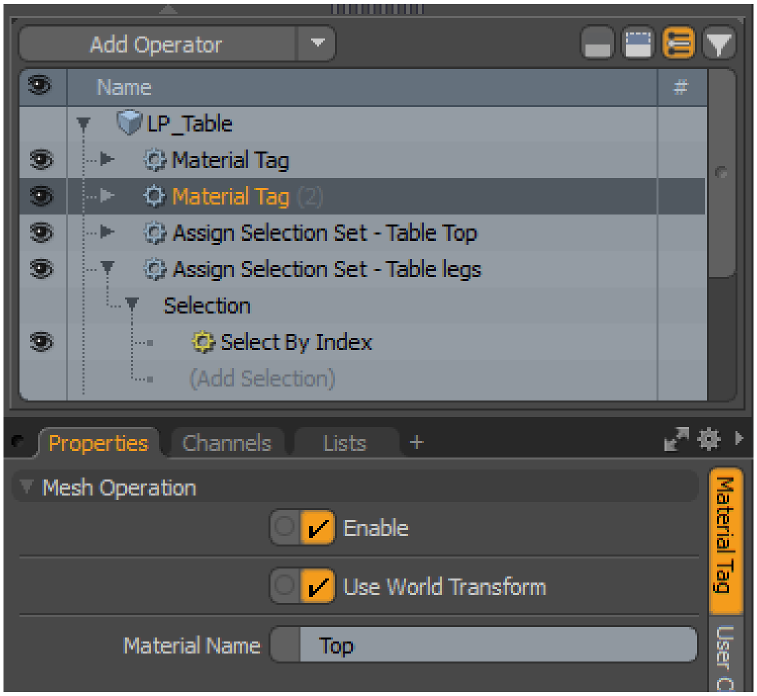
☑ Enable

Selection Type: Polygon

Name: Top

11. Click **Add Operator**, expand **Mesh Operations** > **Polygon**, and double-click **Material Tag**.

12. In the **Material Name** field, type **Top**.


Add Operator
Name
#
LP_Table
Material Tag
Material Tag (2)
Assign Selection Set - Table Top
Assign Selection Set - Table legs
Selection
Select By Index
(Add Selection)
Properties
Channels
Lists
+
Mesh Operation
Material Tag
User C

☑ Enable

☑ Use World Transform

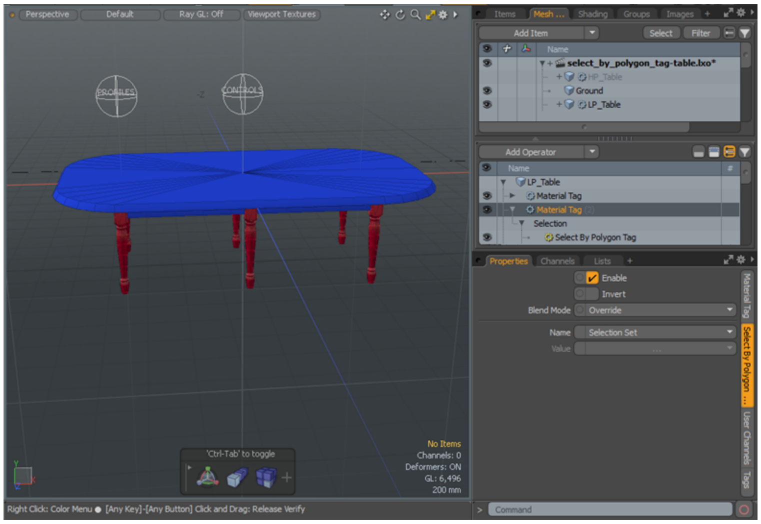
Material Name: Top

13. In the **Mesh Ops** tab, expand **Material Tag(2)** > **Selection**, and click **Add Selection**.

    The **Add Operation** dialog displays.

14. Expand **Mesh Operations** > **Selection** and double-click **Select By Polygon Tag**.

    The top of the table top has a newly assigned material.


Perspective
Default
Ray GL: Off
Viewport Textures
PROFILES
CONTROLS
'Ctrl-Tab' to toggle
No Items
Channels: 0
Deformers: ON
GL: 6,496
200 mm
Right Click: Color Menu ● [Any Key]-[Any Button] Click and Drag: Release Verify
Items
Mesh ...
Shading
Groups
Images
+
Add Item
Select
Filter
Name
select_by_polygon_tag-table.lxo*
HP_Table
Ground
LP_Table
Add Operator
Name
#
LP_Table
Material Tag
Material Tag (2)
Selection
Select By Polygon Tag
Properties
Channels
Lists
+
Material Tag
Select By Polygon ...
User Channels
Tags
Command

☑ Enable

☐ Invert

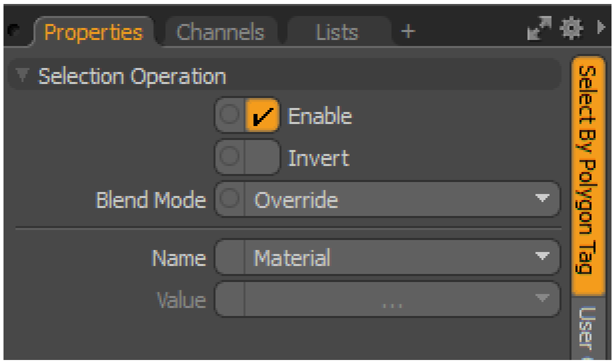
Blend Mode: Override

Name: Selection Set

Value: ...

Select by Polygon Tag Properties

The following options are available for this selection node:



Selection Operation

☑ Enable

☐ Invert

Blend Mode: Override

Name: Material

Value: ...

- **Name** - Allows you to choose the polygon tag type. The following options are available:
  - **Material** - Uses a previously created **Material Tag**

Tip: A **Material Tag** creates a mask for the surfacing using the material and shading properties and applies them to specific surfaces. For more information, see Assigning Material Tags.

  - **Part** - Uses a previously created **Part Tag**.

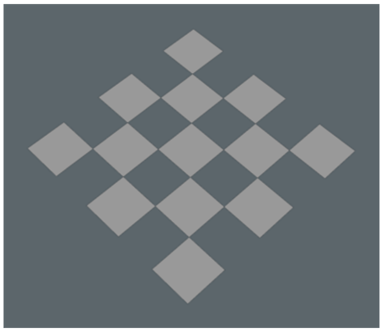
Tip: A **Part Tag** works in a similar fashion to Material Tags, offering an additional means of defining polygon groupings. For more information, see Defining Shaders.

  - **Selection Set** - Uses a previously created component and item-based selection set. For more information, see Using Selection Sets.
- **Value** - Specifies which tag to match to.

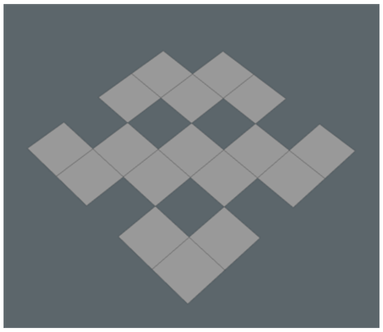
Select by Pattern

The Select by Pattern selection operation creates a selection based on a pattern you define. The pattern selection specifies elements using a "True, False" pattern. Elements whose index is present in the pattern are selected. For example, the pattern "True, False" selects every other element, whereas the pattern "False, False, True" selects every third element.

On a standard grid you get the following patterns:



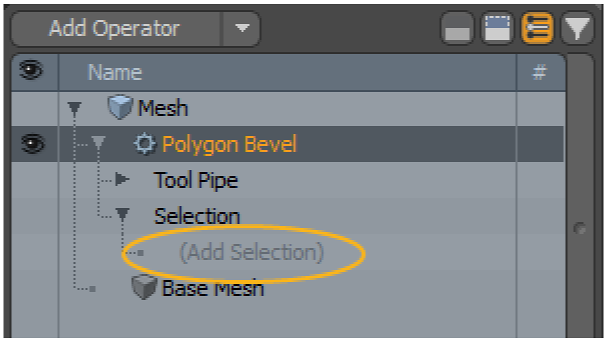
True, False



False, False, True

To use Select by Pattern:

1. In the Mesh Operations list, expand the operation for which you want to define the selection, such as a Polygon Bevel.
2. Expand its **Selection** input and click **(Add Selection)**.

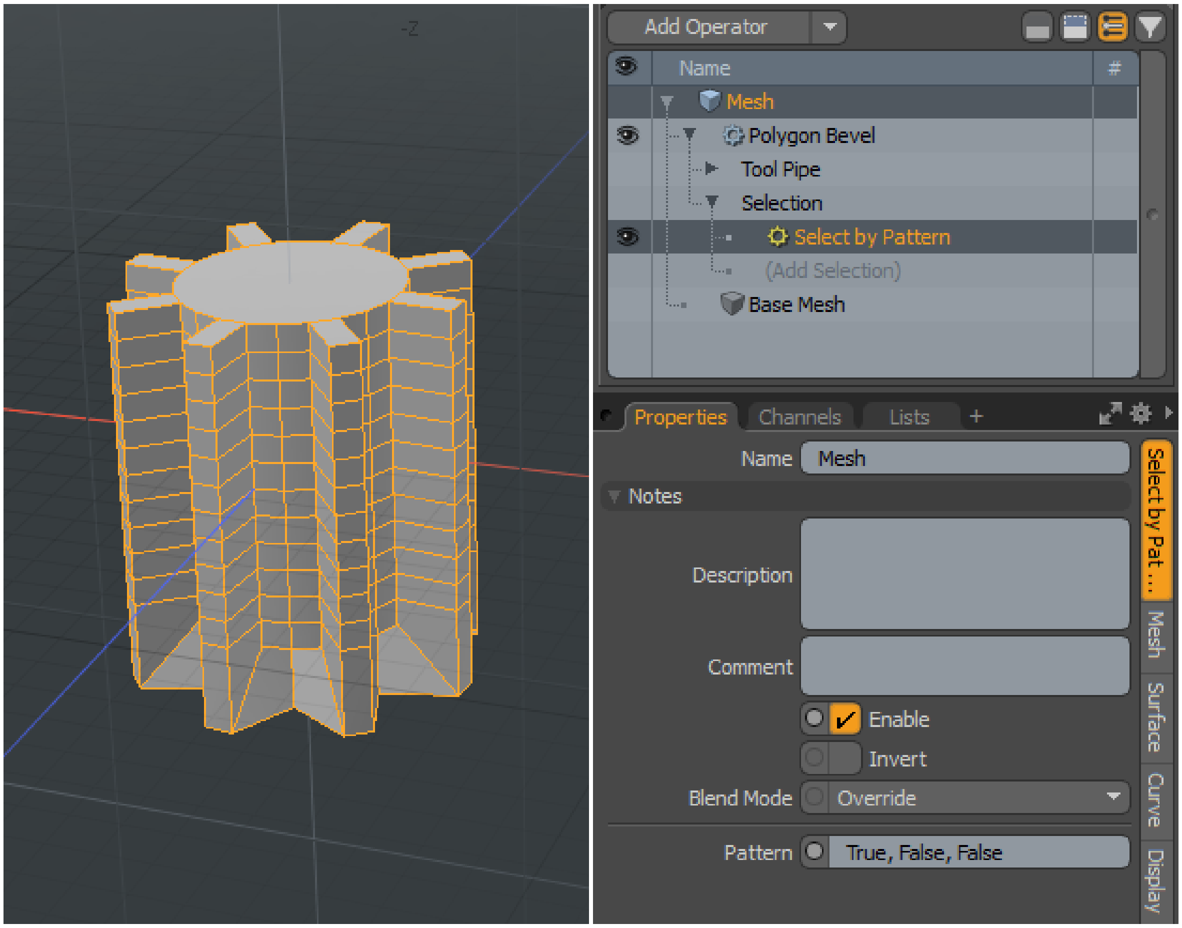


3. In the **Add Operator** window, double-click **Select by Pattern**.

   The selection operation is added to the Mesh Operations list, and its properties open in the lower right panel.
4. In the **Properties** panel, enter the required values in the **Pattern** field. For example, you can enter 'True, False' to select every other polygon.

   You can also enter values like '1, 0' or 'Yes, No', and Modo automatically converts them to True or False.
5. Click the operation that uses the selection, and in the lower right panel, modify its **Properties**, as needed.

   The changes are applied to the polygons according to the pattern you defined. In the example below, the Polygon Bevel operation is applied to the selection defined by the pattern 'True, False, False', which means that every third polygon is selected, starting with the first polygon in the selection.


-Z
Add Operator
Name
#
Mesh
Polygon Bevel
Tool Pipe
Selection
Select by Pattern
(Add Selection)
Base Mesh
Properties
Channels
Lists
+
Notes
Select by Pat...
Mesh
Surface
Curve
Display

Name: Mesh

Description

Comment

☑ Enable

☐ Invert

Blend Mode: Override

Pattern: True, False, False

Select by Previous Operation

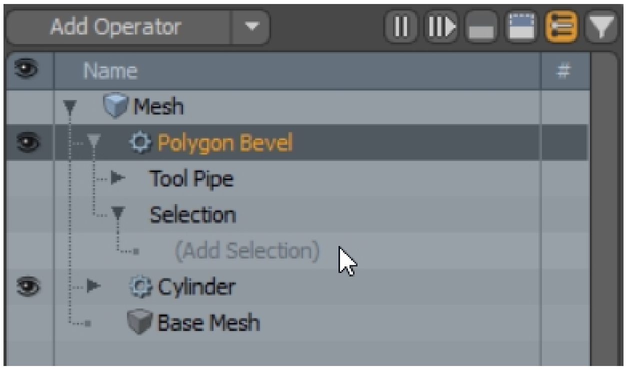
Use this selection operation to automatically leverage the results of mesh operations that are lower in a stack. For instance, if you create a cylinder using the **Cylinder** MeshOp you now have the beginnings of a procedural stack. Then you create a **Polygon Bevel** MeshOp above the **Cylinder** MeshOp. Since you haven't selected anything, the bevel is applied to the entire cylinder. You could use a **Select by Index Selection** op to manually select exactly which polygons you want to bevel. However, since the **Select by Previous Operation** now supports aspects of the Cylinder MeshOp, you can now use that instead to easily and reliably select aspects of the cylinder primitive. In this case, the **Caps**, **Top**, **Bottom**, and **Sides** can be selected.

> **Note:** Procedural operation options vary from one mesh operation to another. For example, a **Bevel** operation may provide separate lists for the back, side, and top polygons, whereas a **Curve Fill** operation may present different lists for each filled shape.

Select Elements Automatically Using Select by Previous Operation

In this example, we'll create a cylinder primitive at the bottom of the stack and then we'll create a **Polygon Bevel** MeshOp to bevel part of the cylinder. Instead of manually selecting what to bevel, we'll use **Select by Previous Operation** to automatically select the **Caps** of the cylinder lower down in the procedural stack.

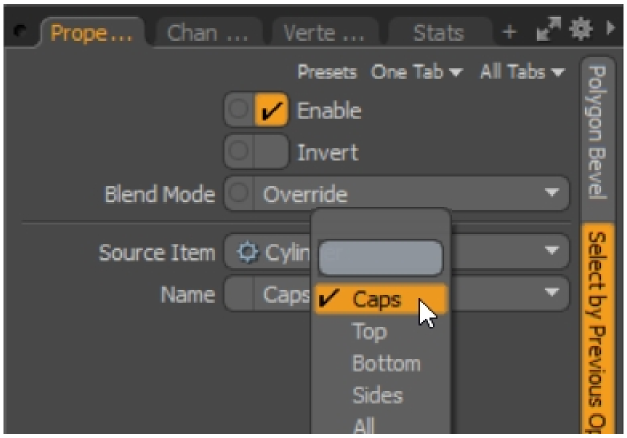
1. In the Mesh Operations viewport, click **Add Operator**, select **Create** > **Primitives**, and then double-click **Cylinder** to add the mesh to the 3D viewport.
2. Click **Add Operator** again, select **Polygon**, and then double-click **Polygon Bevel** to add the operation to the stack.
3. Adjust the bevel handles in the viewport to see the result. The bevel affects the whole mesh, which results in scaling the whole cylinder.
4. To automatically select the cylinder caps, drill down in the **Polygon Bevel** MeshOp to the **Selection** component.



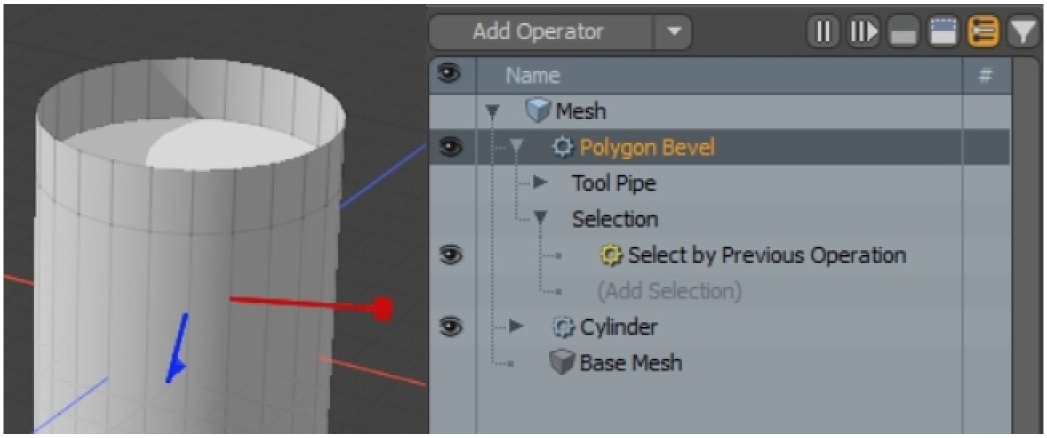
5. Click **(Add Selection)** and double-click **Select by Previous Operation** to add the selection op.
6. In the **Properties** viewport, click the **Source Item** dropdown and select **Cylinder**.
7. Click the **Name** dropdown to see a list of components you can select to drive operations higher up the tree. In this case, the **Polygon Bevel**.

> **Note:** Select by Previous options vary depending on the MeshOp that's being referenced. In this example, **Polygon Bevel** provides separate options for the **Caps**, **Top**, **Bottom**, and **Sides**, whereas a **Curve Fill** operation presents different options for each filled shape.

8. Select **Caps** to affect only the caps of the cylinder.

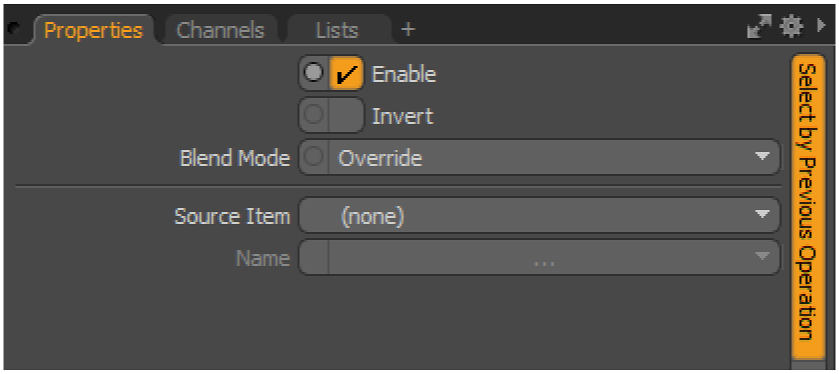

Prope ...
Chan ...
Verte ...
Stats
Presets
One Tab
All Tabs
Polygon Bevel
Enable
Invert
Blend Mode
Override
Source Item
Cylin
Name
Caps
Caps
Top
Bottom
Sides
All
Select by Previous Op

9. Now, when you select the **Polygon Bevel** op, you can see that the bevel is only applied to the caps of the cylinder, rather than the whole mesh.


Add Operator
Name
#
Mesh
Polygon Bevel
Tool Pipe
Selection
Select by Previous Operation
(Add Selection)
Cylinder
Base Mesh

Select By Previous Operation Properties

The following options are available:


Properties
Channels
Lists
+
Enable
Invert
Blend Mode
Override
Source Item
(none)
Name
...
Select by Previous Operation

Option	Description
Enable	Enables or disables the **Select by Previous Operation** function.
Invert	Inverts a single selection operation or all selected operations in the **Mesh Operations** list.
Blend Mode	Affects the blending between different layers of the same effect type. The following options are available: • **Override** - The default behavior. The current selection overrides all selection operations below it in the **Mesh Operations** list. Previous selections are discarded and the new selection is used instead. • **Add** - Adds a new selection to the previous selection. Elements are treated as de-selected if they are de-selected by all selection operations in the **Mesh Operations** list. • **Subtract** - Removes any elements that are selected by the selection operation from the previous selection. • **Multiply** - Only selects elements from the new selection operation that are also selected by the first, filtering the previous selection.
Source Item	Specifies the operation source item to apply the **Select by Previous Operation** to.
Name	Specifies the filter to apply to selected the operation. See the following table for **Name** filters available for each procedural operation. N/A denotes that there are no filters available for that operation.

Procedural Operation	Select by Previous Operation Name (where applicable)
Axis Slice	**Select by Previous Operation** can select **New Edges** from this operation.
Booleans	N/A
Create Polygons	**Select by Previous Operation** can select **new** from this operation.
Create Vertex	**Select by Previous Operation** can select **new** from this operation.
Curve Fill	N/A
Curve Slice	N/A
Edge Bevel	N/A

<table>
<tr><th colspan="2">Procedural Operation</th><th>Select by Previous Operation Name (where applicable)</th></tr>
<tr><td colspan="2">Edge Extrude</td><td>N/A</td></tr>
<tr><td colspan="2">Split</td><td>Select by Previous Operation can select Caps, Cap Side A, and Cap Side B from this operation.</td></tr>
<tr><td colspan="2">Edge Extrude</td><td>Select by Previous Operation can select New Curves from this operation.</td></tr>
<tr><td colspan="2">Polygon Bevel</td><td>N/A</td></tr>
<tr><td colspan="2">Polygon Extrude</td><td>N/A</td></tr>
<tr><td colspan="2">Unify</td><td>Select by Previous Operation can select Unified Polygons from this operation.</td></tr>
<tr><td colspan="2">Slice</td><td>N/A</td></tr>
<tr><td colspan="2">Smooth Shift</td><td>N/A</td></tr>
<tr><td colspan="2">Solid Drill</td><td>N/A</td></tr>
<tr><td colspan="2">Sweep Effector</td><td>N/A</td></tr>
<tr><td colspan="2">Subdivide</td><td>Select by Previous Operation can select New Edges from this operation.</td></tr>
<tr><td colspan="2">Triangulate</td><td>Select by Previous Operation can select New Edges from this operation.</td></tr>
<tr><td colspan="2">Vertex Bevel</td><td>N/A</td></tr>
<tr><td colspan="2">Vertex Extrude</td><td>N/A</td></tr>
<tr><td colspan="3">All 3D Primitives</td></tr>
<tr><td>Capsule</td><td>Ellipsoid</td><td rowspan="4">Select by Previous Operation can select Caps, Top, Bottom, Sides, and All from all primitive operations, where appropriate.</td></tr>
<tr><td>Cone</td><td>Sphere</td></tr>
<tr><td>Cube</td><td>Text</td></tr>
<tr><td>Cylinder</td><td>Torus</td></tr>
<tr><td colspan="3">All 2D Primitives</td></tr>
</table>

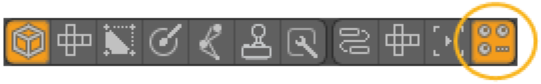
Procedural Operation		**Select by Previous Operation Name (where applicable)**
Circle	N-Sided	**Select by Previous Operation** can select **Curve** from all 2D primitive operations.
Diamond	Rectangle	
Ellipse	Star	
All Clone Tools		
Array	Mirror	**Select by Previous Operation** can select all **Cloned Polygons** or individual **Clone *n*** from all Clone operations.
Clone	Duplicate Geometry Using Particle Clone	
Procedural Clone Effector	Pen Clone	
Curve Clone	Scatter Clone	

Select Connected

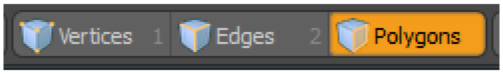
With this operation, you can quickly locate connected geometry to a polygon, vertex, or edge selection. This Mesh Operation uses a previous selection method, such as **Select By Index**, to compute all vertices, edges, and polygons linked to the previous selection. This is useful if you have a complex model and you want to remove a selection from the parent mesh item.

The following example demonstrates applying **Select Connected** command to a previous selection method.

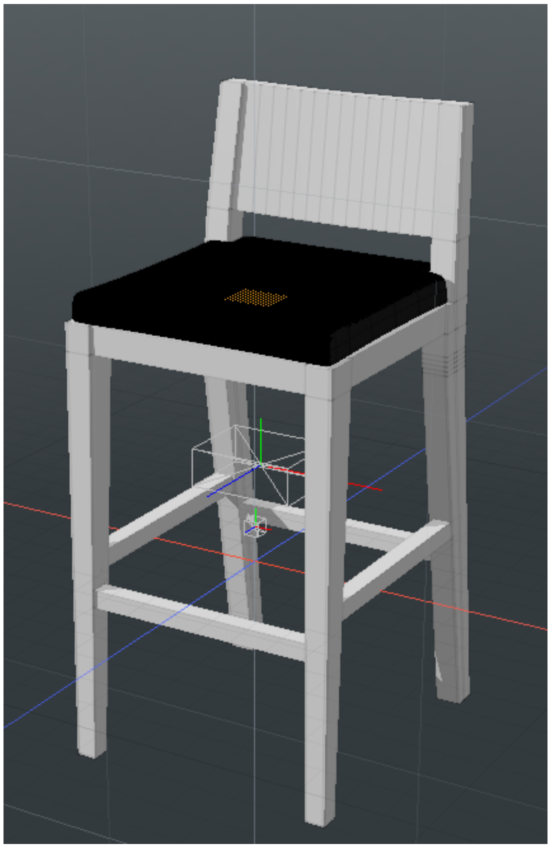
1. On the top left corner of the interface, click the Preset Browser icon.



2. On the left panel of the Preset Browser, expand **Cloud Assets** > **Meshes** > **Interior**, and double-click **Afton_Stool**.
3. Under the layout menu, click **Polygons**.

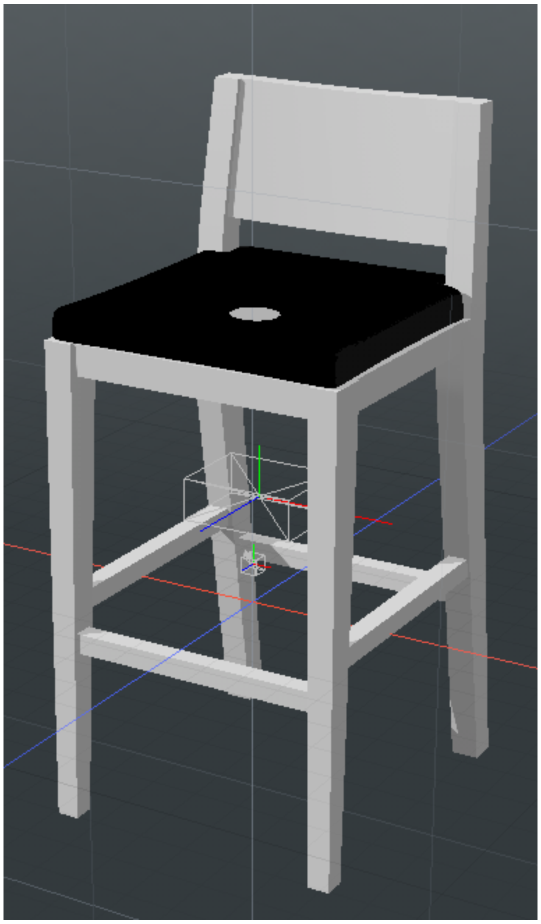


4. Select one polygon on the seat cushion.

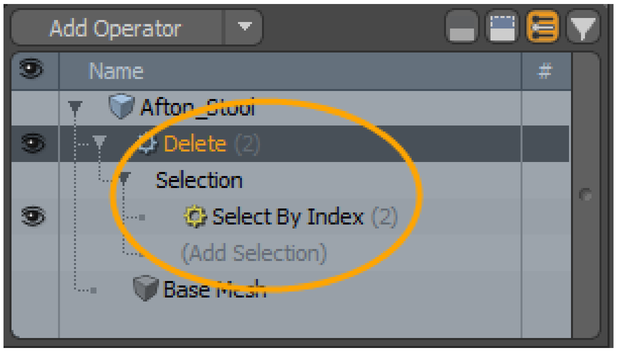


5. On the right panel, click **Add Operator**, double-click **Mesh Operations** > **Edit** > **Delete**.

   The selected polygon is removed from the cushion.

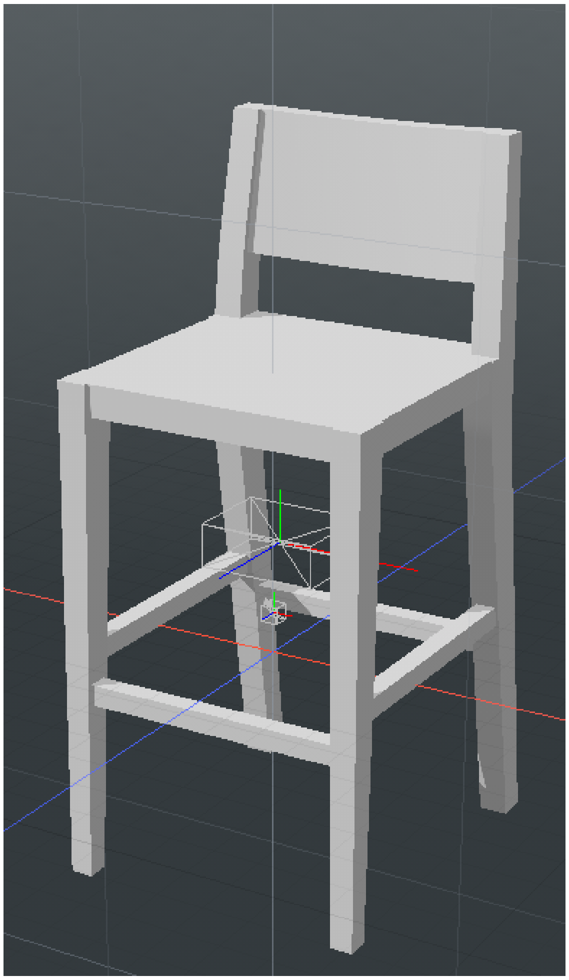


6. Expand the **Delete** mesh operation and click **(Add Selection)**.


Add Operator
Name
#
Afton_Stool
Delete (2)
Selection
Select By Index (2)
(Add Selection)
Base Mesh

7. In the **Add Operations** dialog, double-click **Mesh Operation** > **Selection** > **Select Connected**.

   All of the polygons connected to the cushion are selected and deleted.



Select Coplanar Polygons

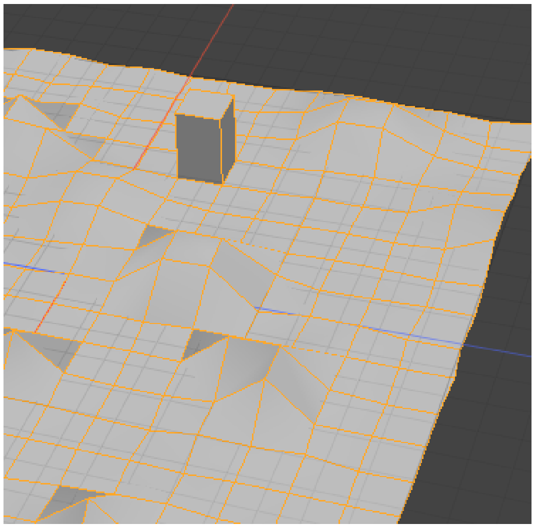
Select Coplanar Polygons is the procedural equivalent of the Coplanar Polygons Tool. It allows you to select neighboring polygons on the same plane as the original selection.

To use the operation:

1. In the 3D viewport, select one or more polygons on your mesh.
2. In the Mesh Operations list, add a mesh operation that you want to apply to the selection. For instance, if you wanted to bevel the selected polygons, add the Polygon Bevel mesh operation to the list.

> **Note:** For more information on the Polygon Bevel mesh operation, see Procedural Polygon Bevel.

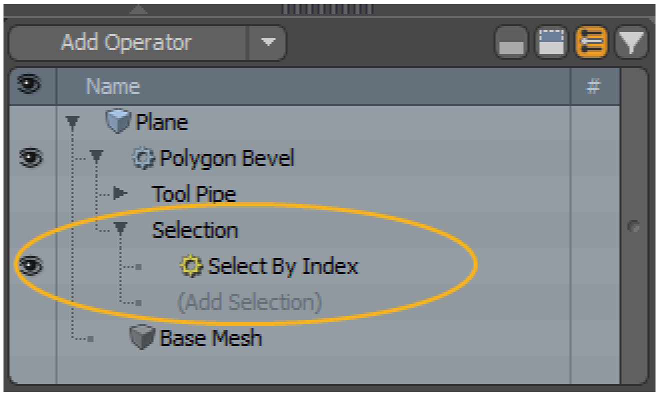
Adding the Polygon Bevel applies the operation to the selected polygon.



3. In the Mesh Operations list, click the arrow in front of the operation's name to expand it.

   This reveals the operation's **Selection** input.
4. Expand the **Selection** input by clicking the arrow in front of it.

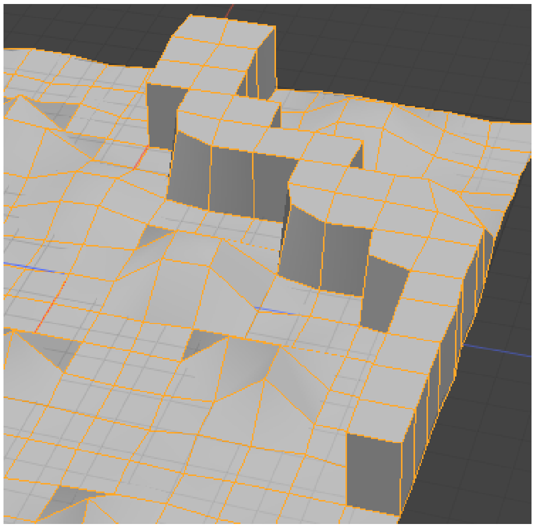
   You can see a Select by Index already in the list. This was added when you selected the initial polygon in the 3D viewport.
5. To select polygons that are coplanar with the original selection, click **(Add Selection)**.


Add Operator
Name
#
Plane
Polygon Bevel
Tool Pipe
Selection
Select By Index
(Add Selection)
Base Mesh

   This opens the **Add Operation** dialog.
6. Under **New** > **Selection**, double-click **Select Coplanar Polygons**.

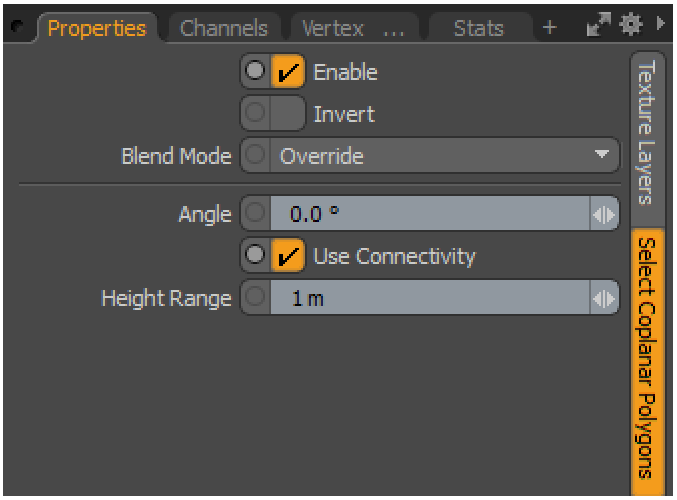
   This adds the Select Coplanar Polygons mesh operation to the list, and applies the connected mesh operation to all coplanar polygons. The image below shows the selected polygons with the default settings.

Note: If no polygons are coplanar to the selection, you won't see any changes.



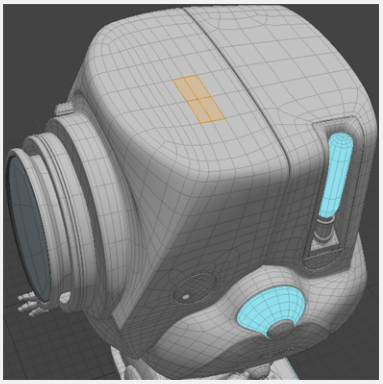
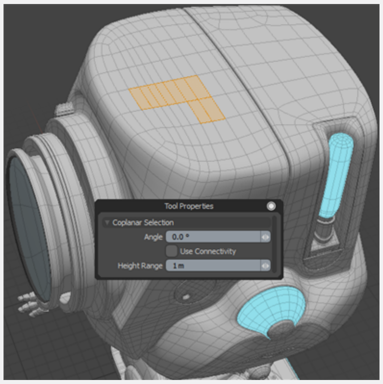
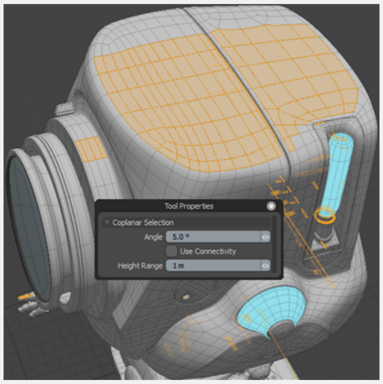
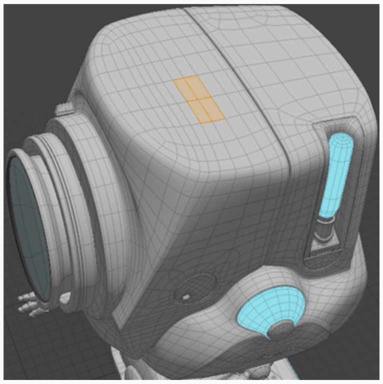
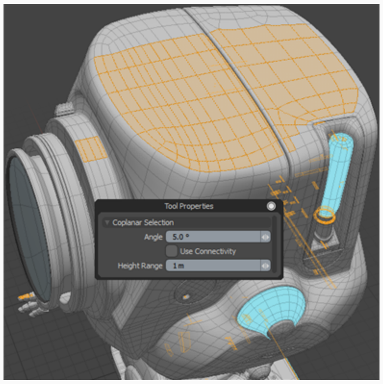
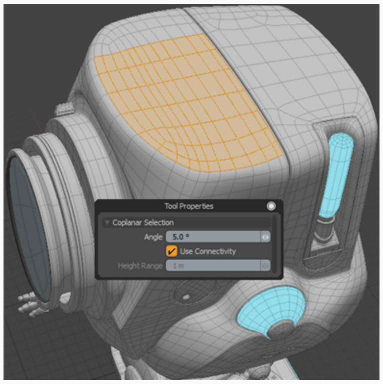
To modify the Select Coplanar Polygons mesh operation's properties, select it in the Mesh Operations list.

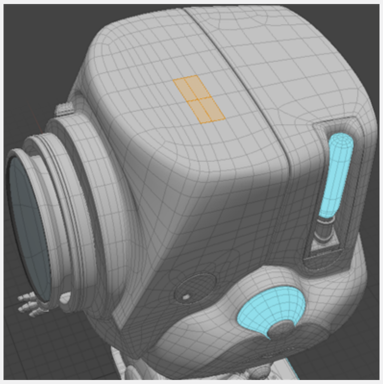
This displays the **Properties** in the lower right panel.


Properties
Channels
Vertex ...
Stats
+
Texture Layers
Select Coplanar Polygons

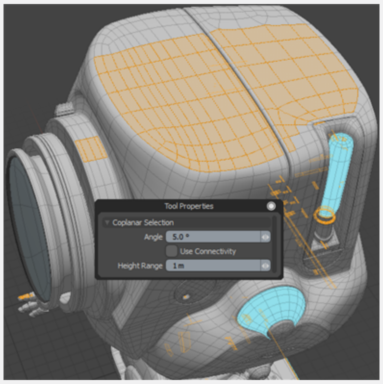
☑ Enable
☐ Invert
Blend Mode: Override
Angle: 0.0 °
☑ Use Connectivity
Height Range: 1 m

Angle	Determines the range outside the selected area to attenuate the effect of the tool. Increasing the value selects more polygons.

	 Original Polygon Selection  Tool Properties Coplanar Selection Angle 0.0 ° ☐ Use Connectivity Height Range 1 m Coplanar polygons with default values  Tool Properties Coplanar Selection Angle 5.0 ° ☐ Use Connectivity Height Range 1 m **Angle** at 5°
Use Connectivity	When enabled, only connected elements are selected. Unconnected elements within range are ignored.  Original Polygon Selection  Tool Properties Coplanar Selection Angle 5.0 ° ☐ Use Connectivity Height Range 1 m **Use Connectivity** disabled  Tool Properties Coplanar Selection Angle 5.0 ° ☑ Use Connectivity Height Range 1 m **Use Connectivity** enabled
Height Range	Determines the maximum threshold to include polygons with the same normal, but not on the same infinite plane. This is used when **Use Connectivity** is disabled. At 0 m, the tool selects only polygons on the infinite plane of the source polygons. When **Height Range** is higher than 0, the tool selects polygon with same normal direction with different height position in the specific range. The default value is 1 m.

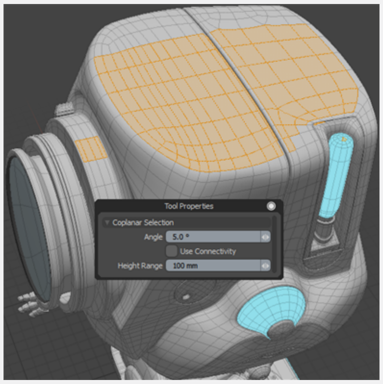


Original Polygon Selection



Tool Properties
Coplanar Selection
Angle 5.0 °
Use Connectivity
Height Range 1 m

Height Range at 1 m



Tool Properties
Coplanar Selection
Angle 5.0 °
Use Connectivity
Height Range 100 mm

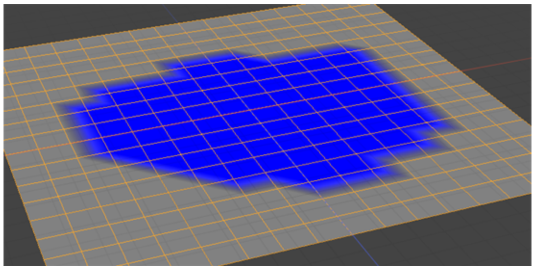
Height Range at 100 mm

Select Vertex Map Value

This selection operation allows you to enter a Vertex map value and select vertices where the value is greater than or equal to the supplied value.

To use the selection operation:

1. In the Mesh Operations list, add a mesh operation that you want to apply to the selection. For instance, if you want to extrude the selected vertices, add the Vertex Extrude mesh operation to the list. In this example, we apply Vertex Extrude to the blue area.



Note: If you're using a mesh operation that requires an edge or polygon selection, you have to add a Convert selection operation in addition to Select Vertex Map Values, to convert your vertex selection to an edge or polygon selection. For more information, see Convert.

2. In the Mesh Operations list, click the arrow in front of the operation's name to expand it.

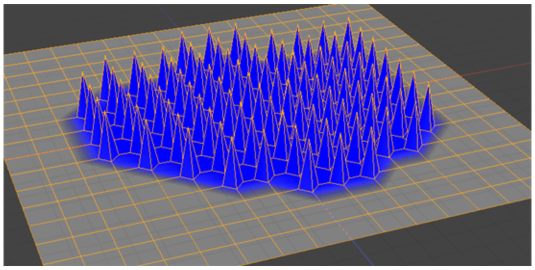
   This reveals the operation's **Selection** input.
3. Expand the **Selection** input by clicking the arrow in front of it, then click **(Add Selection)**.

   This opens the **Add Operation** dialog.

4. Under **New** > **Selection**, double-click **Select Vertex Map Value**.

   This adds the Select Vertex Map Value selection operation to the list and opens its **Properties** on the right pane.

5. In the **Properties**, select the relevant **Map Type** and **Map Name**, and enter the **Min Value to Select**. In our example, we use the **Component Method**, with **Component** set to **B** and the value is 2.0.
   Now the mesh operation only affects vertices that match the criteria.



Select Vertex Map Value Properties

Enable	Enables or disables Select Vertex Map Value.
Invert	Inverts the selection operation.
Blend Mode	Affects the blending between different layers of the same effect type. The following options are available: • **Override** - The default behavior. The current selection overrides all selection operations below it in the **Mesh Operations** list. Previous selections are discarded and the new selection is used instead. • **Add** - Adds a new selection to the previous selection. Elements are treated as de-selected if they are de-selected by all selection operations in the **Mesh Operations** list. • **Subtract** - Removes any elements that are specified by the selection operation from the previous selection. • **Multiply** - Only selects elements from the new selection operation that are also selected by the first, filtering the previous selection.
Map Type	The type of map to use the selection on. The available options are: • **Weight** - Stores weight strength values for falloffs and texturing. • **Subdivision** - A type of weight map controlling the rounding of edges on subdivision surfaces. • **RGB** - Defines a vertex color map based on red, green, and blue color values. • **RGBA** - Defines a vertex color map based on red, green, and blue color values and an

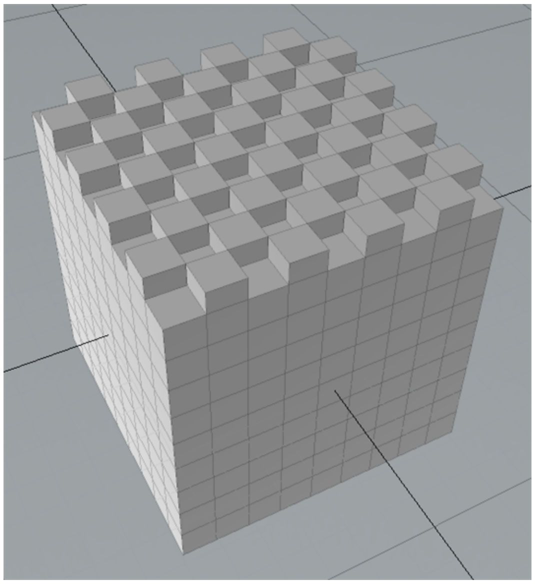
	additional transparency (Alpha) value. **Note:** For more information on Vertex Map types, see Working with Vertex Maps.
Map Name	The name of the map from which the value is read.
Method	The way the value is read: • **Component** - Reads the component of the map. Only applicable for RGB and RGBA maps. Weight and subdivision assume the first component. • **Sum** - The sum total of the components.
Component	When **Method** is set to **Component**, sets the component to select. This is only applicable to **RGB** and **RGBA** maps.
Min Value to Select	If the value is equal to or greater than this value, then the element is selected.

Select by Range

With this operation, you can define a selection using a range of indices matching the current element index. Each mesh element is given an index between zero and the total number of elements in the mesh. If the element index falls within the range defined by the selection operation, then the element is selected. You can also set up your selection to select, for example, every second or every third element in the range.

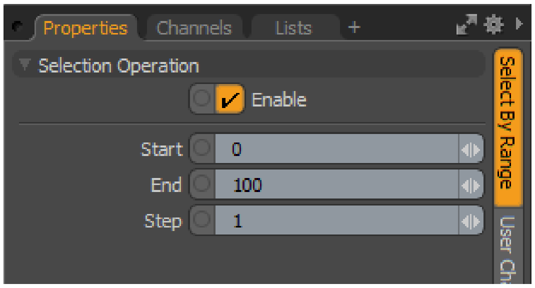
Note: As the input mesh changes, the index mapping may change, and the selected elements may be different.

The image below shows a Polygon Bevel using the Select By Range operation. The **Start** index is 0, the **End** index is 80, and the **Step** value is 2.



Select by Range Properties

The following options are available for this selection node:



Selection Operation

☑ Enable

Start 0

End 100

Step 1

- **Start** - Specifies the first element in the range.
- **End** - Specifies the last element in the range.
- **Step** - Determines the step value over the range. For example, a value of **1** means that every value between the **Start** and **End** values is in the range, whereas a value of **2** means that every other value is in the range, and so on.

Select by Hard Edges

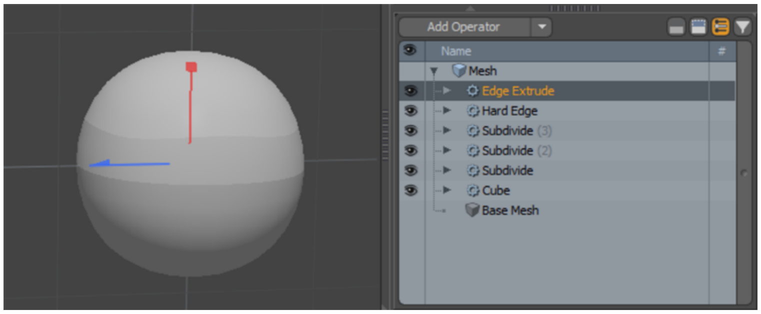
Use this selection operation to select all hard or soft edges on a geometry.

To learn how to create hard edges procedurally, see Procedural Hard Edges.

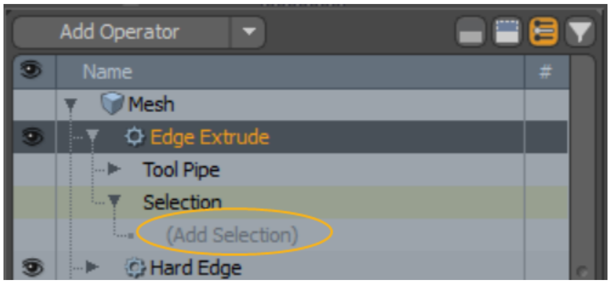
To use this selection type:

1. Add the mesh operation you want to apply to the selection to the Mesh Operations list. In the **Mesh Ops** tab on the right panel, click **Add Operator**, then select the mesh operation.

   In this example, we use an **Edge Extrude** operation.


Add Operator
Name
#
Mesh
Edge Extrude
Hard Edge
Subdivide (3)
Subdivide (2)
Subdivide
Cube
Base Mesh

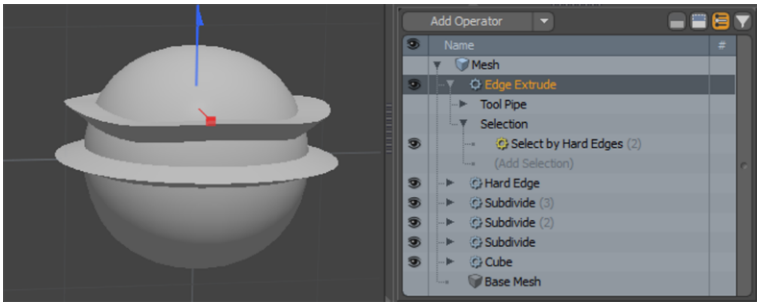
2. In the Mesh Operations list, Expand the **Edge Extrude** to reveal its **Selection** input. Expand **Selection** too, and click **(Add Selection)**.


Add Operator
Name
#
Mesh
Edge Extrude
Tool Pipe
Selection
(Add Selection)
Hard Edge

   This opens the **Add Mesh Operation** dialog.

3. Under **Modifier Mesh Operations** > **New** > **Selection**, double-click **Select by Hard Edges**.

   The mesh operation's selection input is populated with the **Select by Hard Edges** operation. If you now start using the mesh operation, it will only affect the hard edges.


Add Operator
Name
#
Mesh
Edge Extrude
Tool Pipe
Selection
Select by Hard Edges (2)
(Add Selection)
Hard Edge
Subdivide (3)
Subdivide (2)
Subdivide
Cube
Base Mesh

To select the soft edges of a mesh only:

1. Select the **Select by Hard Edges** operation in the Mesh Operations list.

   Its properties are displayed in the **Properties** tab in the lower right panel.

2. Set **Type** to **Soft**.

   Now the selection applies to all soft edges on your mesh.

Select by Hard Edges Properties

This selection type has the following properties.

Option	Description
Enable	Enables or disables **Select by Hard Edges**.
Invert	Inverts a single selection operation or all selected operations in the **Mesh Operations** list.
Blend Mode	Affects the blending between different layers of the same effect type. The following options are available: • **Override** - The default behavior. The current selection overrides all selection operations below it in the **Mesh Operations** list. Previous selections are discarded and the new selection is used instead. • **Add** - Adds a new selection to the previous selection. Elements are treated as deselected if they are deselected by all selection operations in the **Mesh Operations** list. • **Subtract** - Removes any elements that are selected by the selection operation from the previous selection. • **Multiply** - Only selects elements from the new selection operation that are also selected by the first, filtering the previous selection.
Type	Specifies whether the selection applies to **Hard** or **Soft** edges.

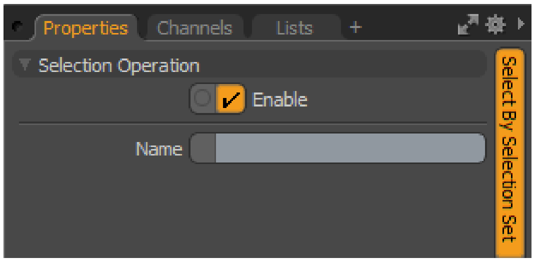
Select by Selection Set

This selection operation marks any elements that exist within a specified selection set. Combined with the **Assign Selection Set** mesh operation, this allows selection to be stored and then later used further up in the **Mesh Operations** list. For details on assigning selection sets, see Selection Operations.

Note: The selection set should be defined first, before adding the operation you want to use it with. So the **Assign Selection Set** operation should be below the other operation in the **Mesh Operations** list.

Select by Selection Set Properties

The following options are available for this selection node:



- **Name** - Specifies the name of the selection set to use.

Selection Operators

Selection Operators are a special kind of selection operation that test the element selection using an external Modo rig with customizable rules.

On each selection operator, the information about the element is provided as an output channel. For example, on the Polygon Selection Operator, information such as the position, normal, area, and index is provided as a channel, and a single input channel called **Selection** allows the selection state to be driven by a rig.

You can build any kind of Modo rig to take the output values from the selection operators, and determine the input value, specifying whether selection should be performed. When the selection operator is evaluated, the rig is evaluated for each mesh element and a different selection state can be returned for each element.

The example below guides you through setting up a simple rig using a Polygon Selection Operator. A Polygon Bevel operation is applied to polygons on a toroid, but only the ones facing the camera are selected.

1. Add a toroid to your scene from the **Basic** sub-tab of the **Model** toolbox, and subdivide it by pressing **D** on the keyboard for a more visible effect.
2. Go to the **Setup** layout, select the camera item in your scene, and in the **Schematic** viewport, click **Add Selected** to add it to the viewport. Then drag its **World Position** channel to the **Schematic** from the **Channels** list on the right.

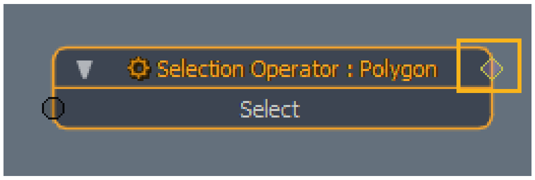
Note: To make sure that the camera always faces the toroid, select your camera item and toroid, and in the camera's **Properties**, click **Set Target**.

3. In the **Mesh Operations** list on right side of the interface, add a Polygon Bevel by clicking **Add Operator**, and under **Mesh Operations** > **Polygon**, double-clicking **Polygon Bevel**. Adjust the **Shift** amount by using the handles in the 3D viewport, or by editing the tool's **Properties**.
4. Expand the Polygon Bevel operation in the list to reveal its **Selection** input, then click **(Add Selection)** and in the Preset Browser, click **Selection Operator: Polygon**.

   This turns off the effect of the Polygon Bevel, as nothing is selected at this point.

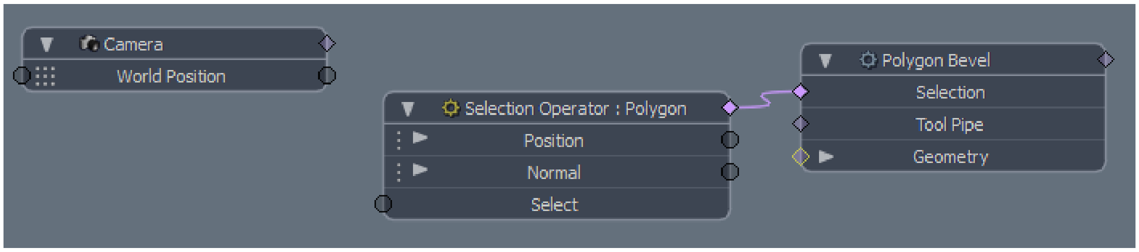
Tip: If a Mesh Operation is selected, **Shift**+**S** opens the **Add Selection Operator** panel. **Shift**+**F** opens the **Add Falloff** panel. `(backtick) opens the mesh operation tree.

5. Add the Selection Operator to the **Schematic** viewport, then double-click its diamond connector to show the connected Polygon Bevel operation.



6. Add the Selection Operator's **Position X**, **Y**, **Z** and **Normal X**, **Y**, **Z** channels to the **Schematic** by dragging them in from the **Channels** list.

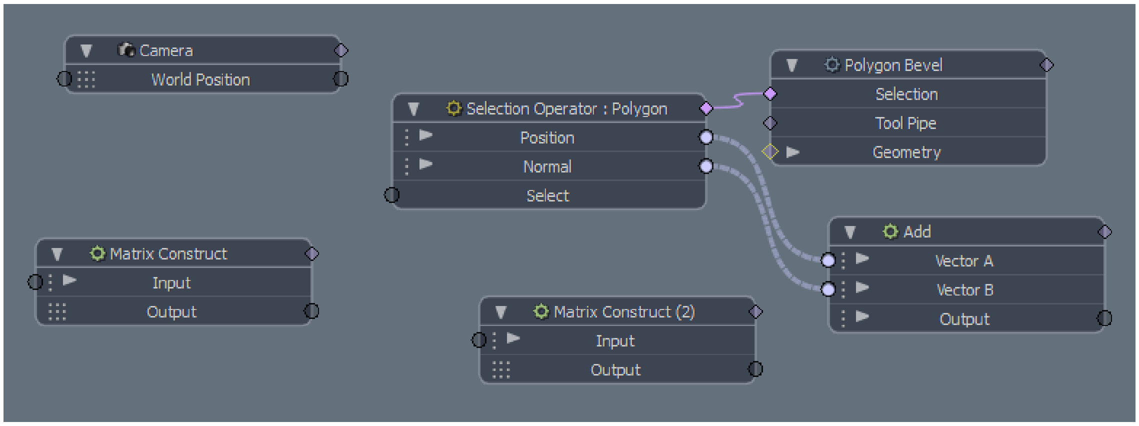
   At this stage, the Schematic viewport looks like this:



Now that we've added the camera, the Polygon Bevel, and the Selection Operator, we have to calculate the angle from the geometry to the camera.

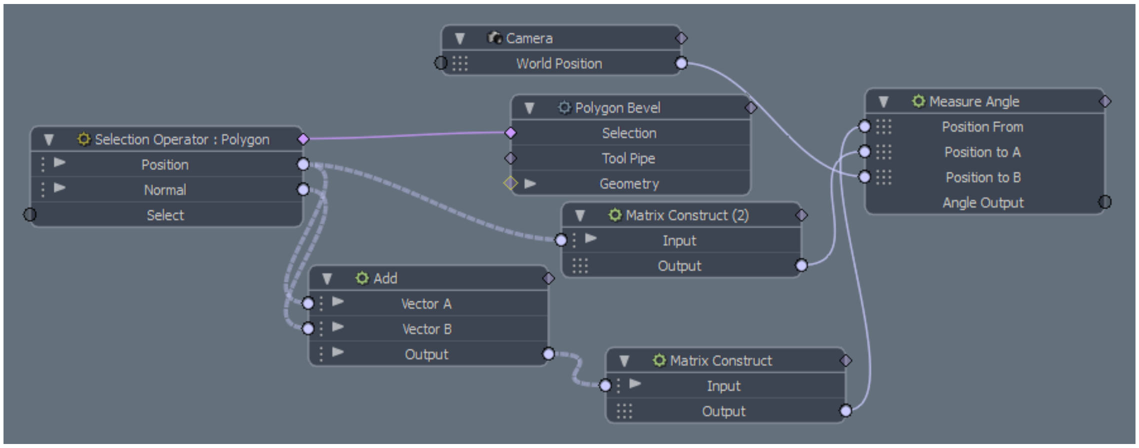
1. In the **Schematic**, add two Matrix Constructs by clicking **Add...** at the top of the viewport, then under **Channel Modifiers** > **Matrix**, double-click **Matrix Construct**.
2. Add the **Vector Maths:Add** node, then connect the **Position** and **Normal** channels of the Selection Operator to the **Vector A** and **Vector B** channels of the Add vector respectively, to add them together.

   You have the following connections in the **Schematic** now:


Camera
World Position
Selection Operator : Polygon
Position
Normal
Select
Polygon Bevel
Selection
Tool Pipe
Geometry
Matrix Construct
Input
Output
Add
Vector A
Vector B
Output
Matrix Construct (2)
Input
Output

3. Connect the Add node's **Output** to the first Matrix Construct's **Input**, then connect the Selection Operator's **Position** channel to the other Matrix Construct's **Input**.
4. Add a Measure Angle channel modifier to the **Schematic**, and make the following connections:
   - First Matrix Construct's **Output** channel to Measure Angle's **Position to A**.
   - Second Matrix Construct's **Output** channel to Measure Angle's **Position From**.
   - Camera's **World Position** channel to **Position to B**.

   The **Schematic** at this point looks like this:

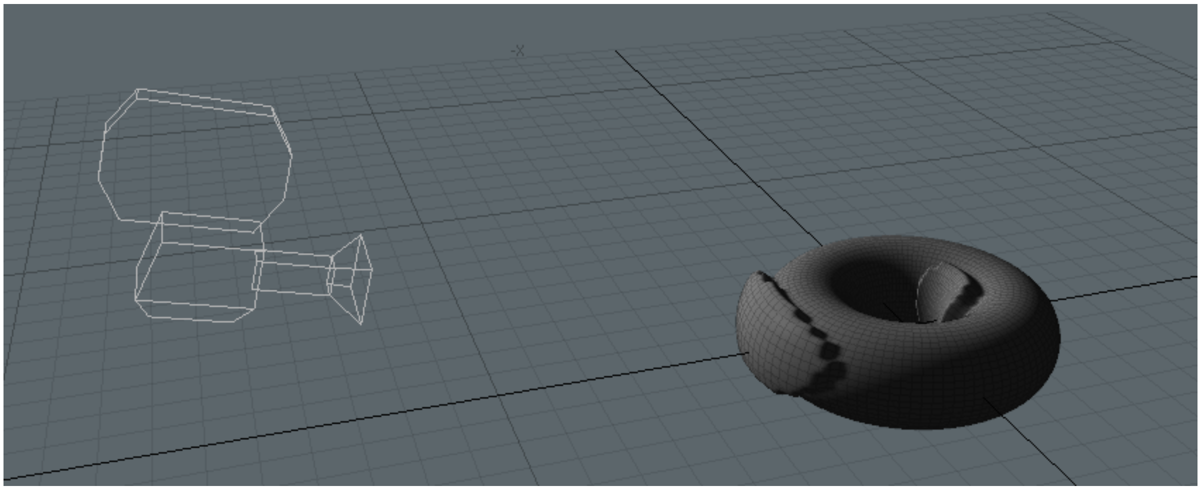

Camera
World Position
Polygon Bevel
Selection
Tool Pipe
Geometry
Measure Angle
Position From
Position to A
Position to B
Angle Output
Selection Operator : Polygon
Position
Normal
Select
Matrix Construct (2)
Input
Output
Add
Vector A
Vector B
Output
Matrix Construct
Input
Output

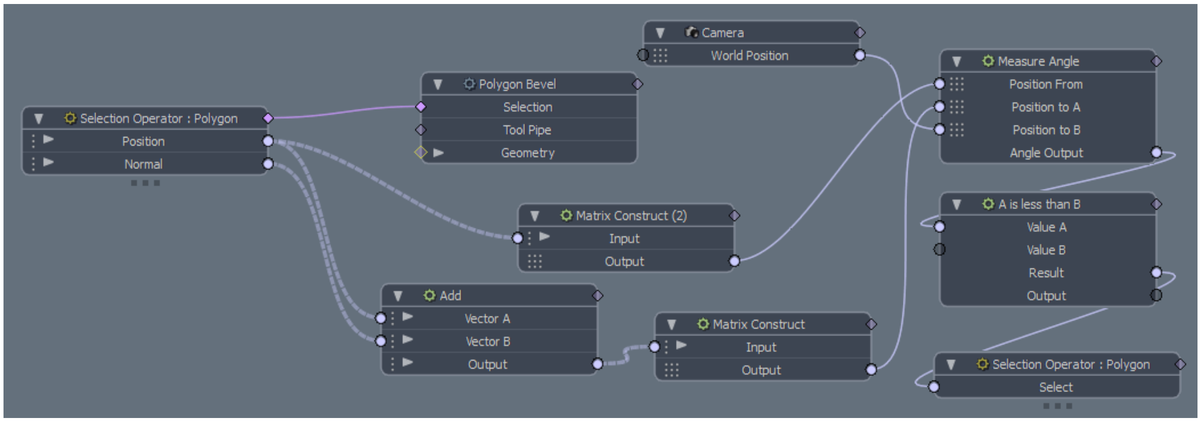
Next, we add a conditional node that we can use to select polygons which have an angle of less than, say, 45 degrees.

1. Optional: To be able to drive the selection, and to make the connections a bit clearer: separate the Selection Operator's **Select** channel by right-clicking on it and clicking **Separate Channel**.
2. Click **Add...** at the top of the **Schematic** viewport, and under **Channel Modifiers** > **Conditional**, double-click **Logic:A is less than B**.
3. Connect the Measure Angle's **Angle Output** to the conditional node's **Value A** channel, and connect the conditional node's **Result** Channel to the Selection Operator's **Select** channel.
4. Select the **A is less than B** conditional node, and in the **Channels** list on the right side of the interface, set **Value B** to 45.

Modo now bevels the polygons that are facing the camera. As you move the camera around, or change the angle value, the affected area changes.

The images below final result in the 3D viewport and the corresponding **Schematic**:




Camera
World Position
Measure Angle
Position From
Position to A
Position to B
Angle Output
Polygon Bevel
Selection
Tool Pipe
Geometry
Selection Operator : Polygon
Position
Normal
Matrix Construct (2)
Input
Output
A is less than B
Value A
Value B
Result
Output
Add
Vector A
Vector B
Output
Matrix Construct
Input
Output
Selection Operator : Polygon
Select

Selection Operator Channels

Name	Type	Description	Minimum	Maximum
Selection Operator: Edge				
Index	Integer	The index of the edge in the mesh.	0	
Position	Vector XYZ	The edge position.		
Normal	Vector XYZ	The edge normal.		

Name	**Type**	**Description**	**Minimum**	**Maximum**
Length	Distance	The distance from one vertex on the edge to the other.	0.0	
Angle	Angle	The angle between the two polygons connected to the edge.	0.0	360.0
Is Border	Boolean	Outputs true if the edge is on the boundary of a polygon island.	False	True
Vertex A	Vector XYZ	Provides the position of the first vertex in the edge.		
Vertex B	Vector XYZ	Provides the position of the second vertex in the edge.		
Selection Operator: Polygon				
Index	Integer	The index of the polygon in the mesh.	0	
Position	Vector XYZ	The centroid position of the polygon.		
Normal	Vector XYZ	The polygon normal.		
Area	Distance	The area of the polygon.	0.0	
Flatness	Percent	The polygon flatness, determined by how coplanar a polygon is.	0.0	1.0
Vertex Count	Integer	Outputs the number of vertices that define the polygon.	2	
Is Border	Boolean	Outputs true if the polygon is on the boundary of a polygon island.	False	True
Selection Operator: Vertex				
Index	Integer	The index of the vertex in the mesh.	0	
Position	Vector XYZ	The vertex position.		

Name	Type	Description	Minimum	Maximum
Normal	Vector XYZ	The vertex normal.		
Edge Count	Integer	The number of edges connected to this vertex.	0	
Is Border	Boolean	Outputs true if the vertex is on the boundary of a polygon island.	False	True
Select	Boolean	Specifies whether the mesh element should be selected or not.	False	True

Modifying Selections

You can use procedural selection modifiers to change your selection, similarly to traditional selection modifiers. They are selection operations that take the previous selection and modify it. You can convert polygon to boundary selection, grow, shrink, or invert your selection.

Note: For more information on traditional selection modifiers, see Using Selection Modifiers.

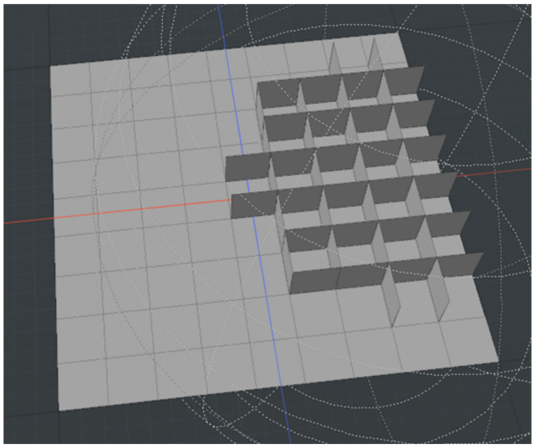
Boundary Edges

This selection modifier converts a polygonal selection into an edge selection by using the boundary of the polygon selection to determine the edges to select.

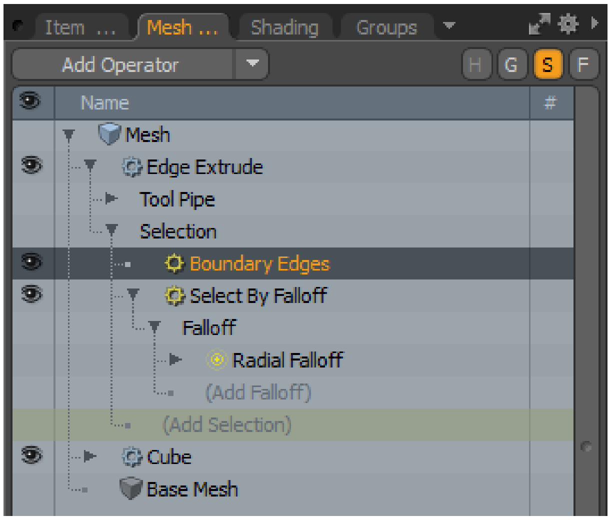
You can add this selection operation above the selection you want to modify in the **Mesh Operations** list:

1. Add a mesh operation that has a selection input, then add a selection operation to it, as described in Procedural Selection.

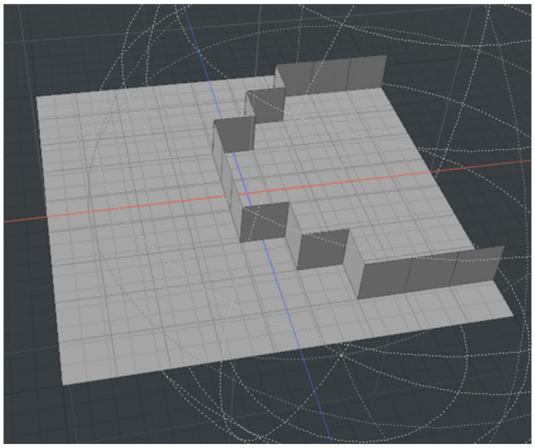
   The image below shows Edge Extrude applied to a plane using a Select By Falloff selection operation with a Radial Falloff.



2. Add a Boundary Edges selection operation to the list, then drag it above the selection you want to modify in the **Mesh Operations** list:


Item ...
Mesh ...
Shading
Groups
Add Operator
H
G
S
F
Name
#
Mesh
Edge Extrude
Tool Pipe
Selection
Boundary Edges
Select By Falloff
Falloff
Radial Falloff
(Add Falloff)
(Add Selection)
Cube
Base Mesh

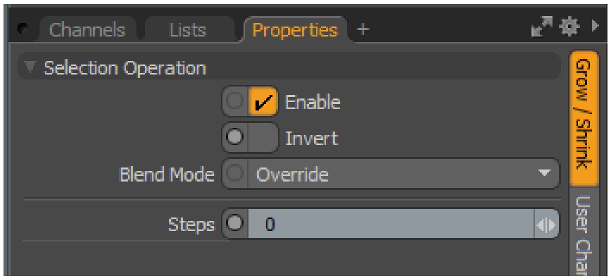
This selects all the edges that were the boundary of the previous selection:



Growing and Shrinking Selections

You can increase or decrease your selection using the Grow/Shrink selection operation.

1. In the **Mesh Operations** list, add the selection above the selection operation you want to modify.
2. In the Grow/Shrink operation's **Properties** panel, increase or decrease the **Steps** value to grow or shrink your selection, respectively.



Selection Operation

☑ Enable

☐ Invert

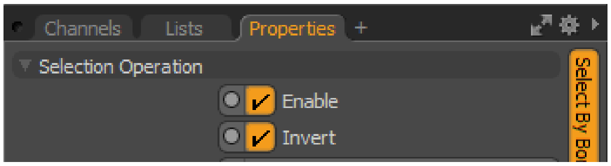
Blend Mode Override

Steps 0

Inverting a Selection

Modo allows you to invert a single selection operation, or all selection operations in the list.

To reverse a single selection operation, in the selection operation's **Properties** panel, enable the **Invert** checkbox.



Selection Operation

☑ Enable

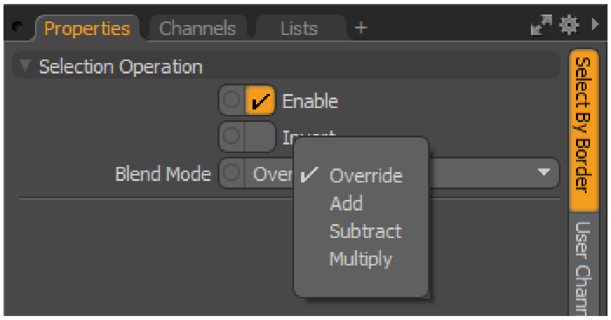
☑ Invert

To reverse a list of operations, use the Invert selection modifier. Place it above the selections you want to invert in the **Mesh Operations** list.

Using Multiple Selection Operations

You can add multiple selection operations to a mesh operation and specify how Modo should evaluate them. When you add a selection operation to the **Mesh Operations** list, its properties are displayed in the **Properties** panel under the list.

Blend Mode allows you to specify how Modo treats the preceding selections.


Properties
Channels
Lists
+
Select By Border
User Chann

Selection Operation

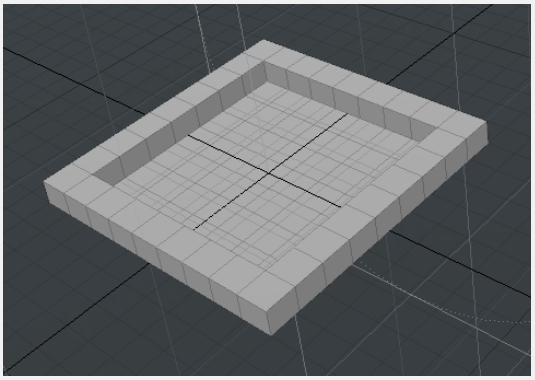
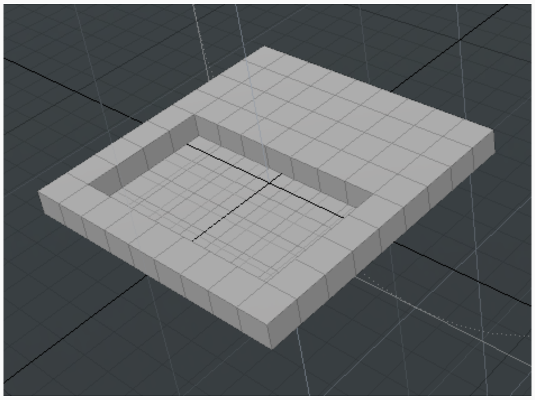
☑ Enable

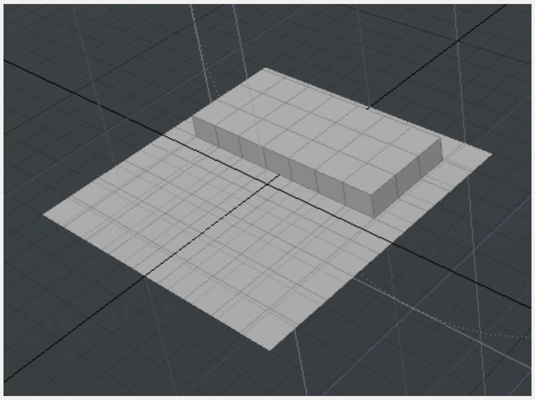
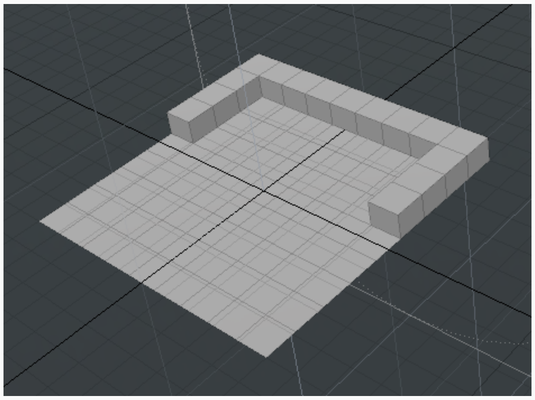
☐ Invert

Blend Mode Override

✓ Override
Add
Subtract
Multiply

Adding a Select by Falloff (using a Linear Falloff) and a Select by Border selection operation produces the following results:

Blend Modes		
	Override	The default behavior. The current selection overrides all selection operations below it in the list. Previous selections are discarded, and the new selection is used instead.
	Add	Adds the new selection to the previous selection. Elements are treated as de-selected if they are de-selected by all selection operations in the list.

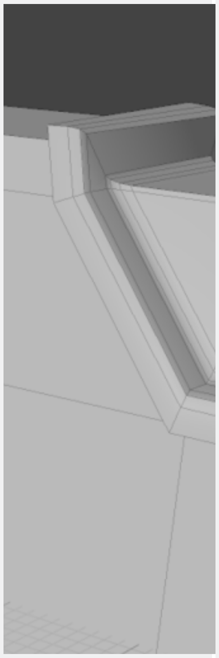
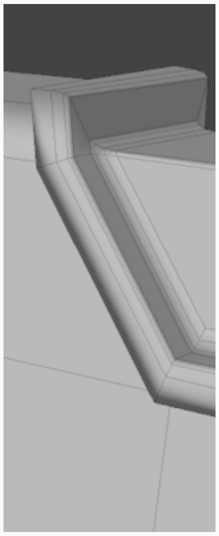
Blend Modes		
	Subtract	Removes any elements that are selected by the selection operation from the previous selection.
	Multiply	Only selects elements from the new selection operation that are also selected by the first, filtering the previous selection.

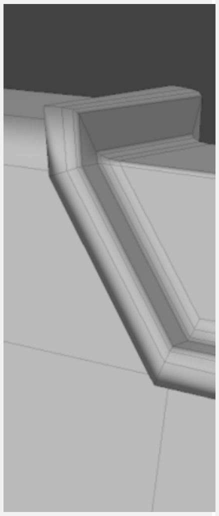
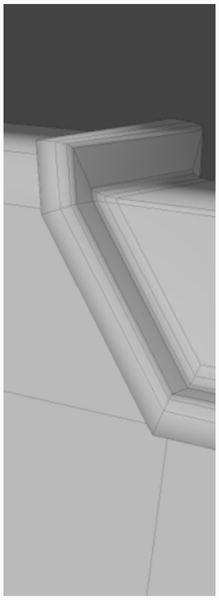
Using The Set Vertex Normals MeshOp to Adjust the Appearance of Your Mesh

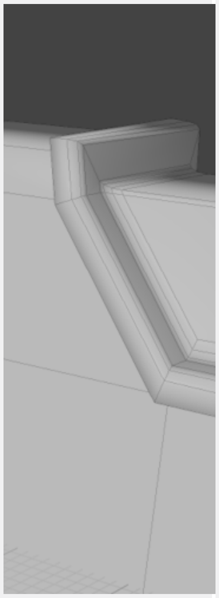
You can set and adjust the edge smoothness of a mesh and give the appearance of a high poly, subdivided mesh using the **Set Vertex Normals** MeshOp.

> **Note:** For more information on Vertex Normals, and Modo's direct modeling Vertex Normal toolset see Vertex Normals Tools.

There are five methods of smoothing edges available within the **Set Vertex Normals** MeshOp:

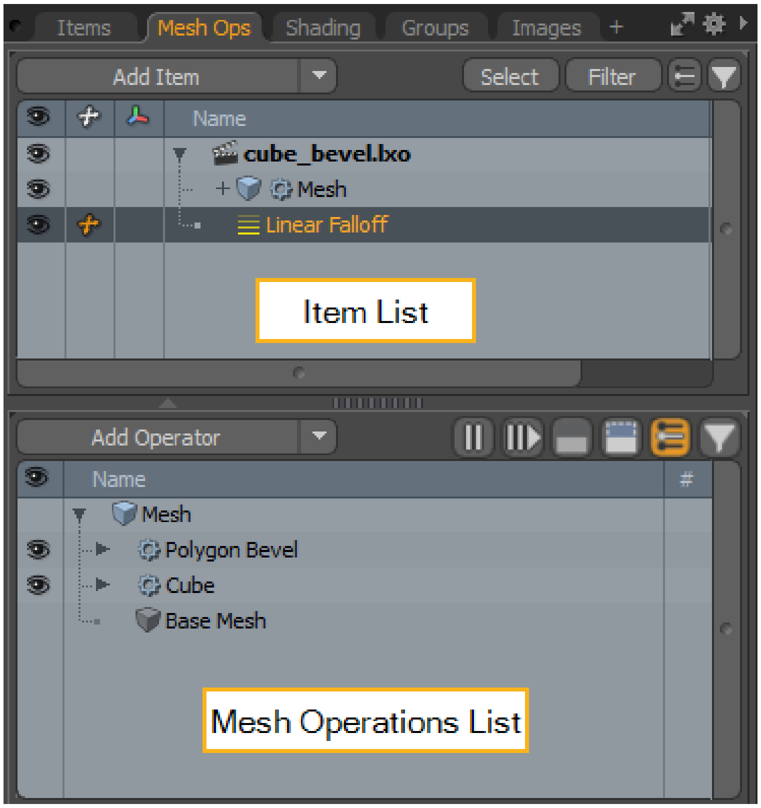
Method		
Standard		Vertex normals are baked from the Viewport. **Note:** Changes to Material Smoothing do not automatically refresh, so the MeshOp must be toggled to reflect changes in smoothing.
Largest		Vertex normals are determined by the largest polys on your mesh.

Method		
Biased		Uses the power value to control the weight given to each polygon. This means that the higher the power, the more weight is given to larger polys.
Falloff		Uses the polygon's falloff value to determine the weight that the polygon has in its area weighting calculation.

Method		
Color Map		Uses the red channel of a supplied color map to drive the weight value for that polygon. **Note:** To use this parameter, create a Vertex Color Map and assign a red color to the polys you want to use to drive the smoothing. For more information on Vertex Color Maps and the Color Tool, see Vertex Color Tool.

Using the Mesh Ops Tab

The procedural modeling tools are available in the **Model** layout, in the **Mesh Ops** tab. You can find it on the right-side of the interface, next to the **Item List**.


Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
Name
cube_bevel.lxo
Mesh
Linear Falloff
Item List
Add Operator
Name
#
Mesh
Polygon Bevel
Cube
Base Mesh
Mesh Operations List

The tab consists of two lists. You can find the **Item List** on top, and the **Mesh Operations** list below it.

Note: For more information on the **Item List**, see Items (Scenes) List.

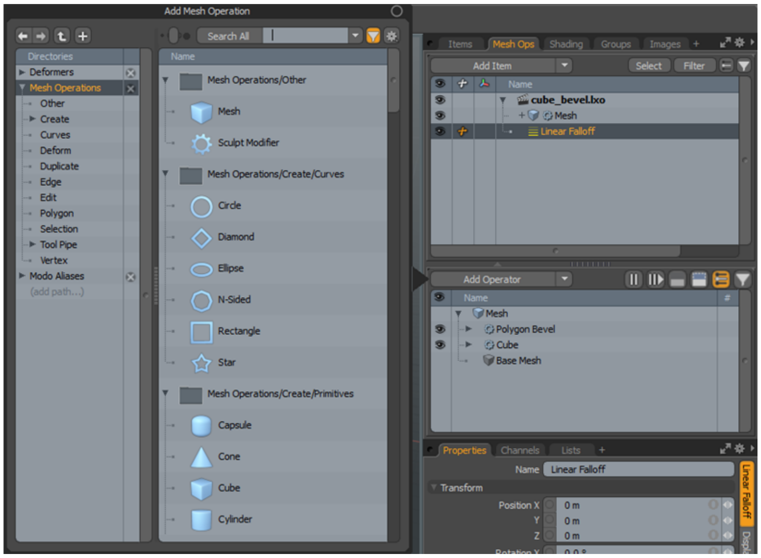
Mesh Operations List

The **Mesh Operations** list contains a list of the mesh operations in your scene. These can be primitives, like a Cube or a Cylinder, or they can be tools, like the Smooth or the Bevel tool. The order of evaluation is determined by the order of mesh operations. Modo evaluates mesh operations at the bottom first and proceeds upwards in the list.

You can add operations to the list by clicking the **Add Operator** button above it or if a procedural Item is selected, press **tab** to open the **Add Operator** panel. This opens the Procedural Preset Browser, where you can find various operations organized into categories: you can add deformers, mesh operations, and assemblies. You can use the scale slider to change the size of the icons. The category headers have folder icons and the tools have their individual icons. You can find a subset of the procedural versions of the traditional modeling tools in this menu.

Alternatively, you can click the small arrow next to the **Add Operator** button to reveal a dropdown menu of the available items.

Tip: You can pin the **Preset Browser** to your workspace by dragging it to a different place in the interface, or clicking the small circle in the top-right corner. To unpin, click the circle in the right-hand corner.


Add Mesh Operation
Search All
Directories
Deformers
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Edit
Polygon
Selection
Tool Pipe
Vertex
Modo Aliases
(add path...)
Name
Mesh Operations/Other
Mesh
Sculpt Modifier
Mesh Operations/Create/Curves
Circle
Diamond
Ellipse
N-Sided
Rectangle
Star
Mesh Operations/Create/Primitives
Capsule
Cone
Cube
Cylinder
Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
cube_bevel.lxo
Mesh
Linear Falloff
Add Operator
Mesh
Polygon Bevel
Cube
Base Mesh
Properties
Channels
Lists
Name
Linear Falloff
Transform
Position X
Y
Z
0 m
0 m
0 m
Rotation X

For example, you can add a cube the following way:

1. Select the base mesh in the **Mesh Operations** list.
2. Click the **Add Operator** button and navigate to **Mesh Operations** > **Create** > **Primitives**.

Tip: Alternatively, you can search for the required mesh operation by typing its name in the search bar at the top of the window. You can click the button on the left side of the search bar to adjust your search. **Search All** searches all mesh operations, and **Search View** searches through the currently-visible operations.

3. Double-click the **Cube** mesh operation from the list on the right of the panel.

   The cube primitive is added to your **Mesh Operations** list and shows up in the 3D viewport. You can then edit the properties of the cube in the **Properties** panel below the **Mesh Operations** list.

Tip: When selecting a mesh operation, the **meshop.haul** tool is activated. This is similar to the channel haul tool (C), except it only uses the hauling assignments from the mesh operation directly, and does not draw the UI. In practice this means that you can haul as soon as you add a mesh operation, rather than having to activate channel haul.

Creating instances of items in the Mesh Ops Tab

You can quickly create an instance from a selected item in the **Mesh Ops** tab. The instance is only relevant for the selected Items. The result is a duplicate of the selected item that is simply a reference to the original item. Instances can be transformed at the item level and can have their own materials and textures applied to them using item-level masks. However, they can not be edited at the instance-level, as they derive all information from the original Item. The advantage is that all item changes made to the original Item propagate to its instances.

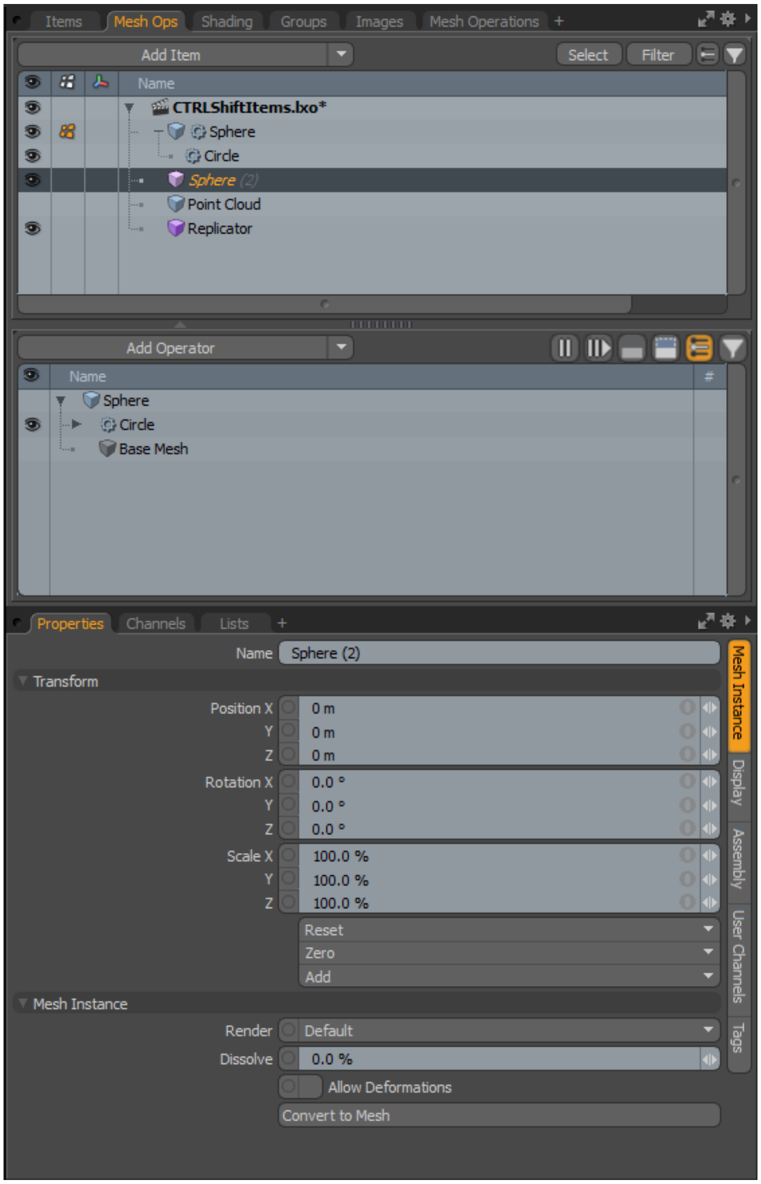
You can make an instance of all items in the **Mesh Ops** tab.

Tip: You can also use the **Ctrl/Cmd** + **Shift** keys to create instances in the Mesh Operations List and add keyframes in the Graph Editor.

To create an instance:

1. In the **Mesh Ops** tab, press **Ctrl/Cmd** + **Shift**, click on a item, and drag the item to anywhere in the **Mesh Ops** tree.

An instance is created and the properties are displayed in the **Properties**.


Items
Mesh Ops
Shading
Groups
Images
Mesh Operations
Add Item
Select
Filter
Name
CTRLShiftItems.lxo*
Sphere
Circle
Sphere (2)
Point Cloud
Replicator
Add Operator
Name
Sphere
Circle
Base Mesh
Properties
Channels
Lists
Mesh Instance
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Mesh Instance
Allow Deformations
Convert to Mesh

Name Sphere (2)

Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

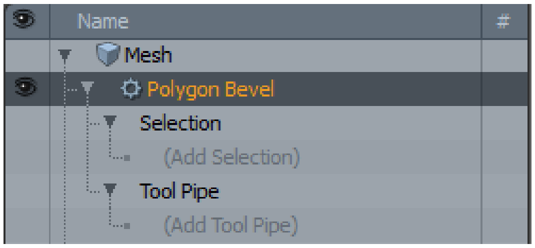
Render Default
Dissolve 0.0 %

Mesh Operation Inputs

Once a mesh operation is added to your list, you can add other operations to it to modify its behavior. The items you can add depend on the type of mesh operation. The most common inputs are Tool Pipe and Selection. For example, the Cube primitive only has a Tool Pipe input, but a Polygon Bevel also has a Selection input. A Curve Rebuild tool, on the other hand, calls for a Curve input.

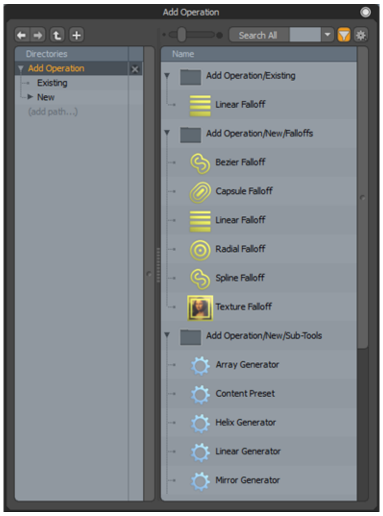
You can reveal a mesh operation's inputs by clicking the small arrow in front of the operation in the **Mesh Operations** list.

The image below shows the Selection and Tool Pipe inputs of the Polygon Bevel.


Name
#
Mesh
Polygon Bevel
Selection
(Add Selection)
Tool Pipe
(Add Tool Pipe)

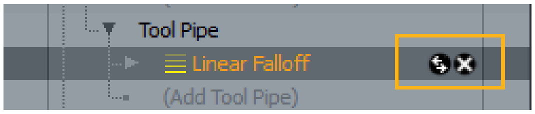
Clicking **(Add Selection)** lets you add a selection operation. For more information on procedural selection, see Procedural Selection.

Clicking **(Add Tool Pipe)** opens a Preset Browser, where you can select mesh operation modifiers to add to your scene.


Add Operation
Search All
Directories
Add Operation
Existing
New
(add path...)
Name
Add Operation/Existing
Linear Falloff
Add Operation/New/Falloffs
Bezier Falloff
Capsule Falloff
Linear Falloff
Radial Falloff
Spline Falloff
Texture Falloff
Add Operation/New/Sub-Tools
Array Generator
Content Preset
Helix Generator
Linear Generator
Mirror Generator

The panel has two categories: the items you've already added to your scene can be found in the **Existing** category, while available and compatible items are under **New**. Once you add any item under **New** to your scene, it appears under **Existing**. To add an item to the list, double-click it in the list.

The modifier appears in the selected mesh operation's Tool Pipe. By hovering the mouse over the right side of the field, two additional controls appear:

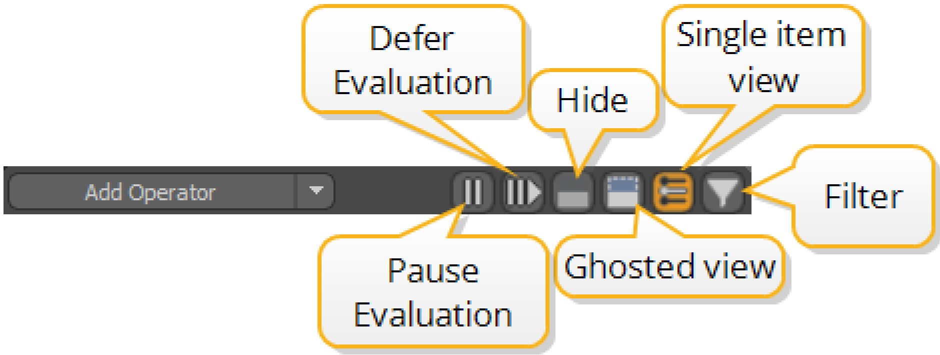

Tool Pipe
Linear Falloff
(Add Tool Pipe)

Clicking the **x** icon removes the item from the list. Clicking the arrows opens the **Add** panel, and you can change the existing modifier to a different one.

Note: Modo displays a Modifier Dependency Loop Warning when you try to add a tool or mesh operation to its own input.

Deferring, Ghosting, Single Item View, Filtering

In the top right corner of the list, you can find four buttons to control how the **Mesh Operations** list displays information.



Deferring and Pausing Evaluation

The evaluation of a mesh's deformers and MeshOps can be deferred or paused to improve UI interaction. This can be controlled globally, or on a per-mesh item basis. By default, all meshes behave according to the global settings available in the MeshOps list, but any non-default setting in the mesh item's properties override the global options.

See Deferring and Pausing Evaluation for more information.

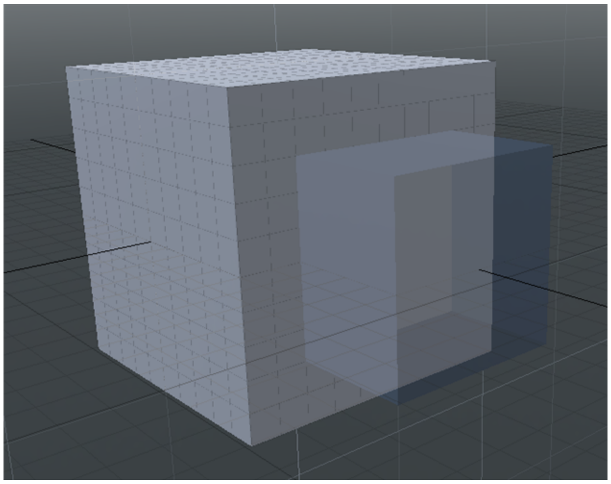
Ghosted View

Ghosting is automatically activated when selecting a base mesh in the **Items** list. Alternatively, clicking the Ghosted view button enables ghosting. In ghosted view, when you select a mesh operation, Modo evaluates and displays the list up until that operation, and displays a ghosted view of the whole list over the top. By default, the mesh appears shaded blue, but you can customize the look of the ghosted view in **System** > **Preferences** > **Defaults** > **Procedural Modeling** > **Ghosted Mesh Drawing**.

In addition, in the **Item List**, right-click on a mesh and select **Edit Base Mesh** from the pop-up contextual menu. This command allows you to quickly edit the base mesh.

Note: For more information on ghosted mesh properties, see Procedural Modeling.

The image below shows a Cube with a Polygon Bevel operation. Selecting the Cube in the **Mesh Operations** list with ghosting enabled, the Polygon Bevel is displayed with blue shading.



Modo also allows you to hide the non-evaluated parts completely. You can do this by clicking the Hide button, or you can set the **Drawing Style** to **Hide** in the **Procedural Modeling** preferences.

Note: Ghosted View and **Hide** are active by default. Select the base mesh element of a MeshOp stack and Ghosted View and Hide become active for the entire stack, so any time you select another item in the mesh operations list below the top item, a ghost view preview of the operations higher up is visible.

Solo Mode

By default, you are viewing the **Mesh Operations** list in single item view. This means that only the deformers and mesh operations applied to the selected mesh layer are displayed in the list. You can toggle this view by clicking the Single item view button.

Filtering

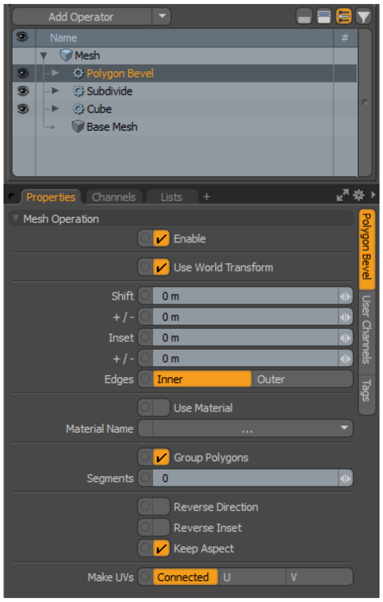
When having a long list of items, you can search through them by clicking the Filter button. Clicking the button reveals a search bar, where you can type in the name of the required operation.

Procedural Mesh Operation Properties

Procedural mesh operations also have a **Properties** panel, but as opposed to traditional tool properties, these are located on the lower-right of the interface, below the **Mesh Operations** list.

To display the properties of an operation, select it in the **Mesh Operations** list.

The image below shows the **Properties** panel of the Polygon Bevel:



Mesh Operation

☑ Enable

☑ Use World Transform

Shift: 0 m

+ / -: 0 m

Inset: 0 m

+ / -: 0 m

Edges: Inner | Outer

☐ Use Material

Material Name: ...

☑ Group Polygons

Segments: 0

☐ Reverse Direction

☐ Reverse Inset

☑ Keep Aspect

Make UVs: Connected | U | V

Here you can manually edit an operation's properties and see the updated results in the 3D viewport.

Mesh Operation Context Sensitive Shortcuts

If a mesh operation is selected:

Shift+S opens the Add Selection Operator panel

Shift+F opens the Add Falloff panel

Backtick (`) opens the mesh operation tree

Creating Geometry

Modo offers a variety of different tools for quickly generating geometry. Primitive items, such as spheres and cubes, are a good way to build basic forms, which can then be edited, added to, modified, and otherwise molded into the form you want, using any of the deform or mesh editing tools. Drawn curves are useful for creating wireframe skeletons of a form, where patches are then used to define the final surface. Functions, such as **Solid Sketch**, allow you to explore forms more readily without needing to manipulate specific geometry.

Primitives		
	Cube	The **Cube** primitive provides a simple method for creating boxes or cubes. **Tip:** The **Cube** primitive tool can also be used to create 2D quadrilaterals by clicking and dragging in the viewport.
	Sphere	The **Sphere** primitive provides a simple method for creating sphere or balls.
	Cone	The **Cone** primitive provides a simple method for creating conical shapes.
	Cylinder	The **Cylinder** primitive provides a simple method for creating cylinders or tubes.
	Capsule	The **Capsule** primitive provides a simple method for creating capsules or rounded-edge tubes.
	Torus	The **Torus** primitive provides a simple method for creating toroids or "doughnuts".
	Ellipsoid	The **Ellipsoid** primitive provides a simple method for creating ellipses or "pills".
	Tube	The **Tube** primitive provides a simple method for creating spline curve-based tubes.
	Teapot	The **Teapot** primitive provides a simple method for creating a teapot.

Drawing		
	Pen	The **Pen** tool allows you to create geometry vertex by vertex. You can use the **Pen** tool to create polygons, lines, vertices, spline patches, or subdivision

Drawing		
		surface polygons.
	Adding Curves	The **Curve** tool allows you to create a spline curve. This is a hermite curve type and can be very useful for creating modeling guides, quickly roughing out a shape, or setting up an outline for lofting with the extrude, lathe, or other multiply tools.
	Bezier	The **Bezier** tool allows you to create a type of curve defined by off-curve control handles.
	B-Spline	The **B-Spline tool** allows you to draw B-spline curves, which are spline curves defined by at least four vertices and a weight map.
	Sketch	The **Sketch** tool allows you to quickly create a curve by drawing in the 3D view.
	Arc	The **Arc** tool allows you to make closed segments of a disc (like the shape of a pie wedge, or a pie with a piece missing).

Curves		
○	Circle	The **Circle** tool allows you to create a circle using a Bezier or Hermite spline curve.
☆	Star	The **Star** tool allows you to create a star shape using a Bezier curve or a polyline.
⬡	N-Sided	The **N-Sided** tool allows you to create a regular polygon with a give number of sides. it can be drawn using a Bezier curve or a polyline.
□	Rectangle	The **Rectangle** tool allows you to create a rectangle or square, using a Bezier curve or a polyline.
⬭	Ellipse	The **Ellipse** tool allows you to create an ellipse using a Bezier or Hermite spline curve.
◇	Diamond	The **Diamond** tool allows you to create a diamond shape using a Bezier curve or a polyline.

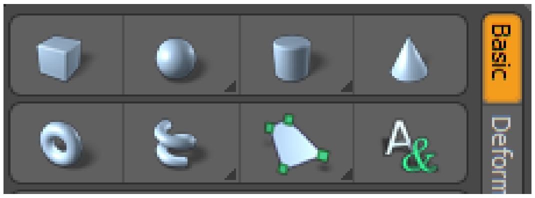
Other		
	Solid Sketch	The **Solid Sketch** tool provides a quick method for roughing out organic shapes - building up basic geometry.
A&	Text	The **Text** tool simplifies the process of creating 3D text using system fonts.
	Vertex	The **Vertex** tool provides a method for quickly plotting points in 3D space.
	Make Polygon	The **Make Polygon** tool uses the currently-selected vertices to create a polygon. Alternatively, use the Make tool to automatically create a polygon from two selected edges.
	Patch Curves	The **Patch Curves** tool generates a smooth surface from three or four curves that define the contour edges of the surfaces.

Adding Primitives

Various geometric primitives available in Modo.

Adding Primitives to Your Scene

You can find the Primitive tools in the **Basic** sub-tab of the **Modeling** toolbox. The toolbox is available in most other layouts by clicking the **Model** button in the top left corner of the layout. You can also press **F2** to open the toolbox.



To activate a tool, click the appropriate icon. You can then place the primitive in your scene in a number of ways:

- Click and drag in the 3D viewport to create the initial plane for the primitive, then adjust the dimensions using the tool handles in the 3D viewport.

- To drag out a uniform-sized primitive, hold **Ctrl** while dragging in the 3D viewport.
- Click the primitive's icon to activate the tool. In the item's **Properties** tab, edit the properties, then click the **Apply** button at the bottom of the panel.
- **Ctrl**+click a primitive's icon to place a unit primitive in your existing mesh layer.
- **Shift**+click a primitive's icon to create a unit primitive in a new mesh layer.

Note: You can also add the same primitives from the menu bar, by clicking **Item** > **Primitive**.

Adding Procedural Primitives

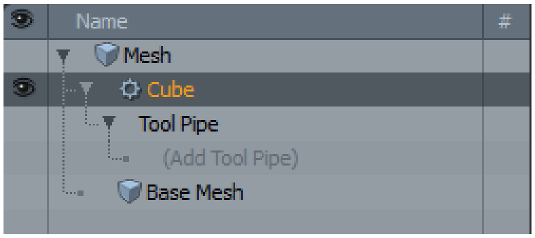
As opposed to traditional primitives, using procedural primitives and modeling tools allows you to make non-destructive changes to your mesh.

Procedural primitives are available in the **Model** layout's **Mesh Ops** tab on the right-side of the interface. You can see the list of your procedural operations in the **Mesh Operations** list, in the lower part of the panel.

Note: For more information on procedural modeling, see Procedural Modeling with MeshOps.

To add a primitive in the **Mesh Ops** tab, click the **Add Operator** button above the **Mesh Operations** list. The **Preset Browser** opens, where you can select the required primitive under **Mesh Operations** > **Create** > **Primitives**.

Primitives have a **Tool Pipe** input, to which you can add deformers and sub-tools. The image below shows a Cube in the **Mesh Operations** list. You can add operations to it by clicking **(Add Tool Pipe).**



Note: For details on the **Mesh Ops** tab, see Using the Mesh Ops Tab.

Cube

The **Cube** primitive provides a simple method for creating boxes or cubes.

To activate a tool, click the Cube icon. You can then place the primitive in your scene in a number of ways:

• Click and drag in the 3D viewport to create the initial plane for the primitive, then adjust the dimensions using the tool handles in the 3D viewport.

• To drag out a uniform-sized primitive, hold **Ctrl** while dragging in the 3D viewport.

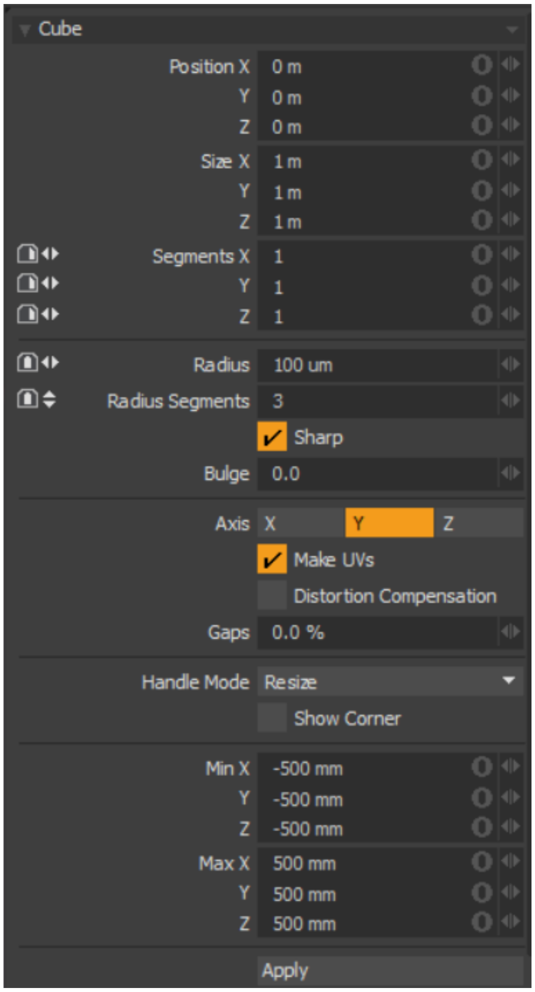
• Click the primitive's icon to activate the tool. In the item's **Properties** tab, edit the properties, then click the **Apply** button at the bottom of the panel.

• **Ctrl**+click a primitive's icon to place a unit primitive in your existing mesh layer.

• **Shift**+click a primitive's icon to create a unit primitive in a new mesh layer.

Note: You can also add the same primitives from the menu bar, by clicking **Item** > **Primitive** > **Cube**.

Cube Properties



Position X 0 m
Y 0 m
Z 0 m
Size X 1 m
Y 1 m
Z 1 m
Segments X 1
Y 1
Z 1
Radius 100 um
Radius Segments 3
☑ Sharp
Bulge 0.0
Axis X [✓ Y] Z
☑ Make UVs
☐ Distortion Compensation
Gaps 0.0 %
Handle Mode Resize
☐ Show Corner
Min X -500 mm
Y -500 mm
Z -500 mm
Max X 500 mm
Y 500 mm
Z 500 mm

Cube	
Position X, **Y**, **Z**	These three values establish the 3D location for the center of the cube.
Size X, **Y**, **Z**	These three values are used to establish the dimensions of the cube. Set to **1 m**, **1 m**, **1 m** (by default), this option provides us with a 1-meter cube. If you want to create a 6-foot plank that is 2 inches by 4 inches, you can type into the **X**, **Y**, and **Z** numeric fields **6' 2" 4"**,respectively, and Modo performs the measurement translation to the current unit mode (Metric, SI, or English).
Segments X, **Y**, **Z**	By default, the primitive box uses a single segment on each axis. By increasing this value, you can increase the number of "cuts" in each side of the box. This

Cube	
	is useful if you plan to deform the cube, as these segments act as hinges during deformation operations.
Radius	This distance value sets a curved edge amount. When radius is set to a positive non-zero value, the edge of the box is rounded off using extra segments, added to the geometry. The number of segments is determined by two additional settings: **Radius Segments** and **Sharp**.
Radius Segments	When the **Radius** value is set to a positive, non-zero value, additional segments are added to round the cube. The **Radius Segments** field allows you to set how many edges are added to smooth out the corners.
Sharp	This adds an extra set of polygons to the edge of the rounded edge so that the adjacent faces are not affected by surface smoothing. This allows the surface material to use smoothing to further round the edges without losing the flat appearance of the faces of the cube.
Bulge	Produces a swelling effect on the cube.
Axis	Defines the major axis for the cubes orientation. On the cube this only affects the direction of **Radius Segments**.
Make UVs	When this button is active a UV map is automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive, as it provides a baseline UV map that you can massage later in the modeling process. In many cases, this can reduce the amount of work required to map the model.
Distortion Compensation	Adjusts the aspect ratio and related area size of UV polygons to reduce the distortion in the primitives UVs. Use this option when you create a **Cube** that does not have equal radius values. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**. Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final

Cube	
	values.
Gaps	Specifies empty space at the boundary of the UV space or between UV parts. The UV unit is specified using a percentage value.
Handle Mode	Provides three types of behavior for corner and side tool handles: • **Resize from Center** - Resizes both sides of the primitive at the center, when dragging a corner or side handle. • **Resize** - Resizes the shape at the opposite side, when dragging a corner or side handle. • **Move All** - Moves the primitive instead of resizing. **Tip:** You can also move the shape by clicking and holding the **Ctrl** key when in another handle mode.
Show Corner	When enabled, displays the bounding box corner handles.
Min X, **Y**, **Z**/**Max X**, **Y**, **Z**	You can also define a cube, based on specific X, Y, and Z bounding box locations in 3D space, which can be specified here. This makes it easy to place a cube on the ground plane for instance by making the **Min Y** value **0** and the **Max Y** value **1m**.

Tip: The **Box** primitive supports symmetrical creation. When Applying Precision is activated, creating the primitive shape creates an identical version across the specified axis.

Sphere

The **Sphere** primitive provides a simple method for creating spheres or balls.

Note: For information on how to add primitives to your scene, see Adding Primitives.

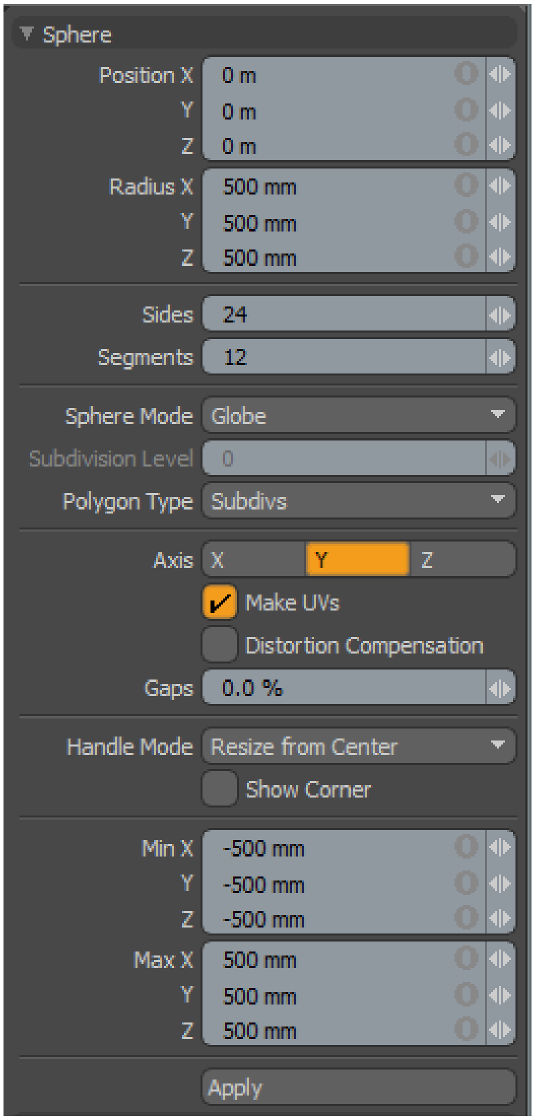
To activate a tool, click the Sphere icon. You can then place the primitive in your scene in a number of ways:

- Click and drag in the 3D viewport to create the initial plane for the primitive, then adjust the dimensions using the tool handles in the 3D viewport.
- To drag out a uniform-sized primitive, hold **Ctrl** while dragging in the 3D viewport.
- Click the primitive's icon to activate the tool. In the item's **Properties** tab, edit the properties, then click the **Apply** button at the bottom of the panel.
- **Ctrl**+click a primitive's icon to place a unit primitive in your existing mesh layer.
- **Shift**+click a primitive's icon to create a unit primitive in a new mesh layer.

Note: You can also add the same primitives from the menu bar, by clicking **Item** > **Primitive** > **Sphere**.

Sphere Properties

When the Sphere tool is active, the following options are available for editing the primitive:



Sphere

Position X 0 m
Y 0 m
Z 0 m
Radius X 500 mm
Y 500 mm
Z 500 mm
Sides 24
Segments 12
Sphere Mode Globe
Subdivision Level 0
Polygon Type Subdivs
Axis X Y Z
☑ Make UVs
☐ Distortion Compensation
Gaps 0.0 %
Handle Mode Resize from Center
☐ Show Corner
Min X -500 mm
Y -500 mm
Z -500 mm
Max X 500 mm
Y 500 mm
Z 500 mm

Position X, Y, Z	These three values establish the 3D location for the center of the ball.
Radius X, Y, Z	These three values are used to establish the dimensions of the sphere. Set to **1m**, **1m**, **1m** (by default), provides us with a 2-meter ball.
Sides	The sides are the latitudinal edges of the sphere running from top to bottom. By default the primitive ball uses 24 sides. By increasing this value you can increase the number of "cuts" in the ball. This is useful if you plan to deform the ball as these segments act as hinges during deformation operations. This also increases the smoothness of the sphere by adding extra facets. If you plan to apply subdivision surfaces to the mesh, you can use a lower number of segments as the subdivision smooths out the shape.

Segments	The segments are the longitudinal edges of the sphere running from right to left. By default the primitive ball uses 12 segments. By increasing this value you can increase the number of "cuts" in the ball. This is useful if you plan to deform the ball as these segments act as hinges during deformation operations. It also increases the smoothness of the sphere by adding extra facets. If you plan to apply subdivision surfaces to the mesh, you can use a lower number of segments as the subdivision smooths out the shape.
Sphere Mode	Options for various **Sphere** types. • **Globe** - the default setting creates a ball, with the controls described above. • **Quadball** - this option creates an SDS sphere made entirely of polygons containing only four vertices (quads). When this option is active the **Sides** and **Segments** settings are disabled and the **Subdivision Level** control activates. Increasing this value adds complexity to the quadball primitive surface. • **Tesselated** - by default the ball primitive is composed of mostly-quad polygons with some triangles at the poles. Activating the **Tesselated** flag creates the sphere from all triangles. When this is active the **Tesselation Level** sets the resolution of the mesh created. Using a tesselated sphere with a high tesselation level creates a more perfect sphere than the default sides and segments, but with many more polygons. Tesselated spheres are not recommended if you plan to use subdivision surfaces.
Subdivision Level	For the **Quadball** and **Tesselated** sphere mode options, the **Subdivision Level** sets the number of subdivisions, which determines the number of polygons used to define the overall shape. Larger numbers produce more polygons that are rounder with a smoother surface.
Polygon Type	You can use this option to determine the type of polygons created: • **Face** - creates standard unsmoothed polygon faces. • **Subdivs** - creates Subdivision Surface smoothed polygons. • **Catmull-Clark** - creates Pixar Catmull-Clark smoothed polygons.
Axis	Defines the major axis direction, most useful for a **Globe** type sphere, as it controls the direction of the segments along the radius.
Make UVs	When this button is active a UV map is automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV

	map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.
Distortion Compensation	Adjusts the aspect ratio and related area size of UV polygons to reduce the distortion in the primitives UVs. Use this option when you create a **Sphere** that does not have equal radius values. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**. Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final values.
Gaps	Specifies empty space at the boundary of the UV space or between UV parts. The UV unit is specified using a percentage value.
Handle Mode	Provides three types of behavior for corner and side tool handles: • **Resize from Center** - Resizes both sides of the primitive at the center, when dragging a corner or side handle. • **Resize** - Resizes the shape at the opposite side, when dragging a corner or side handle. • **Move All** - Moves the primitive instead of resizing. **Tip:** You can also move the shape by clicking and holding the **Ctrl** key when in another handle mode.
Show Corner	When enabled, displays the bounding box corner handles.
Min X, **Y**, **Z/Max X**, **Y**, **Z**	You may also define a sphere based on specific X, Y, and Z bounding box locations in 3D space, which can be specified here. This makes it easy to place a sphere on the ground plane, for instance, by making the **Min Y** value **0** and the **Max Y** value **1m**.

Tip: The **Sphere** primitive supports symmetrical creation. When Applying Precision is activated, creating the primitive shape creates an identical version across the specified axis.

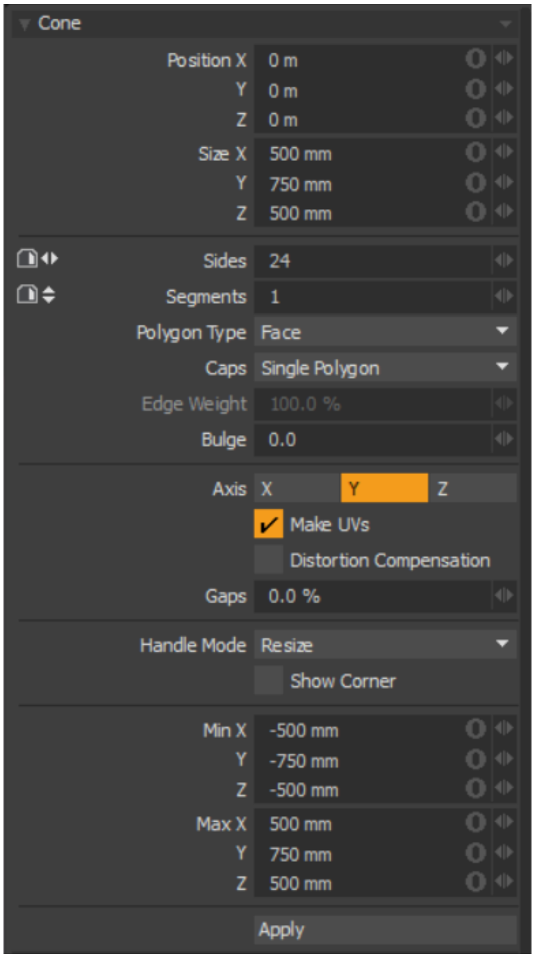
Cone

The **Cone** primitive provides a simple method for creating conical shapes.

Note: For information on how to add primitives to your scene, see Adding Primitives.

When the Cone tool is active, the following options are available for editing the primitive:

Note: The properties allow you to generate subdivided, quad polygon cones that can be used in MeshFusion.



Cone

Position X 0 m
Y 0 m
Z 0 m
Size X 500 mm
Y 750 mm
Z 500 mm
Sides 24
Segments 1
Polygon Type Face
Caps Single Polygon
Edge Weight 100.0 %
Bulge 0.0
Axis X Y Z
☑ Make UVs
☐ Distortion Compensation
Gaps 0.0 %
Handle Mode Resize
☐ Show Corner
Min X -500 mm
Y -750 mm
Z -500 mm
Max X 500 mm
Y 750 mm
Z 500 mm

Cone	
Position X, **Y**, **Z**	These three values establish the 3D location for the center of the cone.
Size X, **Y**, **Z**	These three values are used to establish the dimensions of the cone. Set to **50 cm**, **75 cm**, **50 cm**, by default. As a radius they produce a cone that is .75 meter tall and 1-meter across at the base.
Sides	The sides are the latitudinal edges of the cone running from top to bottom. By default, the primitive cone uses 24 sides. By increasing this value you can increase the number of "cuts" in the cone. This also increases the smoothness of the cone by adding extra facets.
Segments	The segments are longitudinal edges of the sphere running from right to left. By default, the primitive cone uses 12 segments. By increasing this value you can increase the number of "cuts" along the cone. This is useful if you plan to deform the cone, as these segments act as hinges during deformation operations.
Polygon Type	Selects the type of polygons to use for the cylinder. Choose from: • **Face** - Create standard, unsmoothed polygon faces. • **Subdivs** - Create smoothed subdivision surfaces. • **Catmull Clark** - Create smoothed subdivision surfaces using Pixar's Catmull-Clark algorithm.
Caps	Choose how to fill the bottom side (cap) of the cylinder: • **None** - Leave the cap empty. • **Single Polygon** - Fill the cap with a single N-sided polygon. • **Quad Grid** - Fill the cap with multiple quadrangles. This ensures the top is compatible with MeshFusion.
Edge Weight	Sets the sharpness of the edges around the bottom cap. At **100 %** the cap is flat and forms a sharp edge with the side. The **Edge Weight** only applies when the cone has been generated using **Subdivs** or **Catmull Clark** as the **Polygon Type**.
Bulge	Produces a swelling effect on the cone.
Axis	This X, Y, Z choice allows you to quickly change the orientation of the cone.
Make UVs	When this button is active, a UV map is automatically generated for the

Cone	
	geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive, as it provides a baseline UV map that you can massage later in the modeling process. In many cases, this can reduce the amount of work required to map the model.
Distortion Compensation	Adjusts the aspect ratio and related area size of UV polygons to reduce the distortion in the primitives UVs. Use this option when you create a **Cone** that does not have equal radius values. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**. Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final values.
Gaps	Specifies empty space at the boundary of the UV space or between UV parts. The UV unit is specified using a percentage value.
Handle Mode	Provides three types of behavior for corner and side tool handles: • **Resize from Center** - Resizes both sides of the primitive at the center, when dragging a corner or side handle. • **Resize** - Resizes the shape at the opposite side, when dragging a corner or side handle. • **Move All** - Moves the primitive instead of resizing. **Tip:** You can also move the shape by clicking and holding the **Ctrl** key when in another handle mode.
Show Corner	When enabled, displays the bounding box corner handles.
Min X, **Y**, **Z/Max X**, **Y**, **Z**	You can also define the shape based on specific X, Y, and Z bounding box locations in 3D space, which can be specified here. This makes it easy to place a cone on the ground plane for instance by making the **Min Y** value **0** and the **Max Y** value **1m**.

Tip: The **Cone** primitive supports symmetrical creation. When Applying Precision is activated, creating the primitive shape creates an identical version across the specified axis.

Cylinder

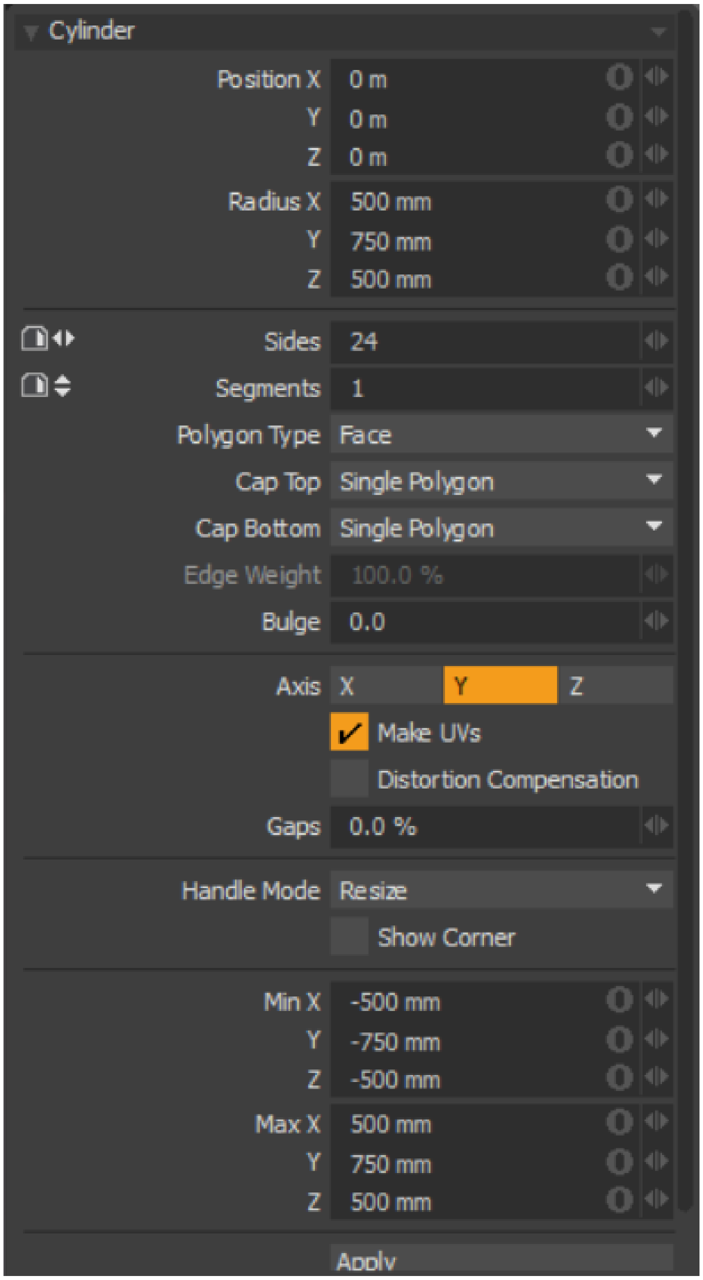
The **Cylinder** primitive provides a simple method for creating cylinders or tubes.

Note: For information on how to add primitives to your scene, see Adding Primitives.

When the Cylinder tool is active, the following options are available for editing the primitive:

Note: The properties allow you to generate subdivided, quad polygon cylinders that can be used in MeshFusion.

Cylinder Properties



Cylinder

Position X 0 m
Y 0 m
Z 0 m
Radius X 500 mm
Y 750 mm
Z 500 mm

Sides 24
Segments 1
Polygon Type Face
Cap Top Single Polygon
Cap Bottom Single Polygon
Edge Weight 100.0 %
Bulge 0.0

Axis X [✓ Y] Z
☑ Make UVs
☐ Distortion Compensation
Gaps 0.0 %

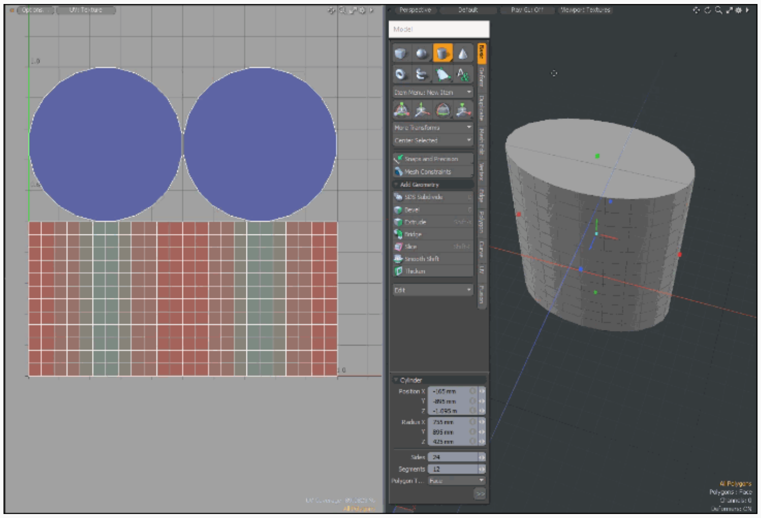
Handle Mode Resize
☐ Show Corner

Min X -500 mm
Y -750 mm
Z -500 mm
Max X 500 mm
Y 750 mm
Z 500 mm

The following **Cylinder** options are available for editing the primitive created:

Cylinder	
Position X, Y, Z	These three values establish the location for the center of the cylinder.

Cylinder	
Radius X, **Y**, **Z**	These three values establish the dimensions of the cylinder, based on the radius.
Sides	This value determines how many edges are used to define the cylinder's circumference. The larger the number of sides, the smoother the cylinder appears. If you plan to use subdivision surfaces on the cylinder, use a low number.
Segments	By default, the primitive cylinder uses one segment. By increasing this value you can increase the number of "cuts" along the cylinder. This is useful if you plan to deform the cylinder, as these segments act as hinges during deformation operations.
Polygon Type	Selects the type of polygons to use for the cylinder. Choose from: • **Face** - Create standard, unsmoothed polygon faces. • **Subdivs** - Create smoothed subdivision surfaces. • **Catmull Clark** - Create smoothed subdivision surfaces using Pixar's Catmull-Clark algorithm.
Cap Top	Choose how to fill the top side (cap) of the cylinder: • **None** - Leave the top empty. • **Single Polygon** - Fill the top with a single N-sided polygon. • **Quad Grid** - Fill the top with multiple quadrangles. This ensures the top is compatible with MeshFusion.
Cap Bottom	Choose how to fill the bottom side of the cylinder. This has the same options as **Cap Top**.
Edge Weight	Sets the sharpness of the edges around the top and bottom caps. At **100 %** the cylinder tops are flat and form square edges with the side. The **Edge Weight** only applies when the cylinder has been generated using **Subdivs** or **Catmull Clark** as the **Polygon Type**.
Bulge	Produces a swelling effect on the cylinder.
Axis	This X, Y, Z choice allows you to quickly change the orientation of the primitive.
Make UVs	When this button is active, a UV map is automatically generated for the

Cylinder	
	geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive, as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.
Distortion Compensation	Adjusts the aspect ratio and related area size of UV polygons to reduce the distortion in the primitives UVs. Use this option when you create a **Cylinder** that does not have equal radius values. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**. Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final values. Click on the image below to view an animation. 
Gaps	Specifies empty space at the boundary of the UV space or between UV parts. The UV unit is specified using a percentage value.
Handle Mode	Provides three types of behavior for corner and side tool handles:

Cylinder	
	• **Resize from Center** - Resizes both sides of the primitive at the center, when dragging a corner or side handle. • **Resize** - Resizes the shape at the opposite side, when dragging a corner or side handle. • **Move All** - Moves the primitive instead of resizing. **Tip:** You can also move the shape by clicking and holding the **Ctrl** key when in another handle mode.
Show Corner	When enabled, displays the bounding box corner handles.
Min X, **Y**, **Z/Max X**, **Y**, **Z**	You can also define a cylinder based on specific X, Y, and Z bounding box locations in 3D space, which can be specified here. This makes it easy to place the cylinder on the ground plane, for instance, by making the **Min Y** value **0** and the **Max Y** value **1m**.

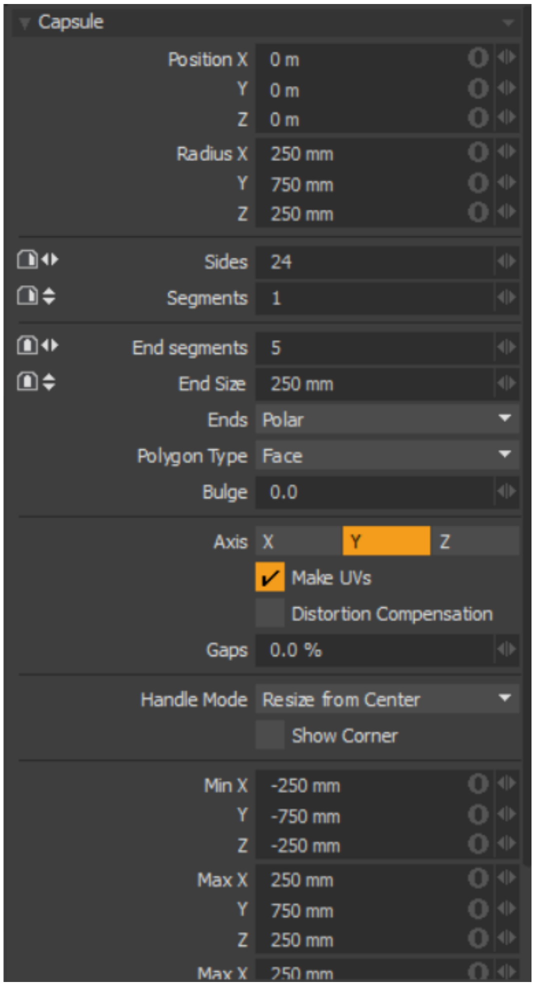
Tip: The **Cylinder** primitive supports symmetrical creation. When Applying Precision is activated, creating the primitive shape creates an identical version across the specified axis.

Capsule

The **Capsule** primitive provides a simple method for creating capsules or rounded edge tubes.

Note: For information on how to add primitives to your scene, see Adding Primitives.

When the Capsule tool is active, the following options are available for editing the primitive:



Capsule

Position X 0 m
Y 0 m
Z 0 m
Radius X 250 mm
Y 750 mm
Z 250 mm
Sides 24
Segments 1
End segments 5
End Size 250 mm
Ends Polar
Polygon Type Face
Bulge 0.0
Axis X [✓ Y] Z
☑ Make UVs
☐ Distortion Compensation
Gaps 0.0 %
Handle Mode Resize from Center
☐ Show Corner
Min X -250 mm
Y -750 mm
Z -250 mm
Max X 250 mm
Y 750 mm
Z 250 mm
Max X 250 mm

Capsule	
Position X, **Y**, **Z**	These three values establish the 3D location for the center of the primitive.
Radius X, **Y**, **Z**	These three values are used to establish the overall dimensions of the capsule. Set to **25cm**, **75cm**, **25cm** (by default); being radius value, they produce a capsule that is .5 meters wide and 1.5 m long.
Sides	The sides are the latitudinal edges of the capsule running from top to bottom. By default, the primitive capsule uses 24 sides. By increasing this value you can increase the number of "cuts" in the shape. This is useful if you plan to deform the capsule, as these segments act as hinges during deformation operations. This also increases the smoothness of the capsule by adding extra facets.
Segments	The segments are the longitudinal edges of the capsule running from right to

Capsule	
	left. By default, the primitive uses 8 segments. By increasing this value you can increase the number of "cuts" in the capsule. This is useful if you plan to deform the shape, as these segments act as hinges during deformation operations. It also increases the smoothness of the shape by adding extra facets.
End Segments	This value is used for creating the rounded end-caps on the capsule. This value sets the number of longitudinal edges on the end-cap. Increasing this number produces a more round cap. The default value is **5**.
End Size	Determines the height of the cap. The default value is **50cm**.
Ends	Choose a way to fill the ends of the capsule: • **Polar** - Create triangles around the polars of the capsule. • **Quad Grid** - Fill each end with a quadrangle grid. This makes the capsule compatible with MeshFusion. **Note:** Set **Ends** to **Quad Grid** for compatibility with MeshFusion.
Polygon Type	You can use this option to determine the type of polygons created: • **Face** - creates standard, unsmoothed polygon faces. • **Subdivs** - creates Subdivision Surface smoothed polygons. • **Catmull-Clark** - creates Pixar Catmull-Clark smoothed polygons.
Bulge	Produces a swelling effect on the capsule.
Axis	This X, Y, Z choice allows you to quickly change the orientation of the primitive.
Make UV	When this button is active, a UV map is automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive, as it provides a baseline UV map that you can massage later in the modeling process. In many cases, this can reduce the amount of work required to map the model.
Distortion Compensation	Adjusts the aspect ratio and related area size of UV polygons to reduce the distortion in the primitives UVs.

Capsule	
	Use this option when you create a **Capsule** that does not have equal radius values. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**. Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final values.
Gaps	Specifies empty space at the boundary of the UV space or between UV parts. The UV unit is specified using a percentage value.
Handle Mode	Provides three types of behavior for corner and side tool handles: • **Resize from Center** - Resizes both sides of the primitive at the center, when dragging a corner or side handle. • **Resize** - Resizes the shape at the opposite side, when dragging a corner or side handle. • **Move All** - Moves the primitive instead of resizing. **Tip:** You can also move the shape by clicking and holding the **Ctrl** key when in another handle mode.
Show Corner	When enabled, displays the bounding box corner handles.
Min X, **Y**, **Z/Max X**, **Y**, **Z**	You can also define the shape based on specific X, Y, and Z bounding box locations in 3D space, which can be specified here. This makes it easy to place a capsule on the ground plane, for instance, by making the **Min Y** value **0** and the **Max Y** value **1m**.

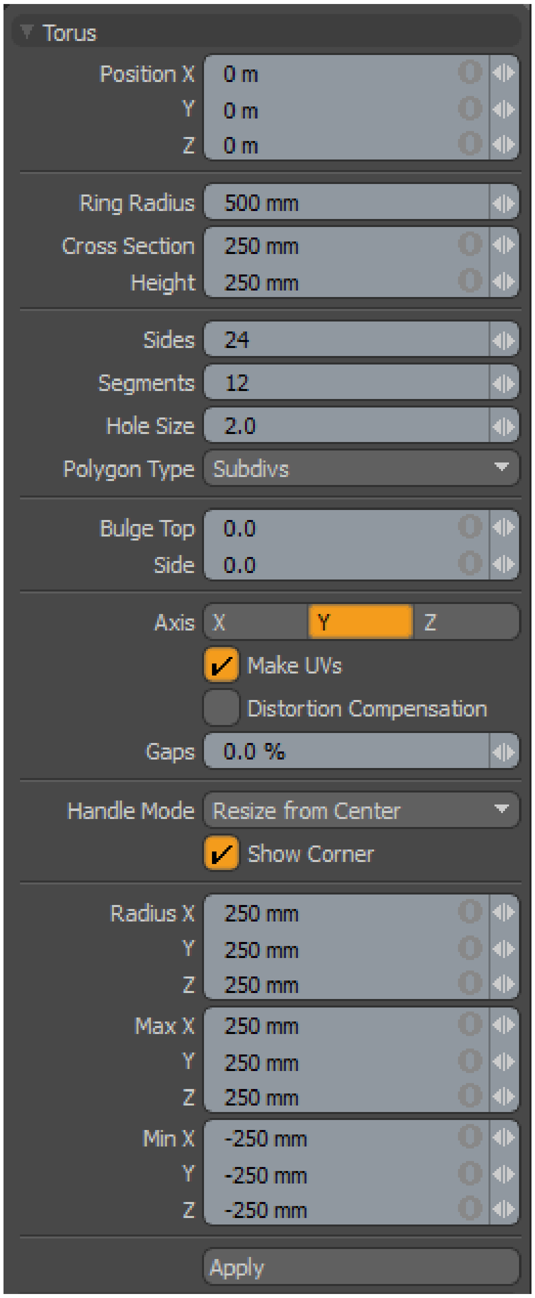
Tip: The **Capsule** primitive supports symmetrical creation. When Applying Precision is activated, creating the primitive shape creates an identical version across the specified axis.

Torus

The **Torus** primitive provides a simple method for creating toroids or "doughnuts".

Note: For information on how to add primitives to your scene, see Adding Primitives.

When the Torus tool is active, the following options are available for editing the primitive:



Torus
Position X 0 m
Y 0 m
Z 0 m
Ring Radius 500 mm
Cross Section 250 mm
Height 250 mm
Sides 24
Segments 12
Hole Size 2.0
Polygon Type Subdivs
Bulge Top 0.0
Side 0.0
Axis X Y Z
☑ Make UVs
☐ Distortion Compensation
Gaps 0.0 %
Handle Mode Resize from Center
☑ Show Corner
Radius X 250 mm
Y 250 mm
Z 250 mm
Max X 250 mm
Y 250 mm
Z 250 mm
Min X -250 mm
Y -250 mm
Z -250 mm
Apply

Torus	
Position X, Y, Z	These three values establish the 3D location for the center of the primitive.

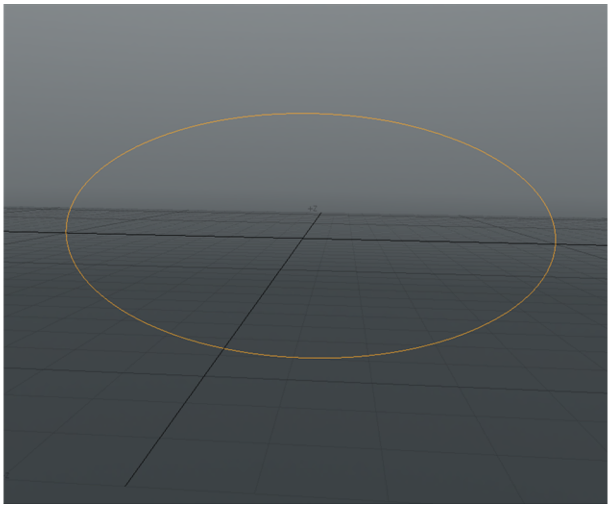
Torus	
Ring Radius	Determines the overall outside width of the torus.
Cross Section/Height	A **Cross Section** width and **Height** setting, determining the sizing of the ring cross section of the torus.
Radius X, **Y**, **Z**	These three values establish the dimensions of the torus. The tool is set to **50cm**, **50cm**, **50cm** by default.
Sides	The sides are the latitudinal edges of the torus running from top to bottom. By default, the primitive uses 24 sides. By increasing this value, you can increase the number of "cuts" in the shape. This is useful if you plan to deform the mesh, as these segments act as hinges during deformation operations. This also increases the smoothness of the capsule by adding extra facets. If you plan to apply subdivision surfaces to this shape, you can use a lower number of sides.
Segments	The segments are the longitudinal edges of the torus running from right to left. By default, the primitive uses 12 segments. By increasing this value, you can increase the number of "cuts" in the shape. This is useful if you plan to deform the shape, as these segments act as hinges during deformation operations. It also increases the smoothness of the shape by adding extra facets. If you plan to apply subdivision surfaces to this shape, you can use a lower number of segments.
Hole Size	This value sets the radius of the hole in the middle of the torus.
Polygon Type	You can use this option to determine the type of polygons created: • **Face** - creates standard, unsmoothed polygon faces. • **Subdivs** - creates Subdivision Surface smoothed polygons. • **Catmull-Clark** - creates Pixar Catmull-Clark smoothed polygons.
Bulge Top/Side	Raising this value above zero (**0**) attenuates the shape towards a more cubic or box-like form.
Axis	This X, Y, Z choice allows you to quickly change the orientation of the primitive.
Make UVs	When this button is active, a UV map is automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive, as it provides a baseline UV map that you can massage later in the modeling process. In many cases, this can reduce the amount of work required to map the model.

Torus	
Distortion Compensation	Adjusts the aspect ratio and related area size of UV polygons to reduce the distortion in the primitives UVs. Use this option when you create a **Torus** that does not have equal radius values. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**. Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final values.
Gaps	Specifies empty space at the boundary of the UV space or between UV parts. The UV unit is specified using a percentage value.
Handle Mode	Provides three types of behavior for corner and side tool handles: • **Resize from Center** - Resizes both sides of the primitive at the center, when dragging a corner or side handle. • **Resize** - Resizes the shape at the opposite side, when dragging a corner or side handle. • **Move All** - Moves the primitive instead of resizing. **Tip:** You can also move the shape by clicking and holding the **Ctrl** key when in another handle mode.
Show Corner	When enabled, displays the bounding box corner handles.
Radius	Determines the size from the center of the Torus to the outer bounding box edge - the radius - of the torus.
Min XYZ/Max XYZ	You can also define the shape based on specific X, Y, and Z bounding box locations in 3D space, which can be specified here. This makes it easy to place the torus on the ground plane, for instance, by making the **Min Y** value **0** and the **Max Y** value **1m**.

Tip: The **Torus** primitive supports symmetrical creation. When Applying Precision is activated, creating the primitive shape creates an identical version across the specified axis.

Ellipse

The **Ellipse** primitive is a closed elliptical **Bezier** or **Spline** curve. The **Bezier** type is equivalent to a curve made by the Bezier tool. The **Spline** type is a Hermite Cubic spline, equivalent to that made by the Adding Curves



Note: You can fill the shape using the Curve Fill tool.

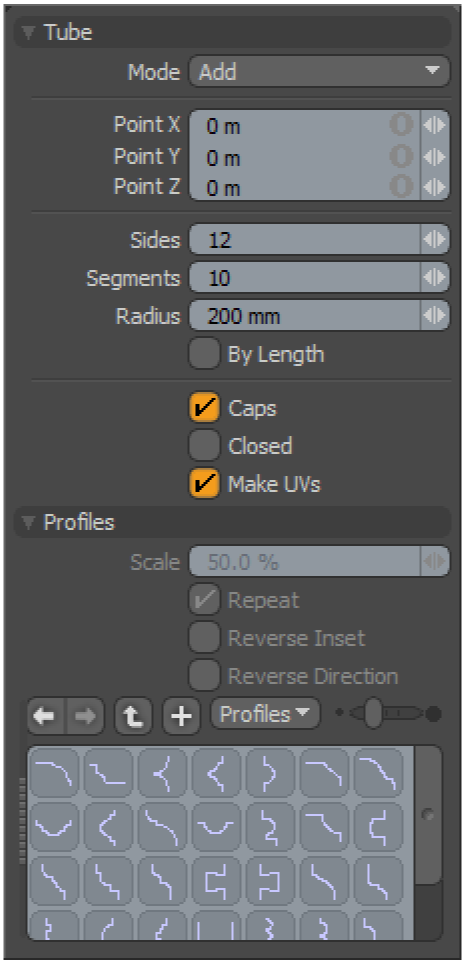
Ellipse Properties

Ellipse	
Enable	Toggles the primitive.
Type	Selects the curve type: • **Bezier** - Draw using Bezier control points. • **Spline** - Draw using Hermite Cubic spline type. This locates all vertices along the path of the ellipse.
Position X, Y	Determines the X, Y position along the plane defined by **Axis**.
Size X, Y	Sets the width and height.

Axis	Chooses the plane on which to draw the ellipse: • **X** - Align to the YZ plane. • **Y** - Align to the XZ plane. • **Z** - Align to the XY plane.

Tube

The **Tube** primitive provides a simple method for creating spline curve-based tubes. Using the **Tube** primitive you can quickly click down a number of control points and Modo creates a tube, based on the spline curve interpolated through them.


Tube
Mode Add
Point X 0 m
Point Y 0 m
Point Z 0 m
Sides 12
Segments 10
Radius 200 mm
☐ By Length
☑ Caps
☐ Closed
☑ Make UVs
Profiles
Scale 50.0 %
☑ Repeat
☐ Reverse Inset
☐ Reverse Direction

The following **Tube** options are available for editing the primitive created:

Tube	
Mode	In **Add** mode (default), each click on the 3D viewport creates a new control point in the tube. Clicking on an existing control point allows you to move it freely in space. To add a new control point between two existing points, select the first point of the desired pair, then click between them. **Edit** mode disables the ability to create new control points, so that existing control points can be edited quickly without the concern of accidentally adding a new control point. The **Delete** mode removes control points when directly clicked on.
Point X, **Y**, **Z**	This value triplet is set interactively as you drag a control point in the 3D viewport. If you wish to set a control point to a very specific location you can also use the tool property sheet to enter the X, Y, and Z coordinates directly.
Sides	This value determines how many polygons (sides) are used to make the ring of the tube. If you use a low number such as 3, you would create a triangular tube, whereas a higher number such as the default **12**, would create a very round tube. If you plan to use subdivision surfaces on the tube, once it is created, you can lower the number of sides to 4 or 5, and the resulting mesh is still very smooth (once subdivision surfaces are active).
Segments	This value determines the number of polygons that are used to span the gap between each control point. If you set this value to one, you may see that the tube looks very linear, as there are not enough connecting polygons to interpolate a smooth curve. Lower numbers, such as one or two, can work if you plan to use subdivision surfaces after creating the tube. Using larger numbers creates a smooth flowing tube, at the expense of additional geometry.
Radius	This value sets the thickness of the tube. It's important to keep in mind that if you plan to use subdivision surfaces, the resulting mesh is somewhat smaller than the original cage polygons. In this case, you may wish to increase the radius prior to completing the tube, so that when you activate subdivision surfaces, the resulting mesh maintains your intended thickness.
By Length	When enabled, tube sections are recomputed as equal lengths across the curve, rather than by the control point locations.
Caps	Determines whether or not the ends of the tube are closed. By default, this setting is turned on so that the tube is closed on both ends. You can remove the caps by turning this button off.
Closed	Activating the **Closed** flag creates a curve loop, rather than a curve line, so that the resulting tube has its first and last points connected together using another set of

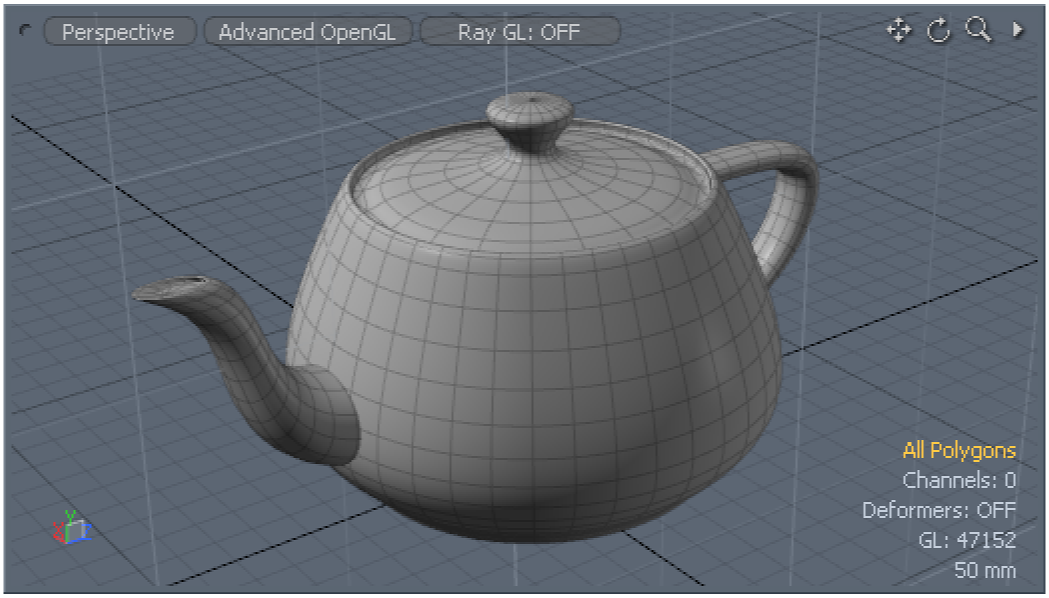
Tube	
	segments.
Make UVs	When this button is active, a UV map is automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive, as it provides a baseline UV map that you can massage later in the modeling process. In many cases, this can reduce the amount of work required to map the model.
Profiles - Additionally, there is a mini-**Preset Browser** for viewing various profiles. It works the same as the Preset Browser. Profiles, when selected, apply themselves along the length of the tube. The following **Profiles** options are available:	
Scale	With this setting, you can adjust the size of the profile applied to the tube primitive.
Repeat	When this option is toggled, it enables/disables the repeating of the profile between each curve point.
Reverse Inset	This option toggles the inset direction of the profile, reversing it horizontally. Areas that were initially pushed in, when activated, are pushed out.
Reverse Direction	This option toggles the direction of the profile, reversing it vertically, providing a flipped orientation for the profile.

Teapot

The **Teapot** primitive provides a simple method for creating a teapot. The teapot primitive is included with Modo because of its historical significance.

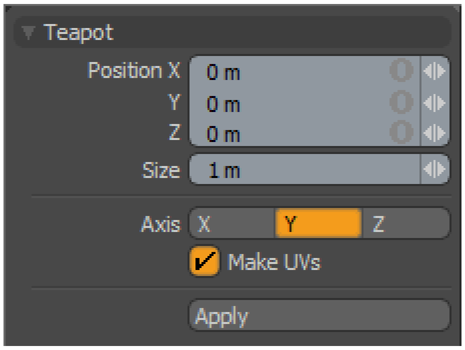
In the early years of 3D modeling and rendering, Martin Newel created a computer-generated three dimensional teapot. The original "Utah teapot" was actually taller than the famed teapot we see in so many places today. Legend has it that during a demo, using a system with non-square pixels, Jim Blinn scaled the teapot rather than scaling the image. As this dataset became more widely distributed than the original, his edit changed the aspect of teapots in 3D software for years to come.

Tip: If you are interested in following teapot lore, simply type "Utah Teapot" into your favorite search engine.


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 47152
50 mm

Click **Geometry** > **Primitive Tools** > **Teapot** and set the properties to add it to your scene.

When the tool is active, the following options are available for editing the primitive:


Teapot
Position X 0 m
Y 0 m
Z 0 m
Size 1 m
Axis X Y Z
Make UVs
Apply

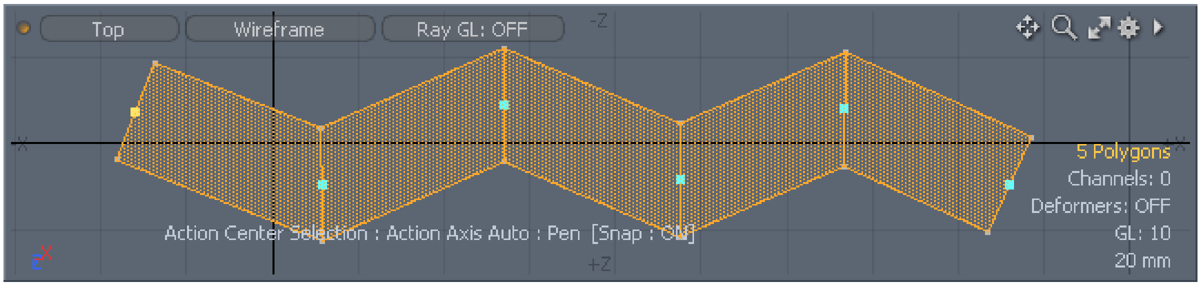
Teapot	
Position X, Y, Z	Defines the center point for the teapot's bounding box.
Size	Defines the overall size of the teapot's longest bounding edge.
Axis	This X, Y, Z choice allows you to quickly change the orientation of the primitive.
Make UVs	When this button is active, a UV map is automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive, as it provides a baseline UV map that you can massage later in the modeling process. In many cases, this can reduce the amount of work required to map the model.

Using Drawing Tools

The following topics provide details about Modo's drawing tools.

Pen

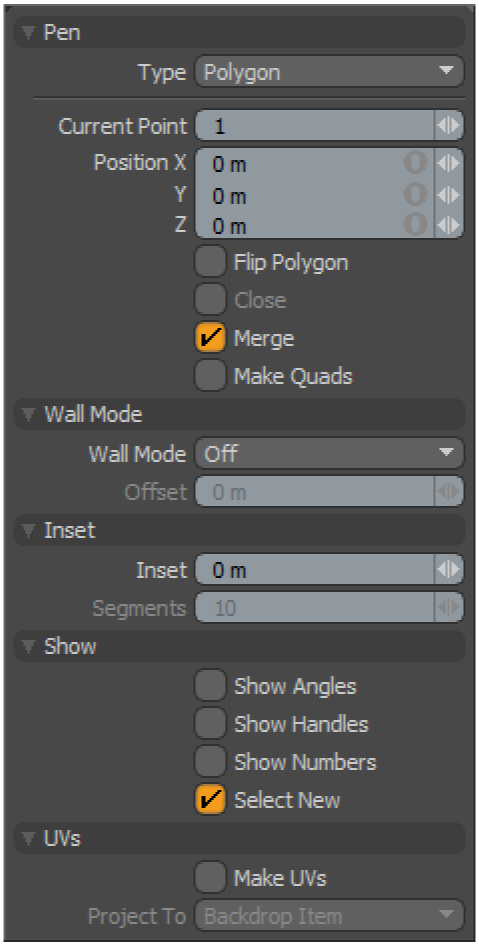
The **Pen** tool allows you to create geometry vertex by vertex, by clicking in the 3D viewport. One can use the **Pen** tool to create polygons (default), lines, vertices, spline patches, or subdivision surface polygons.



Usage

The **Pen** tool is found in several locations within the default interface. It can be found in the **Model** layout **Toolbox**, under the **Polygon** sub-tab. It is also found in the **Topology** layout, under the **Tools** sub-tab. Clicking the tool icon activates the tool. Clicking in the 3D viewport sets the **Pen** tool to interactive mode. Depending on the **Type** option selection, slightly different components draw in the interface. When the tool is active, you can also click directly on any of the vertices it has created in the current session, and drag them in 3D space for direct editing. They turn yellow when the mouse is directly over them. Dragging them onto any previous vertex welds them together. Clicking away from a vertex creates a new vertex in that location.

To start a new line without dropping the tool, press **Shift**+click. To insert a vertex between two existing ones, highlight a previously created vertex and click away from it. The new vertex is inserted after the selected vertex. When creating **Quad Strips** with the **Make Quads** option enabled, an occasional triangle can be created by pressing the **Ctrl** key. When the tool is active, the following options are available in the tool's **Properties** panel.



Pen

Type: Polygon

Current Point: 1

Position X: 0 m

Y: 0 m

Z: 0 m

☐ Flip Polygon

☐ Close

☑ Merge

☐ Make Quads

Wall Mode

Wall Mode: Off

Offset: 0 m

Inset

Inset: 0 m

Segments: 10

Show

☐ Show Angles

☐ Show Handles

☐ Show Numbers

☑ Select New

UVs

☐ Make UVs

Project To: Backdrop Item

The following options are available for adjusting the **Pen** tool:

Pen	
Type	Allows you to select between different modes. • **Polygons** - Allows you to draw individual polygons vertex by vertex. The first click of the mouse creates a new vertex and each subsequent click continues adding vertices to the active polygon face. Pressing **Shift** starts a new polygon. • **Lines** - Allows you to create a continuous string of two point polygon line segments. • **Vertices** - Allows you to simply create new vertices with each click, leaving you with a vertex (point) cloud. • **Spline patches** - Acts exactly like **Polygon** mode but creates polygons of type **Spline Patch**. • **Subdivision Surfaces** - Allows you to simply create polygons tagged as **SDS**.

Pen	
	• **Polyline** - Allows you to create a string of **Polylines**, similar to curves (in many ways) but made out of straight line segments. These are used as long hair guides. • **Catmull-Clark** - Allows you to create Catmull-Clark subdivision surfaces by clicking down points in the 3D viewport. • **B-Spline** - Allows you to draw B-spline curves. When this option is selected, the points you draw define a B-Spline polygon. B-splines are spline curves that are defined by at least four vertices and a Weight Map. They provide better continuity than other spline curve types, such as Hermite and Bezier, because they do not go through any of their control points.
Current Point	You can change this setting to change the current vertex to be edited by the numeric values listed as **Position X**, **Y**, and **Z**. Since the tool may have created several vertices, it's useful to be able to choose one numerically. The vertices are numbered in sequence as they are generated by the tool. Clicking on any vertex in the 3D viewport (from the current **Pen** tool session) also displays its numeric value in the sequence.
Position X, **Y**, **Z**	This numeric entry field allows you to precisely edit each vertex created by the tool. Simply set the **Current Point** value to the vertex you wish to edit and enter the numeric values for the **X**, **Y**, and **Z** positions. Clicking any vertex in the 3D viewport also displays its current position.
Flip Polygon	This option simply inverts the surface normal of the polygon(s) created.
Close	Automatically draws a line between the first and last vertex created making a closed loop.
Merge	Automatically merges, or combines, vertices within close proximity to each other when created.
Make Quads	Automatically places vertices to create quads with each new click after the first two vertices have been set. This mode makes it very simple to create polygon strips. Pressing the **Ctrl** key creates a single triangle. This option is only compatible with the **Type** > **Polygons** setting.
Wall Mode	
Wall Mode	With mode type set to **Polygons**, **Wall Mode** allows you to draw out polygon

Pen	
	strips that look like a building floor plan, effectively creating the foundation for rooms. There are four settings: **Off** disables wall mode, **Inner** draws the wall segment on the inside of the polygon edge line, **Outer** draws the wall segments on the outside of the polygon edge line, and **Both** extends the wall segments on both sides of the polygon edge line. Once the tool is dropped, the resulting **Wall Mode** polygons can be thickened into upstanding walls.
Offset	• Represents the thickness of the wall, when in **Wall Mode**. **Note:** When **Wall Mode** is set to **Both**, the offset is doubled. This value can also be set interactively by right-clicking in the viewport, much like adjusting brush size for painting.
Inset	
Inset	When **Wall Mode** is enabled, this option allows each new vertex to be beveled into multiple segments for quickly flattening or rounding corners.
Segments	This option is only available when **Inset** is non-zero. A value determines how many vertices to create in order to facilitate the inset value. Increasing this value makes more complex and rounded corners. Using a value of **1** results in flattened corners.
Show	
Show Angles	Toggles the display of corner angles between opposing edges in degrees, drawn in ghosted form over active geometry.
Show Handles	Toggles the display of tool handles offering you the ability to adjust the previously-created vertex by axis.
Show Numbers	Toggles display of the vertex point order values in the viewport.
Select New	Toggles the selection state of the most recently-created polygon, while the tool is still active.
UVs	
Make UVs	When enabled, this setting automatically makes UVs for the created geometry further defined by the **Project To** setting.

Pen	
Project To	Determines the way that the UV map is created: • **Action Axis** - The **Pen** tool determines the UV map projection axis, based on the current work plane for the created polygon. UVs are created in space, relative to where they are drawn in 3D space, so only points drawn with the 0m to 1m world space appear within the 0-1 UV space. • **Backdrop Item** - Useful for drawing polygons over a backdrop item. The UVs are projected onto the backdrop item to match position. So, for instance, if you created a polygon over the tire of car backdrop image, the new polygon would inherit the UV position of the tire's location in the image, relative to UV space. • **U/V Direction** - When in **Wall Mode**, this setting automatically generates a row of regular quad UV coordinates for the polygons, along either the U direction (horizontal) or V direction (vertical), as specified.

Curves

This topic provides information on curve-based primitive creation tools.

Adding Curve-based Primitives to Your Scene

You can add curves to your scene from the menu bar:

1. Under **Geometry**, click **Primitive Tools**, and click the required primitive in the list.
2. Click in the 3D viewport to place the curve.
3. If needed, adjust the dimensions of the curve, using the tool handles in the 3D viewport or in the tool's **Properties** panel.

Adding Procedural Curves to Your Scene

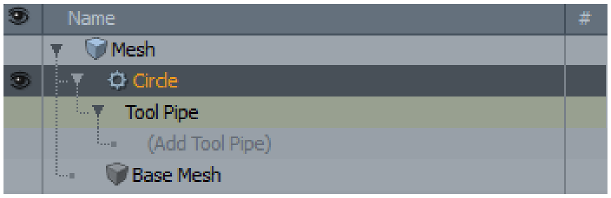
You can use the new procedural modeling tools to add procedural curves to your scene. You can find the **Mesh Ops** tab on the right side of the interface.

To add a curve in the **Mesh Ops** tab, click the **Add Operator** button above the **Mesh Operations** list.

The **Preset Browser** opens, where you can select the required curve under **Mesh Operations** > **Create** > **Curves**.

Curves have a **Tool Pipe** input, to which you can add deformers and sub-tools.

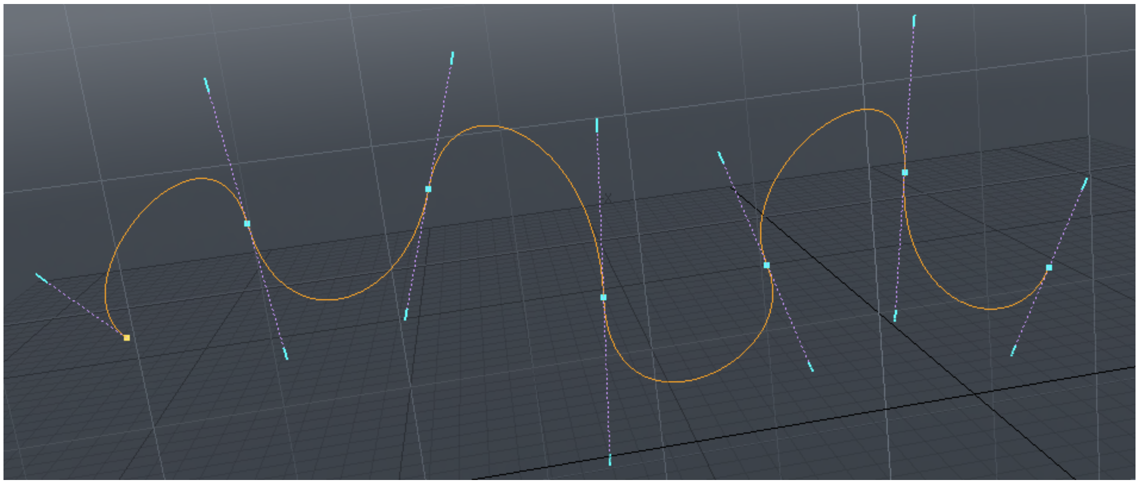
The image below shows a Circle in the **Mesh Operations** list. You can add operations to it by clicking **(Add Tool Pipe)**.



Note: For details on the **Mesh Ops** tab, see Using the Mesh Ops Tab. For more information on procedural selection, see Procedural Selection.

Bezier

The Bezier tool allows you to create a type of curve defined by off-curve control handles. Bezier curves were originally developed for computer modeling in automotive design, but are popular in many vector drawing applications. These curves provide a familiar means of generating curves, as they allow you to interactively control the amount of incoming and outgoing curvature for each control point.



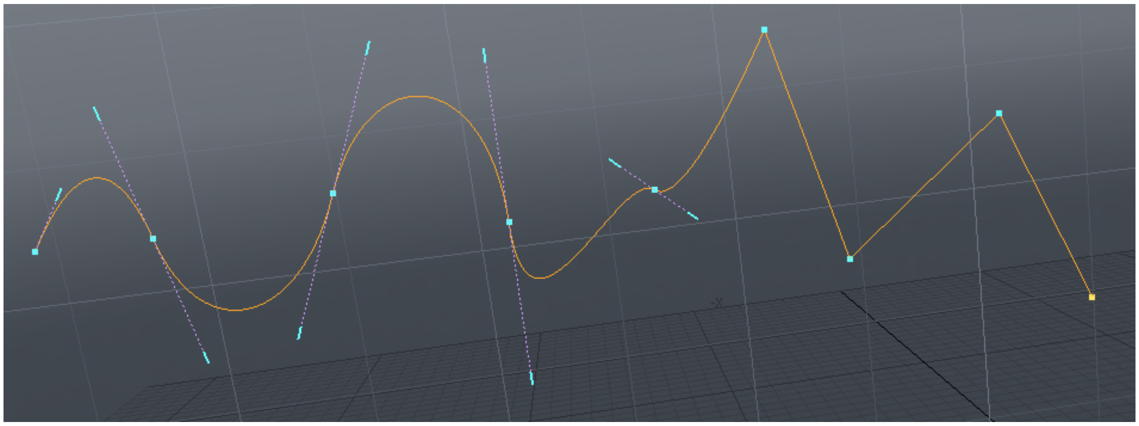
You can find the Bezier tool in the **Create** sub-tab of the modeling toolbox and in the menu bar under **Geometry** > **Curve Palette**.

To use the tool:

1. In the **Create** sub-tab of the modeling toolbox, click the **Bezier Curve** button to activate the tool.

   Alternatively, you can access the tool in the menu bar under **Geometry** > **Curve Palette**.
2. Click in the 3D viewport to position the first control point, then continue to drag the mouse cursor to create the Bezier handle.
3. Release the mouse button to set the handle length.

   Subsequent clicking and dragging draws additional control points and handles with a curve segment between each control point. Control points are always created at the intersection of the work plane and the mouse button click. The curvature of the segments between control points is defined by the Bezier handles, which are called endpoint tangent vectors.
4. To create straight line segments, click in the viewport without dragging. This creates control points with zero-length tangents. When you move these control points later, the corners remain sharp and the lines straight. You can see the two control point types below:



5. You can smooth sharp corners by switching to **Edit** mode in the Bezier tool's **Properties** panel. **Edit** mode reveals handles for the control points that you can drag out to create round corners:

6. While the tool is active, you can re-position the control points by clicking and dragging them. Holding **Shift** and moving a point affects all points downstream of the selected control point as a single unit.

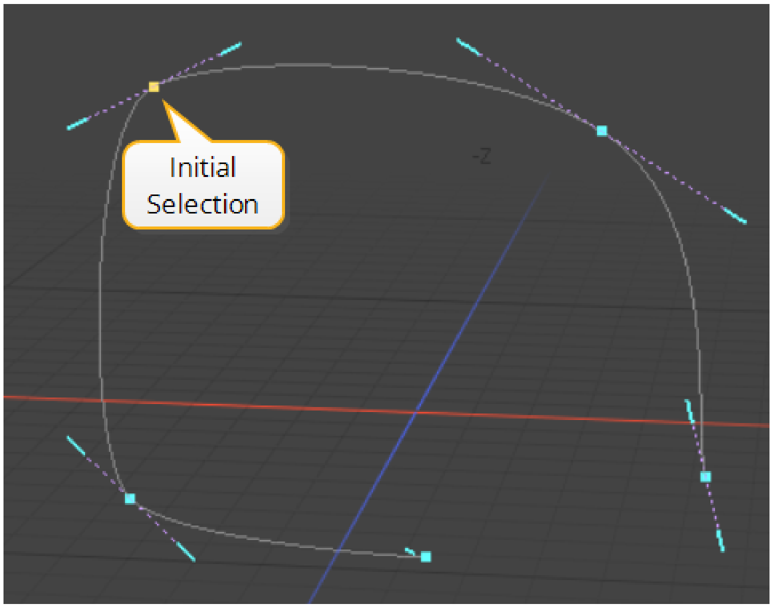
Note: Point order is defined by the initial order in which the curve was created. Press **F** before activating the tool to invert the order.

Tip: To create a closed curve, right-click on the first control point. Modo adds a curve segment between the first and last control points.

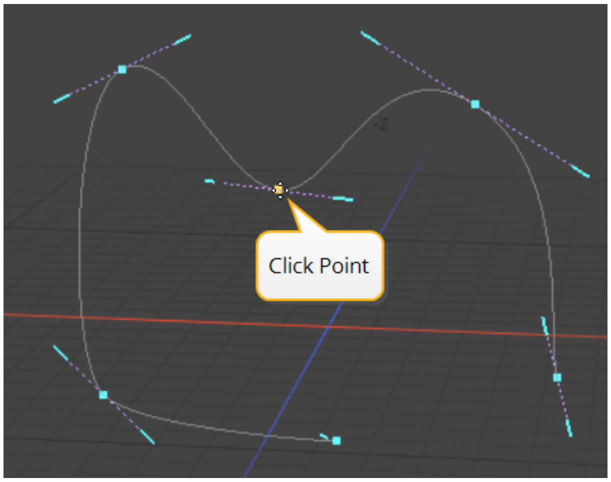
Points always move on the two axes relative to the current work plane, so rotating the viewport to change an axis may be necessary.

You can edit the handles in a similar way. Dragging on the end of one handle breaks tangency, allowing opposing handles to be different lengths and/or angles. You can restore tangency and edit both handles in unison by pressing and holding **Shift** while adjusting the handles.

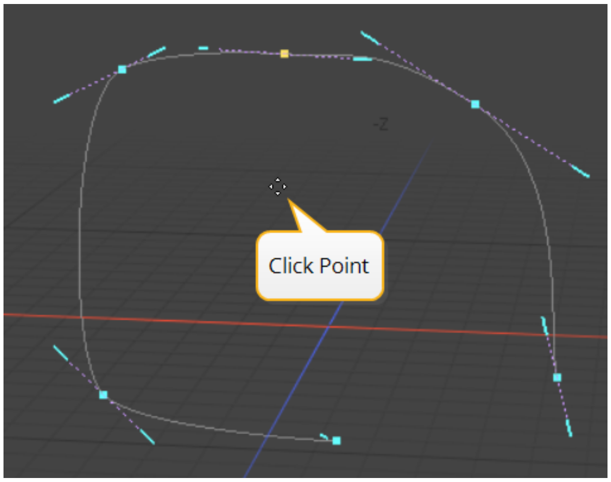
7. You can add points to a curve using the **Add** and **Insert** modes:
   - **Add** always creates a control point relative to the currently selected point on the curve and warps the curve to accommodate the new point.
   - **Insert** creates a control point on the curve closest to the click-point and does not warp the curve.



The initial bezier shape and control point selection



Add mode creates a point and warps the curve

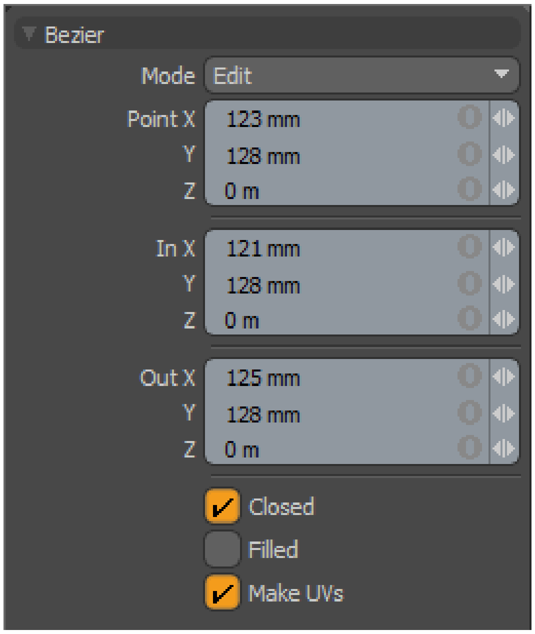


Insert mode creates a point on the existing curve

8. **Delete** mode removes the selected point and warps the curve to accommodate the change.

Bezier Tool Properties

Once you drop the tool by pressing the **spacebar**, interactive handle editing within the tool itself is lost but you can re-activate the editing ability by selecting the curve in **Polygons** selection mode. This displays the editing handles. Control points along the Bezier curve can also be positioned precisely by selecting the target vertex and using the **Point X**, **Y**, **Z** fields of the tool's **Properties** panel.



Bezier

Mode: Edit

Point X: 123 mm
Y: 128 mm
Z: 0 m

In X: 121 mm
Y: 128 mm
Z: 0 m

Out X: 125 mm
Y: 128 mm
Z: 0 m

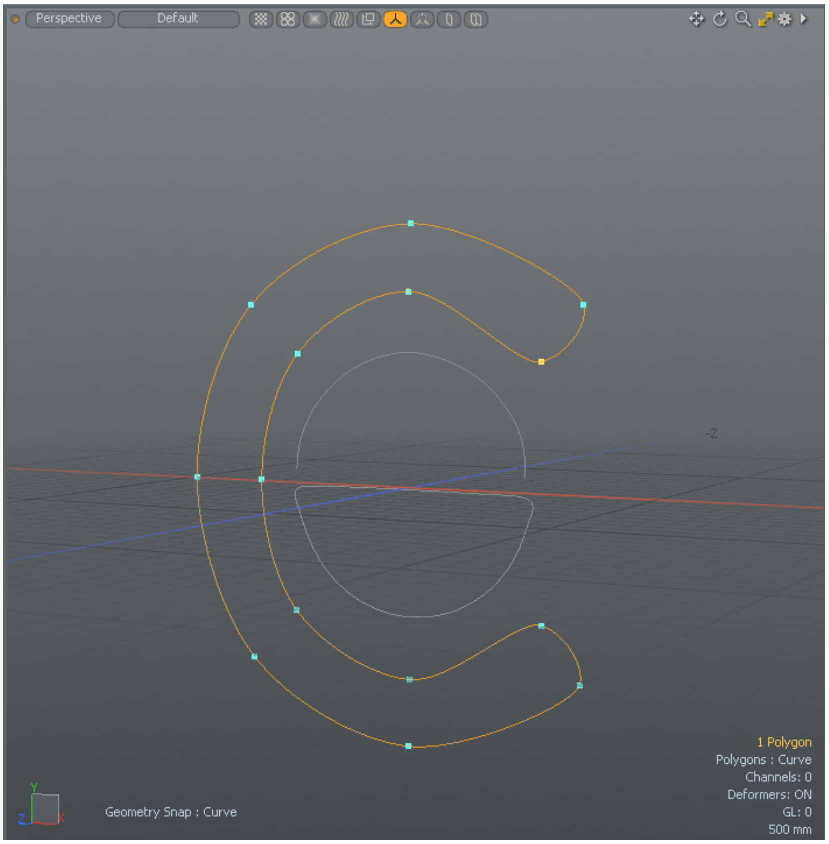
☑ Closed
☐ Filled
☑ Make UVs

The following **Bezier** options are available for adjusting the Bezier tool:

Bezier	
Mode	There are four Bezier curve mode options: • **Add** - The default mode. Creates a control point relative to the currently selected point on the curve and warps the curve to accommodate the new point. • **Edit** - Click and drag any of the control points or handles along the curve to change the look to the desired shape. Pressing and holding the **Shift** key allows you to edit both handles of a control point simultaneously. • **Delete** - Click any control point along the curve to remove it from the Bezier. • **Insert** - Creates a control point on the curve closest to the click-point and does not warp the curve.
Point X, **Y**, **Z**	You can assign specific X, Y, and Z values in these input fields for the currently-selected control point, allowing for finer point control.
In X, **Y**, **Z/Out X**, **Y**, **Z**	You can assign specific X, Y, and Z values in these **In** and **Out** input fields for the currently-selected (highlighted) handle, allowing for finer handle control.
Closed	Adds an automatic curve segment between the first and last control point positions, producing a closed curve.
Filled	When the **Closed** option is enabled, you can enable this to create a renderable flat surface defined by the outlining curve. The resulting surface can be tagged like a polygon for adding material definitions.
Make UVs	Activates auto-generation of UV texture coordinates along the curve. The generated UV values are of a single vertical line (V axis in UV), positioning all the control point vertices evenly between 0 and 1. For example, this can be useful for applying a gradient to a rendered curve. The **Render Curves** option is available in the Mesh Item properties.

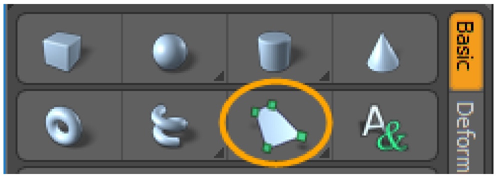
B-Spline

The B-Spline tool allows you to draw B-spline curves, which are spline curves defined by at least four vertices and a weight map. They provide better continuity than other spline curve types because they don't go through any of their control points.

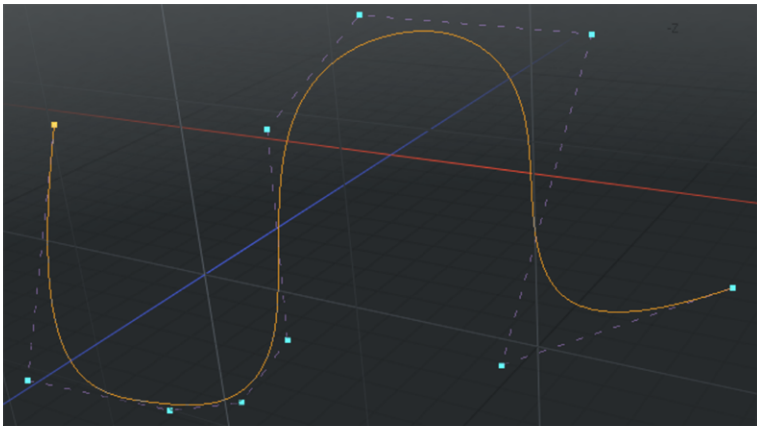

Perspective
Default
-Z
1 Polygon
Polygons : Curve
Channels: 0
Deformers: ON
GL: 0
500 mm
Geometry Snap : Curve

Creating a B-Spline Curve

1. Activate the B-Spline tool in one of the following ways:
   - In the **Basic** sub-tab of the modeling toolbox, right-click the Pen tool to reveal additional tool options, and select **B-Spline**.


Basic
Deform

   - In the **Curve** sub-tab of the modeling toolbox, click **B-Spline**.
   - In the menu bar, under **Geometry** > **Curve Palette**, click **B-Spline**.
2. Start clicking in the viewport to define control points for your curve.



3. Press the **Space** key to drop the tool.

Tip: To create a closed curve, right-click on the first control point. Modo adds a curve segment between the first and last control points.

Editing a B-Spline Curve

Once you drop the tool, interactive editing within the tool itself is lost but you can re-activate the editing ability by selecting the curve in **Polygons** selection mode.

For more information, see Polygon Selection Mode

Changing a B-Spline Curve to Different Curve Type

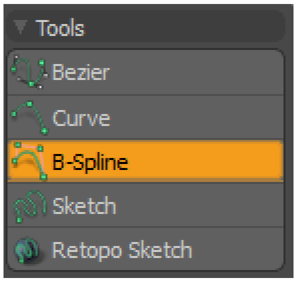
You can change B-Spline curves to B-Spline curves or standard Adding Curves depending on what you want to achieve in your scene. The **Curve Convert** mesh operation allows you to convert your selected curves to another specified curve type.

For more information, see Curve Convert.

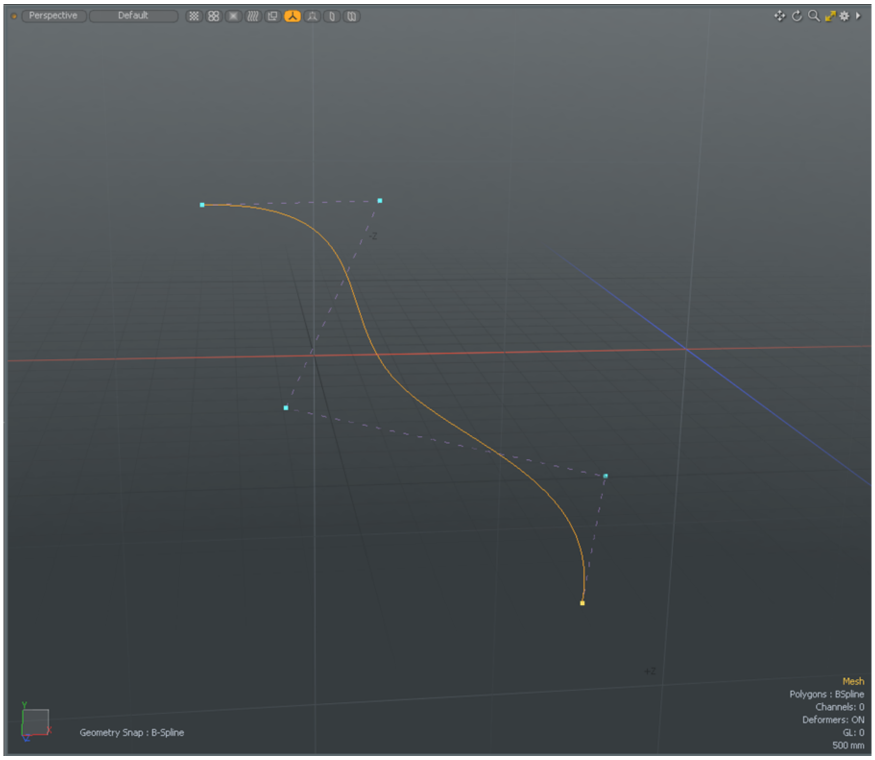
Example

To change the curve type:

1. In the **Model** layout, open the **Curve** tab in the left panel, and click **B-Spline**.

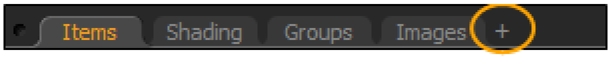

Tools
Bezier
Curve
B-Spline
Sketch
Retopo Sketch

2. In the 3D viewport, click in several locations to create a B-Spline curve, then press the **Space** key to drop the tool.

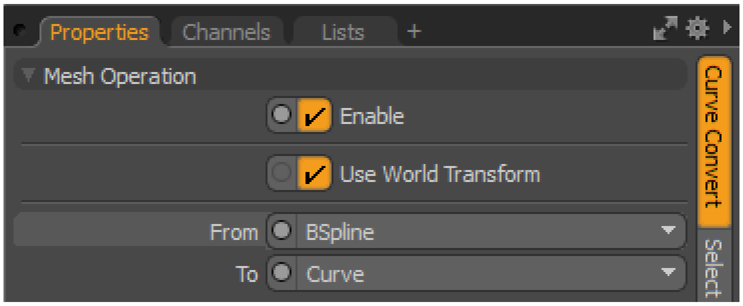

Perspective
Default
Geometry Snap : B-Spline
Mesh
Polygons : BSpline
Channels: 0
Deformers: ON
GL: 0
500 mm

3. In the **Mesh Ops** tab in the right panel, click **Add Operator**, and under **Curves**, double-click **Curve Convert**.

Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **New Tab** > **Data Lists** > **Mesh Ops**.


Items
Shading
Groups
Images
+

4. Select the source and target curve type. In this example, on the **Properties** tab, set **From** to **BSpline** and **To** to **Curve**.


Properties
Channels
Lists
Curve Convert
Select

Mesh Operation

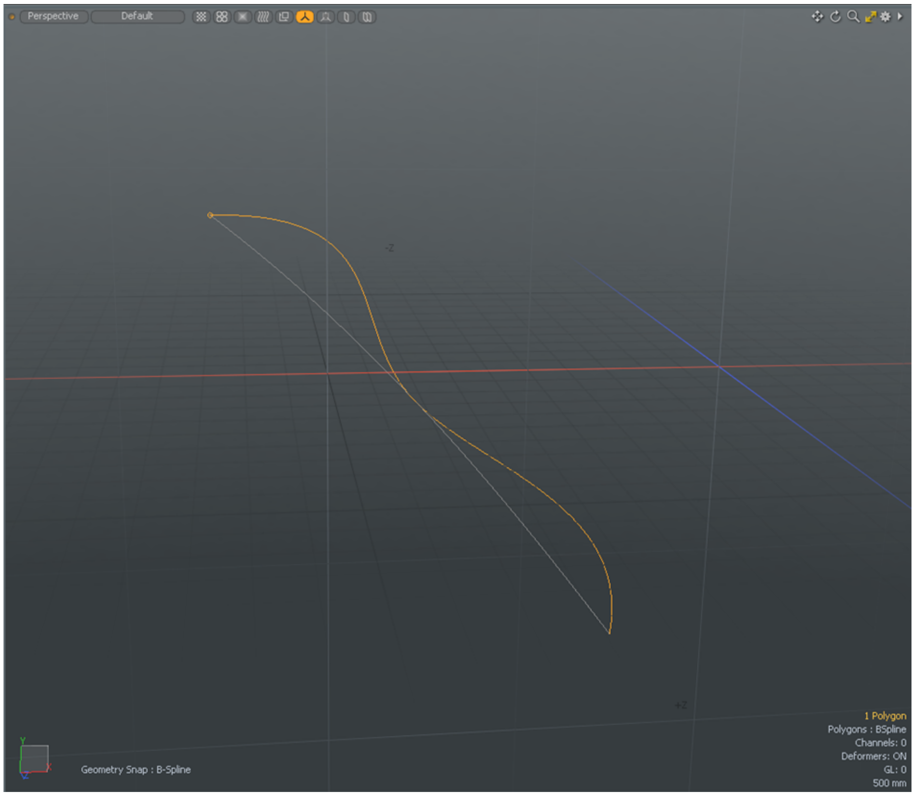
☑ Enable

☑ Use World Transform

From BSpline

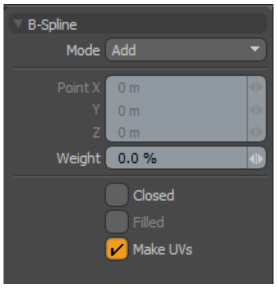
To Curve

The selected B-Spline curve is converted to a standard curve.


Perspective
Default
1 Polygon
Polygons : BSpline
Channels: 0
Deformers: ON
GL: 0
500 mm
Geometry Snap : B-Spline

B-Spline Properties

You can find the tool's properties in the bottom left of the UI.



B-Spline

Mode Add

Point X 0 m

Y 0 m

Z 0 m

Weight 0.0 %

☐ Closed

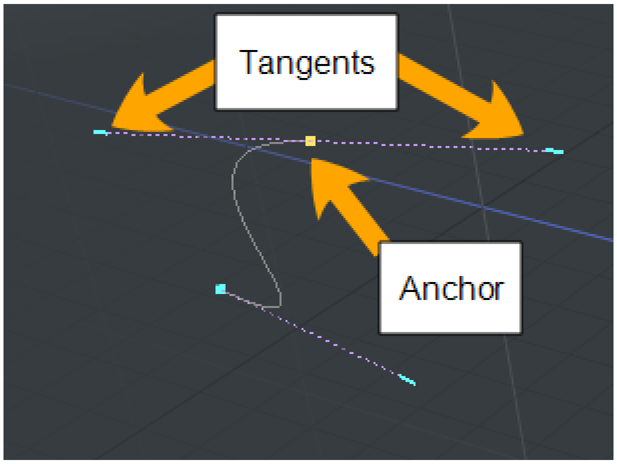
☐ Filled

☑ Make UVs

Mode	There are several B-Spline curve mode options: • **Add** - The default mode. When you click in the viewport, additional points are added to the curve. At each control point position, two handles extend out that allows you to adjust the curvature of the curve segment between vertices. While drawing, you can hover over any point or handle where it turns yellow. You can then click-and-drag the control point or handle to further edit the curve. Control points may be added mid-curve by selecting the preceding control, highlighting it yellow, and clicking the position where the new control point is desired (point order is defined by the initial order in which the curve was created. Press **F** before activating the tool to invert the order). • **Edit** - Click-and-drag any of the control points along the curve to change the look to the desired shape. • **Delete** - Click any of the control point along the curve to remove it from the B-Spline.
Point X, **Y**, **Z**	If you want finer point control, you can assign specific X, Y, and Z values in these input fields for the currently-selected control point.
Weight	Specifies the weight of individual vertices.
Closed	Adds an automatic curve segment between the first and last control point positions, producing a closed curve.
Filled	When the **Closed** option is enabled, you can additionally enable the **Filled** option to create a renderable flat surface that is defined by the outlining curve itself. The resulting surface can be tagged like a polygon for adding material definitions.
Make UVs	Activates auto-generation of UV texture coordinates along the curve. The generated UV values are of a single vertical line (V axis in UV), positioning all the control point vertices evenly between 0 and 1. For example, this can be useful for applying a gradient to a rendered curve (the **Render Curves** option is available in the Mesh Item properties.

Bezier

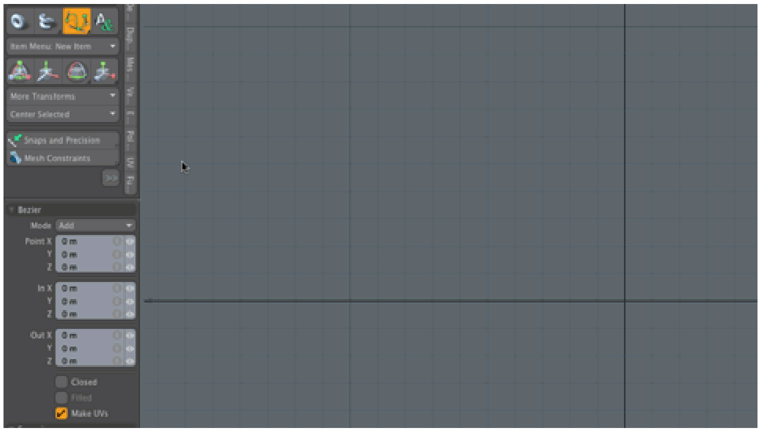
The **Bezier** tool allows you to create a curve defined by control vertices. This tool works like the Pen tool. Bezier curves are a subset of Non-uniform Rational B-spline (NURBS) curves that are composed of two types of control vertices, anchors and tangents. Anchors lie on the curve and determine the origin of the tangents. The tangents determine the shape of the curve leading to an adjacent anchor. Depending on the **Mode** option, slightly different components draw in the interface.



Each time you click in the viewport with the Bezier Curve tool activated, you create an anchor which has no tangents and a sharp corner is created on your curve. When you continue to click in the viewport, additional points are added to the curve.

When you click and drag in the viewport, an anchor and a pair of tangents are created. Dragging moves the tangents to shape the curve. By default, both sides of the tangent move together.

The following animation illustrates the workflow.

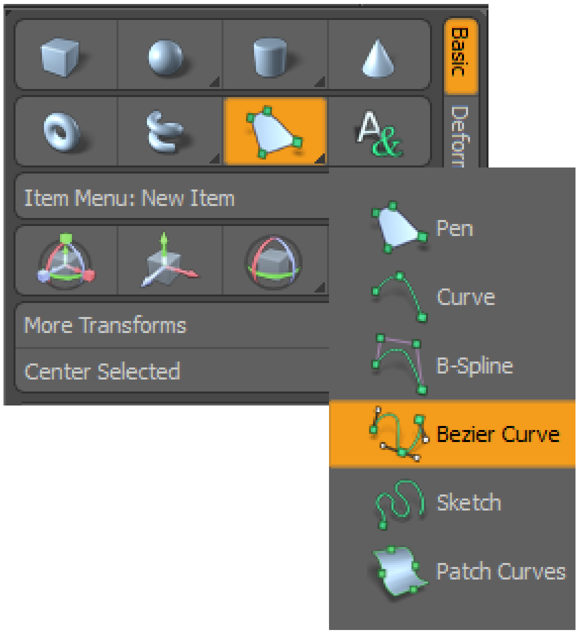


Mode Add
Point X 0 m
Y 0 m
Z 0 m
In X 0 m
Y 0 m
Z 0 m
Out X 0 m
Y 0 m
Z 0 m
☐ Closed
☐ Filled
☑ Make UVs

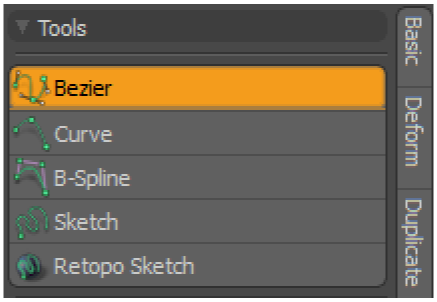
Activating the Bezier Curve Tool

To activate the direct modeling Bezier tool:

- In the **Basic** sub-tab of the modeling toolbox, right-click the Pen tool to reveal additional options, and click **Bezier Curve**.


Basic
Deform
Item Menu: New Item
More Transforms
Center Selected
Pen
Curve
B-Spline
Bezier Curve
Sketch
Patch Curves

- Alternatively, in the **Curve** sub-tab of the modeling toolbox, under **Tools**, click **Bezier**.

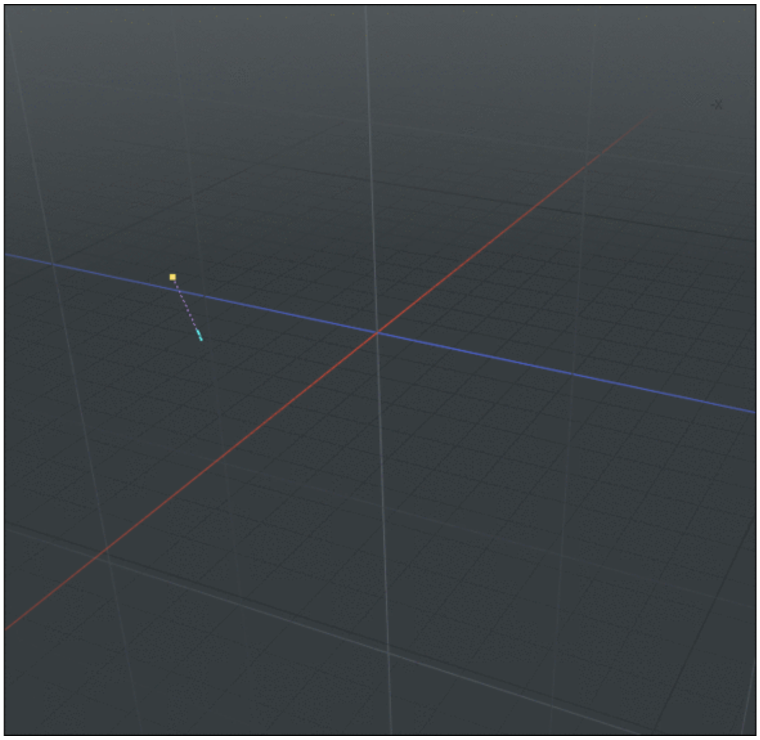

Tools
Bezier
Curve
B-Spline
Sketch
Retopo Sketch
Basic
Deform
Duplicate

Adding a Bezier Curve

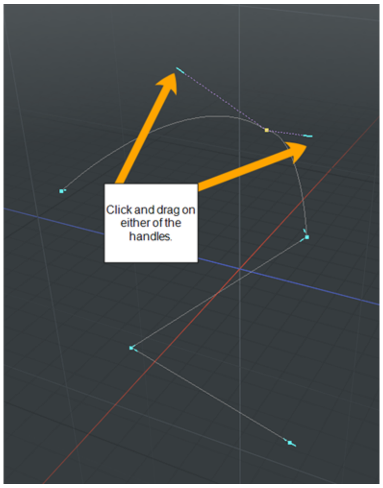
1. Click the **Model** layout on the switcher bar at the top of the UI.
2. In the Toolbox, open the **Basic** sub-tab.
3. Right-click the Pen tool and select **Bezier Curve**.
4. Click and drag in the viewport to define the first anchor for your curve.

   This splits the corner anchor into two points, that move out along the segments, to create a rounded curve section between the straight segments.
5. Click and hold in another position and drag out the tangent handles to shape the curve. Repeat to add additional points on your curve.

   Click on the image below to view an animation.



To change the tangent angle at an anchor, enable **Polygons** selection mode, click on a tangent handle, and drag it into a new position. The sharpness of the corner changes. To reposition an anchor, enable **Polygons** selection mode, click on an anchor, and drag it into a new position.



Editing a Bezier Curve

While drawing, you can hover over any point or handle where it turns yellow. You can then click-and-drag the anchor or handle to further edit the curve.

To move only one side of a tangent, click on the tangent handle and drag it into another position.

Once you drop the tool by pressing the **Space** key, interactive editing is lost but you can re-activate the editing ability by selecting the curve in **Polygons** selection mode.

Deleting Control Points

To remove an anchor:

1. In the right panel, click **Tool Properties**.

   This opens the **Tool Properties** panel.
2. Set **Mode** to **Delete**.
3. Click any of the control points to delete them.

Changing a Bezier Curve to Different Curve Type

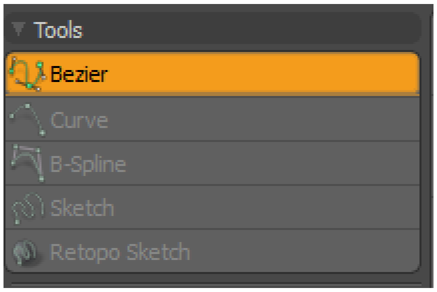
You can change Bezier curves to B-Spline curves or standard curves depending on what you want to achieve in your scene. The Curve Convert mesh operation allows you to convert your selected curves to another specified curve type.

For more information, see Curve Convert.

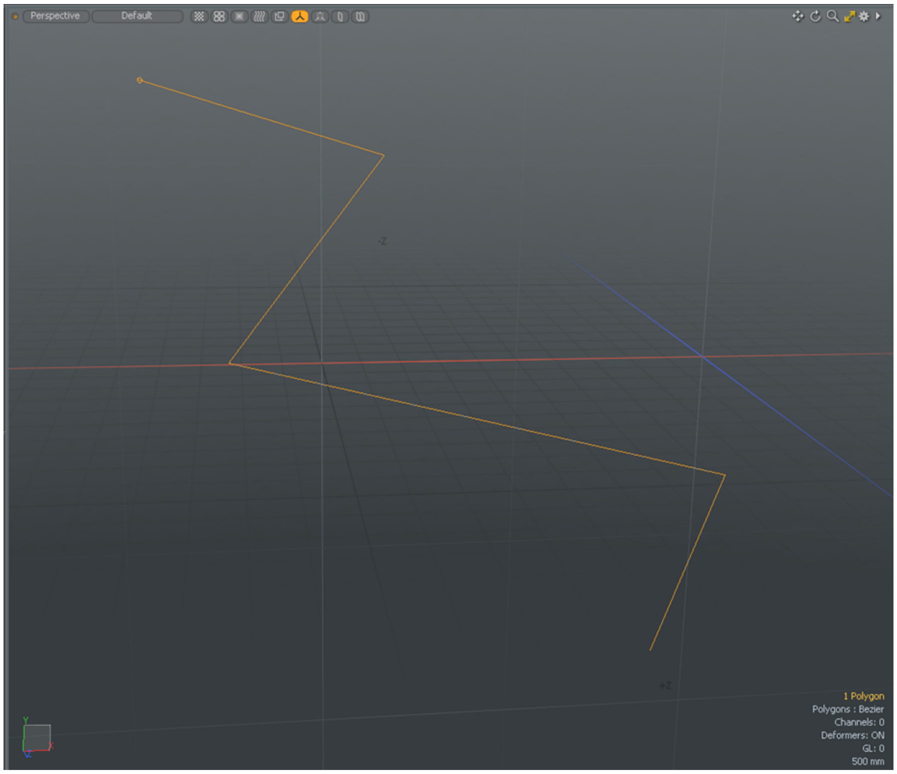
Example

To change the curve type:

1. In the **Model** layout, open the **Curve** tab on the left panel, and click **Bezier**.

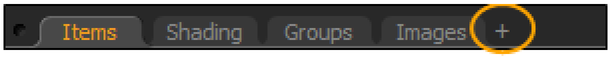


2. In the 3D viewport, click in several locations to create a Bezier curve, and press the **Space** key to drop the tool.

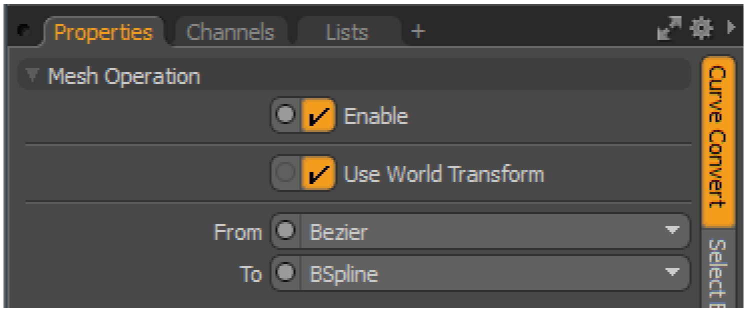

Perspective
Default
1 Polygon
Polygons : Bezier
Channels: 0
Deformers: ON
GL: 0
500 mm

3. Open the **Mesh Ops** tab on the right panel and click **Add Operator** > **Curves** > **Curve Convert**.

> **Tip:** If the **Mesh Ops** tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **New Tab** > **Data Lists** > **Mesh Ops**.


Items
Shading
Groups
Images
+

4. Select the source and target curve type you wish to transform. In this example, on the **Properties** tab, set **From** to **Bezier** and **To** to **B-Spline**.


Properties
Channels
Lists
+
Curve Convert
Select

Mesh Operation

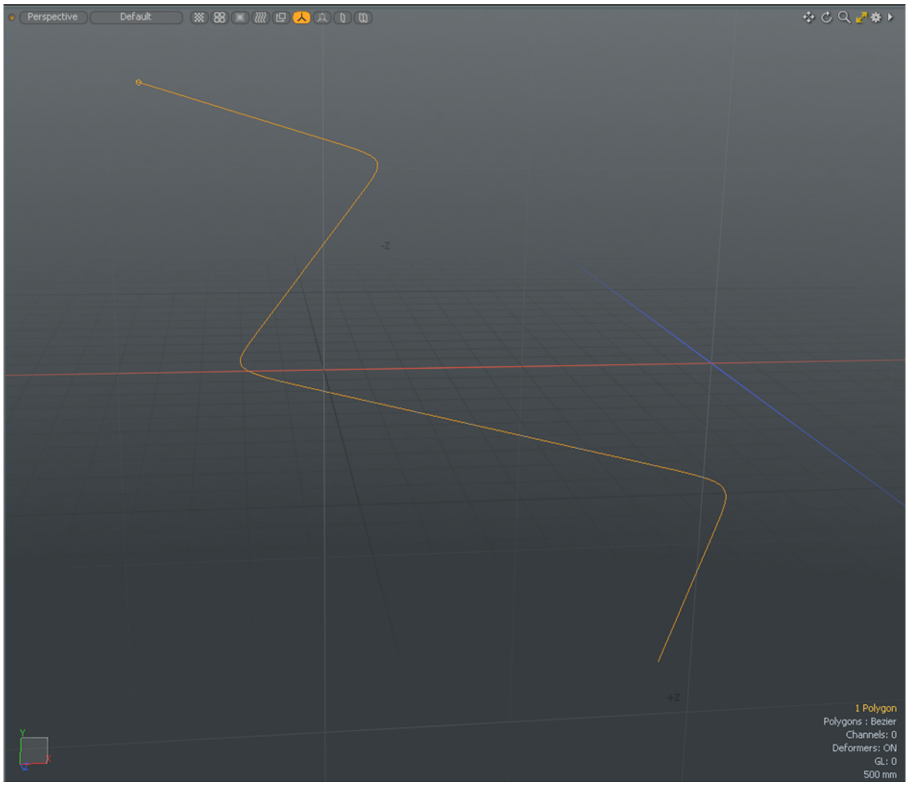
☑ Enable

☑ Use World Transform

From: Bezier

To: BSpline

The Bezier curve is converted to a B-spline curve.

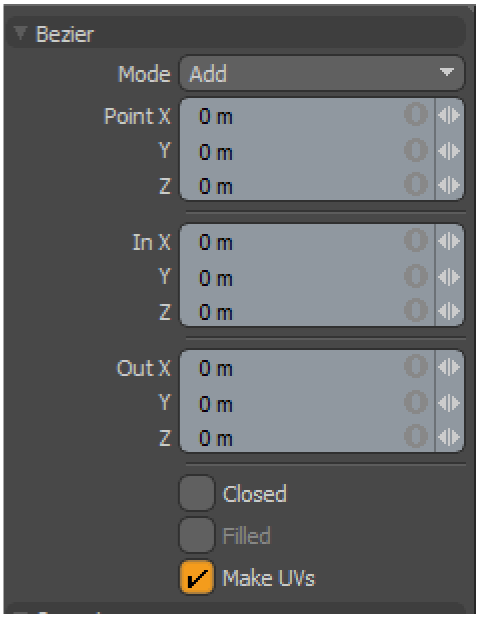

Perspective
Default
1 Polygon
Polygons : Bezier
Channels: 0
Deformers: ON
GL: 0
500 mm

Editing a Bezier Curve

Once you drop the tool, interactive editing is lost but you can re-activate it by selecting the curve in **Polygons** selection mode.

For more information, see Polygon Selection Mode

Bezier Curve Properties



Bezier
Mode Add
Point X 0 m
Y 0 m
Z 0 m
In X 0 m
Y 0 m
Z 0 m
Out X 0 m
Y 0 m
Z 0 m
☐ Closed
☐ Filled
☑ Make UVs

f

Mode	Specifies the Bezier Curve mode. The following options are available: • **Add** - Adds anchors along a curve. This is the default mode. • **Edit** - Freely click-and-drag any of the anchors along the curve to change the look to the desired shape. • **Delete** - Click any of the anchors along the curve to remove it.
Point X, **Y**, **Z**	Allows finer point control by assigning specific X, Y, and Z values in these input fields for the currently-selected anchor.
In X, **Y**, **Z**	Allows finer handle control by assigning specific X, Y, and Z values in these **In** input fields for the currently selected (highlighted) anchor.
Out X, **Y**, **Z**	Allows finer handle control by assigning specific X, Y, and Z values in these **Out** input fields for the currently selected (highlighted) anchor.
Closed	Adds an automatic curve segment between the first and last anchor positions, producing a closed curve. **Tip:** Alternatively, to create a closed curve, right-click on the first anchor in the viewport.

Filled	When the **Closed** option is enabled, you can additionally enable the **Filled** option to create a renderable flat surface that is defined by the outlining curve itself. The resulting surface can be tagged like a polygon for adding material definitions.
Make UVs	Activates auto-generation of UV texture coordinates along the curve. The generated UV values are of a single vertical line (V axis in UV), positioning all the anchor vertices evenly between 0 and 1. For example, this can be useful for applying a gradient to a rendered curve (the **Render Curves** option is available in the Mesh Item properties.

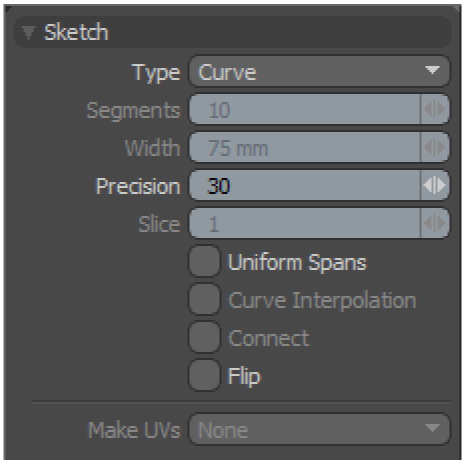
Sketch

The **Sketch** tool allows you to quickly create a curve by drawing in the 3D view. To use the tool, click and drag in any 3D viewport. Use a mouse or tablet to draw the curve and Modo converts your sketch into the curve type you've chosen. Strokes, while active, have a couple widgets associated with them: a plus icon at the root, or base, and a small dot at the tip. Strokes can be manipulated by clicking on the plus icon to move the entire curve, while clicking on the dot at the tip allows you to rotate and scale the tip around the base. With **Constrain to Background** enabled, the sketch tool is a quick and easy way to add polygons, re-topologizing any background geometry.

The **Sketch** tool can be found in the **Model** layout, under the **Curve** sub-tab. In addition, this tool is also found in the **Topology** layout, under the **Tools** sub-tab.

Sketch Tool Options

The following options are available:



Sketch

Type: Curve

Segments: 10

Width: 75 mm

Precision: 30

Slice: 1

☐ Uniform Spans

☐ Curve Interpolation

☐ Connect

☐ Flip

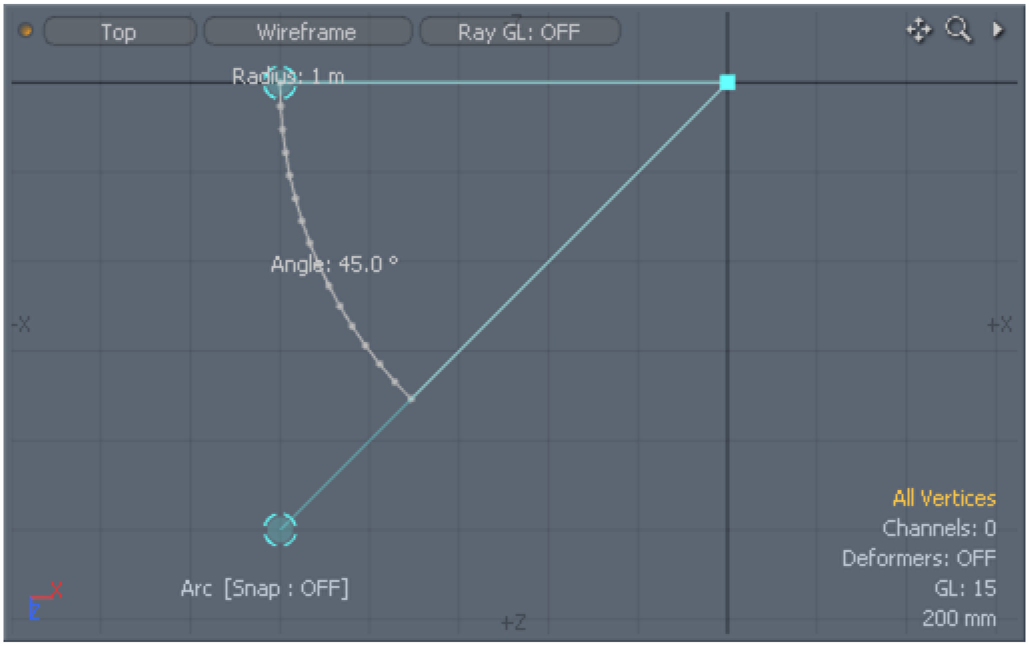
Make UVs: None

Sketch	
Type	Allows you to choose between the various sketch tool types: • **Curve** - The default; the **Sketch** tool creates a spline curve based on the sketch path you create. • **Face** - The **Sketch** tool creates a polygon by drawing a straight line from the end of your sketch back to the beginning, so as to close the loop. • **Lines** - The **Sketch** tool generates a series of 2-vertex polylines that follow the contour of your sketched curve. • **Quads** - The **Sketch** tool creates a strip of quad (4-sided) polygons, centered along the contour of the sketched curve. • **Bezier** - The tool creates a Bezier curve based on the sketch you draw. • **B-Spline** - The tool creates a B-Spline curve based on the sketch you draw. B-splines are spline curves that are defined by at least four vertices and a Weight Map. They provide better continuity than other spline curve types, such as Hermite and Bezier, because they do not go through any of their control points.
Segment	When set to **Quads**, this value specifies the number of polygons in a segment strip. This is somewhat like uniform spans, in that subsequent strokes, if longer or shorter, may contain more or less segments. However, the spans roughly equal those in the initial segment value until changed.
Width	Sets the width of the polygon strip. This setting only affects the **Quads** type. Right-click in the viewport to interactively set the width value.
Precision	Sets how finely Modo interpolates the drawn stroke. Lower values produce strokes with greater numbers of points, higher values produce strokes with

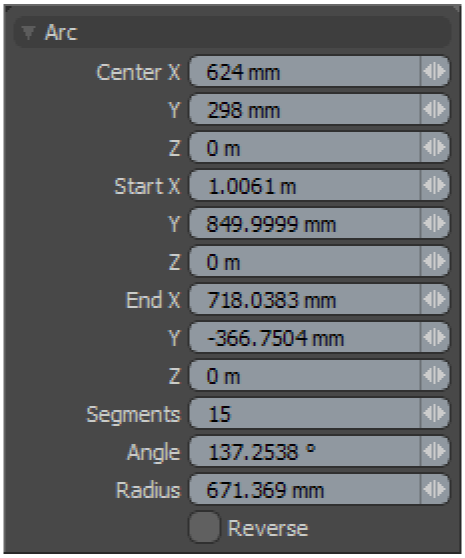
Sketch	
	lower numbers of points. Think of it roughly like the distance, in pixels, the pointer travels before creating a new point.
Slice	When set to **Quads**, this value specifies the number of slices in a polygon strip, so a 16-polygon strip with a slice value of 2, produces 32-polygons > 16 long by 2 wide. • **Uniform Spans** - For curves and polylines, this setting creates a uniform distance between generated vertices, regardless of stroke speed or shape. When set to **Quads**, **Uniform Spans** creates straight line quad strips between the start and end points of the drawn stroke. • **Curve Interpolation** - Continuously evaluates the stroke, when set to the **Quads** type with **Uniform Spans** enabled; the polygon strips follow along the drawn path. • **Connect** - When drawing out quad strip polygons, with **Connect** enabled, adjoining strips in close enough proximity to each other, joins them together. • **Flip** - Reverses the normal facing direction of polygons. If you're drawing out polygons and you don't see anything, they are probably facing away from you. Use the **Flip** option to reverse their facing direction.
Make UVs	When creating **Quads** polygon strips, **Make UVs** automatically makes appropriate UVs. Choose which direction you wish the lengths to align with, U (horizontal) or V (vertical).

Arc

The **Arc** tool provides a means to make closed segments of a disc, like the shape of a pie wedge (or a pie with a piece missing).



To use the **Arc** tool, select the command from the menu bar, under **Geometry** > **Primitive Tools** > **Arc** to activate it. Once activated, you can click-and-drag to define the radius, with the initial click determining the center of the wedge. Once releasing the mouse button, you can then click-and-drag any of the tool handles to make further adjustments. Handles change to a yellow color when hovering, to indicate they can be adjusted. The first handle determines the overall angle of the wedge as well as its radius, while the second handle indicates the angle of the wedge itself. Dragging its handle further from the center allows increasingly finer control over the angle. The center position determines the center location of the wedge. Pressing and holding **Shift** when clicking in the viewport initiates a new wedge. When the tool is active, the following options in the tool's properties panel are available for modifying the currently-active arc.


Arc

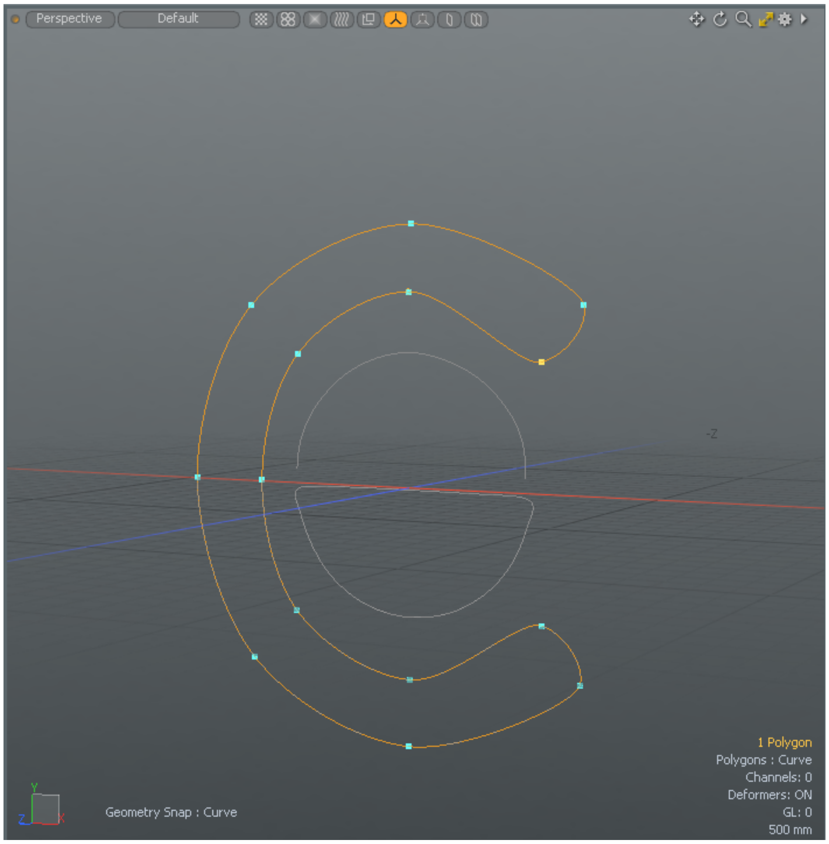
Center X	624 mm
Y	298 mm
Z	0 m
Start X	1.0061 m
Y	849.9999 mm
Z	0 m
End X	718.0383 mm
Y	-366.7504 mm
Z	0 m
Segments	15
Angle	137.2538 °
Radius	671.369 mm
☐ Reverse	

The following options are available for adjusting the **Arc** tool:

Arc	
Center X, Y, Z	The X, Y, and Z coordinate values that determine the center position of the arc. You can interactively adjust this value by clicking and dragging the handle in the 3D viewport.
Start X, Y, Z	The X, Y, and Z coordinate values that determine the origin angle of the arc. You can interactively adjust this value by clicking and dragging the handle in the 3D viewport.
End X, Y, Z	The X, Y, and Z coordinate values that determine the termination angle of the arc. You can interactively adjust this value by clicking and dragging the handle in the 3D viewport.
Segments	Determines the number of edge segments that define the curve. Higher values produce a smoother, more round-looking arc.
Angle	Determines the total length of the arc as a number of degrees from **Start** to **End**. A full circle has 360°.
Radius	Determines the length or size of the arc from its **Center** point to the edge of the arc.
Reverse	Inverts the arc value, creating the reverse shape and making it possible to create circles missing a wedge shape instead of creating just a wedge piece.

Adding Curves

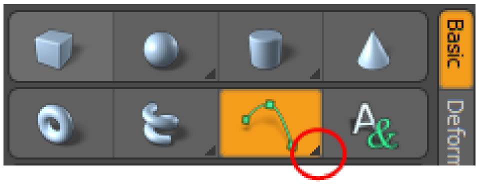
The Curve tool allows you to create a spline curve by clicking down a series of control points through which the curve is interpolated. This is a hermite curve type, which can be very useful for creating modeling guides, quickly roughing out a shape, or setting up an outline for lofting with the extrude, lathe, or other multiply tool.


Perspective
Default
-Z
Geometry Snap : Curve
1 Polygon
Polygons : Curve
Channels: 0
Deformers: ON
GL: 0
500 mm

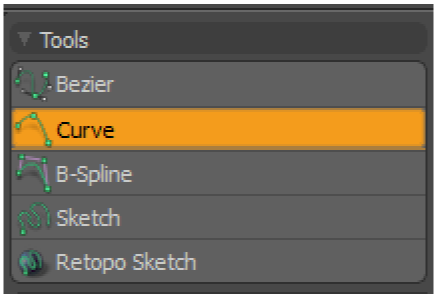
Activating the Curve Tool

1. Activate the Curve tool in one of the following ways:

- In the Basic sub-tab of the modeling toolbox, right-click the Pen tool to reveal additional tool options, and select **Curve**.


Basic
Deform

- In the **Curve** sub-tab of the modeling toolbox, click **Curve**.


Tools
Bezier
Curve
B-Spline
Sketch
Retopo Sketch

- In the menu bar, under **Geometry** > **Curve Palette**, click **Curve**.

2. Click in the 3D viewport to position your control points.

   Adding new control points to the curve is simple. Click anywhere on the screen without selecting an existing control point. This adds a new control point to the curve.

Note: The new control point is added after the currently-selected point, so if you want to add a point to a specific location on the curve, first select the point before it.

3. Once you have created several control points, you can click directly on an existing control point to drag it around, allowing you to change the shape of the curve.

Tip: To create a closed curve, right-click on the first control point. Modo adds a curve segment between the first and last control points.

4. Drop the tool by pressing the **Space** key.

Editing a Curve

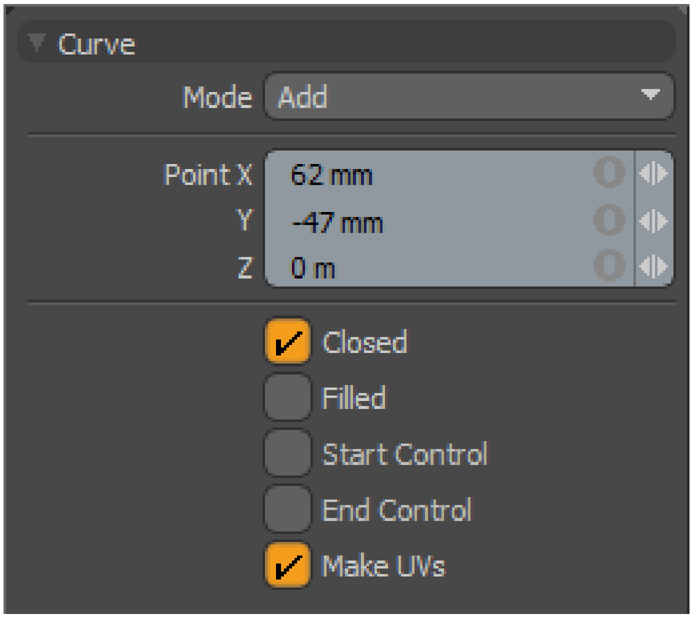
Once you drop the tool, interactive editing within the tool itself is lost but you can re-activate the editing ability by selecting the curve in **Polygons** selection mode.

For more information, see Polygon Selection Mode

Tip: You can convert selected curves to either a Adding Curves, B-Spline, or Bezier curve type. For more information, see Curve Convert

Curve Properties

You can find the **Curve** tool's properties in the bottom left of the UI.


Curve

Mode Add

Point X 62 mm

Y -47 mm

Z 0 m

☑ Closed

☐ Filled

☐ Start Control

☐ End Control

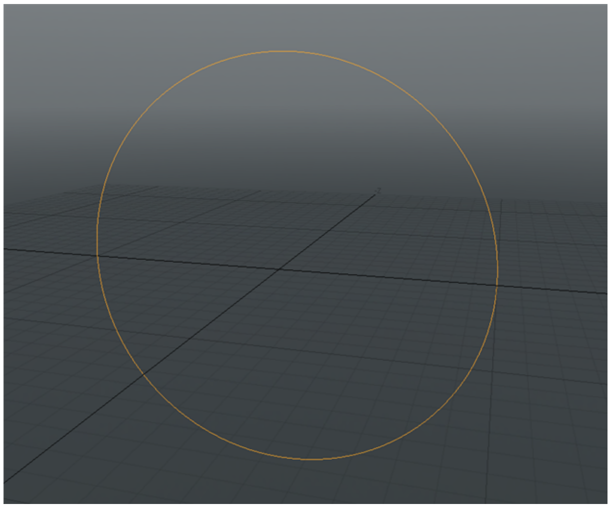
☑ Make UVs

Mode	Allows you to change the behavior of the cursor. The following options are available: • **Add** - Adds a curve. This is the default setting. • **Edit** - click-and-drag any of the control points along the curve to change the look to the desired shape. • **Delete** - Click any of the control points along the curve to remove it from the B-Spline.
Point X, **Y**, **Z**	The X, Y, and Z values of the currently-active point can be set precisely by typing in the desired 3D coordinates.
Closed	When enabled, adds an automatic curve segment between the first and last control point positions, producing a closed curve.
Filled	When the **Closed** option is enabled, you can enable the **Filled** option to create a renderable flat surface that is defined by the outlining curve itself. The resulting surface can be tagged as a polygon for adding material definitions.
Start Control	Converts the first point in the curve to a control point that does not contribute to the actual curve length, but acts as a handle so that the curvature at the new first point can be modified. When activated, the curve length from the control point to the next point is drawn in the handle color to indicate that it's no longer part of the geometry but simply a handle.
End Control	Converts the last point in the curve to a control point that does not contribute to

	the actual curve length, but acts as a handle so that the curvature at the new end point can be modified. When activated, the curve length from the control point to the next point is drawn in the handle color to indicate that it is no longer part of the geometry but simply a handle.
Make UVs	Activates auto-generation of UV texture coordinates along the curve.

Circle

The **Circle** primitive is a closed circular **Bezier** or **Spline** curve. The **Bezier** type is equivalent to a curve made by the Bezier tool. The **Spline** type is a Hermite Cubic spline, equivalent to that made by the Adding Curves tool.



Note: You can fill the shape using the Curve Fill tool.

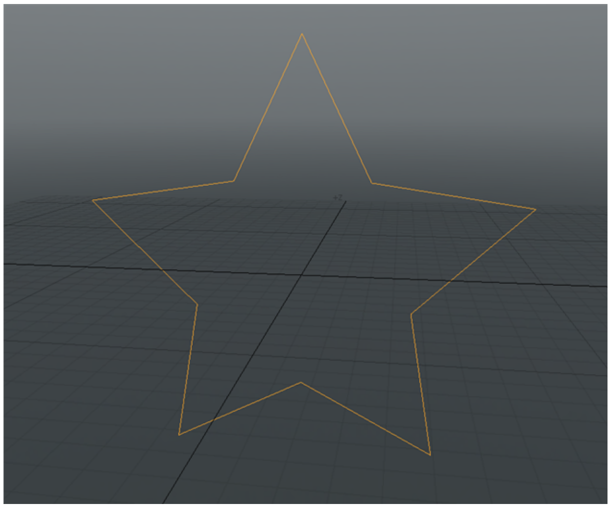
Circle Properties

Circle	
Enable	Toggles the primitive.
Type	Selects the curve type:

	• **Bezier** - Draw using Bezier control points. • **Spline** - Draw using Hermite Cubic spline type. This locates all vertices along the path of the circle.
Position X, Y	Determines the X, Y position along the plane defined by **Axis**.
Radius	Sets the circle radius.
Axis	Chooses the plane on which to draw the circle: • **X** - Align to the YZ plane. • **Y** - Align to the XZ plane. • **Z** - Align to the XY plane.

Star

The **Star** primitive is a closed star-shaped **Bezier** or **Polyline** curve. The Bezier type is equivalent to a curve made by the Bezier tool. The **Polyline** type is a connected sequence of line segments.



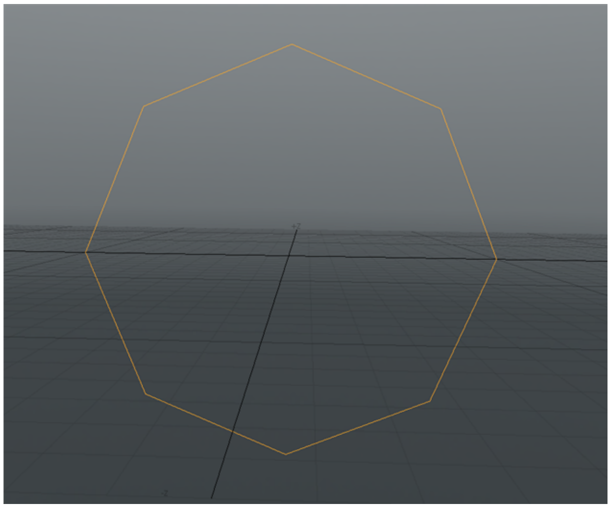
Note: You can fill the shape using the Curve Fill tool.

Star Properties

Circle	
Enable	Toggles the primitive.
Type	Selects the curve type: • **Bezier** - Draw using Bezier curve control points. • **Polyline** - Draw using a connected series of lines.
Position X, Y	Sets the X, Y position along the plane defined by **Axis**.
Size X, Y	Sets the width and height.
Scale	Sets the scale of the distance of the inner points compared to the outer points. At **100%** the primitive will result in a regular polygon that has twice the number of sides given by **Sides**.
Sides	Sets the number of arms for the star, in other words, the number of outer points.
Axis	Chooses the plane on which to draw the star: • **X** - Align to the YZ plane. • **Y** - Align to the XZ plane. • **Z** - Align to the XY plane.

N-Sided

The **N-Sided** primitive is a regular polygon with a given number of sides. The polygon can be made from a **Bezier** curve or a **Polyline**. The Bezier type is equivalent to a curve made by the Bezier tool. The **Polyline** is a connected sequence of line segments. You can adjust the number of sides using the **Sides** setting, which can range from 3 - 128.



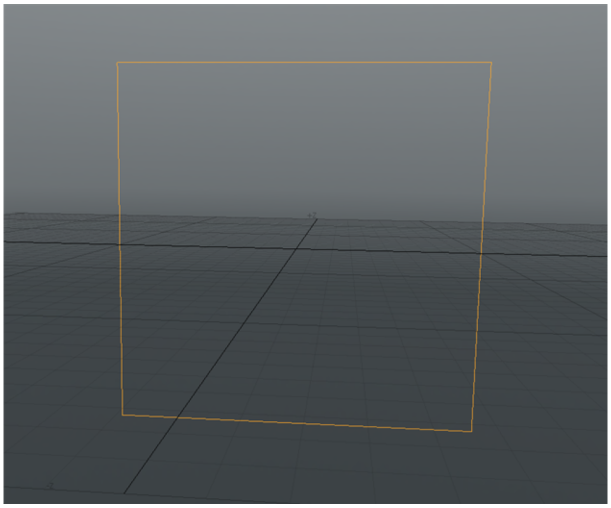
Note: You can fill the shape using the Curve Fill tool.

N-Sided Properties

N-Sided	
Enable	Toggles the primitive.
Type	Selects the curve type: • **Bezier** - Draw using Bezier curve control points. • **Polyline** - Draw using a connected series of lines.
Position X, Y	Sets the X, Y position along the plane defined by **Axis**.
Size X, Y	Sets the width and height.
Sides	Sets the number of sides for the polygon.
Axis	Chooses the plane on which to draw the polygon: • **X** - Align to the YZ plane. • **Y** - Align to the XZ plane. • **Z** - Align to the XY plane.

Rectangle

The **Rectangle** primitive is a rectangle or square. The polygon can be made from a **Bezier** curve or a **Polyline**. The Bezier type is equivalent to a curve made by the Bezier tool. The **Polyline** is a connected sequence of line segments. The width and height of the polygon are adjusted using the **Size X** and **Size Y** settings.



Note: You can fill the shape using the Curve Fill tool.

Rectangle Properties

Rectangle	
Enable	Toggles the primitive.
Type	Selects the curve type: • **Bezier** - Draw using Bezier curve control points. • **Polyline** - Draw using a connected series of lines.
Position X, Y	Sets the X, Y position along the plane defined by **Axis**.

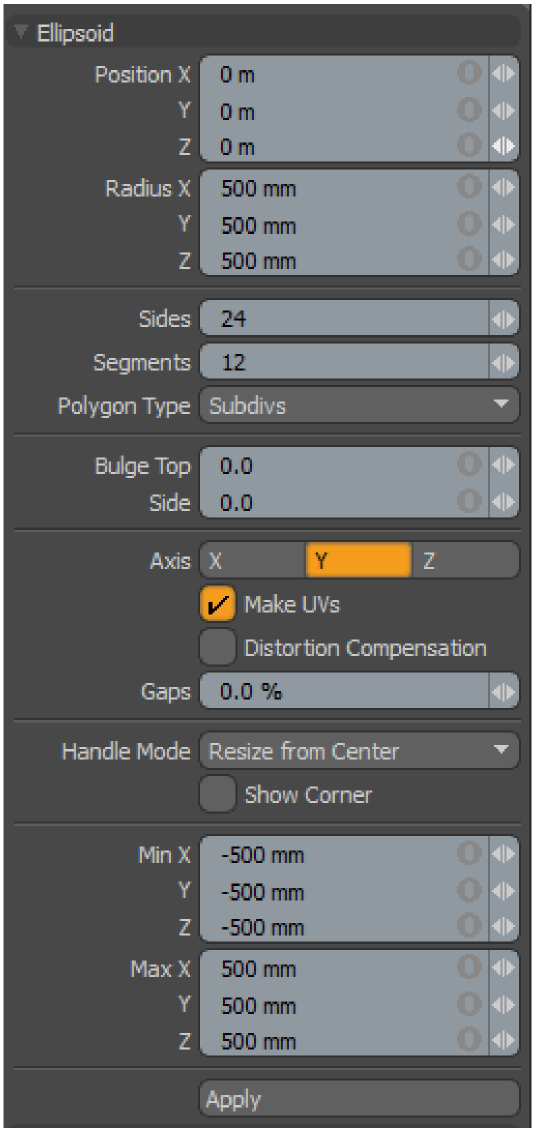
Size X, Y	Sets the width and height.
Axis	Chooses the plane on which to draw the polygon: • **X** - Align to the YZ plane. • **Y** - Align to the XZ plane. • **Z** - Align to the XY plane.

Ellipsoid

The **Ellipsoid** primitive provides a simple method for creating ellipses or "pills".

Note: For information on how to add primitives to your scene, see Adding Primitives.

When the Ellipsoid tool is active, the following options are available for editing the primitive:



Ellipsoid

Position X 0 m
Y 0 m
Z 0 m
Radius X 500 mm
Y 500 mm
Z 500 mm
Sides 24
Segments 12
Polygon Type Subdivs
Bulge Top 0.0
Side 0.0
Axis X [✓ Y] Z
☑ Make UVs
☐ Distortion Compensation
Gaps 0.0 %
Handle Mode Resize from Center
☐ Show Corner
Min X -500 mm
Y -500 mm
Z -500 mm
Max X 500 mm
Y 500 mm
Z 500 mm

Ellipsoid	
Position X, **Y**, **Z**	These three values establish the 3D location for the center of the primitive.
Radius X, **Y**, **Z**	These three values are used to establish the dimensions of the ellipsoid. Set to **1m**, **1m**, **1m** (by default), provides us with a 2-meter ball.
Sides	The sides are the latitudinal edges of the ellipsoid running from top to bottom. By default, the primitive ellipsoid uses 24 sides. By increasing this value, you can increase the number of "cuts" in the shape. This is useful if you plan to deform the ellipsoid, as these segments act as hinges during deformation operations. This also increases the smoothness of the shape by adding extra facets.

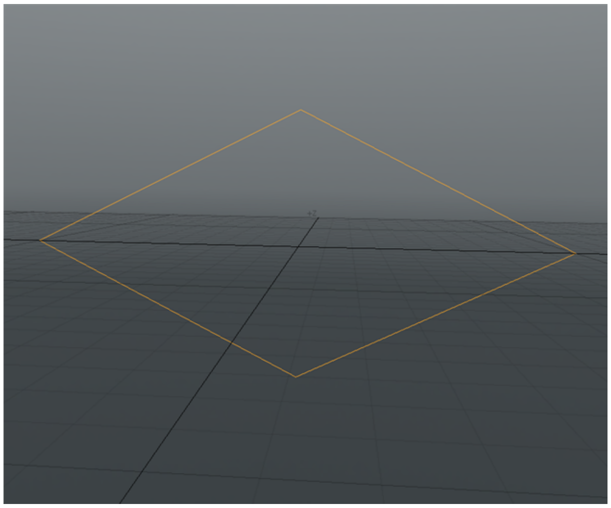
Ellipsoid	
Segments	The segments are the longitudinal edges of the ellipsoid running from right to left. By default, the primitive uses 12 segments. By increasing this value, you can increase the number of "cuts" in the primitive. This is useful if you plan to deform the shape, as these segments act as hinges during deformation operations. It also increases the smoothness of the shape by adding extra facets.
Polygon Type	You can use this option to determine the type of polygons created: • **Face** - creates standard, unsmoothed polygon faces. • **Subdivs** - creates Subdivision Surface smoothed polygons. • **Catmull-Clark** - creates Pixar Catmull-Clark smoothed polygons.
Bulge Top	Increasing this value changes the shape from an elongated sphere to a rounded, capped cylinder.
Side	Increasing this value pushes the edges to the corners of the shape, converting it into a flat-sided ellipse.
Axis	This X, Y, Z choice allows you to quickly change the orientation of the primitive.
Make UVs	When this button is active a UV map is automatically generated for the geometry created with the tool. This is a very useful option if you plan to UV map the model you are creating from the primitive as it provides a baseline UV map that you can massage later in the modeling process. In many cases this can reduce the amount of work required to map the model.
Distortion Compensation	Adjusts the aspect ratio and related area size of UV polygons to reduce the distortion in the primitives UVs. Use this option when you create a **Ellipsoid** that does not have equal radius values. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**. Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final values.
Gaps	Specifies empty space at the boundary of the UV space or between UV parts. The UV unit is specified using a percentage value.

Ellipsoid	
Handle Mode	Provides three types of behavior for corner and side tool handles: • **Resize from Center** - Resizes both sides of the primitive at the center, when dragging a corner or side handle. • **Resize** - Resizes the shape at the opposite side, when dragging a corner or side handle. • **Move All** - Moves the primitive instead of resizing. **Tip:** You can also move the shape by clicking and holding the **Ctrl** key when in another handle mode.
Show Corner	When enabled, displays the bounding box corner handles.
Min XYZ/Max XYZ	You can also define the shape based on specific X, Y, and Z bounding box locations in 3D space, which can be specified here. This makes it easy to place an ellipsoid on the ground plane, for instance, by making the **Min Y** value **0** and the **Max Y** value **1m**.

Tip: The **Ellipsoid** primitive supports symmetrical creation. When Applying Precision is activated, creating the primitive shape creates an identical version across the specified axis.

Diamond

The **Diamond** primitive is a diamond shaped regular polygon. The diamond can be made from a **Bezier** curve or a **Polyline**. The Bezier type is equivalent to a curve made by the Bezier tool. The **Polyline** is a connected sequence of line segments. The width and height of the diamond are adjusted using the **Size X** and **Size Y** settings.



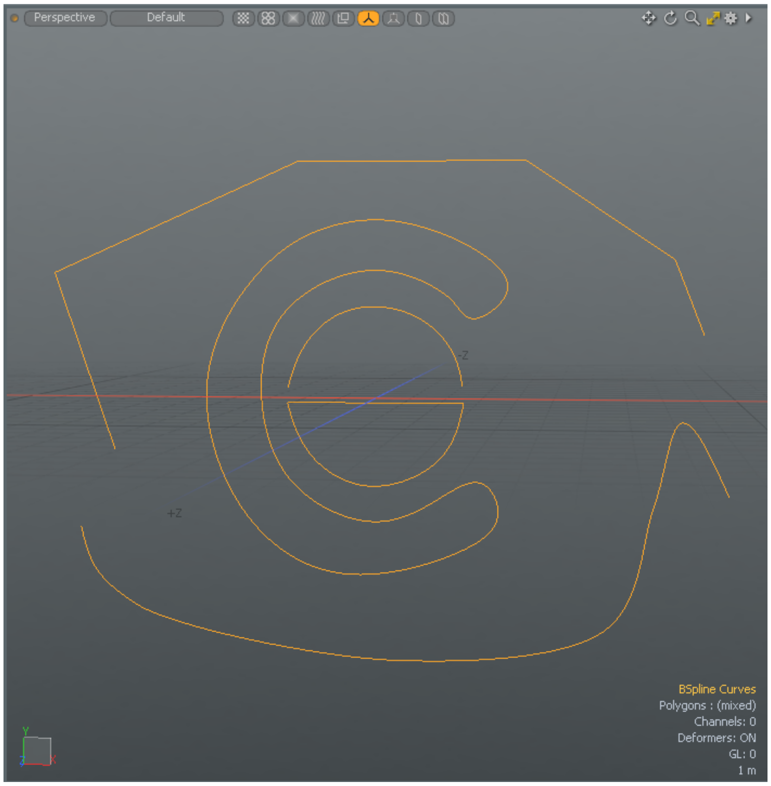
Note: You can fill the shape using the Curve Fill tool.

Diamond Properties

Diamond	
Enable	Toggles the primitive.
Type	Selects the curve type: • **Bezier** - Draw using Bezier curve control points. • **Polyline** - Draw using a connected series of lines.
Position X, Y	Sets the X, Y position along the plane defined by **Axis**.
Size X, Y	Sets the width and height.
Axis	Chooses the plane on which to draw the polygon: • **X** - Align to the YZ plane. • **Y** - Align to the XZ plane. • **Z** - Align to the XY plane.

Curve Convert

The **Curve Convert** procedural mesh operation allows you to convert selected curves to either a Adding Curves, B-Spline, or Bezier curve type. Each curve type has its own benefits, depending on what you want to achieve in your scene.



- **Bezier** - Creates a curve defined by control vertices. These curves are composed of two types of control vertices, anchors and tangents of which you can manipulate.
- **B-Spline** - Creates a curve with at least four vertices and a weight map. This curve provides better continuity than other spline curve types because it doesn't go through any of its control points.
- **Curve** - Creates a standard curve, which can be very useful for creating modeling guides, quickly roughing out a shape, or setting up an outline for lofting with the extrude, lathe, or other multiply tools.

Benefits of Using the Curve Convert Procedural Tool

- Convert multiple curves (closed or open) to be the same type or a multitude of different types.
- Use the converted curves with other mesh operations.

- Convert curves imported from other file formats, such as Illustrator.
- Use selection sets to convert a specific curve. For more information, see Using Selection Sets.

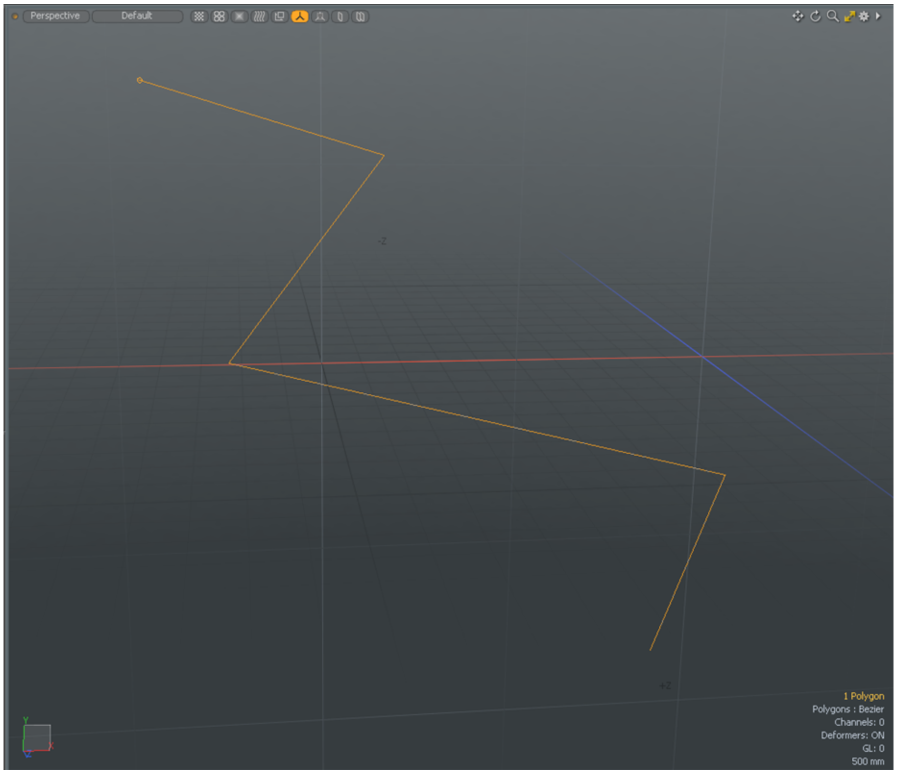
Activating the Curve Convert Tool

- In the **Mesh Ops** tab, click **Add Operator**, and under **Mesh Operations** > **Curves**, double-click **Curve Convert**.
- In the **Schematic Fusion** layout or the **Setup** layout's **Schematic** view, click **Add...**, and under **Mesh Operations** > **Curves**, double-click **Curve Convert**.

Example

To change the curve type:

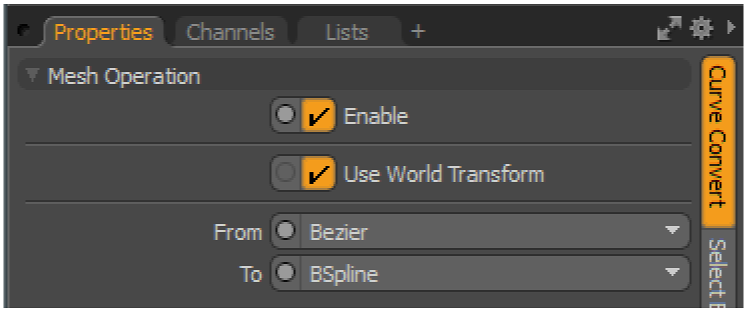
1. In the **Model** layout, open the **Curve** tab on the left panel, and click **Bezier**.
2. In the 3D viewport, click in several locations to create a Bezier curve, and press the **Space** key to drop the tool.



3. Open the **MeshOps** tab on the right panel and click **Add Operator** > **Curves** > **Curve Convert**.

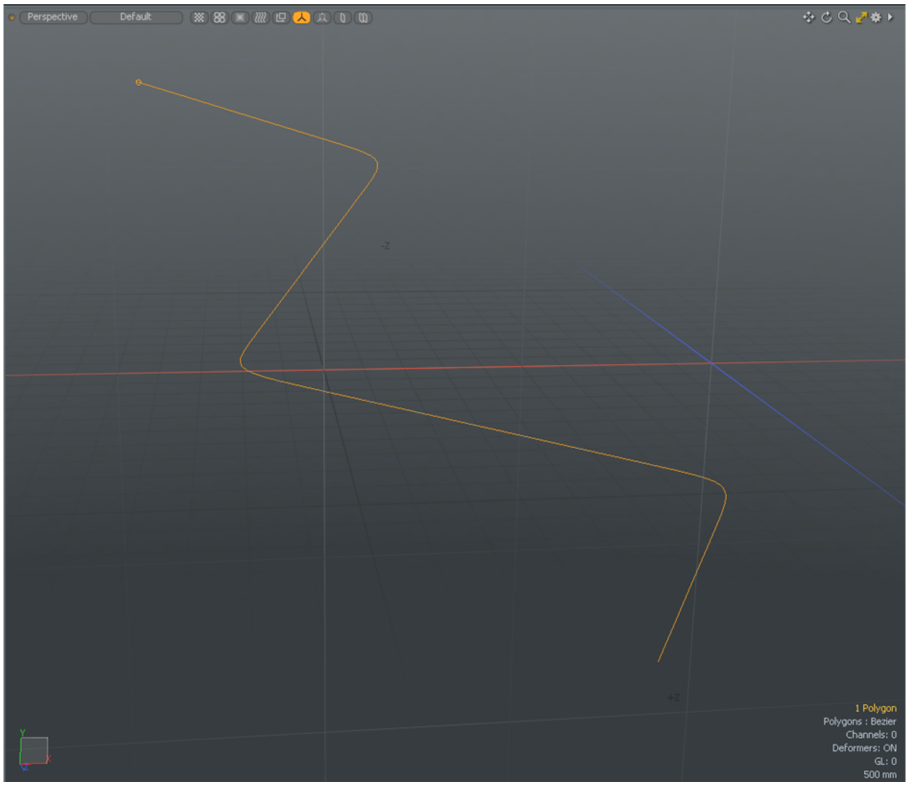
Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **New Tab** > **Data Lists** > **Mesh Ops**.

4. Select the source and target curve type you wish to transform. In this example, in the **Properties** tab, set **From** to **Bezier** and **To** to **BSpline**.


Properties
Channels
Lists
+
Mesh Operation
Curve Convert
Select

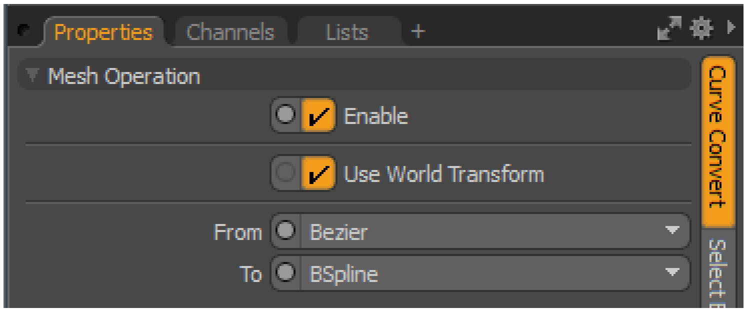
☑ Enable
☑ Use World Transform
From: Bezier
To: BSpline

The selected Bezier curve is converted to a B-Spline curve.


Perspective
Default
1 Polygon
Polygons : Bezier
Channels: 0
Deformers: ON
GL: 0
500 mm

Tip: Once you drop the tool by pressing the **Space** key, you can no longer edit your curve. To re-activate editing, select the curve in Polygon Selection Mode

Curve Convert Properties



Enable: ☑
Use World Transform: ☑
From: Bezier
To: BSpline

Enable	Toggles Curve Convert on and off.
Use World Transform	When enabled, the mesh operation maintains the world space on the input layer.
From/To	**From** specifies the original curve type for the selected curve(s), and **To** specifies the curve type you want to transform your selected curve(s) to. The following options are available: • **Bezier** - Creates a curve defined by control vertices. These curves are composed of two types of control vertices, anchors and tangents that you can manipulate. • **B-Spline** - Creates a curve with at least four vertices and a weight map. They provide better continuity than other spline curve types, because they don't go through any of their control points. • **Curve** - Creates a standard curve, which can be very useful for creating modeling guides, quickly roughing out a shape, or setting up an outline for lofting with the extrude, lathe, or other multiply tools.

Curve to Bezier Mesh Operation

The **Curve to Bezier** mesh operation is a specialized operation that allows you to convert an existing curve to a Bezier curve, with the ability to control the tangent length of the Bezier handles, so you can sharpen or smooth the points on a curve, and animate the value using the **Tangent Length** channel.

The normals used to create the Bezier tangents usually come from an array generated using the Array Modifiers, but you can use any valid vector array input.

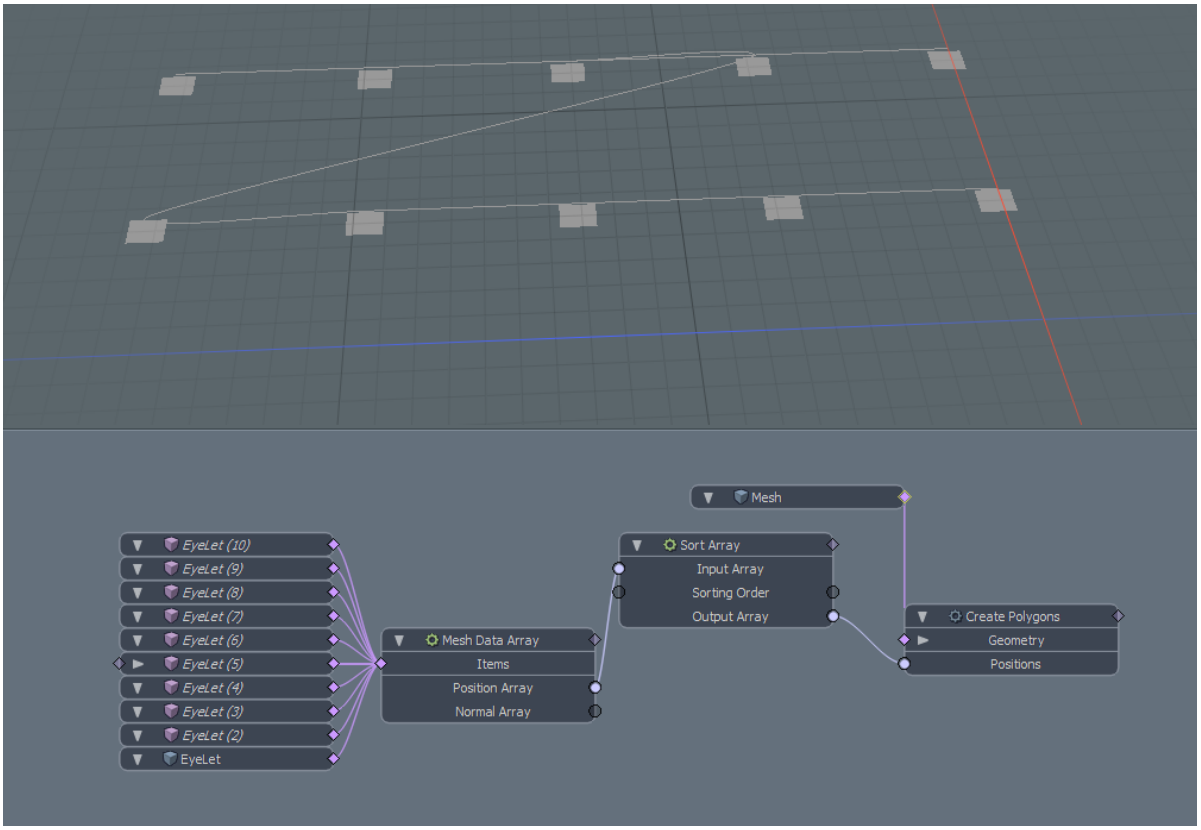
Note: For more information on arrays and array modifiers, see Array Modifiers.

Note: In order for the mesh operation to work, the number of normals in the input tangent normal array must match the number of existing points in the curve, or the normal array must be empty. When using normals from a Mesh Data Array, this happens automatically.

A typical example for using the Curve to Bezier Mesh operation is as follows:

A Mesh Data Array is generated based on the position and normals of items in your scene. In the scene below, the **Eyelet** item and its instances are fed into the **Mesh Data Array**, which uses the items to build an array with **Y** specified as the **Item Axis**. Then we use a **Sort Array** to define the order of the elements.

The sorted array is used by the Create Polygons mesh operation to generate a curve. This is the curve that the Curve to Bezier node will operate on.

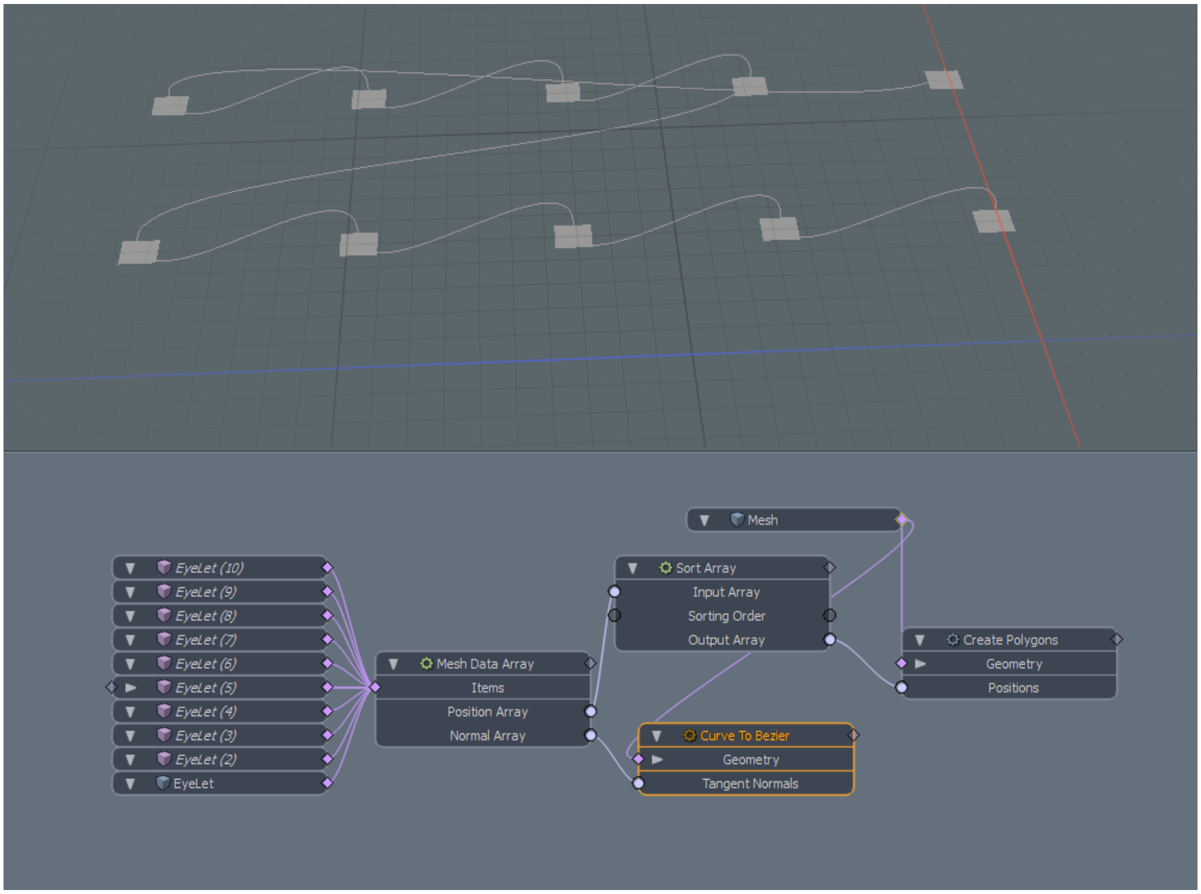

Mesh
EyeLet (10)
EyeLet (9)
EyeLet (8)
EyeLet (7)
EyeLet (6)
EyeLet (5)
EyeLet (4)
EyeLet (3)
EyeLet (2)
EyeLet
Mesh Data Array
Items
Position Array
Normal Array
Sort Array
Input Array
Sorting Order
Output Array
Create Polygons
Geometry
Positions

You can add Curve to Bezier in the Schematic by clicking **Add...**, then under **Mesh Operations** > **Curves**, double-click **Curve to Bezier**.

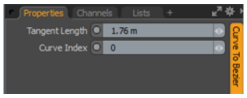
In this example, the **Curve to Bezier** node is connected to the **Mesh Data Array**'s **Normal Array** channel, and uses the values from that channel to generate tangent handles. It is also connected to the **Mesh** layer containing the curve.

The curve is converted into a Bezier curve.

To adjust its tangent length, select the **Curve to Bezier** node in the Schematic viewport. Its properties open in the **Properties** panel on the right, and you can edit the **Tangent Length** value. Higher values sharpen the points, while lower values smooth them.


Mesh
EyeLet (10)
EyeLet (9)
EyeLet (8)
EyeLet (7)
EyeLet (6)
EyeLet (5)
EyeLet (4)
EyeLet (3)
EyeLet (2)
EyeLet
Mesh Data Array
Items
Position Array
Normal Array
Sort Array
Input Array
Sorting Order
Output Array
Create Polygons
Geometry
Positions
Curve To Bezier
Geometry
Tangent Normals

Curve to Bezier Properties


Properties
Channels
Lists
+
Curve To Bezier

Tangent Length 1.76 m

Curve Index 0

The Curve to Bezier mesh operation has the following properties:

Tangent Length	Set the tangent length to sharpen or smooth the points on the curve.

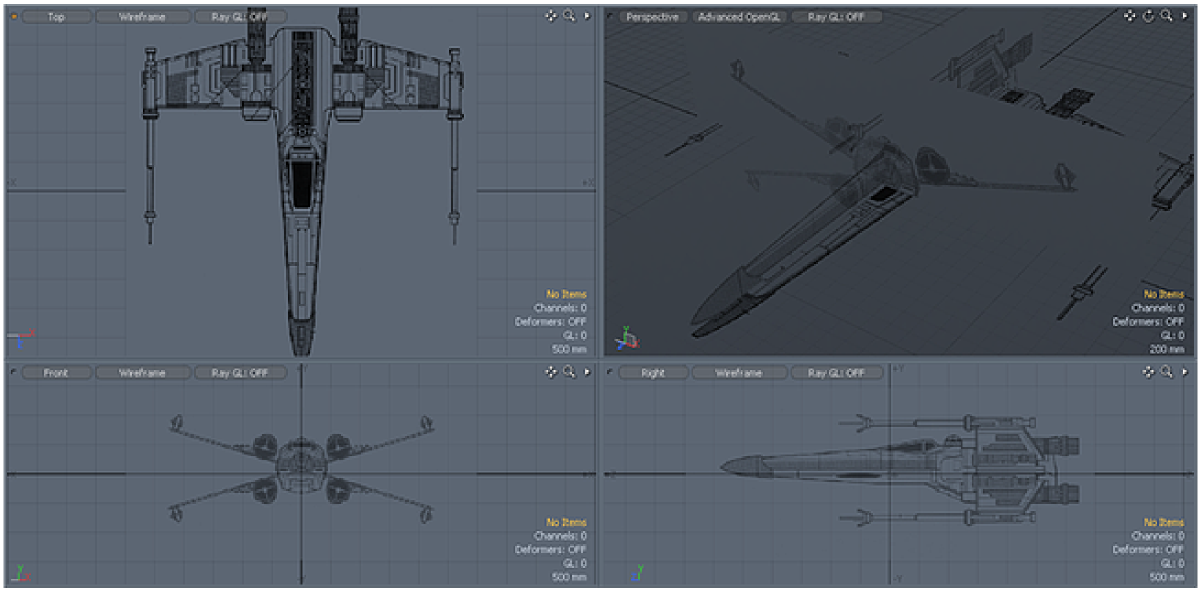
Curve Index	When you have multiple curves in a mesh layer, this specifies which one to use.

Using Other Geometry Tools

Learn about additional geometry creation tools in Modo.

Modeling with Images

Backdrop items are modeling aids for you to add images into a background plane. The backdrop image acts as reference for geometry creation and can be a photograph, a line drawing, or even a blueprint. Because backdrop items are only meant for reference, they do not contribute to the rendered output. They are items in the **Item List**, and Modo saves them with the scene. You can manipulate backdrop items with the standard transform tools.


Top
Wireframe
Ray GL: OFF
No Items
Channels: 0
Deformers: OFF
GL: 0
500 mm
Perspective
Advanced OpenGL
Ray GL: OFF
No Items
Channels: 0
Deformers: OFF
GL: 0
200 mm
Front
Wireframe
Ray GL: OFF
No Items
Channels: 0
Deformers: OFF
GL: 0
500 mm
Right
Wireframe
Ray GL: OFF
No Items
Channels: 0
Deformers: OFF
GL: 0
500 mm

To add a background image:

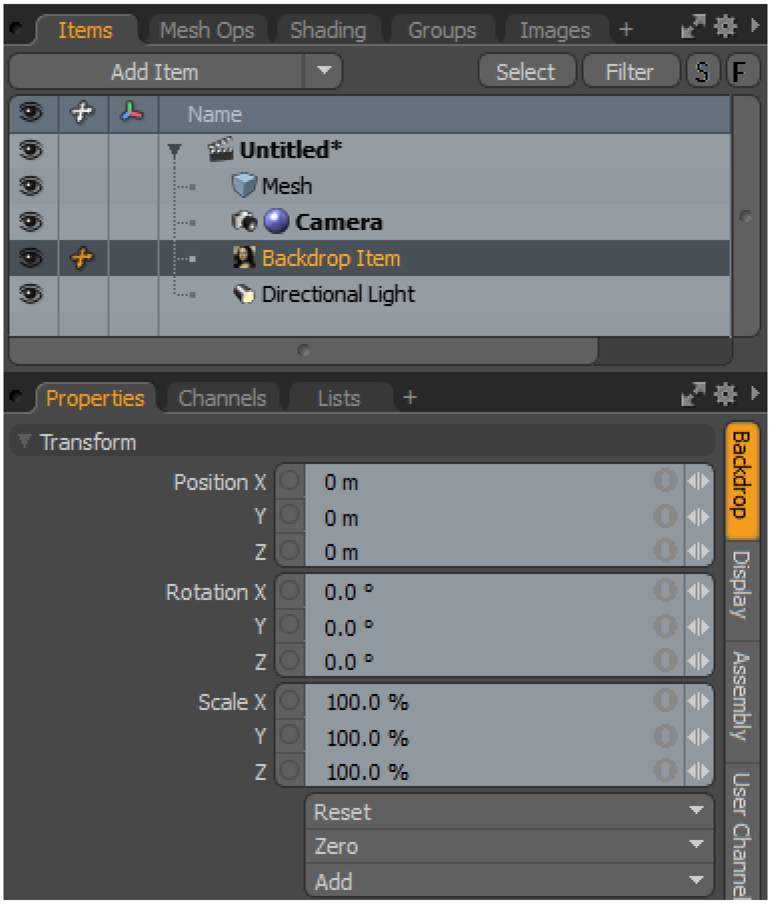
1. Click **Item > Create Backdrop**.

   Alternatively, on the right panel, click **Add Item** > **Backdrop Item**.
2. On the right panel, in the Backdrop properties, click **Images** tab, and click **Add Clip** > **(load image)**.
3. Navigate to the image you want to use and click **Open**.
4. In the **Items** list, select **Backdrop Item**.

5. On the right panel, in the **Properties** tab, open the **Backdrop** tab.
6. Set the appropriate properties.

Tip: For multiple viewports (such as front, side, and back), you create three different backdrop items with the appropriate image and Projection Type specified for each.

Backdrop Item Transform Properties

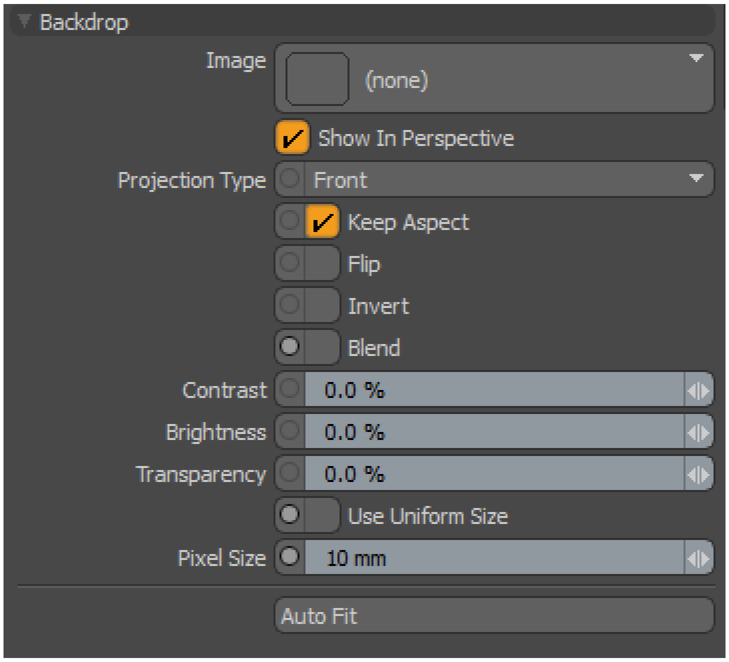

Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
S
F
Name
Untitled*
Mesh
Camera
Backdrop Item
Directional Light
Properties
Channels
Lists
Transform
Backdrop
Display
Assembly
User Channel
Reset
Zero
Add

Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Position XYZ	Specify the positional values for placing the backdrop image within the viewport. All transforms originate from the plane's center point.
Rotation XYZ	Adjust the backdrop item's rotation in the viewport. However, in most instances, consider adjusting the **Projection Type** option to position the image toward a specific viewport instead of rotating the image.

Scale XYZ	Adjust the size of the backdrop items.
Reset	Resets the selected transform values to (0,0,0) to return the items to the World Space center position.
Zero	Returns the item's center position to the World Space center (0,0,0) without changing the position of the backdrop item itself.
Add	Adds the selected set of transforms to the **Channel** list while keeping the default 0,0,0 values. (To override a channel, the transforms must first exist and be referenced.) Transform items are channel groups associated with an item that store the item's transform values (controlling its position, rotation, or scale). By default, new items do not have transform items associated with them (even though they are visible within the **Properties** panel). Modo adds only the necessary transforms on an as-needed basis to reduce scene overhead. There are several ways to add transforms. You can adjust the target item with one of the transform tools or specify values. With this action Modo adds the transform item automatically to the **Channels** viewport list.

Backdrop Item Backdrop Properties



Backdrop
Image (none)
☑ Show In Perspective
Projection Type Front
☑ Keep Aspect
☐ Flip
☐ Invert
☐ Blend
Contrast 0.0 %
Brightness 0.0 %
Transparency 0.0 %
☐ Use Uniform Size
Pixel Size 10 mm
Auto Fit

Image	Specifies an image or uses the load command to load an image from disk.

Show in Perspective	Backdrop items can be a hindrance when the 3D viewport is in **Perspective** view, as they are flat along a cardinal axis. This option enables/disables Backdrop items in **Perspective** view.
Projection Type	Specifies which viewport the backdrop image faces toward as well as a Camera projection mode that appears automatically in the background of any viewport that is set to **Camera** as the type. For multiple viewports (such as front, side, and back), you create three different backdrop items with the appropriate image and projection type specified for each.
Keep Aspect	When enabled, retains the aspect ratio of an image when you scale the image.
Flip	When enabled, mirrors the image's pixels across the vertical axis. This is useful when using the same image for front-and-back or left-and-right pairs.
Invert	When enabled, inverts the RGB values for the backdrop image. This is useful to invert a black-and-white line drawing to make the lines more visible.
Blend	When enabled, produces a small amount of smoothing between neighboring pixels to reduce the jagged appearance of low-resolution images.
Contrast	Determines the apparent visual difference between light and dark values in the backdrop image. When the value is greater than **0%**, Modo increases the contrast; when the value is less than **0%**, Modo decreases the contrast.
Brightness	Determines the degree or amount of light or luminosity in the backdrop image. With a value greater than **0%**, Modo increases the brightness of the backdrop image; with a value less than **0%**, Modo decreases the brightness.
Transparency	Determines the opacity of the backdrop image. A value greater than **0%** increases the transparency. With a value of **100%**, the image is completely invisible.
Use Uniform Size	When this option is enabled, backdrop items ignore image size, and base the scale of the backdrop on the item's **Scale** value.
Pixel Size	Specifies the pixel size of the backdrop image.
Auto Fit	Scales the backdrop item automatically to match the height of the currently active Mesh Items (or the height and width when **Keep Aspect** is disabled). **Note:** This setting only works if the **Use Uniform Size** option is enabled.

Backdrop Size	Specifies the width and height of the image used for the Backdrop item.
Backdrop Size Keep Aspect	Specifies the width of the image used for the Backdrop item and maintains aspect ratio.

Tip: Within Modo's preferences, you can adjust some options related to backdrop items and how Modo scales images when placing them. For more information, see the Preferences topic.

Solid Sketch

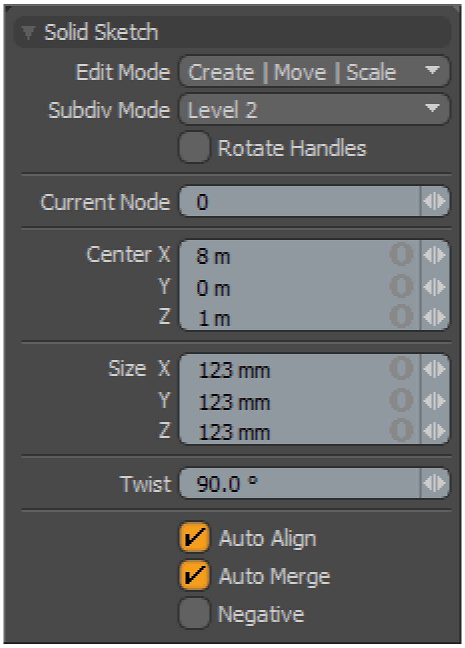
The **Solid Sketch** tool provides a quick method for "roughing out" organic shapes - building up basic geometry with just a few clicks. You can click to place individual nodes that interconnect automatically. The results are something like a subdivided cube that has been automatically extruded or bridged between each node as it is placed. The geometry then becomes the basis for more complex or detailed forms with further editing. Individual nodes can be selected and edited, even after the tool has been dropped.

The tool is found in the **Modeling** layout, under the **Basic** sub-tab as an alternate option of the **Tube** tool button. Right-click to open the pop-up menu and click to select it. It can also be activated from the menu bar under **Geometry** > **Draw** > **Solid Sketch**. When activated, you can click in the viewport to place nodes.

Nodes are always created and positioned relative to Using the Work Plane. Click and drag to scale the node as it's created. After releasing the mouse, clicking and dragging again on an active node, allows you to adjust the position of it relative to any other nodes that may be present.

Active nodes are indicated by the yellow color of the center dot and always surrounded by the tool's handles. These handles allow you to squash and stretch the node, and to rotate it. Click over the desired handle dot and drag to apply the action. Previously-created nodes can be selected by clicking over its center dot. Right-clicking on the selected node scales it. **Shift**+click and drag repositions the node and any child nodes further down the chain. New nodes always connect to the previously-selected node. To create an additional branch, select the desired parent node and then draw the new node by clicking. **Ctrl** + click to create a new node between an existing one. Individual nodes can be removed by middle-clicking.

It should be noted that any mesh created with **Solid Sketch** also places a series of two point polylines inside the mesh so that when the tool is re-activated, it recovers its previous state from the information the polylines contain. It should be noted that any additional geometry in the layer at that time the tool is re-activated is deleted when re-initializing the tool with this information. When the tool is active, the following attributes are available in the **Properties** viewport.


Solid Sketch

Edit Mode: Create | Move | Scale

Subdiv Mode: Level 2

☐ Rotate Handles

Current Node: 0

Center X: 8 m

Y: 0 m

Z: 1 m

Size X: 123 mm

Y: 123 mm

Z: 123 mm

Twist: 90.0 °

☑ Auto Align

☑ Auto Merge

☐ Negative

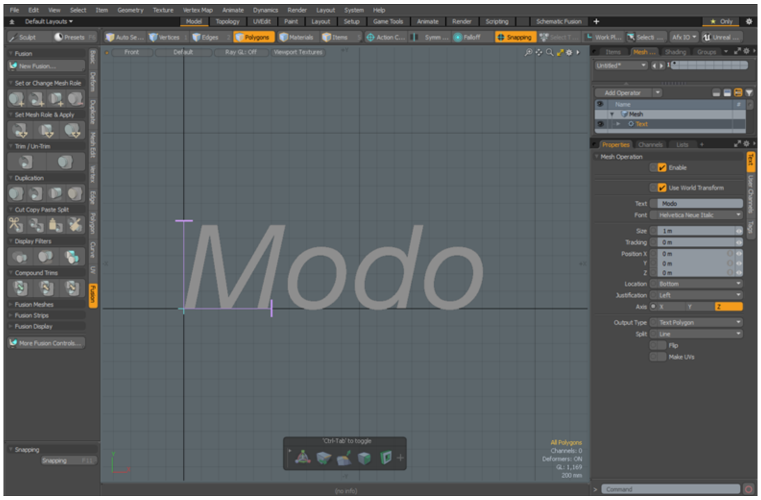
The following options are available for adjusting the **Solid Sketch** tool:

Edit Mode	The default value is **Create|Move|Scale**, which allows you to create and scale the node on initial click+drag, and then use the various handles to control the node further. In this mode, you can delete a node by middle-clicking directly on the node. You can also set the **Edit Mode** to **Destroy**, if you want to quickly click on several nodes to remove them.
Subdiv Mode	Determines how many tessellation levels to apply to the mesh using the SDS algorithm. Setting this value to **0** reveals the raw polygonal cage. Increasing the value to two creates additional edge loops at each node. This can be particularly useful when creating complex branching from a single node, as it produces smoother results.
Rotate	When this option is enabled, the rotate handles are drawn as Red, Green, and Blue rings around the selected node. These represent rotation around the Z, Y, and X axes respectively. To rotate the node, simply click and drag on the desired ring handle. The affect of using the rotation handles is to actually move the child nodes of the current node, as if the selected node were being re-oriented.
Current Node	Nodes are internally numbered in the order they are created, the current selection is always reflected in this field; you can change the value to select a different node manually.

Center X, Y, Z	These values define the position of the selected node. Clicking and dragging in the 3D viewport adjusts the position interactively, or you can adjust these value manually for finer control.
Size X, Y, Z	These values define the basic size of the selected node. Right-clicking and dragging on a selected node in the 3D viewport adjusts the size interactively, or you can adjust these values manually for finer control.
Twist	The protruding side handle on the selected node allows you to apply a simple twisting effect to the node interactively. This value can be manually adjusted here for finer control. Twisting the node does not affect the position of the nodes connected to it.
Auto Align	With this option enabled, each node re-orients the geometry it creates to more smoothly blend with the surrounding geometry segments.
Auto Merge	Since each node typically generates a cube of polygons around it, bunching up nodes close together causes the mesh to pinch. The **Auto Merge** setting reduces a node's created geometry by merging it with other nodes once they are too close together.
Negative	This is disabled by default; this option causes a node to repel neighboring geometry. This can be useful to create divots in the mesh.

Text

The **Text** tool simplifies the process of creating 3D text using system fonts.



Mesh Operation
☑ Enable
☑ Use World Transform
Text: Modo
Font: Helvetica Neue Italic
Size: 1 m
Tracking: 0 m
Position X: 0 m
Y: 0 m
Z: 0 m
Location: Bottom
Justification: Left
Axis: ○ X ○ Y ● Z
Output Type: Text Polygon
Split: Line
☐ Flip
☐ Make UVs

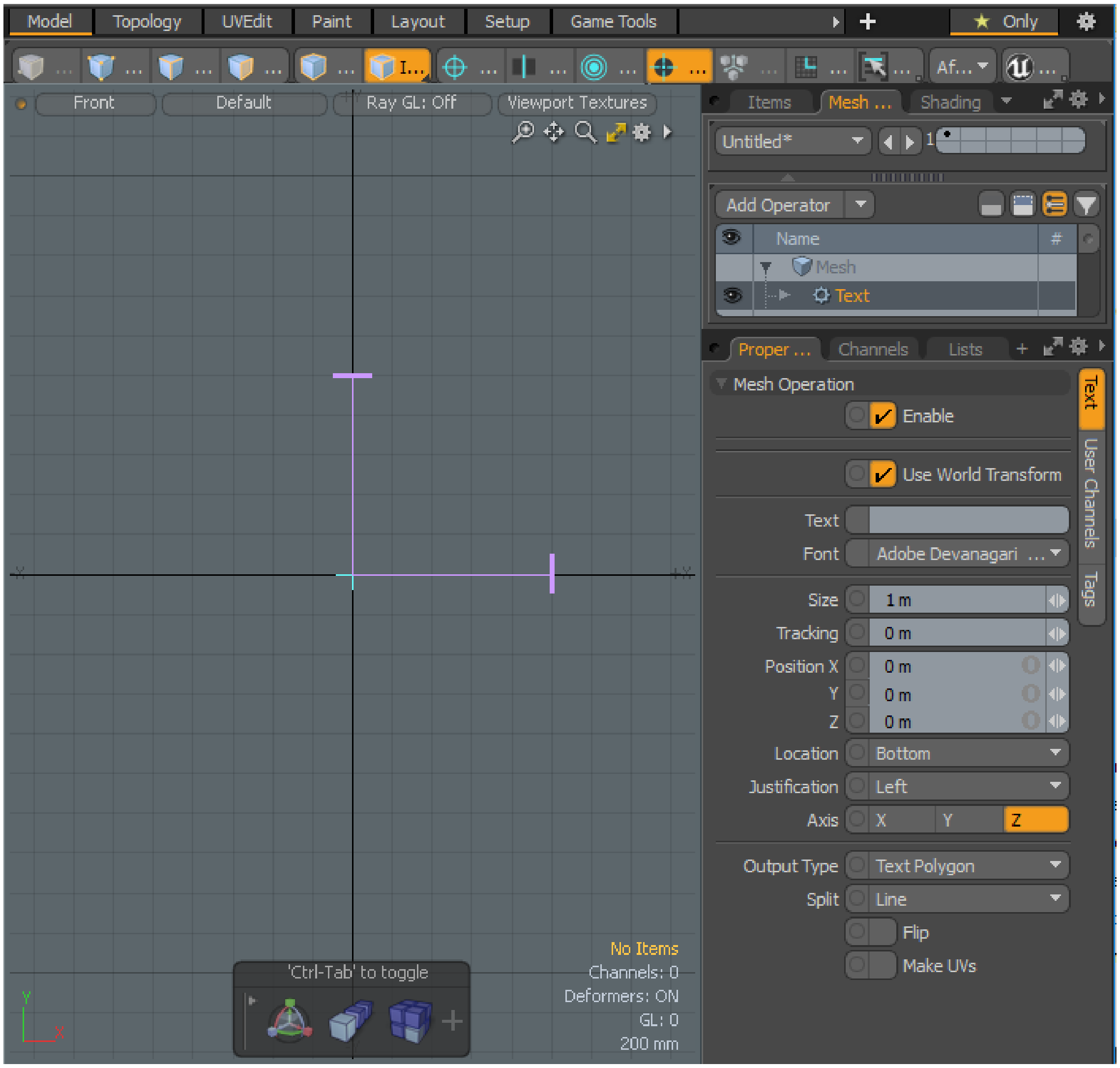
Text polygons are created, which contain metadata in the form of polygon tags that store all the information about the font used, size, kerning amount, and so on. As long as the text remains as this text polygon, the **Text** tool can be re-activated with the text polygon selected, and any of these attributes can be edited live. For example, when the text polygon is beveled or extruded, Modo converts them to "face" polygons and the text metadata is purged.

Tip: Use the Merge Meshes tool to control which text polygons are merged. For more information, see Applying Merge Meshes to Text.

Applying the Text Tool

1. On the right panel, open the **Mesh Ops** tab and click **Add Operator**.
2. Expand **Mesh Operations** > **Create**, and double-click **Text**.

   The properties displays on the right panel and handles display in the 3D viewport.


Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Only
Front
Default
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Untitled*
Add Operator
Name
Mesh
Text
Proper ...
Channels
Lists
User Channels
Tags
'Ctrl-Tab' to toggle
No Items
Channels: 0
Deformers: ON
GL: 0
200 mm

Mesh Operation

☑ Enable

☑ Use World Transform

Text:

Font: Adobe Devanagari ...

Size: 1 m

Tracking: 0 m

Position X: 0 m

Y: 0 m

Z: 0 m

Location: Bottom

Justification: Left

Axis: X Y [✓ Z]

Output Type: Text Polygon

Split: Line

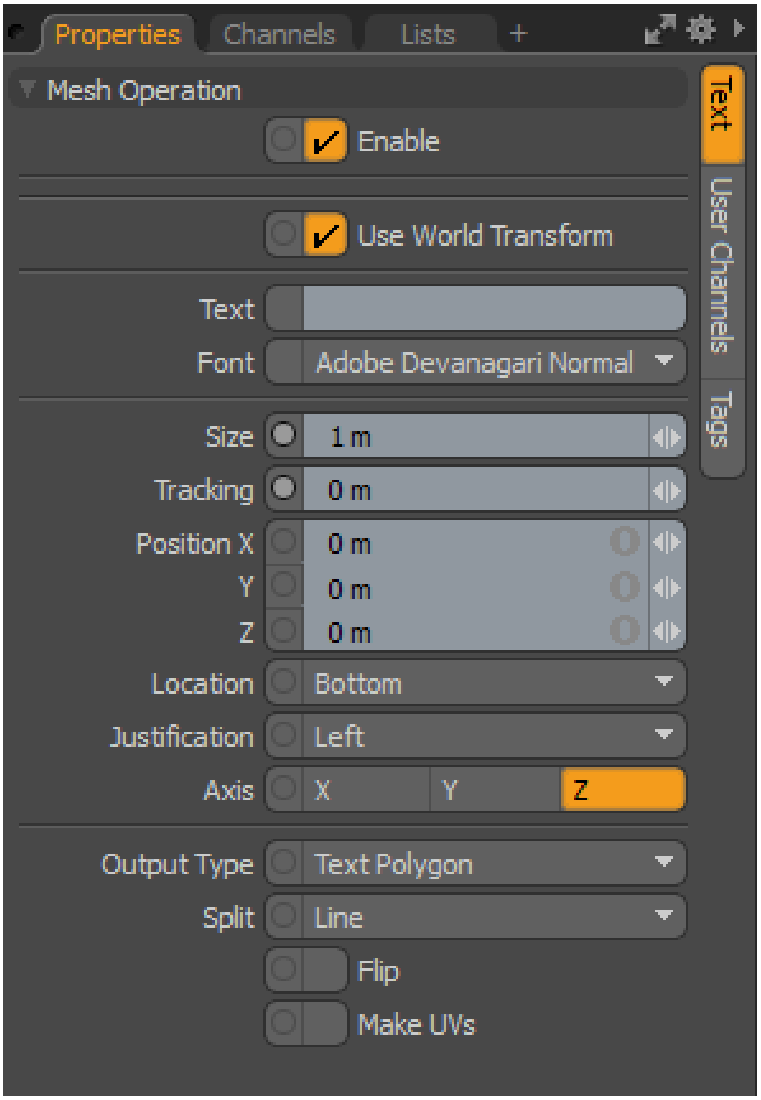
☐ Flip

☐ Make UVs

3. Update the Text as required.

Tip: In the 3D viewport, click-and-drag the handles to interactively update the Size and Tracking options.

Text Tool Options



☑ Enable

☑ Use World Transform

Text

Font: Adobe Devanagari Normal

Size: 1 m

Tracking: 0 m

Position X: 0 m

Y: 0 m

Z: 0 m

Location: Bottom

Justification: Left

Axis: X Y **Z**

Output Type: Text Polygon

Split: Line

☐ Flip

☐ Make UVs

Option	Description
Enable	Enables or disables the Text tool.
Use World Transform	Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Text	Allows you to enter the desired text to display in the scene.
Font	Gives access to all fonts installed on the system. The font used for the text string can be

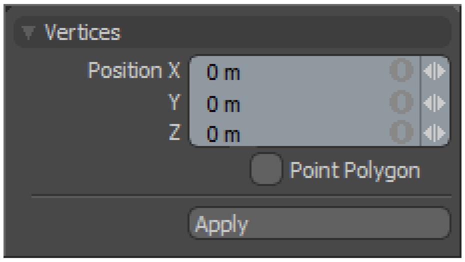
	changed at anytime using this pop-up, as long as the text remains in the native text polygon type.
Size	Sets the scale for the text.
Tracking	Allows fine tuning of the space between each letter.
Position X, **Y**, **Z**	Determines the starting XYZ positions for the text string in the 3D space.
Location	Shifts the position of the font up or down, relative to the start location. The options include **Bottom**, **Middle**, and **Top**. The default value of **Bottom** sets the bottom-left-most point on the font, at the start location. Changing the value to **Middle** positions the font at its middle (in height) and the **Top** setting pushes the font down such that the top-left-most point on the font rests at the start location.
Justification	Adjusts the text position to the right and left. The options include **Left**, **Center**, and **Right**. The default value of **Left** puts the left-most edge of the text at the start location. Changing this value to **Center** positions the text such that the center of the text string rests at the start location. Using the **Right** option pushes the text string to the left so that the right-most edge of the text rests at the start location.
Split	Determines how many text polygons to create. By default, the entire text string is a single text polygon, using the **Line** setting. Changing this setting to **Word** breaks the string into a single text polygon for each discreet word as determined using spaces. For instance "Hello World" would be two text polygons - one for "Hello" and one for "World". Changing the vale to **Character** creates a text polygon for each letter in the text string. The **Word** and **Character** settings are very useful when combining the text tool with various generators, such as **Curve** or **Scatter**, as they provide better granularity for shaping the text.
Axis	Determines the direction the text faces.
Output Type	Allows you to choose to display your text as **Text Polygons**, or as a **Bezier** outline.
Split	Specifies how your text is split. You can choose **Line**, **Word**, or **Character**.
Flip	Determines the polygon normal direction for the text polygon faces.
Make UVs	Creates a UV map for all polygons. This option is disabled by default to increase the interaction speed while working with the tool.

Vertex

The Vertex tool provides a method for quickly plotting points in 3D space. Found in the **Vertex** sub-tab of the **Modeling** layout in Modo, the Vertex tool can be used to quickly plot points in 3D space. To use the tool, simply click on a 3D viewport to create a point under the mouse pointer. Notice that the point doesn't draw anywhere in the viewport, but only draws where the mouse pointer's position intersects the Work Plane.

In a perspective viewport, changing the point of view can modify the work plane, which, in turn, changes where points are created. To change the position of the most-recent point, either click directly on the current vertex and drag, or hold down the **Shift** key and drag anywhere in the 3D viewport. To numerically edit the most-recently created point, you can change the X, Y, or Z values in the tool's properties panel by adjusting the **Position** attribute. The **Point Polygon** option generates single point polygons instead of regular vertices.

On their own, vertices won't render but they can be useful in creating object positions, when used as a point source for a Replicators, or can be used as sources to create positions for Blobs, Sprites, and Volumes items. There are also some commands that complement the Vertex tool for making geometry.



Position X: 0 m
Y: 0 m
Z: 0 m
☐ Point Polygon

You can use the following commands with the Vertex tool.

- **Make Polygon** command - uses the currently-selected vertices to create a polygon. Refer to Make Polygon for more information.
- **Make Open Curve** command - uses the currently-selected vertices to generate an open spline curve. When using the Vertex tool, the first drawn vertex determines the base of the curve. If created from a selection of vertices, the first vertex selected would then become the base. Determining the base of a curve is important for **Fur Guides**. The command may be invoked by pressing the **Shift**+**O** keyboard shortcut, or from the **Curve** palette available from the menu bar **Geometry** > **Curve Palette**.
- **Make Closed Curve** command - uses the currently-selected vertices to generate a closed spline curve, joining the first and last vertices into a continuous loop. The command may be invoked by pressing the **Ctrl**+**P** keyboard shortcut, or from the **Curve Palette** available from the menu bar **Geometry** > **Curve Palette**.

Tip: Sometimes vertices may not be visible in the viewport until selected, though this can be easily remedied by changing the visibility option. Hover the mouse pointer over the window for which you want to modify visibility, and press the **O** key. This opens the drawing and visibility options panel. Toggle the **Show Vertices** option under the **Visibility** sub-tab to make them visible. This a per viewport setting.

Patch Curves

Patching is a means of generating a smooth surface from 3 or 4 curves that define the contour edges of the surfaces. Multiple patches can be combined to create a closed volume (like a contoured bottle). The **Patch Curves** tool can be found under the **Duplicate** tab of the toolbox, or within the **Curves Palette**.

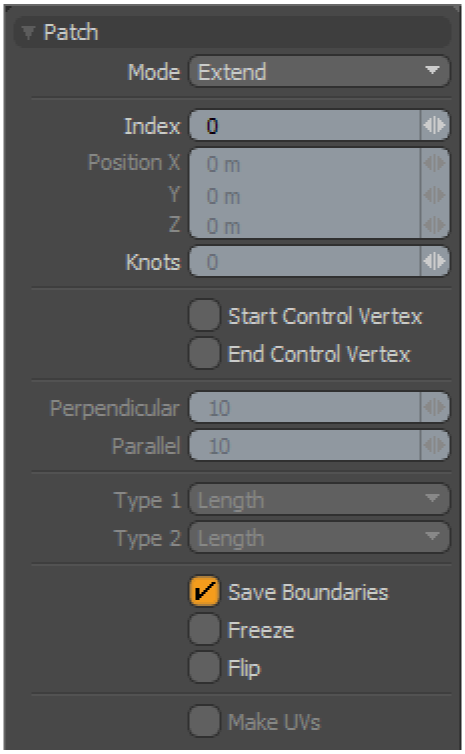
To make a Spline Patch:

1. Open the Model layout and use the **Curve** drawing tool to draw curves that share endpoints (those that are merged).
2. On the left panel, open the **Duplicate** sub-tab and click **Patch Curves**.
3. In the viewport, click to activate the tool.
4. Click the curve edge points. By clicking on a sequence of edges, you can define a patch.
5. Edges can also be clicked and dragged, extending patch edges, if necessary.

   You can interactively adjust the control points on the curve while the patches are re-made to follow the new contours. You can pick edges and set the number of segments for that edge as the patching updates to match. Patching generates UVs that follow the grid of quads generated for each patch.

Patch Curve Options

The following options are available for adjusting the **Patch Curves** tool.



Patch

Mode Extend

Index 0

Position X 0 m

Y 0 m

Z 0 m

Knots 0

☐ Start Control Vertex

☐ End Control Vertex

Perpendicular 10

Parallel 10

Type 1 Length

Type 2 Length

☑ Save Boundaries

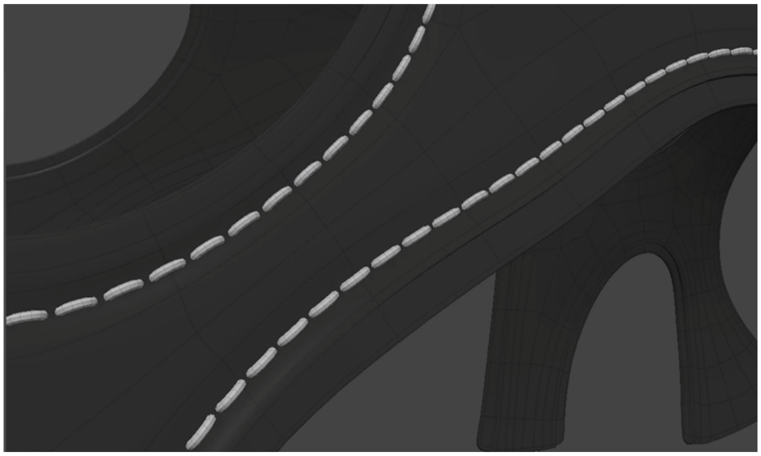
☐ Freeze

☐ Flip

☐ Make UVs

Mode	Interactively select edges to define the patch surface. • **Define Patch** - Define the patch position and other tool options. • **Extend** - Drag edge handles to extend patch edges. • **Move Knot** - Interactively adjust specific knots in the patch. • **Move Edge** - Move the selected edge. • **Edit Edge** - Select edges and modify the number of knots.
Knots	Defines the number of divisions along the length of a curve for editing.
Start/End Control Vertex	The flag for side curves to extend, allowing control over curvature for end segments.
Perpendicular	The number of divisions along a perpendicular side.
Parallel	The number of divisions along a parallel side.
Save Boundaries	Retains the boundary curves.
Freeze	When the tool is dropped, the **patch** surface becomes editable polygons.
Flip	Reverses the normal direction of the patched surface.
Make UVs	Generates an automatic UV map for the patched surface for texturing.

Stitching



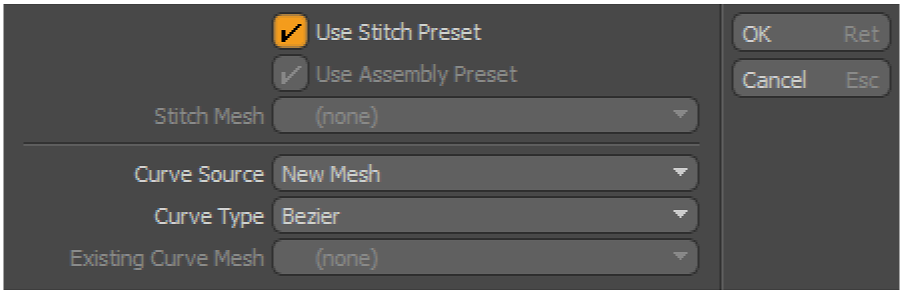
Modo allows you to add stitches to existing mesh items. You can do this by using the **Stitch Mesh Setup** command available from the **Geometry** menu. This uses a Stitch Particle Generator to create the stitches along a curve. The Stitch Particle Generator is a custom version of the Curve Particle Generator.

Using Stitch Mesh Setup

You can access Stitch Mesh Setup from the menu bar.

> **Note:** If you've applied transforms to your geometry, make sure you freeze it before adding stitches. To do this, select your item in the **Items** list or the 3D viewport, and in the **Properties** panel on the right, in the **Transform** section, click **Freeze** > **All**. Without freezing, the stitches are positioned in relation to the original geometry.

1. Click **Geometry** > **Stitch Mesh Setup**.



☑ Use Stitch Preset

☑ Use Assembly Preset

Stitch Mesh: (none)

Curve Source: New Mesh

Curve Type: Bezier

Existing Curve Mesh: (none)

The **Stitch Mesh Setup** dialog appears. You have to specify what stitches to use and the curve along which they should be placed.

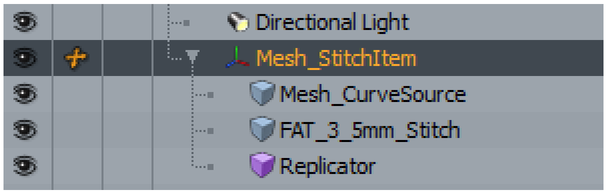
2. Select the type of stitch you want to use:
   - **Use Stitch Preset** - The default option. Select this to choose a stitch from the **Stitches** category of the Preset Browser.
   - **Use Assembly Preset** - Select this to use an existing assembly for stitching. You can pick an assembly from the Preset Browser.
   - **Stitch Mesh** - Select this to use a mesh from the current scene. You can also leave this set to **(none)** and add the stitch later in the item's **Properties** tab or the Schematic.
3. In the lower half of the dialog, specify the curve along which the stitches are placed.
   - **Curve Source** - Select how the curve should be created. You can draw a curve using the default **New Mesh** option, or choose an existing curve in your scene by selecting **Existing Mesh**.

     If your model has edges at the required location, you can use them to define the curve. **Selected Edges (Procedural)** creates a curve along the edges you selected. The curve is procedurally linked to the mesh and as you edit the mesh, the curve and the stitches change with it. **Selected Edges (Unlinked)** adds a curve to the selected edges that is not linked to the geometry and therefore it's not affected by transforms applied to the mesh, but you can edit the curve manually.
   - **Curve Type** - You can use a regular or a Bezier curve.
   - **Existing Mesh** - When **Curve Source** is set to **Existing Mesh**, you can pick the mesh to be used here.
4. Click **OK**.
5. If you've selected **Use Stitch Preset** or **Use Assembly Preset**, the Preset Browser appears. Select a stitch or assembly, then click **OK**.
6. If you've set **Curve Source** to **New Mesh**, the curve tool is activated and you can start drawing your curve. Stitches are added automatically as you draw.

Tip: You can create multiple curves. To start a new curve, hold **Shift** while clicking to create the first control point.

If you've selected an existing curve, the stitches are added to it. In the case of selected edges, a curve and stitches are added along the selected edges.

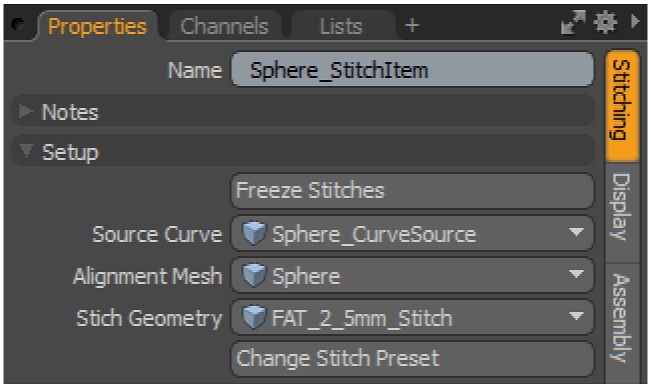
In the Item Tree, a Stitch Item appears. It consists of a curve that is the path for the stitches, a stitch, used as the prototype mesh, and a replicator that propagates the prototype along the curve.


Directional Light
Mesh_StitchItem
Mesh_CurveSource
FAT_3_5mm_Stitch
Replicator

Editing Stitches

You can edit your stitching in the Stitch Item's **Properties** panel. To open it, click the Stitch Item in the Item Tree. The **Properties** tab in the lower right panel is populated with the Stitch Item's properties.

Setup



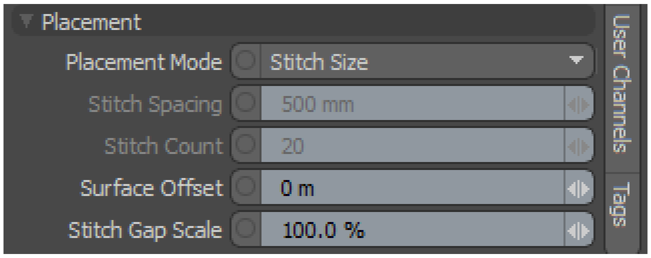
Name: Sphere_StitchItem
Source Curve: Sphere_CurveSource
Alignment Mesh: Sphere
Stich Geometry: FAT_2_5mm_Stitch

In this section, you can edit some of the properties you specified when setting up your stitches using **Stitch Mesh Setup**. You can change the curve, the stitch mesh, and the alignment of the stitches. In addition, you can also freeze your stitches, turning them into exportable geometry.

Freeze Stitches	As stitches are replicated items, it might be difficult to export them. Click **Freeze Stitches** to convert them to geometry that you can easily export. Once a stitch is frozen, a mesh item with the name **Stitches_Alignment Mesh Name** is created. The **Alignment Mesh Name** suffix is dependent on the name of the alignment mesh assigned to your stitches. For example, if the alignment mesh is called tongue, the frozen mesh item is called **Stitches_Tongue**. Custom names can be created by holding down Alt when clicking **Freeze Stitches** and

	typing a name in the dialog that appears. **Note:** Once the stitches are frozen, their properties can't be edited.
Source Curve	Change the curve along which the stitches are replicated.
Alignment Mesh	Change the mesh to which the stitches are aligned.
Stitch Geometry	Change the stitch mesh to be replicated.
Change Stitch Preset	Click this to choose a new preset to be used as **Stitch Geometry**.

Placement

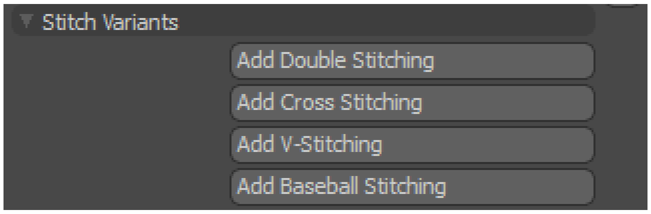

Placement
Placement Mode: Stitch Size
Stitch Spacing: 500 mm
Stitch Count: 20
Surface Offset: 0 m
Stitch Gap Scale: 100.0 %

These properties determine how the stitches are placed along the curve.

Placement Mode	Determines how stitches are placed along the curve. The following options are available: • **Stitch Count** - Places a given number of stitches along each curve. • **Spacing Distance** - Places stitches approximately this far apart. • **Stitches Per Inch** - Fits a given number of stitches into one inch of the curve length. • **Stitch Size** - Uses the bounding box size of the stitches so that stitches lay end to end along the curve.
Stitch Spacing	Increase or decrease the spacing between stitches when **Placement Mode** is set to

	Spacing Distance.
Stitch Count	Increase or decrease the number of stitches when **Placement Mode** is set to **Stitch Count**.
Surface Offset	Specify how far stitches are pushed into the mesh surface.
Stitch Gap Scale	Increase or decrease the spacing between stitches when **Placement Mode** is set to **Stitch Size**.

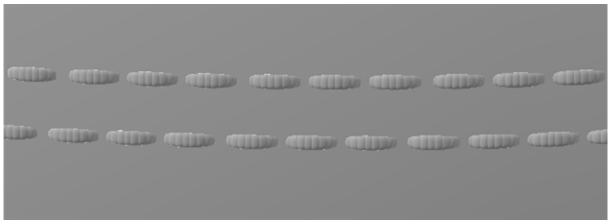
Stitch Variants



You can create your own rigs to manipulate the stitching look, but these properties allow you to create some common looks automatically.

Stitch variants can be combined, but some of them negate other types. For example, cross stitching can't be combined with v-stitching or baseball stitching.

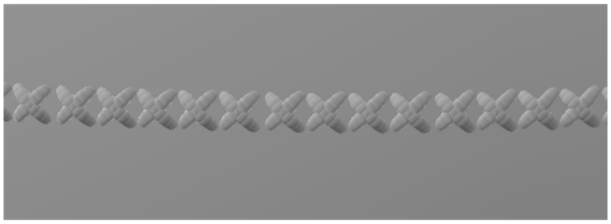
Double Stitching



Add/Remove Double Stitching	Click to add or remove double stitching. When enabled, additional properties appear.
Remove Central Curve	When enabled no stitches are created along the original curve, only on either side of it.

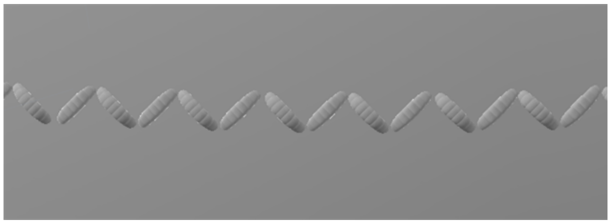
Stitch Radius	Determines how far to offset the two outer curves from the original curve.
Create Positive Side	When enabled, Modo creates the outer curve on the positive side of the central curve.
Create Negative Side	When enabled, Modo creates the outer curve on the negative side of the central curve.

Cross Stitching



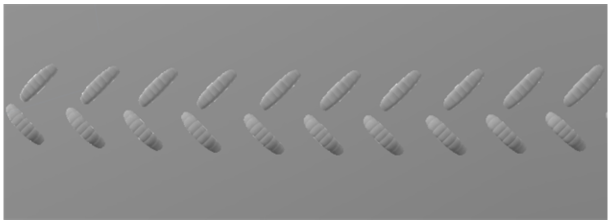
Add/Remove Cross Stitching	Click to add or remove cross stitching. When enabled, additional properties appear.
Rotation	Specify the angle at which each stitch is rotated.
Stitch 1 Inset/Stitch 2 Inset	Specify how much each of the pair of stitches is pushed into the alignment mesh surface.

V-Stitching



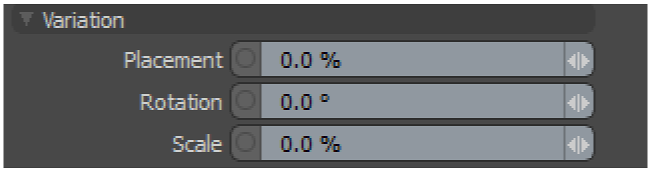
Add/Remove V-Stitching	Click to add or remove v-stitching. When enabled, additional properties appear.
Rotation	Specify the angle at which each stitch is rotated.

Baseball Stitching



Add/Remove Baseball Stitching	Click to add or remove baseball stitching. When enabled, additional properties appear.
Rotation	Specify the angle at which each stitch is rotated.
Stitch Gap Scale	Increase or decrease the spacing between stitch pairs. By default, each pair of stitches is placed one stitch-width from the next pair.

Variation



▼ Variation

Placement 0.0 %

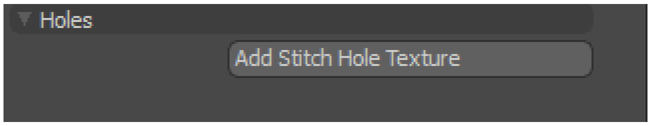
Rotation 0.0 °

Scale 0.0 %

These properties allow you to randomize stitch properties. This can make your model more realistic.

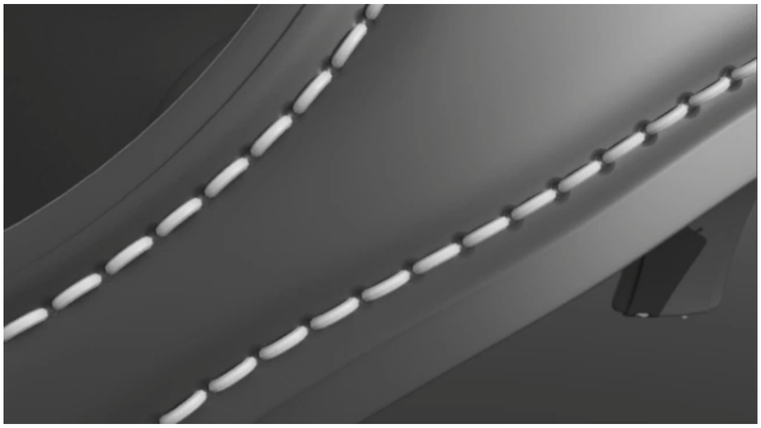
Placement	Move each stitch randomly forward or backward along the curve by the specified amount.
Rotation	Rotate each stitch by a random amount.
Scale	Scale each stitch by a random amount.

Stitch Holes



This is a texturing effect that places dots at the ends of stitch geometry using texture replicators, resembling the holes the stitches go through. When you open the Preview render, you can see the texture with holes.

Note: The holes are only visible on the render, you can't see them in the 3D viewport.



Add/Remove Stitch Hole Texture	Add or remove stitch hole texture.
Shading Effect	Set the texture layer's effect to **Diffuse Amount**, **Bump**, or **Displacement**.
Texture Group	Parent the texture layer to a specific material group in the Shader Tree.
Opacity	Set the opacity of the texture layer. The **Bump** and **Displacement** effects may look a bit strong, so this allows you to tone them down.
Hole Size	Set the texture replicator's particle size. This is the radius of each hole.

Lace Item

The Lace Item is a collection of commands designed to streamline the process of building laces in Modo.

No Items
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 796,960
10 mm

Generating Laces

Before adding a Lace Item, make sure you have a scene containing the geometry you want to add the laces to, including the geometry for the holes the lace goes through. You can also have a lace shape ready in a separate mesh layer. This should be a flat polygonal geometry or a curve.

Accessing the Tool

In the Item List, click **Add Item**, then under **Items** > **Procedural**, double-click **Lace Item**.

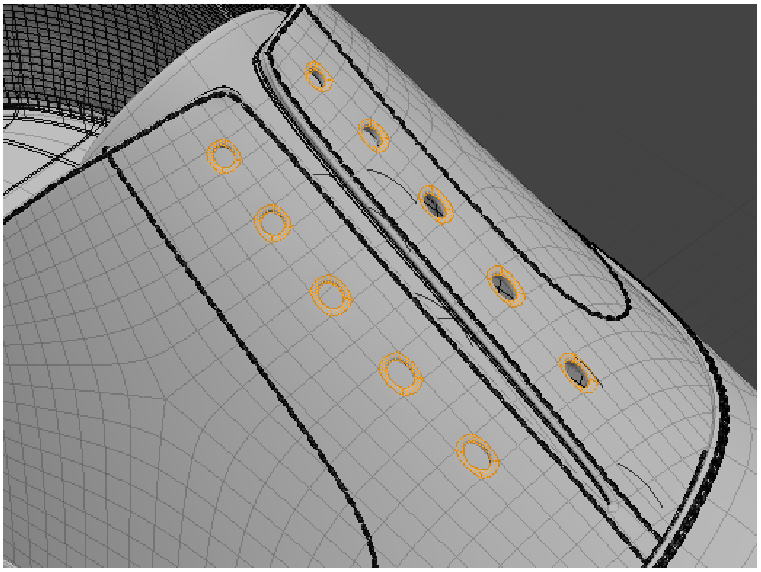
A **Lace Item** is added to your scene and its properties appear on the lower right panel.

Specifying the Eyelets

Next, you need to select the components to be used as the eyelets.

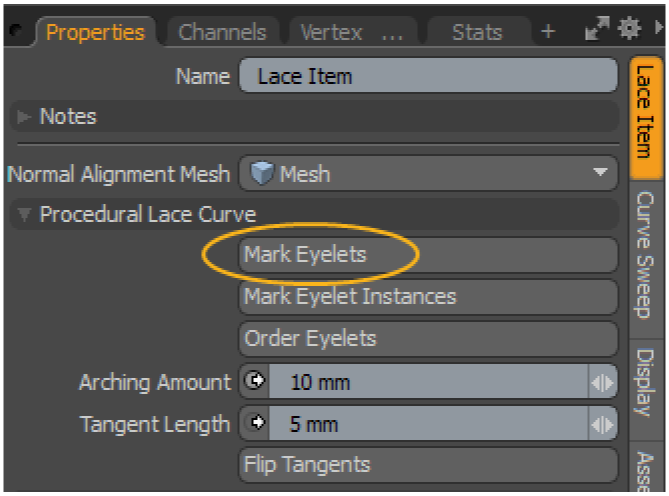
1. In the 3D viewport, select the required polygons or edges. Each contiguous selection is treated as a point on the lacing path, so this works best by selecting edge or polygon loops around the holes a lace will pass through.

Tip: It may be helpful to create a selection set and gradually add all the elements you need. This way, you avoid accidentally losing your selection, and it's easy to quickly select the required elements later. For more information, see Using Selection Sets.



Tip: Hold **Ctrl** and click to unmark a selection.

2. In the Lace Item's **Properties** panel, click **Mark Eyelets**.


Properties
Channels
Vertex ...
Stats
+
Notes
Procedural Lace Curve
Mark Eyelets
Mark Eyelet Instances
Order Eyelets
Flip Tangents
Lace Item
Curve Sweep
Display
Asse

Name: Lace Item

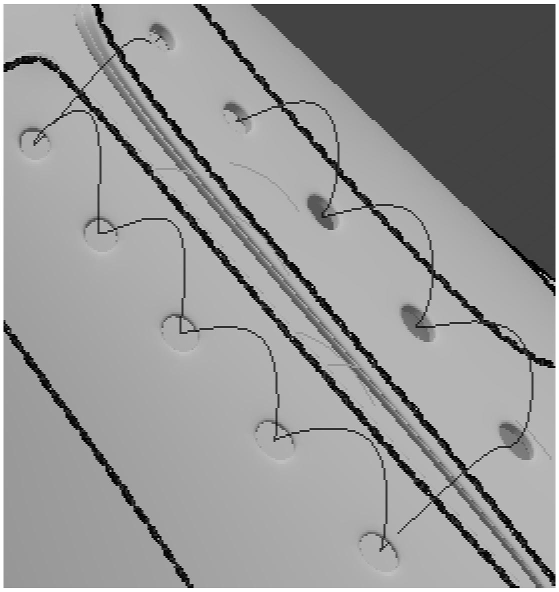
Normal Alignment Mesh: Mesh

Arching Amount: 10 mm

Tangent Length: 5 mm

When your eyelet geometry is made up of instances, you can select them in the Item List, then click **Mark Eyelet Instances**. If you want to select all instances, hold down the **Shift** key. This changes **Mark Eyelet Instances** to **Mark All Instances**. Clicking the button marks all instances in your scene as eyelets.

The selected geometry is now marked as eyelets for the lace. A Lacing Curve is generated and drawn in the 3D viewport.

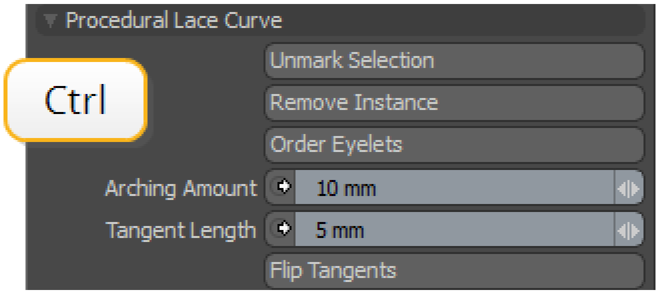


Unmarking Eyelets

To remove your eyelet selection, hold down the **Ctrl** key. This modifies the behavior of the buttons and the following controls appear:

- **Unmark Selection** - Click to remove the eyelet selection.
- **Remove Instance** - Click to unmark the instances selected in the Item List.

Note: If you have selected all instances by **Shift**+clicking **Mark All Instances**, you can't remove them using **Remove Instance**.



Procedural Lace Curve
Ctrl
Unmark Selection
Remove Instance
Order Eyelets
Flip Tangents

Arching Amount: 10 mm

Tangent Length: 5 mm

Ordering Eyelets and Fine-tuning the Curve Path

The resulting curve generally doesn't follow the required path, so you need to re-order the points and adjust the tool properties.

1. Click **Order Eyelets**, and click on the blue points in the order you want the lace to go through them.

The points are numbered in the viewport and the Lace Curve follows the new order. Click the thumbnail to see how eyelets are re-ordered.

2. Adjust the other options as needed:

Arching Amount	Specifies how much the laces arch up above the surface.
Tangent Length	Each eyelet is a Bezier control point, and this option defines how tightly the curve goes into and out of the point.
Flip Tangents	Flips the Bezier tangents at a specific control point. This allows you to control how the curve bends around the control point. Clicking the **Flip Tangents** button selects the control points in the scene. You can then flip the tangents at a particular control point by clicking on it.
Lace Curve	Allows you to select an existing curve mesh along which to extrude the lace geometry.
Freeze/Edit Lacing Curve	The lacing curve built following the steps outlined here is fully procedural, and updates as the mesh geometry is edited or eyelet selections are updated. This is intended to serve as a good baseline for lacing, but any complex surface will likely require manual manipulation of the lace curve. This command converts the lacing curve to a standard editable mesh that can be manipulated.
Lace Geometry	The mesh the lace geometry is built to. When set to **(none)**, Modo creates a mesh under the Lace Item. Here you can specify a different mesh layer.

Building the Laces

1. Click **Build Lace Geometry** to generate the laces.

   A dialog prompts you to set the shape for the lace geometry.

2. Click the **Shape Profile Type** dropdown to select the shape type. The default setting is **Existing Mesh**. Select this to use geometry in an existing mesh layer.

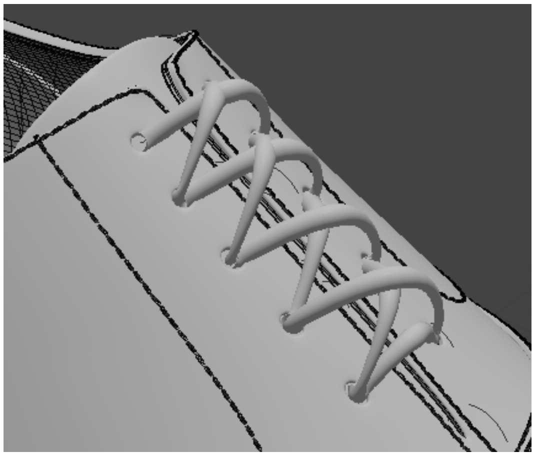
   The other available options are:

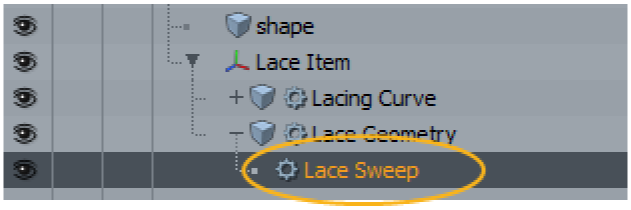
   - **Profile Preset** - Uses one of the Profile presets to generate the laces.
   - **Polygon Strips** - Uses polygon strips to generate the laces.
   - **New Curve Mesh** - Adds a new mesh containing a curve you can edit.

Note: You can modify these options later in the Curve Sweep properties. For more information, see Curve Sweep Properties.

3. If you set **Curve Source** to **Existing Mesh**, click the **Shape Source** dropdown to select the mesh layer where the shape can be found.

   This generates the lace geometry using a Curve Sweep mesh operation. You can see it in the Item List parented under the **Lace Item**, called **Lace Sweep**.





   The Curve Sweep properties are displayed on the lower-right panel. Edit them to further customize the lace geometry.

Note: For more information on the Curve Sweep mesh operation, see Curve Sweep.

Create Polygons

The Create Polygons mesh operation generates new polygons and vertices on a procedural mesh layer, using arrays of position values. It allows you to create both surface-type (Face, Subdivision, Catmull-Clark) and linear-type polygons, such as curves and lines.

The mesh operation can generate polygons in two ways:

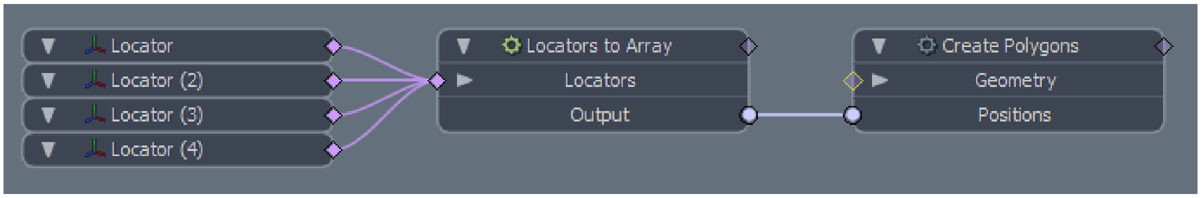
- **Fill** - Treats each array of positions as a single polygon. For example if the input array contains two sub-arrays of positions, two separate polygons are generated for the two arrays.
- **Bridge** - Polygons are generated from one sub-array to the next, creating a bridge between the sub-arrays to generate the surface.

Using Create Polygons

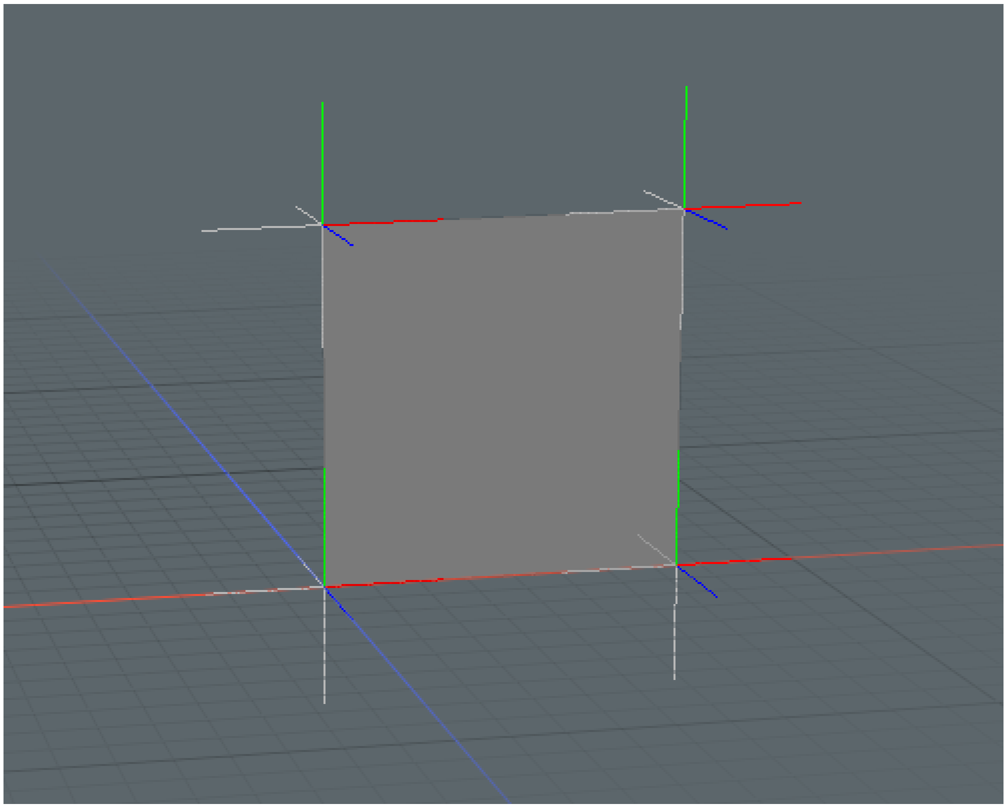
In order to use Create Polygons, make sure you have at least one array of positions in your scene. The example here uses an array created from locator positions, using the Array Modifiers modifier.

1. In the Schematic layout, click **Add...** and under **Mesh Operations** > **Create**, double-click **Create Polygons**.

   This adds the Create Polygons node to the Schematic.
2. Connect the array to the **Create Polygons** node's **Positions** channel. Here, the **Output** from the Locators to Array operation is used as **Positions** for the Create Polygons operation.



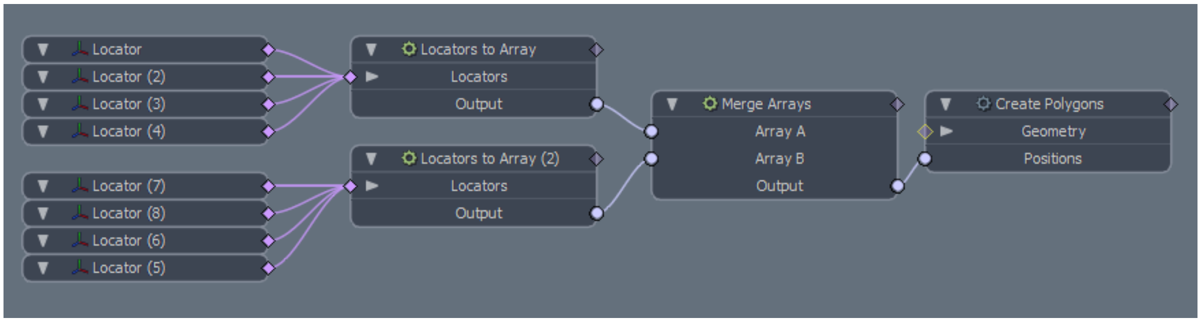
3. Add a new mesh layer by pressing the **n** key or from the **Items** list, and add it to the **Schematic** by selecting and clicking **Add Selected** in the Schematic.
4. Select the **Create Polygons** node to open its properties on the right panel. By default, the operation uses **Fill Mode** and generates a **Face** polygon. Adjust the options to get the result you need.



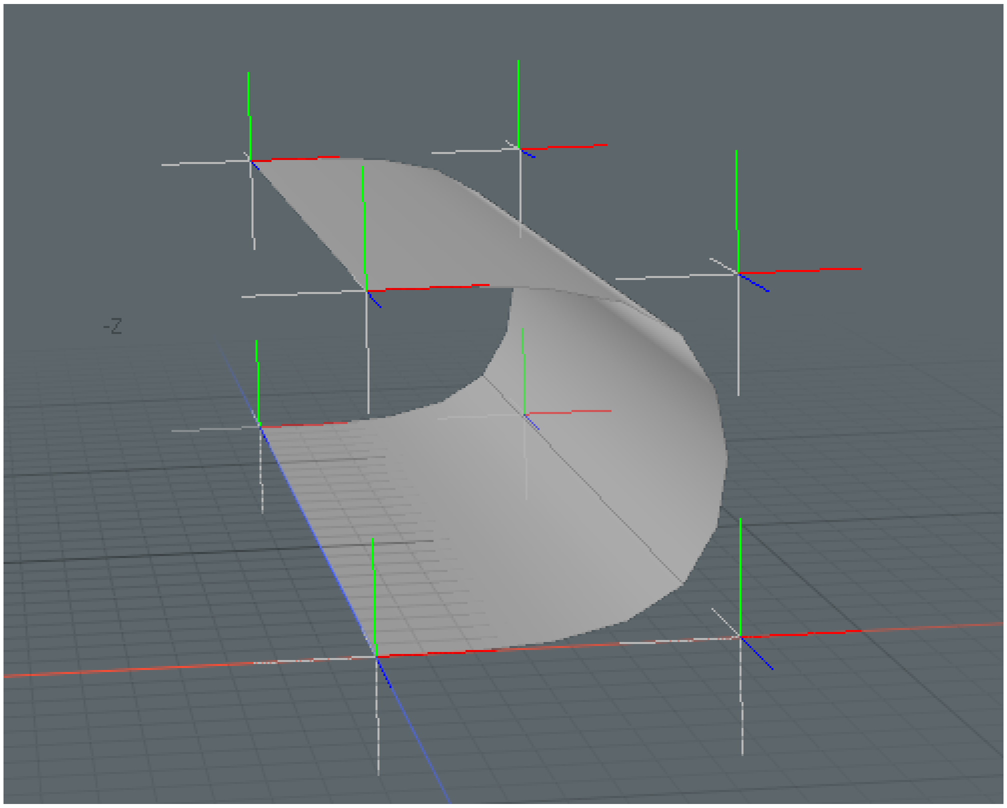
Creating Polygons by Bridging Arrays

You can use Create Polygons to generate polygons from one sub-array to another.

In the example below, Array Modifiers creates two sub-arrays that serve as **Positions** for the Create Polygons operation. When you set Mode to Bridge in the properties, Modo generates polygons that connect the two sub-arrays.


Locator
Locator (2)
Locator (3)
Locator (4)
Locator (7)
Locator (8)
Locator (6)
Locator (5)
Locators to Array
Locators
Output
Locators to Array (2)
Locators
Output
Merge Arrays
Array A
Array B
Output
Create Polygons
Geometry
Positions

With a **Catmull-Clark** polygon type and the **Closed** option set to **None**, the following geometry is generated:



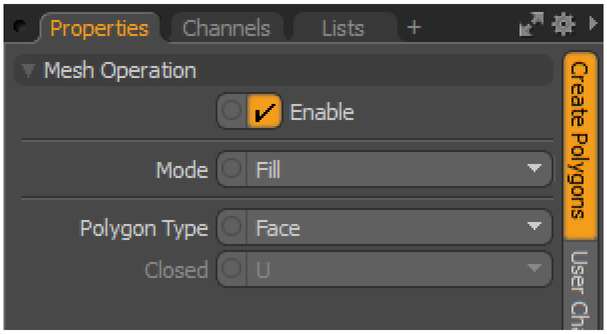
If some of the polygons are not displayed correctly, their normals might be flipped. You can re-order the locators feeding into the Locators to Array modifier, or use the Polygon Flip mesh operation:

1. Select the polygons you want to flip.
2. In the Schematic, click **Add**, then under **Mesh Operations** > **Polygon**, double-click **Polygon Flip**.

   This flips the normals of the selected polygons.

Create Polygons Properties

The Create Polygons mesh operation has the following options:


Properties
Channels
Lists
+
Mesh Operation
Create Polygons
User Ch

☑ Enable

Mode: Fill

Polygon Type: Face

Closed: U

Mode	Specifies how the arrays are used to generate polygons. • **Fill** - Treats each array of positions as a single polygon. For example if the input array contains two sub-arrays of positions, two separate polygons are generated for the two arrays. • **Bridge** - Polygons are generated from one sub-array to the next, creating a bridge between the sub-arrays to generate the surface.
Polygon Type	The type of polygon to generate. The available options are: • **Face** • **Subdivision Surface** • **Catmull-Clark** • **Curve** • **Line**
Closed	Specifies if the generated polygons should be closed, creating a watertight surface. The available options are: • **None** • **U** • **V** • **Both**

Selecting Items

There are seven total selection modes in Modo.

- The first three (vertices, edges, and polygons) are the component selection modes. In these modes you use the pointer to select the component elements of the geometry to manipulate by using Modo's various modeling and transform tools. When in **Items mode**, you can transform an entire item layer as a whole (such as move, rotate, or scale the layer). Think of item layers as a container. Component edits manipulate the contents of the container, and item edits transform the container, itself.
- Next, with the Materials selection mode you can automatically select a Shader Tree layer by clicking a polygonal surface within the 3D viewport. (No guessing about which surface belongs to what geometry.)
- With the last two modes, the Pivot and Center modes, you can select the center and pivot elements of an item layer. Then, you can manipulate them directly by modifying their position and orientation to the associated geometry for animation purposes.

You can directly select any of these modes by using keyboard shortcuts:

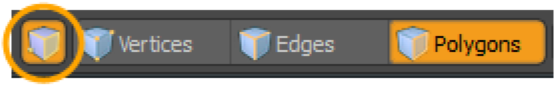
- **1** - Vertices
- **2** - Edges
- **3** - Polygons
- **4** - Materials
- **5** - Items
- **6** - Pivot
- **7** - Center

Cycle Component Selections

This command cycles through the component selection types for vertices, edges, and polygons. In the Modo default configuration this is mapped to the **Spacebar** key. If you have selected a tool previously and you then press the **Spacebar**, Modo drops the tool. If you have not selected a tool, the command cycles the selection sequentially from vertices to edges to polygon and back to vertices in a continuous loop.

Selection Mode Types

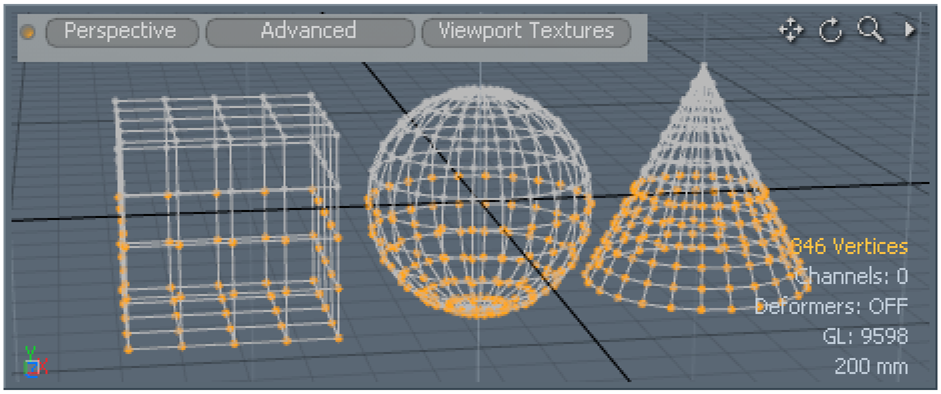
Auto Select



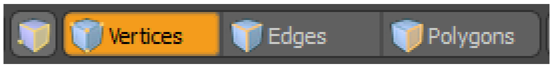
Auto Select mode allows you to quickly change the current selection mode used in your scene. From the modes toolbar, click the **Auto Select** icon to activate the automatic selection mode.

Once activated, deselect items in your scene and select a vertex, an edge, or a polygon to automatically activate the different selection modes. For example, if **Vertices** selection mode is active and there are no vertices currently selected, clicking on an edge will switches to **Edges** selection mode automatically.

Vertices Selection Mode

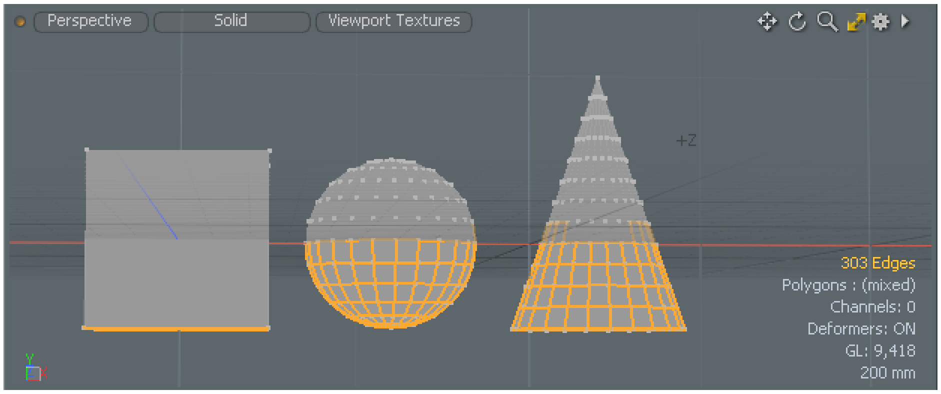

Perspective
Advanced
Viewport Textures
846 Vertices
Channels: 0
Deformers: OFF
GL: 9598
200 mm

A vertex is a single point that represents a position in 3D space. When in Vertices selection mode, you can use the basic paint or lasso selection functions to quickly select one or more vertices for precise editing. Modo indicates selected vertices with an orange color to distinguish them from unselected vertices. By dragging over an object, you can paint a selection of one of more vertices. If you right-click and drag, you can draw a lasso around the selection area to select multiple vertices at once. Modo provides other selection options to use with vertices such as the **Loop Select** and **Select More/Less** functions. (For information about these and other selection functions, see the Using Selection Modifiers topic.) If you wish, you can change the lasso style by choosing **Select > Lasso Style** and specifying a different style.

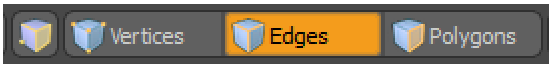

Vertices
Edges
Polygons

You can activate Vertices selection mode by clicking the **Vertices** interface button in the modes toolbar. Alternatively, choose **Select > Vertices**, press **1**, right-click away from any geometry in the 3D viewport and choose **Vertices**, or middle-click any vertex in the viewport.

Edge Selection Mode

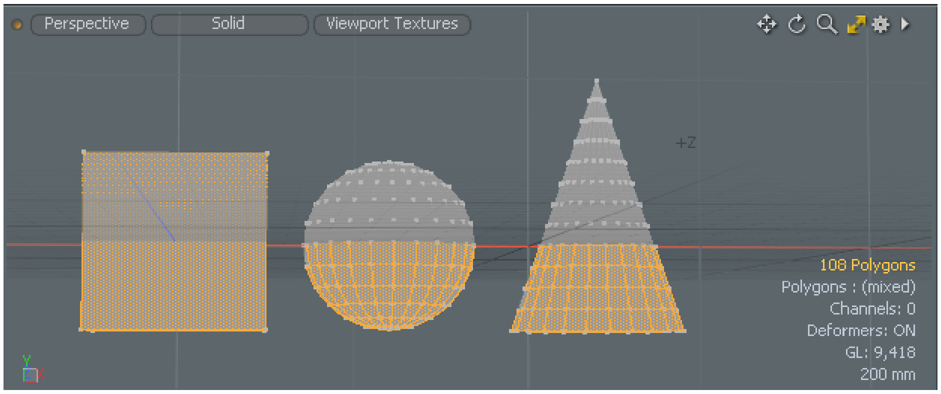

Perspective
Solid
Viewport Textures
+Z
303 Edges
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 9,418
200 mm

Two vertices define an edge, which itself is the line between the vertices. When in Edges selection mode, you can use the basic paint or lasso selection options to quickly select one or more edges of the geometry for precise editing. Modo indicates selected edges with an orange color to distinguish them from unselected edges. By dragging over an object, you can paint a selection of one of more edges. If you right-click and drag, you can draw a lasso around the selection area to select multiple edges at once. Modo provides other selection options to use with edges such as the **Loop Select** and **Ring Select** functions. (For information about these and other selection functions, see the Using Selection Modifiers topic.) If you wish, you can change the lasso style by choosing **Select > Lasso Style** and specifying a different style.

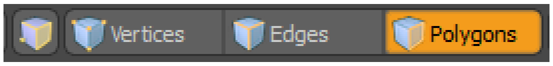


You can activate Edges selection mode by clicking the **Edges** interface button in the modes toolbar. Alternatively, choose **Select > Edges**, press **2**, right-click away from any geometry in the 3D viewport and choose **Edges**, or middle-click any edge in the viewport.

Polygon Selection Mode

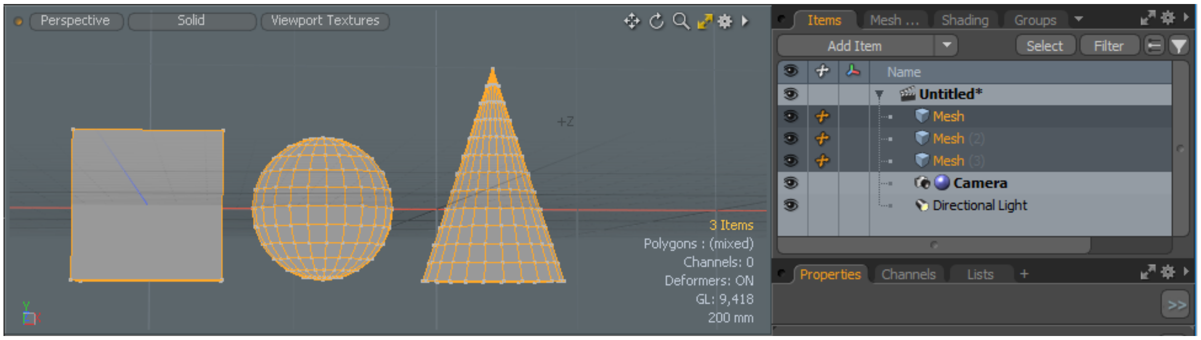


A polygon is a closed plane that is created from three or more edges for a renderable surface. When in Polygons selection mode, you can use the basic paint or lasso selection options to quickly select one or more polygons of the geometry for precise editing. Modo indicates selected polygons with an orange color to distinguish them from unselected polygons. By dragging over an object, you can paint a selection of one of more polygons. If you right-click and drag, you can draw a lasso around the selection area to select multiple polygons at once. Modo provides other selection options to use with polygons such as the **Loop Select** and **Select Connected** functions. (For information about these and other selection functions, see the Using Selection Modifiers topic.) If you wish, you can change the lasso style by choosing **Select > Lasso Style** and specifying a different style.

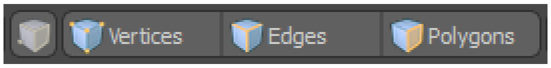


You can activate Polygons selection mode by clicking the **Polygons** interface button in the modes toolbar. Alternatively, choose **Select > Polygons**, press **3**, right-click away from any geometry in the 3D viewport and choose **Polygons**, or middle-click any polygon in the viewport (except in Wireframe draw mode).

Items Selection Mode

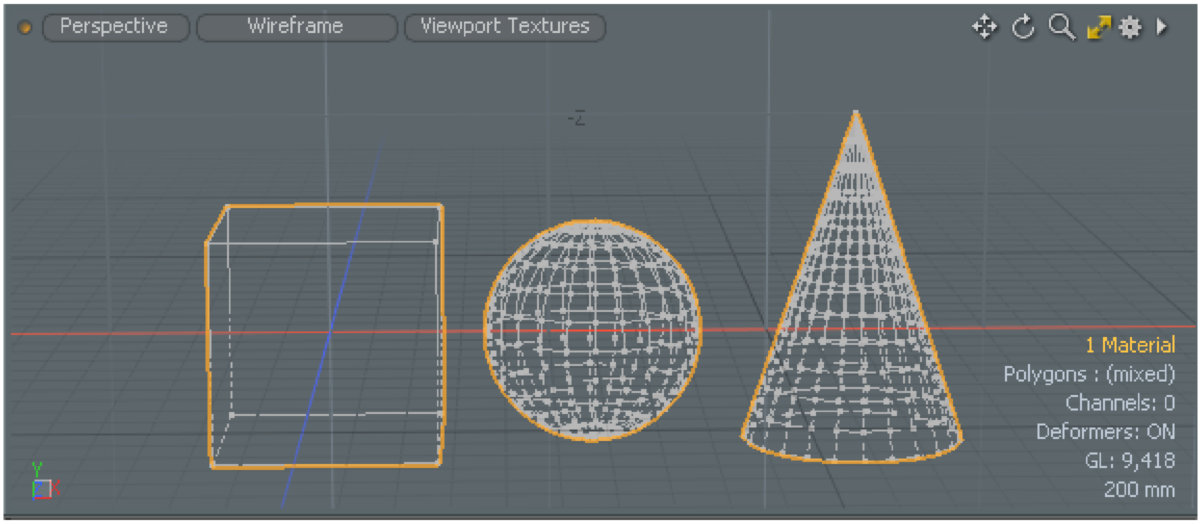

Perspective
Solid
Viewport Textures
+Z
3 Items
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 9,418
200 mm
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Mesh
Mesh (2)
Mesh (3)
Camera
Directional Light
Properties
Channels
Lists
+
>>

An item is an individual object layer in a scene — whether it's a camera, light, locator, mesh, or other type of object. Each one is individually an item. When modeling, you work in one of the component modes (Vertex, Edge, or Polygon mode). In these modes, you modify the contents of the layer items, but while in **Items mode**, you can select and transform entire layers. You can only modify mesh layers in a component mode. You transform all other elements in **Items mode**. For more information about the modes, see the Items vs. Components topic. You set all keyframes for animation while in Items mode. Component animation is only possible by using mesh deformers.


Vertices
Edges
Polygons

You can activate Items selection mode by deselecting the active selection mode buttons, displayed in orange, in the modes toolbar. For example, clicking on the **Polygon** selection button takes you into polygon selection mode as normal and that button is highlighted in orange. To return to **Items** mode click on the orange **Polygon** button to turn it off. Once done, Items selection mode is activated. Alternatively, choose **Select > Items**, press **5**, or right-click away from any geometry in the 3D viewport and choose **Items** from the contextual menu.

Material Selection Mode

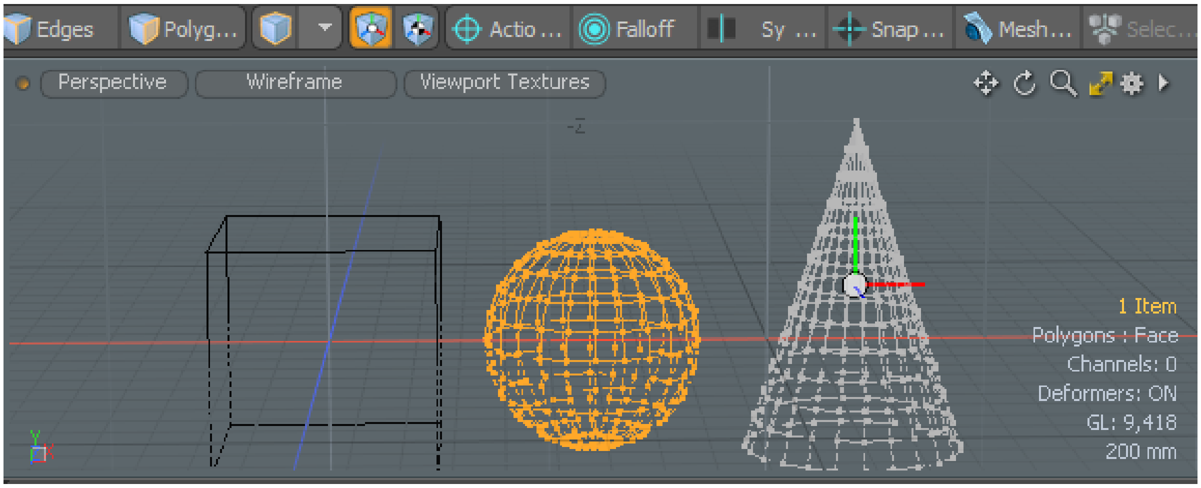


You define materials when you select specific polygons and assign a Material tag. These tags produce a Material Group Item in the **Shader Tree** that represents the tagged polygons for masking. Modo applies any surfacing to the geometry by using this mask. You can then activate Material selection mode to select geometry based on tag definitions. Modo indicates material selections as orange outlines surrounding the area defined by the tags. You can either use the selection to edit the geometry or adjust the surfacing within the **Shader Tree**. When you select a specific material tag in the 3D viewport, Modo also selects the corresponding group item in the **Shader Tree** to make it easier to locate a specific Shader Tree layer in a complex scene.

With Material selection mode active, you can select from several polygon tag types that Modo offers. By choosing **Select > Polygon Tag Type**, you can specify the Materials selection mode to be based on **Material**, **Part**, or **Smoothing Group** polygon tags.

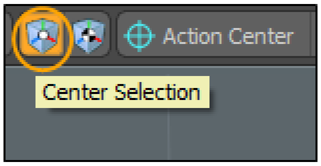
You can activate Materials selection mode by clicking **Select > Materials** in the modes toolbar. Alternatively, choose **Select > Items**, press **4**, or right-click away from any geometry in the 3D viewport and choose **Materials**.

Center Selection Mode


Edges
Polyg…
Actio …
Falloff
Sy …
Snap …
Mesh…
Selec…
Perspective
Wireframe
Viewport Textures
-Z
1 Item
Polygons : Face
Channels: 0
Deformers: ON
GL: 9,418
200 mm

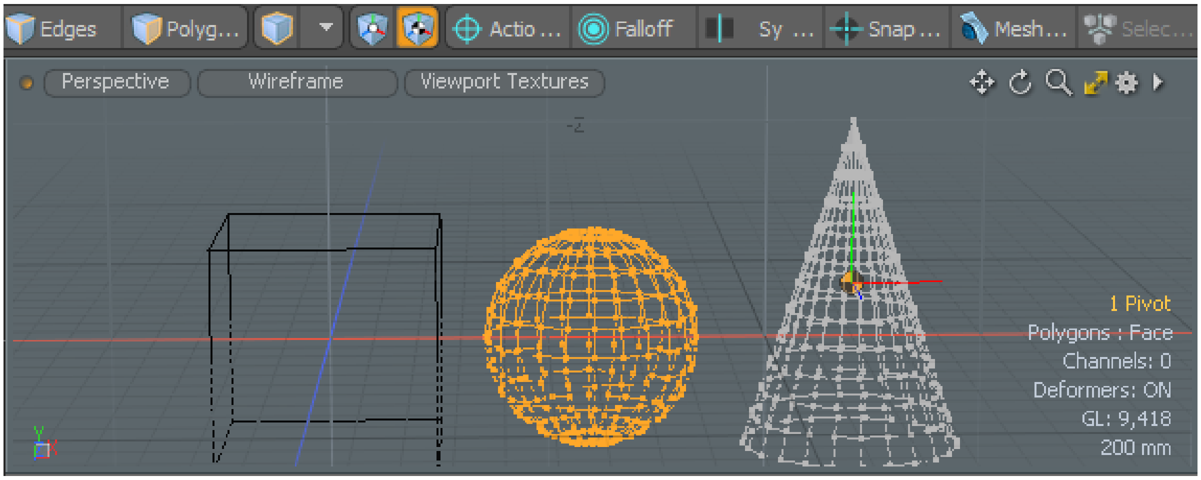
The center element represents the center location of an item layer. Each layer in the **Items** list has its own corresponding center position. Modo uses this as the origin for all Position, Scale and Rotate transforms for that layer. You can use action centers to dynamically define transform origins for modeling operations, but a center defines the origin when animating in Modo. By default, the center is at the World Position origin of 0,0,0 for any item layer. You can activate Centers mode and select the center element by clicking the center icon. You can then use the Move and Rotate transform tools to reposition the center in relation to the geometry without affecting the geometry's position, itself.

You can activate center selection mode by clicking the **Center Selection** icon in the modes toolbar.


Action Center
Center Selection

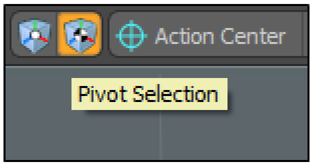
Alternatively, choose **Select > Center**, press **7**, or right-click away from any geometry in the 3D viewport and choose **Centers**.

Pivot Selection Mode


Edges
Polyg...
Actio ...
Falloff
Sy ...
Snap ...
Mesh...
Selec...
Perspective
Wireframe
Viewport Textures
1 Pivot
Polygons : Face
Channels: 0
Deformers: ON
GL: 9,418
200 mm

A pivot is a secondary center-like element that is offset from the center. Typically, the pivot and center positions coincide for any layer. If you modify the pivot's position, then that pivot postion becomes the animation origin and overrides the center position. Modo differentiates pivots from centers in that you can use a keyframe to animate (or rotate) a pivot position. You cannot animate (nor rotate) a center.

You can activate Pivot selection mode by clicking the **Pivot Selection** icon in the modes toolbar.


Action Center
Pivot Selection

Alternatively, choose **Select > Pivot**, press **6**, or right-click away from any geometry in the 3D viewport and choose **Pivots**.

Implicit Selections for Component Modes

Beyond directly selecting elements, with all selection modes you implicitly select vertices, polygons, and edges. For example, when you select a polygon, Modo also implicitly selects the vertices that make up that polygon. In this way you can select a polygon and still apply the Vertex Bevel tool, for example, and the vertices in that polygon are the active mesh elements for that tool. The vertices do not appear as selected in the viewport, but, nonetheless, you can perform vertex operations on them.

In all modes, if you don't select anything directly, Modo implicitly selects all vertices, polygons, or edges.

If you make a direct selection, Modo makes the following implicit selections:

Vertex mode

- selected vertices only

Edge mode

- vertices of selected edges
- selected edges

Polygon mode

- vertices of selected polygons
- contour edges of selected polygons
- selected polygons

When you select several polygons in Polygon mode and then use an edge tool, Modo applies the action to all the edges around the polygon group. This contour edge is a simple, one-step method for selecting certain types of edge loops.

By pressing the **Alt** key, you can convert implicit selections to direct selections. For example, if you have selected 3 polygons, pressing the **Alt** key as you click the **Polygon** button in the modes toolbar changes the viewport selection to edges. Doing the same for an Edge selection converts the selection to vertices. Pressing **Ctrl** changes the **Edge** button to a **Boundary** button for you to convert existing polygon selections to boundary-edge selections.

Selection Styles

Paint Select

You can paint geometry selections by clicking elements in 3D view. Click away from any geometry first to deselect everything. Then drag to have Modo select all elements under the pointer as you move the pointer on the screen. In this manner, you can quickly "paint" a selection by dragging over a desired component.

Press the **Shift** key as you select current selection mode elements to add them to the selection. You can either **Shift**-click single elements or **Shift**-drag across multiple elements to add multiple elements.

Ctrl-click removes items from the selection.

Lasso Select

You can select geometry in 3D view by right-clicking and dragging to enclose the geometry within a lasso. Geometry must be fully enclosed in the lasso to be selected.

In Shaded mode, selecting with the lasso uses a backface cull, and, therefore, Modo does not select polygons, edges, or vertices facing away from the view plane. If you middle-click and drag, Modo selects through the geometry to make selections on the backside as well. This is reversed in Wireframe mode: using the lasso selects through the mesh, but using the middle-click and drag implements the backface cull to select only facing geometry.

To add new geometry to a lasso selection:

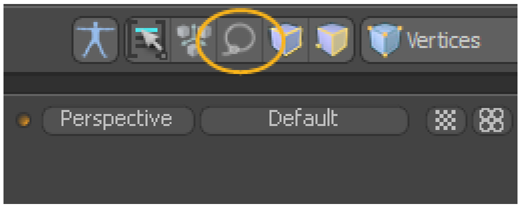
- Press the **Shift** key as you continue using the lasso selection. Modo adds the new selection to the original selection.

To remove geometry from a selection:

- Press **Ctrl** as you continue using the lasso selection. Modo removes the newly selected geometry from the existing selection.

Lasso Partial Mode

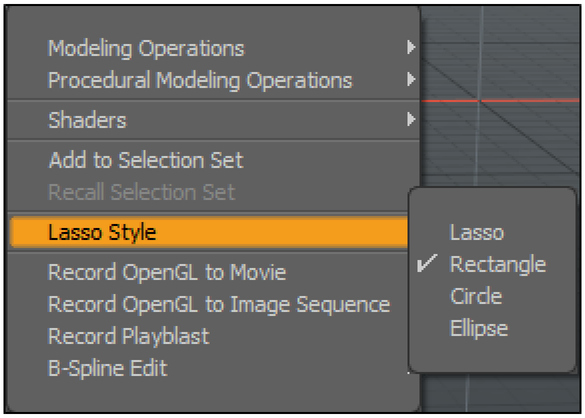
Partial Selection mode for lasso allows you to select entire edges or polygons by only selecting part of it. To enable this, hold down the **8** key when making your selection, or click the Lasso Partial Selection Mode button at the top of the viewport.



Lasso Style specifies the shape of the lasso selection. The style defaults to a regular, freeform lasso, but you can specify **Rectangle**, **Circle**, or **Ellipse**.

To change the lasso style to a rectangle, circle, or ellipse:

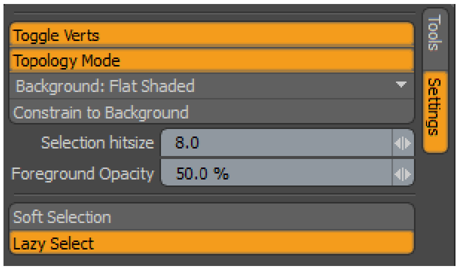
- Click **Select > Lasso Type** and specifying an alternate style.
- Alternatively, right-click in the 3D viewport, select **Lasso Style** from the contextual menu, and select an alternate style.


Modeling Operations
Procedural Modeling Operations
Shaders
Add to Selection Set
Recall Selection Set
Lasso Style
Record OpenGL to Movie
Record OpenGL to Image Sequence
Record Playblast
B-Spline Edit
Lasso
✓ Rectangle
Circle
Ellipse

Lazy Selection

Lazy Selection allows you to select elements by clicking in their proximity, rather than clicking on the elements precisely. When you enable this option in any selection mode, you can click anywhere on your mesh, and the nearby elements get selected. Click and drag in an area to select all nearby elements.

You can enable Lazy Selection in the menu bar, by clicking **Select** > **Lazy Selection**. Alternatively, in the **Topology** layout, you can find it in the **Settings** subtab of the left-side panel:


Toggle Verts
Topology Mode
Background: Flat Shaded
Constrain to Background
Soft Selection
Lazy Select
Tools
Settings

Selection hitsize 8.0

Foreground Opacity 50.0 %

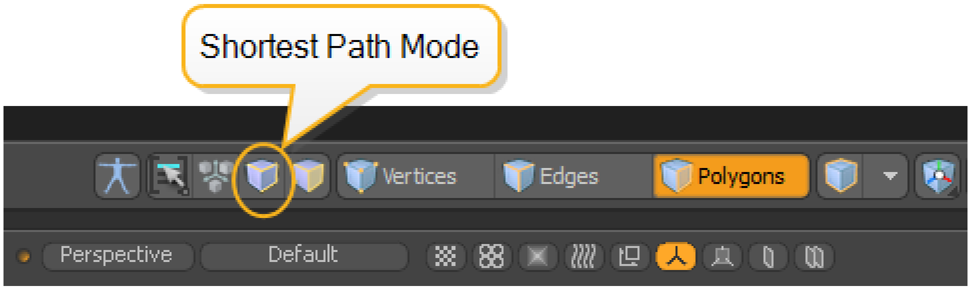
Click the **Lazy Select** button to activate it.

Shortest Path Selection

Shortest Path selection finds the shortest path between two elements on a continuous mesh. It works with vertices, edges, and polygons, and with Symmetry. You can use this as a selection mode for paint selection, or a stand-alone command.

Shortest Path Mode

To activate Shortest Path Mode, click the Shortest Path Mode button at the top of the viewport.



Alternatively, in the menu bar, click **Select** > **Shortest Path Mode**.

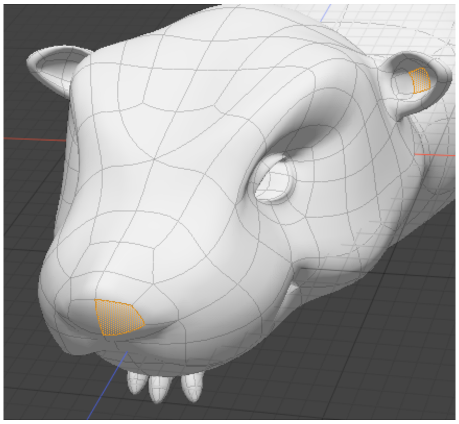
Now if you select an element in the 3D viewport, and move the cursor, the shortest path between the selected element and the current element is highlighted. With an element selected, hold **Shift** and click another element to select it along with the shortest path between the two elements.

To start a new path that's not connected to the previous selection, hold **Ctrl**+**Shift**, and click to select a new component.

Shortest Path Command

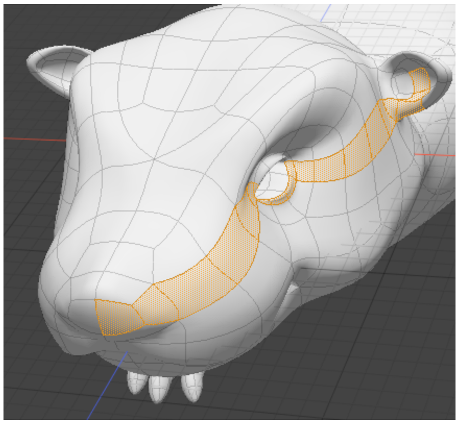
The **Shortest Path** command allows you to select elements between two selected elements. These can be vertices, edges, or polygons. To use the command:

1. Select two elements by holding the **Shift** key and clicking in the viewport.



2. In the Modo layout, go to the **Model** toolbar on the left, click the **Select** tab, and click the **Shortest Path** button.

   This selects the elements on the shortest path between the two original elements.



Dropping Selections

To deselect (or clear) the current selection, click anywhere in a 3D viewport, press the **Spacebar**, or press the **Esc** key. If you click another element of the current selection mode, Modo clears the original selection and selects the element you clicked. Making selections is an undo-able action. You can easily undo a selection state if you accidentally clear a selection.

When nothing is selected, Modo considers everything active for the tool and command functions. This makes dropping selection a quick way to select everything.

Using Vertex Component Tools

Some of Modo's tools are specific to the various component types: vertices, edges, and polygons. For example, there are three component beveling tools, and each has its own settings. The Edge Bevel tool has a rounding option, but the Polygon Bevel tool does not. Some tools are interactive tools, but others are one-shot commands that produce a specific result. You may use some regularly and others only rarely; however, all could be considered indispensable. For information on using a specific tool, click any link below to see its topic.

Vertices		
	Vertex Bevel	Vertex Bevel replaces each selected vertex with 3 vertices, which are moved away from the original position.
	Create Vertex	You can create vertices one at a time by specifying their

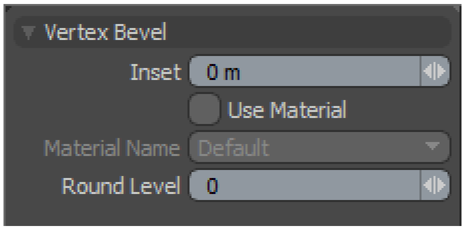
Vertices		
		location in 3D space.
	Join	Welds the selected vertices to the last vertex selected and then moves them all to the position of the last selected vertex.
	Split	Takes a single vertex and creates multiple vertices in the same location based on the number of polygons that share the point.
	Vertex Extrude	Shifts the selected vertex along its normal (away from its original position) and creates three additional vertices and edges around the original vertex.
	Set Position	Repositions an existing vertex (or many) to a specific position by typing new coordinates.
	Remove	Deletes the selected vertices, edges, or polygons from the 3D model view.
	Merge	Measures the distance between vertices in the currently-selected geometry to see if any fall within a specific distance threshold to one another.
	Vertex Merge Tool	Adjusts a distance threshold value by dragging in the 3D viewport or by using a mini-slider.
	Align	Align a selection of vertices into a straight row.
	Collapse	Removes the selected element without destroying the integrity of the geometry.

Vertex Bevel

Vertex Bevel replaces each selected vertex with 3 vertices, which are moved away from the original position. The **Roundness** option generates more vertices to create a rounding effect. This tool is accessed from the

Vertex tab of the Modo **Tools** toolbar. If you select vertices, you can also select this tool from the **Geometry** > **Bevel** menu. Unlike Polygon Bevel, only a single handle is available when beveling vertices.

Vertex Bevel can also be used on curves, except for Bezier.


Vertex Bevel

Inset 0 m

☐ Use Material

Material Name Default

Round Level 0

The following **Vertex Bevel** options are available for adjusting the **Vertex Bevel** tool:

Vertex Bevel	
Inset	Determines the distance of the generated vertices from the original vertex location.
Use Material	When enabled, all new polygons created automatically are assigned a **Material** polygon tag. The **Material** name is chosen from the pop-up just beneath this control.
Material Name	This pop-up is only active when the **Use Material** flag is enabled. Use this pop-up to choose any **Material** names that already exist in the project. This is assigned to all polygons created by the bevel operation.
Round Level	Increasing this value adds additional vertices to the newly-created edges and applies a rounding effect.

Vertex Bevel in the Procedural

Modo has a procedural Vertex Bevel mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the stack in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Vertex**, double-click **Vertex Bevel**.

You can apply the bevel to a selection by making the selection before adding the bevel operation. This adds a **Select by Index** to the **Selection** input of the bevel. Once added, the selection can be edited using the **Properties** panel for the **Select by Index**. For more information, see Procedural Selection.

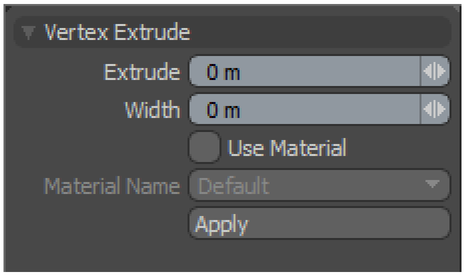
You can expand the Vertex Bevel item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses. Vertex Bevel requires the following inputs:

- **Selection** - Allows you to select the vertices you want to bevel.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Vertex Extrude

Found on the **Vertex** tab of the Modo Tools toolbar, the Vertex Extrude tool shifts the selected vertex along its normal (away from its original position) and creates three additional vertices and edges around the original vertex.



Vertex Extrude
Extrude 0 m
Width 0 m
☐ Use Material
Material Name Default

Vertex Extrude	
Extrude	Determines how far out the vertex is shifted away from its original location.
Width	Determines how far the newly created edges that surround the original vertex are pushed out away from that location.
Use Material	When enabled, determines what material group to assign to the newly created geometry. When disabled, the new polygons do not have a material group and use the base material.
Material Name	Specifies the material group for the newly created geometry after you enable **Use Material**.

Vertex Extrude in the Procedural

Modo has a procedural Vertex Extrude mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the list in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Vertex**, double-click **Vertex Extrude**.

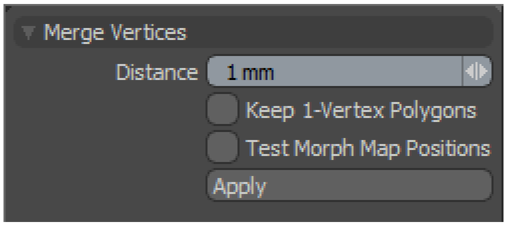
You can expand the Vertex Extrude item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses. Vertex Extrude requires the following inputs:

- **Selection** - Allows you to select the vertices you want to extrude.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Vertex Merge Tool

With the Vertex Merge tool you can interactively adjust a distance threshold value by dragging in the 3D viewport or by using a mini-slider. Modo merges any two or more vertices whose proximity to each other falls within that range.



Merge Vertices

Distance: 1 mm

☐ Keep 1-Vertex Polygons

☐ Test Morph Map Positions

Apply

Merge Vertices	
Distance	Determines if vertices are merged together. Modo merges any vertices whose distance apart is less than or equal to this value.
Keep 1-Vertex Polygons	When enabled, leaves behind single- and two-vertex polygons when merging vertices. This is disabled by default for the tool to clean up error-causing polygons. However, there are some cases where a you may want to use the resulting 1- and 2-vertex polygons as a starting point for additional geometry.
Test Morph Map Positions	When enabled, tests the morph map positions and only performs the merge if the morphed positions are the same.

Vertex Merge in the Procedural

Modo has a procedural Vertex Merge tool available in the **Mesh Operations** list. In the **Model** layout, you can find the stack in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Vertex**, double-click **Vertex Merge**.

You can expand the Vertex Merge item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses. Vertex Merge requires the following inputs:

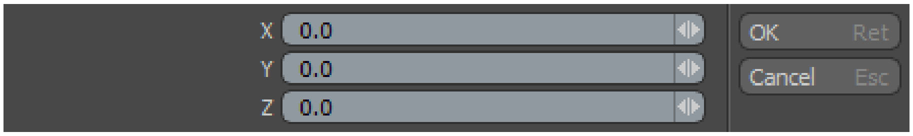
- **Selection** - Allows you to select the vertices to merge.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

The Procedural Vertex Merge tool has two additional options in its **Properties** panel:

- **Allow Self Merge** - When enabled, vertices from the same polygon island are allowed to merge. When disabled, only separate polygon islands can be merged together. This is useful for merging two halves of a shape together.
- **Merge To Closest Vertex** - Traditionally, Vertex Merge finds all vertices within the threshold distance and merges them together. This often results in meshes collapsing to a single vertex if the distance value was high. When this option is enabled, Modo finds the closest vertex for each vertex and merges to that.

Vertex Tools

Create Vertex



You can create vertices one at a time by specifying their location in 3D space. To do so, choose **Geometry > Vertex > Create**.

To create an individual vertex:

1. Choose **Geometry** > **Vertex** > **Create**.
2. In the **Create New Vertex** dialog, specify the **X**, **Y**, and **Z** coordinates for the vertex's location.
3. Click **OK**.

   The vertex is visible in the 3D model view.

Procedural Create Vertex

Create Vertex is also available as a mesh operation. It works the same way as its direct modeling equivalent.

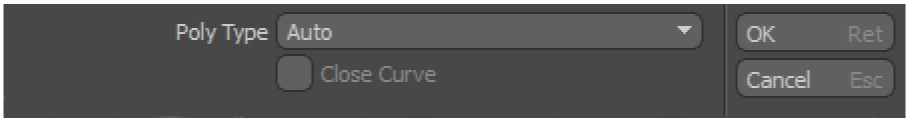
Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

To create a vertex procedurally using a mesh op:

1. Open the **Mesh Ops** tab.
2. Click **Add Operator**.
3. Under **Mesh Operations** > **Create**, double-click **Create Vertex**.
4. In the **Properties** tab, specify the **X**, **Y**, and **Z** coordinates for the vertex's location.

Make Polygon

The **Make Polygon** tool uses the currently-selected vertices to create a polygon. The order in which the vertices were created (or selected) determines the direction of the polygons' normal. When run with an edge loop selection, a new polygon is created from those edges, and the selection order is not important. This command can be triggered by pressing the **P** key or by using **Geometry** > **Polygon** > **Make**, which offers several additional user-definable options.

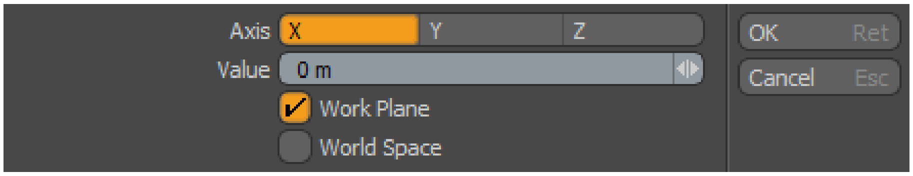

Poly Type: Auto

☐ Close Curve

The following polygon types are available.

Faces	Produces a regular polygon, based on the drawn vertices.
Subdivs	Generates a Subdivision Surface polygon, eliminating the need to toggle the newly-created geometry manually.
Curves	Generates a Spline curve from the points with the additional option to **Close Curve**, joining the in and out vertices into a loop.
SplinePatch	Produces a spline patch curve for further patching; this requires 3 or 4 vertices.
Catmull-Clark	Generates a Pixar Catmull-Clark Subdivision Surfaces polygon, eliminating the need to toggle the newly-created geometry manually.
Polyline	Generates a string of continuous polylines, similar to a curve but made from a string of line segments. These are often used as hair guides.
Auto	Matches the polygon type from a neighbor polygon connected to the selected vertex. If the selected vertices are floating, Modo chooses the **Faces** polygon type.

Set Position



Value: 0 m

☑ Work Plane

☐ World Space

With the **Set Position** command you can reposition an existing vertex (or many) to a specific position by typing new coordinates. For example, you can center an entire loop of vertices on the zero position for a mirror operation.

1. Select the vertices to change.
2. In the Toolbar on the left panel, go to the **Vertex** tab, and under **Commands**, click **Set Position...**.

   The command is also available in the menu bar, under **Geometry** > **Vertex**.
3. Select the **Axis** along which you want to move your vertices, and specify the **Value**.

   To assign the position value based off the **World Space** or the **Work Plane** position, check the appropriate property.
4. Click **OK**.

   The selected vertices are set to the new position.

Procedural Vertex Set Position

Set Position is also available as a mesh operation. It works the same way as its direct modeling equivalent.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

1. Select the vertices to modify.
2. In the **Mesh Ops** tab, click **Add Operator**.

Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **New Tab** > **Data Lists** > **Mesh Ops**.

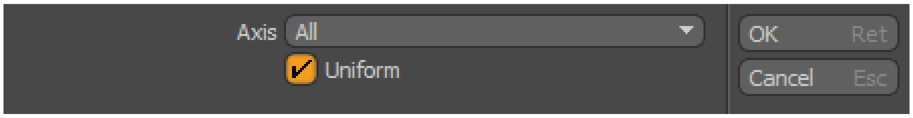
3. Under **Mesh Operations** > **Vertex**, double-click **Vertex Set Position**.

You can also find it in the Mesh Operations list.

4. Select the **Vertex Set Positon** mesh operation in the Mesh Operations list. Its **Properties** appear in the lower-right panel.
5. Enable the axes you want to move the vertices along by checking **Enable X**, **Y**, or **Z** and set the **Position** for the relevant axes.

   The selected vertices are set to the new position.

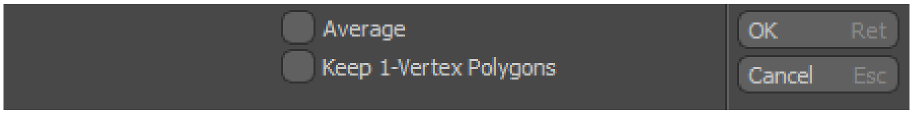
Align



With the **Align** command you can align a selection of vertices into a straight row (in a line between the first and last vertices selected).

Option	Description
Axis	Confines the movement of the repositioned vertices to one or **All** axes. You can also select an axial plane (**XY**, **YZ**, or **ZX**).
Uniform	When enabled, spaces the vertices evenly (uniformly) along the span.

Join



The **Join** command welds the selected vertices to the last vertex selected and then moves them all to the position of the last selected vertex.

You access the **Join** command on the **Vertex** tab of the Modo Tools toolbar.

Option	Description
Average	When enabled, welds the selected vertices into a single vertex positioned at an average distance between the selected points.
Keep 1-Vertex Polygons	When enabled, allows single vertices to exist after the join operation. When disabled, Modo automatically deletes 1-vertex polygons.

Join Averaged

The **Join Average** command welds two (or more) vertices together at a central position between the two vertices.

To join multiple vertices:

1. Select the vertices to join together.
2. On the Modo Tools toolbar, click the **Vertex** tab.
3. In the Commands section, click **Join Averaged**.
4. In the **Join Vertices** dialog, read the message indicating how many vertices were joined into one.
5. Click **OK**.

Remove

The **Geometry Remove** command deletes the selected vertices, edges, or polygons from the 3D model view.

You can use the **Remove** command from the **Vertex** or **Edges** tab of the Modo Tools toolbar or choose **Geometry** > **Remove**. Also, you can use the **Backspace** or **Delete** key to remove vertices, edges, or polygons. **Geometry Remove** works on individual or multiple selected vertices, edges, or vertices.

Collapse

The **Collapse** command removes the selected element without destroying the integrity of the geometry. You can delete any selected polygon, edge, or vertex but, with this command, no hole is left behind. Instead, Modo heals the mesh to close any gaps by merging neighboring elements together.

You access the **Collapse** command in several places:

- **Vertex**: On the **Vertex** tab of the Modo Tools toolbar click **Commands** > **Collapse**.
- **Edge**: On the **Edge** tab of the Modo Tools toolbar click **Commands** > **Collapse Edges**.
- **Polygons**: On the **Polygons** tab of the Modo Tools toolbar click **Commands** > **Reduce** > **Collapse**.
- You can also select whatever geometry to collapse and then choose **Geometry** > **Collapse**.

Split

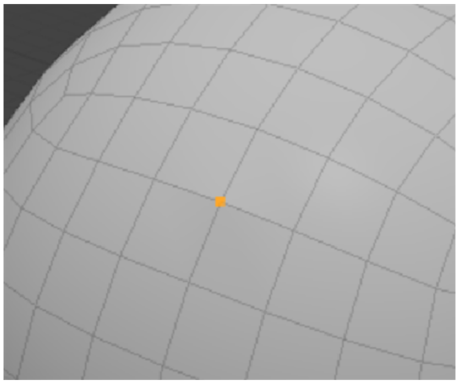
The **Split** command for vertices is the opposite of the **Join** command. Instead of combining multiple vertices into a single vertex, this command takes a single vertex and creates multiple vertices in the same location based on the number of polygons that share the point.

To use the tool:

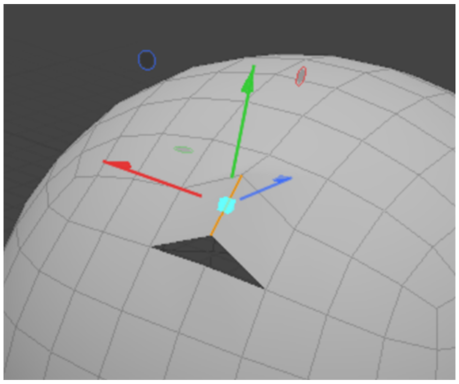
1. Select the vertices to split.
2. In the Toolbar on the left panel, go to the **Vertex** tab, and under **Commands**, click **Split**.

   The command is also available in the menu bar, under **Geometry** > **Vertex**.

   Unlike the **Join** command, **Split** creates the multiple vertices in their original location, it doesn't move them apart. Select any polygon or edge and move it to separate the split vertices.



The vertices after applying **Split**.



Moving an edge reveals that the vertices are split.

Procedural Vertex Split

Vertex Split is also available as a mesh operation. It works the same as its direct modeling equivalent.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

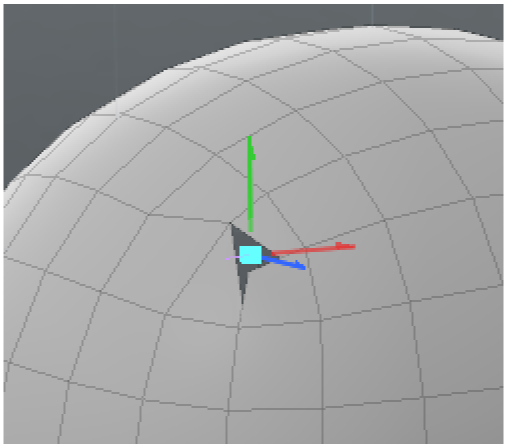
1. Select a vertex to split.

2. In the **Mesh Ops** tab, click **Add Operator**.

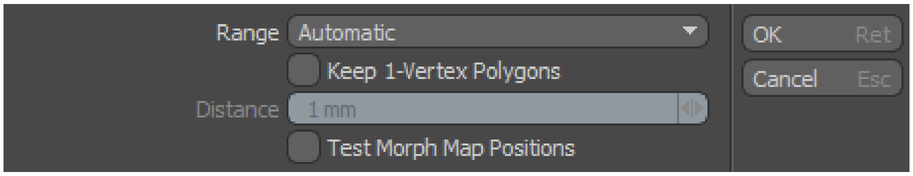
Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

3. Under **Mesh Operations** > **Vertex**, double-click **Vertex Split**.

   The mesh operation is applied to your geometry and you can also find it in the Mesh Operations list. If you then move a connected edge or polygon, the split vertices are revealed. The following example shows the **Edge Slide** mesh operation applied to a connected edge.



Merge



The **Merge Vertices** command measures the distance between vertices in the currently selected geometry to see if any fall within a specific distance threshold to one another. If any vertices are within that range, Modo combines them into a single vertex. It positions the resulting vertex at the site of the most recently selected vertex of the merged vertices.

The **Merge** command is on the **Vertex** tab of the Modo Tools toolbar in the **Tools** section. The following options are available:

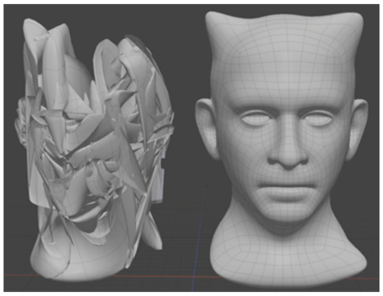
Option	Description
Range	Specifies when to merge vertices. By default, the **Merge Vertices** command uses a **Range** setting of **Automatic**. • **Automatic**: Merges only vertices that share the exact location on all three axes. • **Fixed**: Provides a way to specify a distance threshold. Select to increase the distance allowed between two vertices to be merged. After selecting **Fixed**, you can set a threshold value for the **Distance**.
Keep 1-Vertex Polygons	When enabled, leaves behind any 1-vertex polygons from merging vertices. Often when merging vertices within a certain threshold, polygons become flattened into only one vertex. In this case they become 1-vertex polygons. These 1-vertex polygons can cause rendering anomalies and other issues in the model file when exporting to other systems. When disabled (by default), Modo removes these 1-vertex polygons. However, there are some cases where you may want to keep the resulting 1-vertex polygons.
Distance	Specifies the distance threshold if you selected **Fixed** as the **Range**. For example, if the **Distance** is set to one meter, Modo merges any points that fall within one meter of another point.
Merge Discontinuous UV Values	Enabled by default, merges discontinuous UV values together into continuous data to optimize the UV map.

Vertex Index Order

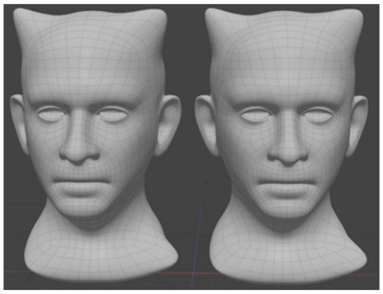
In order for mesh deformation to work correctly with morph influences, Modo assigns each vertex on a mesh a numerical index which becomes part of the vertex index order. This ensures each vertex on the mesh is mapped and stored correctly with the morph influence.

Note: See Morph Influence for more information on creating deformations with Morph Influences.

However, using mesh edit tools on the original mesh can change the index order, causing deformation issues between the edited mesh and the target mesh.



An example of a morph map with mismatched vertex index order.



A fixed vertex index order between two meshes results in a correct morph.

Reordering Vertex Index Order

There are two options in the **Vertex** tab to reorder the vertex order on your meshes:

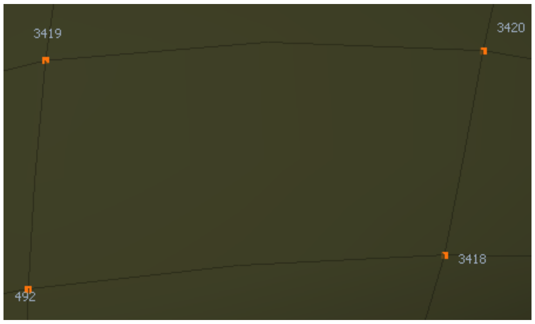
- **Reorder Index** - assigns the selected vertices a new index order based on the **Mode** you choose. This allows you to repair specific vertices on meshes for deformation purposes, or repair game assets that require a strict index order.
- **Sync Vertex Order** - matches the vertex index order of one mesh to another mesh based on the polygons you have selected. You can use this option to fix mismatches in vertex index order between two meshes when you are creating a morph map between the two meshes.

Tip: Enable **Show Indices** in the **3D Viewport Properties** to see the index number of a selected vertex in the viewport.

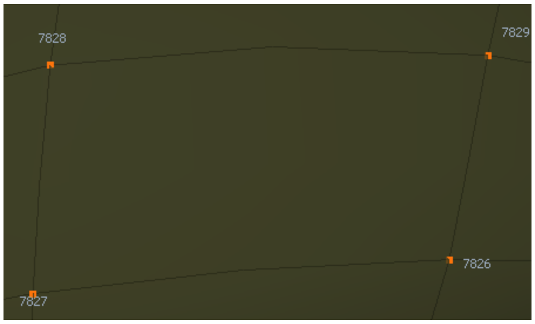
To fix the index order of specific vertices on a single mesh:

1. Click **Vertices** on the **Modo** layout to enter Vertex Selection mode.
2. Select the vertices on the mesh in the order you would like them to appear in the index.
3. Click **Reorder Index** in the **Vertex** tools tab.
4. Select a **Mode** from the dropdown:
   - **Swap** - Assigns the selected vertices a new index order based on the original indices of the selected vertices, and the order in which the vertices are selected. In the following image on the left, the

vertices were selected in the following order: 3420, 3418, 492, 3419. After **Swap** mode was assigned in the image on the right, the selected vertex order became 492, 3418, 3419, 3420.

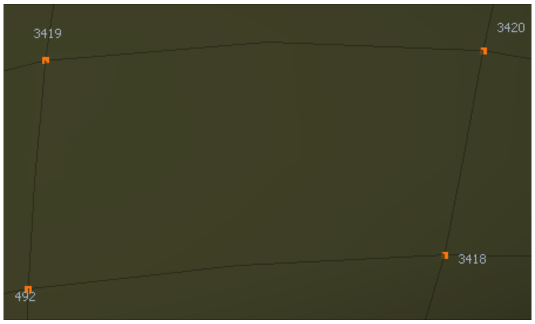


The original index order of the selected vertices.

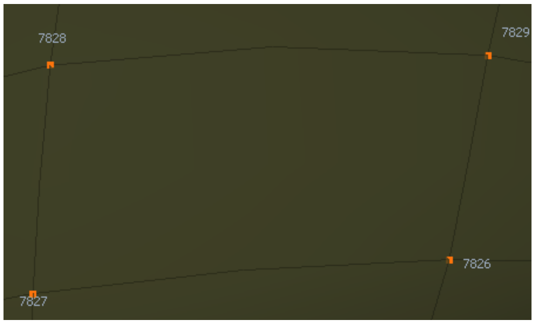


The new index order of the selected vertices when selecting **Bottom** mode.

- **Top** - The selected vertices are moved to the top of the index order based on the order the vertices were selected. The first vertex to be selected is assigned an index of 0. All unselected vertices are appended to the new index. In the following image on the left, the vertices were selected in the following order: 3420, 3418, 492, 3419.

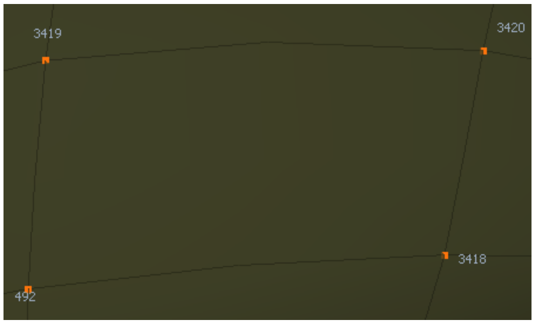


The original index order of the selected vertices.

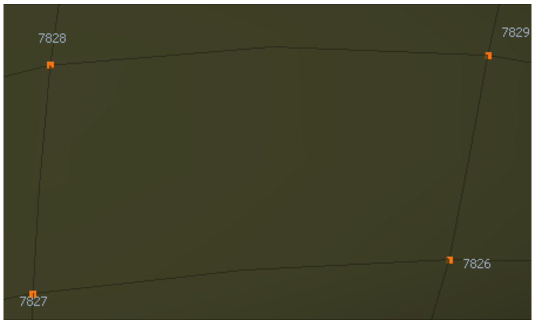


The new index order of the selected vertices when selecting **Bottom** mode.

- **Bottom** - The selected vertices are moved to the bottom of the index order based on the order the vertices were selected. The last vertex to be selected is assigned the final index number in the list on the entire mesh. All unselected vertices on the mesh are appended before the new index. In the following image on the left, the vertices were selected in the following order: 3420, 3418, 492, 3419.



The original index order of the selected vertices.



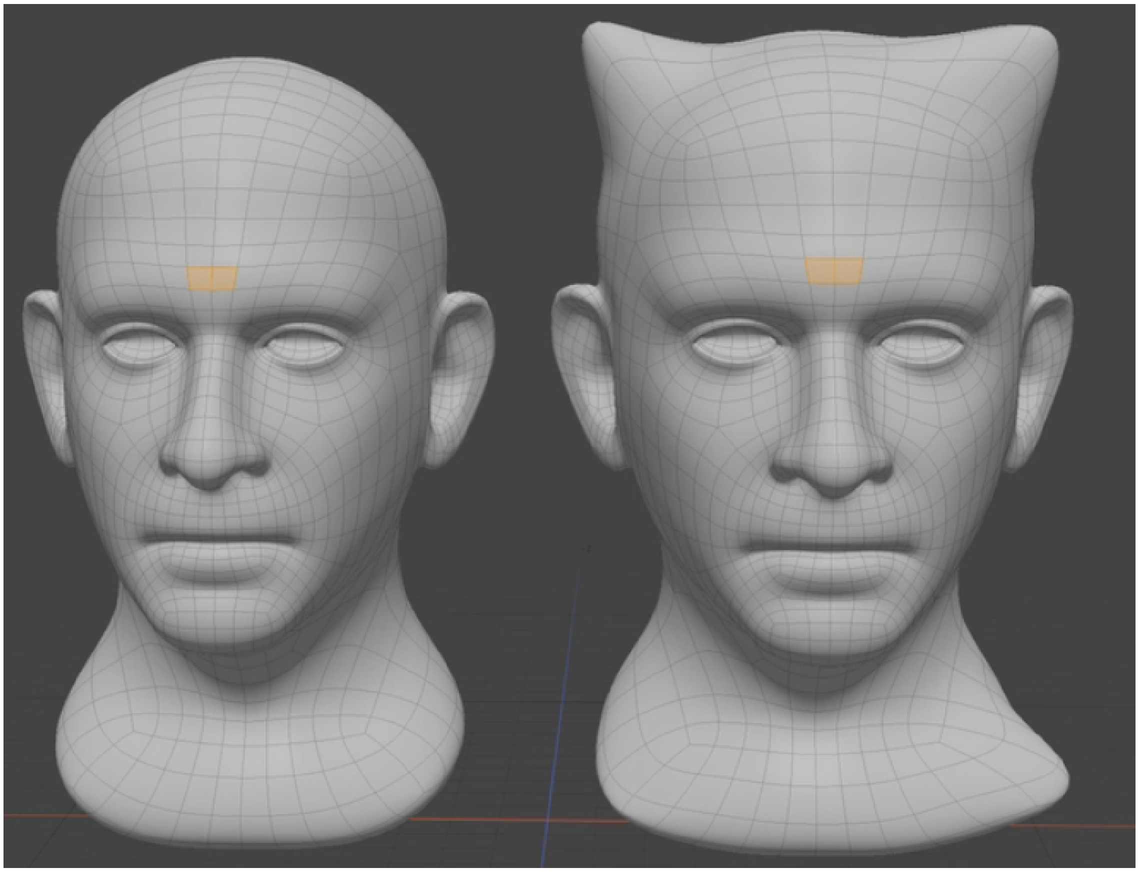
The new index order of the selected vertices when selecting **Bottom** mode.

5. Click **OK**.

To fix a mismatch in vertex index order between two meshes with the same topology:

1. Click **Polygons** on the **Modo** layout to enter Polygon Selection mode.
2. Hold **Shift** and select two polygons on one mesh, then hold **Shift** and select the same polygons on the second mesh.

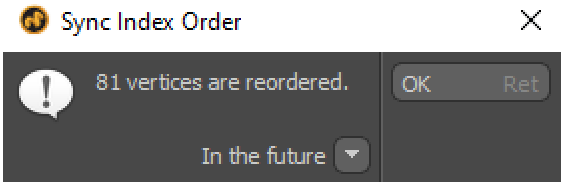
   Four polygons are selected.



The same two polygons on the source and target mesh are selected.

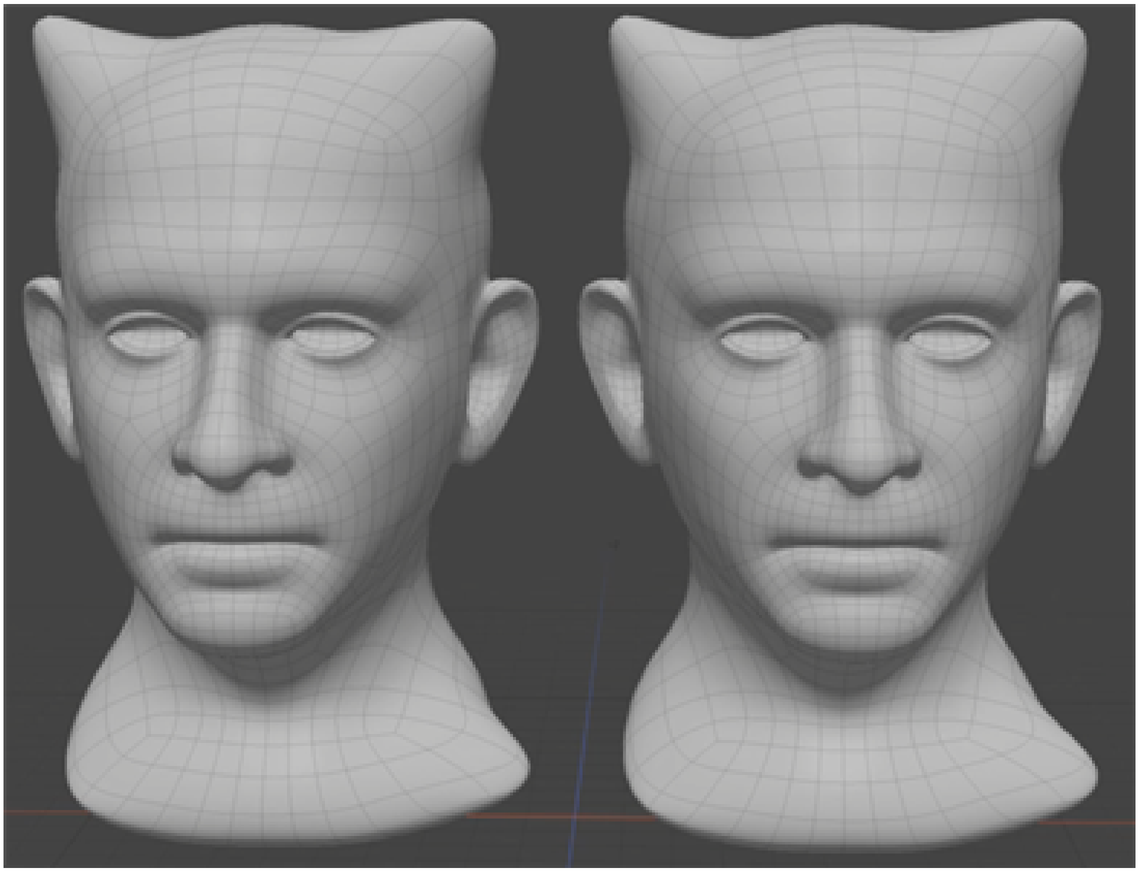
3. Click **Sync Vertex Order** in the **Vertex** tools tab.

The vertex index order on the two models are synchronized, and a dialog appears displaying the amount of vertices that were reordered. In this example, the mesh heads had 81 vertices out of sync with one another.


Sync Index Order

81 vertices are reordered.

The **Sync Index Order** dialog.



The source mesh correctly using the target mesh as a morph influence.

Using Edge Component Tools

Some of Modo's tools are specific to the various component types: vertices, edges, and polygons. For example, there are three component beveling tools, and each has its own settings. The Edge Bevel tool has a rounding option, but the Polygon Bevel tool does not. Some tools are interactive tools, but others are one-shot commands that produce a specific result. You may use some regularly and others only rarely; however, all could be considered indispensable. For information on using a specific tool, click any link below to see its topic.

Edges		
	Edge Bevel	Widens the selected edges into quad strips of a uniform specified width.
	Slide / Edge Slide	Moves selected mesh elements (vertices, edges and polygons) along their connected edges.
	Edge Tools	Turns the current edge or edges to re-organize the way they separate the two adjoining polygons.
	Edge Tools	Welds two edges together and the resulting edges have a position that is an average of the original edge positions.
	Edge Extrude	Shifts edges outward along the average normal of their two neighboring polygons.
	Add Point	Adds a new vertex point along an existing polygon edge, splitting the edge in two, or along a curve wherever you click.
	Edge Relax	Changes the radius of edge bevels.
	Edge Chamfer	Round selected edges of a mesh.
	Edge Tools	Removes the selected element without destroying the integrity of the geometry.

Edges		
	Split	Separates mesh polygons along selected edges.
	Edge Extend	Creates geometry from selected edges by extending them outward.
	Add Loop	Adds a continuous edge through a series of polygons.
	Edge Tools	Deletes edges from the mesh geometry.
	Split Normals	Creates hard or soft transition edges between polygons based on edge selections.
	Edge Tools	Fills any open holes within a mesh with only quad polygons.
	Edges to Curves	Creates Spline curves, Polylines, Bezier curves, or B-spline curves from edges on an existing mesh.

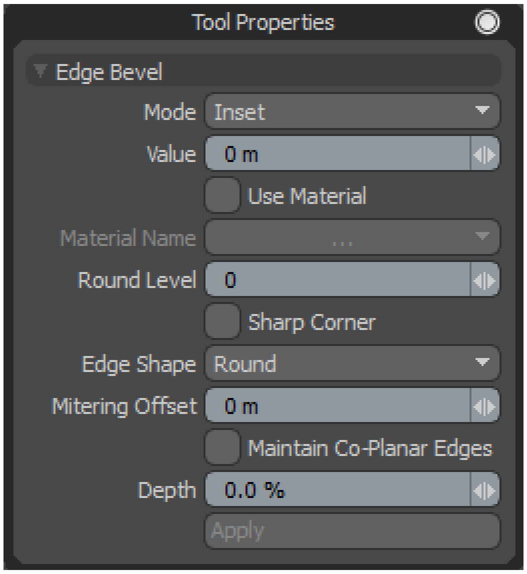
Edge Bevel

When applying **Edge Bevel**, selected edges widen into quad strips of a uniform, specified width. New polygons are centered over the original edge and are oriented with the average normal of the two polygons on either side. Surrounding polygons remain in their same plane, simply shrinking on their affected edges.

For direct modeling, the **Edge Bevel** tool is available in the modeling Toolbox. Open the **Edge** tab and click **Bevel**. The **Edge Bevel** tool is also available as a procedural in the **Mesh Operations** list. Open the **Mesh Ops** tab, click **Add Operator**, and select **Mesh Operations** > **Edge** > **Edge Bevel**.

Edge Bevel Options

The following options are available in the Direct Modeling and Procedural Mesh Operations tools:



Tool Properties

Edge Bevel

Mode Inset

Value 0 m

☐ Use Material

Material Name ...

Round Level 0

☐ Sharp Corner

Edge Shape Round

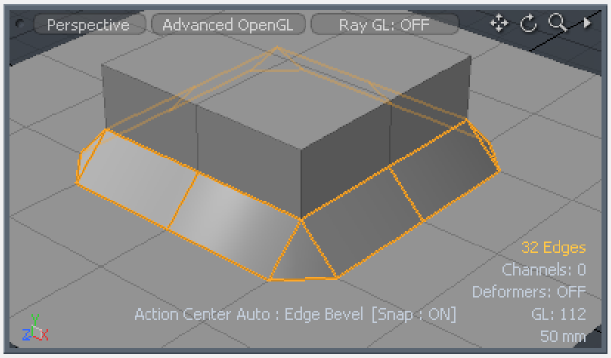
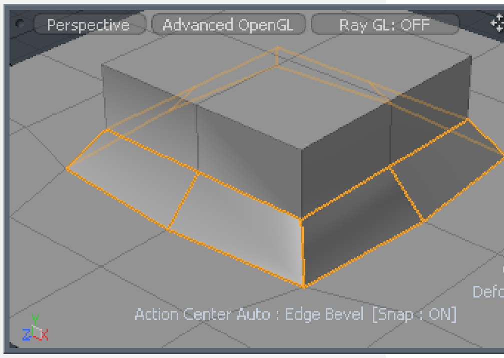
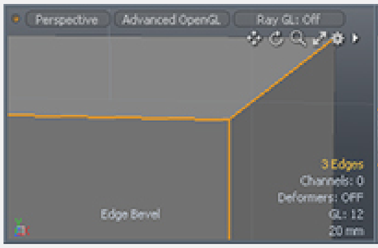
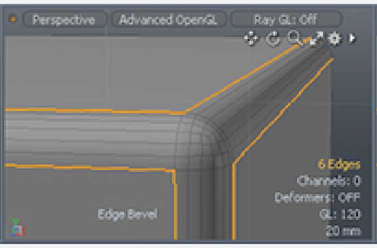
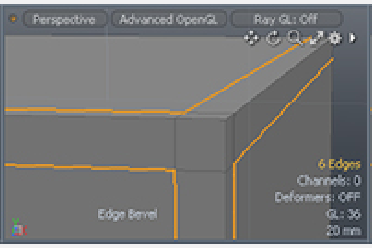
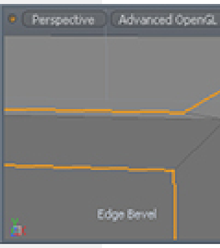
Mitering Offset 0 m

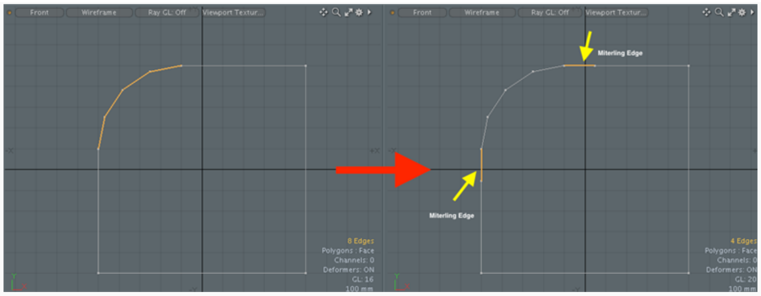
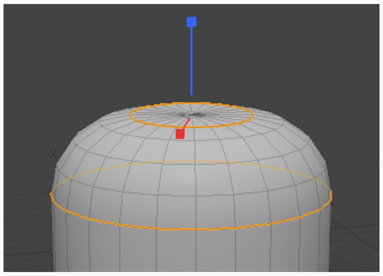
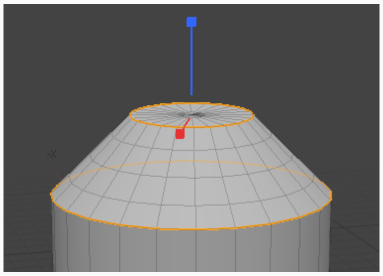
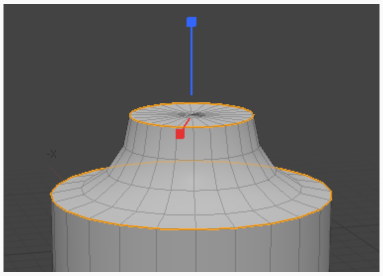
☐ Maintain Co-Planar Edges

Depth 0.0 %

Apply

Edge Bevel	
Enable	Enables or disables the Polygon Bevel tool. **Note:** This option is only available for the procedural Edge Bevel tool.
Use World Transform	Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene. **Note:** This option is only available for the procedural Edge Bevel tool.
Mode	Select either **Inset** or **Width**. **Inset** determines the distance away form the original un-beveled edge. **Width** determines the actual width of the new bevel (prior to the application of any rounding, if applicable).
Value	Controls the amount of the bevel as a distance value, depending on the **Mode** selected.
Use Material	When enabled, you can choose an existing material to apply to all newly created geometry. When disabled, the tool uses the neighboring material.
Material Name	When **Use Material** is enabled, you can select the material to be used from this dropdown.
Round Level	This value allows the tool to create additional edges between the original pair generated. Each new edge is placed in such a way as to create a rounding effect between the outermost

Edge Bevel	
	edges.
Sharp Corner	For edges shared by three or more polygons, Modo groups the polygons into pairs and creates new faces between each pair, ignoring polygons not part of a pair. At junctions between three or more edges, extra polygons are inserted to bridge any gaps created by the extrusion. Extra polygons are also inserted at the bare ends of edge to keep the shape closed and planar as the edge bevels.  Perspective Advanced OpenGL Ray GL: OFF 32 Edges Channels: 0 Deformers: OFF GL: 112 50 mm Action Center Auto : Edge Bevel [Snap : ON] Normal Edge Bevel Corners  Perspective Advanced OpenGL Ray GL: OFF Action Center Auto : Edge Bevel [Snap : ON] Edge Bevel **Sharp Corner** enabled
Edge Shape	Determines the shape of the resulting beveled edge with three options.  Perspective Advanced OpenGL Ray GL: Off 3 Edges Channels: 0 Deformers: OFF GL: 12 20 mm Edge Bevel Base Edge Selection  Perspective Advanced OpenGL Ray GL: Off 6 Edges Channels: 0 Deformers: OFF GL: 120 20 mm Edge Bevel Round Edge Selection  Perspective Advanced OpenGL Ray GL: Off 6 Edges Channels: 0 Deformers: OFF GL: 36 20 mm Edge Bevel Square Edge Shape  Perspective Advanced OpenGL Edge Bevel Sharpe Edge Shape
Mitering Offset	With **Edge Shape** set to **Round**, the **Mitering Offset** option creates extra co-planar polygons beside the outer edges to smooth artifacts on connecting polygons.

Edge Bevel	
	
Maintain Co-Planar Edges	Fixes co-planar edge errors when you are detailing faceted polygons from roughed out shapes. When this option is enabled, the unselected co-planar edges merge with the next unselected sharp edge, creating an even surface.
Depth	This value adjusts the roundness of beveled edges. 100% is the default value for round bevel. 0% makes round positions flat, and -100% produces a negative radius. The following examples show the various **Depth** values with a **Round Level** of **2**.  Depth 100%  Depth 0%  Depth -100%

Edge Bevel in the Procedural

Modo has a procedural Edge Bevel mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the stack in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Edge**, double-click **Edge Bevel**.

You can apply the bevel to a selection by making the selection before adding the bevel operation. This adds a **Select by Index** to the **Selection** input of the bevel. Once added, the selection can be edited using the **Properties** panel for the **Select by Index**. For more information, see Procedural Selection.

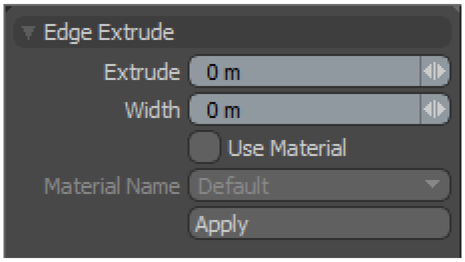
You can expand the Edge Bevel item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses. Edge Bevel requires the following inputs:

- **Selection** - Allows you to select the edges you want to bevel.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

> **Tip:** To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Edge Extrude

With the **Edge Extrude** tool, edges are shifted outward along the average normal of their two neighboring polygons. New faces are added between the edge and their neighbor polygons so that their neighbors can remain in their original plane. Neighbor polygons are shrunk by the same amount and in the same manner as for edge bevel.



Edge Extrude

Extrude 0 m

Width 0 m

☐ Use Material

Material Name Default

Edge Extrude	
Extrude	This distance value determines how far the edge is shifted away from its original position.
Width	This distance value determines how far the newly generated edges shift away from the original edge location.
Use Material	Turn this on to apply a specific material group to the polygons created with the edge extrusion.
Material Name	Specifies the material group for the **Use Material** command.

Edge Extrude in the Procedural

Modo has a procedural Edge Extrude mesh operation available in the **Mesh Ops** tab. In the **Model** layout, you can find the list on the right side of the interface, next to the **Item List**.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Edge**, double-click **Edge Extrude**.

You can expand the Edge Extrude item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses. Edge Extrude requires the following inputs:

- **Selection** - Allows you to select the edges you want to extrude.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

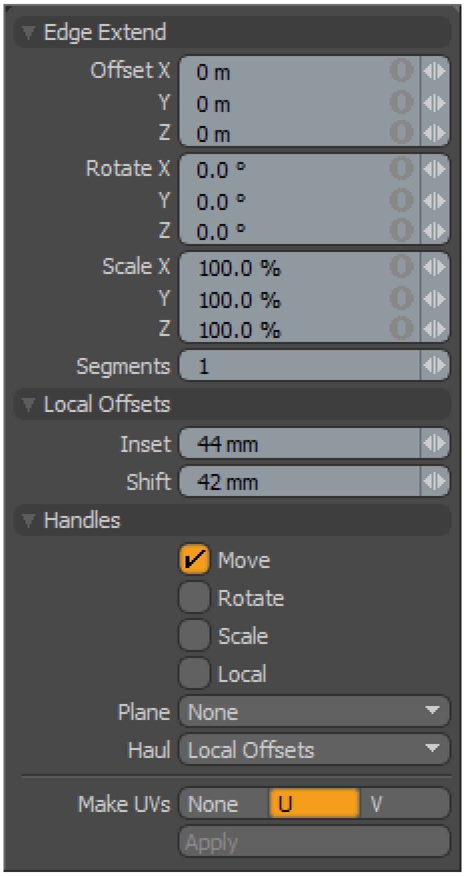
Edge Extend

The **Edge Extend** tool provides a simple method for creating geometry from selected edges by extending them outward with options for **Offsetting**, **Rotating**, and **Scaling** the newly created edges. The **Edge Extend** tool includes a customizable tool handle. By default there are **Offset** and **Scale** handles, but these can be modified in the tools properties panel to include **Rotate** handles, if required. Planar **Offset** and **Scale** handles can also be assigned. This allows you to adapt the tool easily for a particular modeling operation and also means it's possible to generate more complex edge editing without dropping the tool. The **Local Mode** options allow you to control the extension direction for the resulting geometry based on the normal direction of the source polygon.

Usage

Prior to using the **Edge Extend** tool, you may want to select the target series of edges for extending (done in the **Edges** component mode). The **Edge Extend** tool is found in the **Edge** sub-tab of the **Modeling** layout toolbox. Left-clicking the button in the toolbox activates it. It can also be activated by pressing the **Z** keyboard shortcut. Once activated you can assign values in the tools properties panel and then press the **Apply** button to generate the extended edges, but to use the tool in interactive mode you need to left-click in any 3D viewport, placing the tool in interactive mode. From there, you can left-click any of the available tool handles in the viewport to interactively adjust the extension values, or numerically adjust the values in the properties panel with real-time feedback in the 3D viewport.

Left-clicking and dragging off the tool handles (hauling) defines the amount of **Shift** for the edge extension, with the direction determined by the **Local Mode** setting. While the tool is active, you can also press **Shift** and left-click to define a new edge extension value. When the tool is selected, the following attributes are available in the **Tool Properties** panel.



Edge Extend
Offset X 0 m
Y 0 m
Z 0 m
Rotate X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Segments 1
Local Offsets
Inset 44 mm
Shift 42 mm
Handles
☑ Move
☐ Rotate
☐ Scale
☐ Local
Plane None
Haul Local Offsets
Make UVs None U V
Apply

Edge Extend	
Offset X/Y/Z	The **Offset** values determine extension distance of the newly generated edges in any of the three planar directions, or any combination thereof. These values are also adjusted interactively in the viewport while hauling (left-click and drag) when the **Haul** parameter is set to **Global Offsets**.
Rotate X/Y/Z	The **Rotate** values determine the amount of rotation of the newly generated edges in any of the three planar directions, or any combination thereof.

Edge Extend	
ScaleX/Y/Z	The **Scale** values determine the size of the newly generated edges for any of the three planar directions, or any combination thereof.
Segments	The **Segments** value determines the number of divisions added across the length of the extended edges.
Local Offsets	
Inset/Shift	The **Inset** and **Shift** values represent the relative local distance from the initial position of the edge (with the direction based on the adjoining polygons normal). Working much like the same **Bevel** options, **Inset** moves the extended edge parallel to the normal (left and right), while **Shift** moves the extension perpendicular to the normal (up and down/in and out). These values are also adjusted interactively in the viewport while hauling (left-click and drag) when the **Haul** parameter is set to **Local Offsets**.
Handles	
Move/Rotate/Scale/Local	The **Move**, **Rotate**, **Scale** and **Local** options toggle the visibility of the associated tool handles in the 3D viewport when the **Extend** tool is in interactive mode, allowing you to control which handles are preferred when working with edge extensions.
Plane	The **Plane** option allows you to further customize the tool handles by adding small planar control circles that affect two axes at once. You can choose from a **Scale** action for the handles or a **Move** action for them. **None** disables the extra handles.
Haul	**Hauling** is the action of left-clicking away from any tool handle in the 3D viewport to control an attribute. The **Haul** setting defines which tool attributes are affected when hauling. The **Global Offset** option affects the **Offset X/Y/X** controls of the edger extension, while **Local Offsets** controls the **Inset/Shift** values of the **Local Offsets**. Global offsets always move relative to the current Work Plane, local offsets are adjusted by horizontally moving the mouse cursor controlling the **Inset** amount, and vertically moving the cursor adjusting the **Shift** amount.
Make UVs	When the **Make UVs** option is on the **U** or **V** selection, the polygons along the extended edge have automatic UV values added along the defined axis, either the horizontal **U** direction or the vertical **V** direction.

Slide / Edge Slide

The direct modeling **Slide** tool moves selected mesh elements (vertices, edges, and polygons) along their connected edges. The Slide tool’s handles will adapt to the type of component you have selected.

The procedural **Edge Slide** tool allows you to select the **Radial** mode to slide an entire edge loop to widen or narrow your selection, or use the **Linear** mode to specify the distance to move the selected geometry. For more information, see Applying the Procedural Slide Tool.

You can also use the Slide tool and the mesh operation on a UV map by selecting the appropriate UV map in the tool's properties.

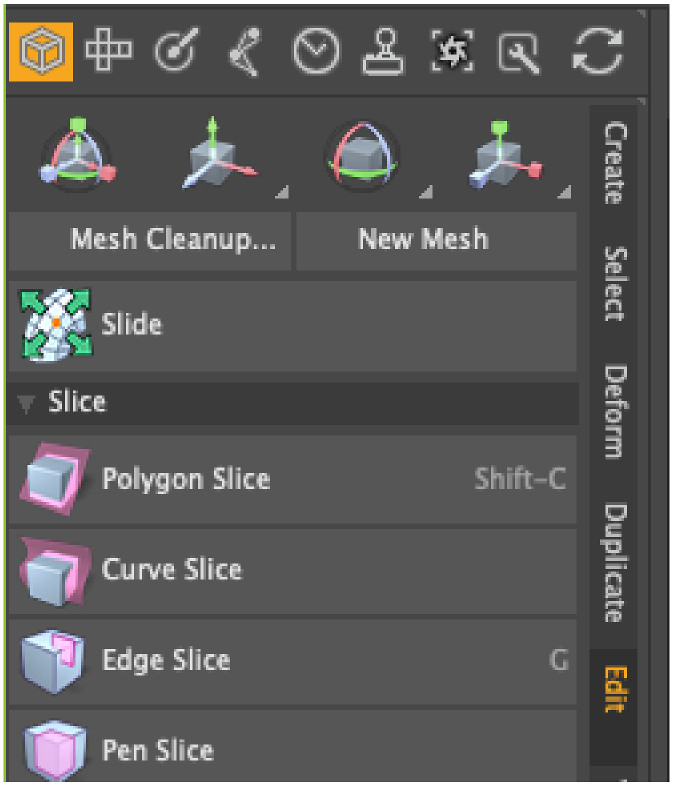
Tip: For more information on direct and procedural modeling, see Modeling Techniques.

Activating the Tool

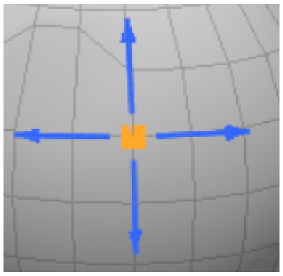
Note: Before applying this tool, you must activate a component selection mode. For more information, see Understanding Items vs. Components.

Direct Modeling Slide Tool:

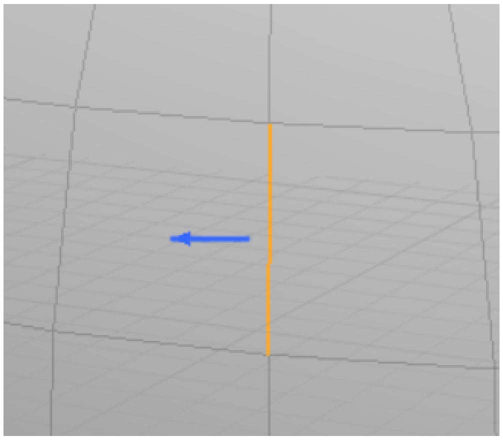
- In the **Model** layout, activate the **Edge** or **Vertices** component selection mode. On the left panel, open the **Edit** section and select **Slide**.


Mesh Cleanup...
New Mesh
Slide
Slice
Polygon Slice
Shift-C
Curve Slice
Edge Slice
G
Pen Slice
Create
Select
Deform
Duplicate
Edit

If you are in **Vertices** component mode, the Slide tool will display **Linear** and **Face** mode handles by default.



If you are in **Edge** component mode, then the Slide tool will display **Radial** mode handles by default.



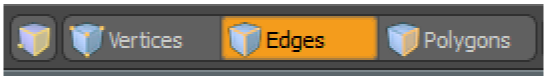
To access the tool while using any component selection mode, you can press **S**.

Procedural Modeling Edge Slide Tool:

- In the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Edge** > **Edge Slide**.
- In the **Schematic** view, click **Add...**, and double-click **Mesh Operations** > **Edge** > **Edge Slide**.
- In the **Topology** layout, on the left panel, open the **Tools** tab, and click **Edge Slide**.

Sliding an Edge

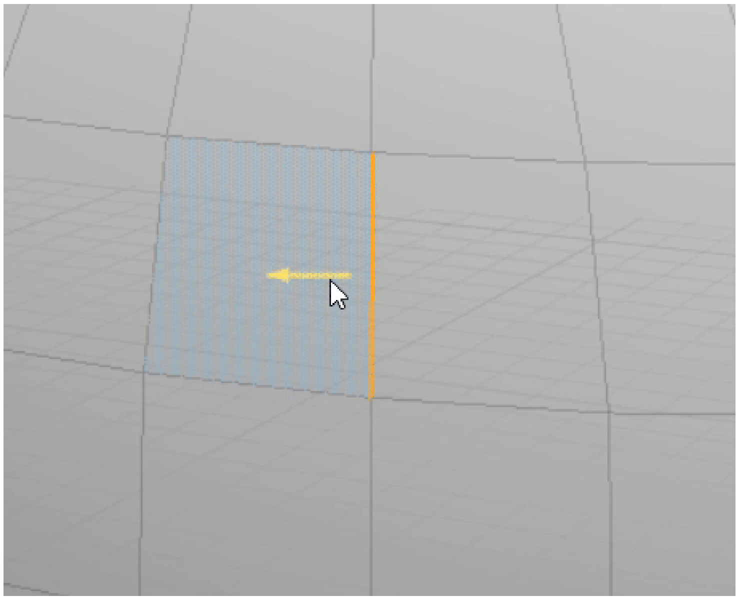
1. Click **Edges** and select an edge.



Tip: Hold **Shift** and click to add more edges to the selection. You can select multiple edges by clicking and dragging or right-click lassoing.

2. Press **S** to activate the Slide tool.

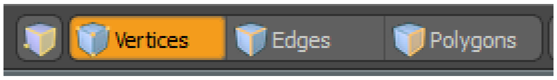
   Alternatively, on the left panel, open the **Edge** tab and click **Slide**.
3. Click and drag on one of the handles to slide the edge(s) into a new position.



4. Press **Esc** to drop the tool.

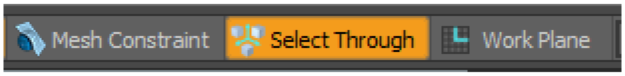
Sliding a Vertex

1. Click **Vertices** and select a vertex on your mesh item.

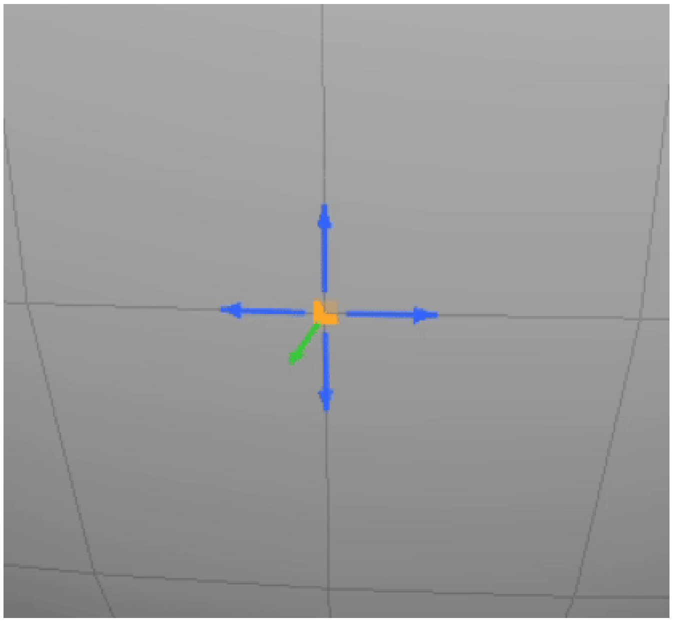


2. Press **S** to activate the Slide tool.

3. Ensure that the **Select Through** button is enabled.



4. Click and drag on one of the handles.



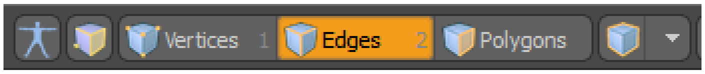
5. Press the **space** key to drop the tool.
6. Hold **Shift** and select a number of additional vertices.
7. Press **S** and click and drag on a handle.

Applying the Procedural Slide Tool

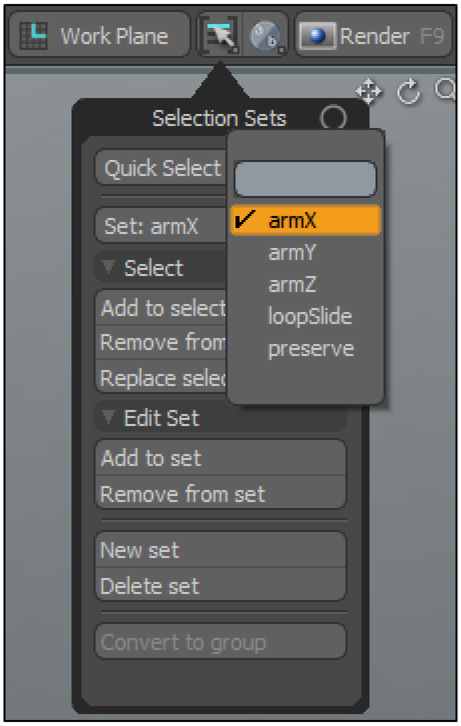
The procedural **Edge Slide** tool allows you to select the **Radial** mode to slide an entire edge loop to widen or narrow your selection, or use the **Linear** mode to specify the distance to move the selected geometry. Specifying other properties allows you to position your selection exactly where you want it.

Example

1. Download our example scenes and open **Edge Slide meshop.lxo** in the **Model** layout.
2. On the right panel, ensure that the **model** item is selected.
3. Under the layout menu, select **Edges**.

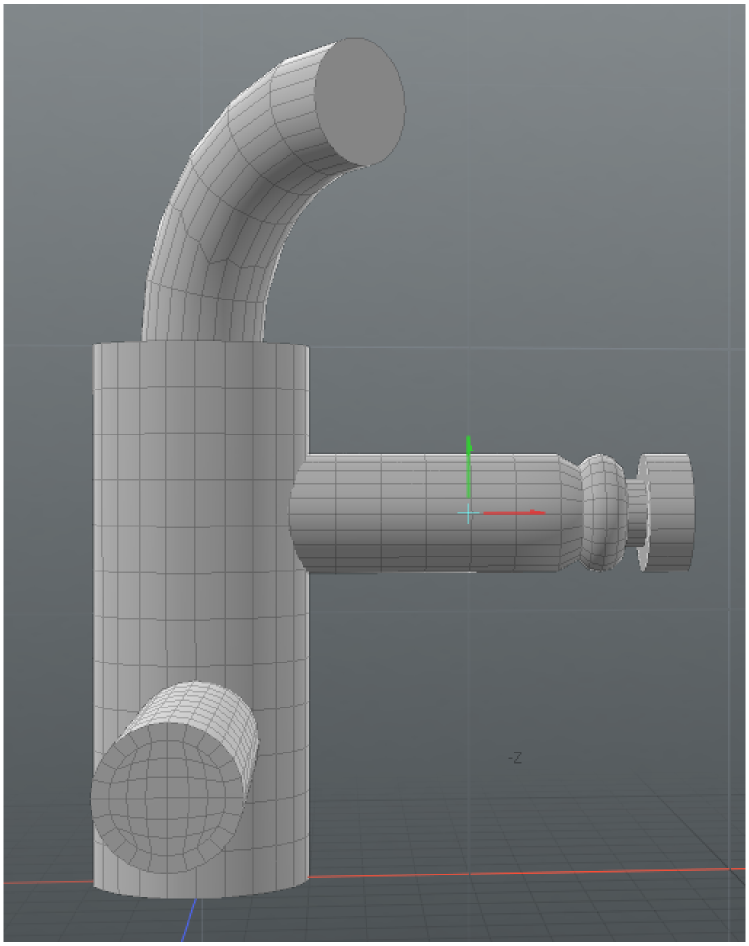

Vertices 1
Edges 2
Polygons

4. On the top right of the interface, click the **Selection Sets** icon, and then click **Quick Select** > **armX**.

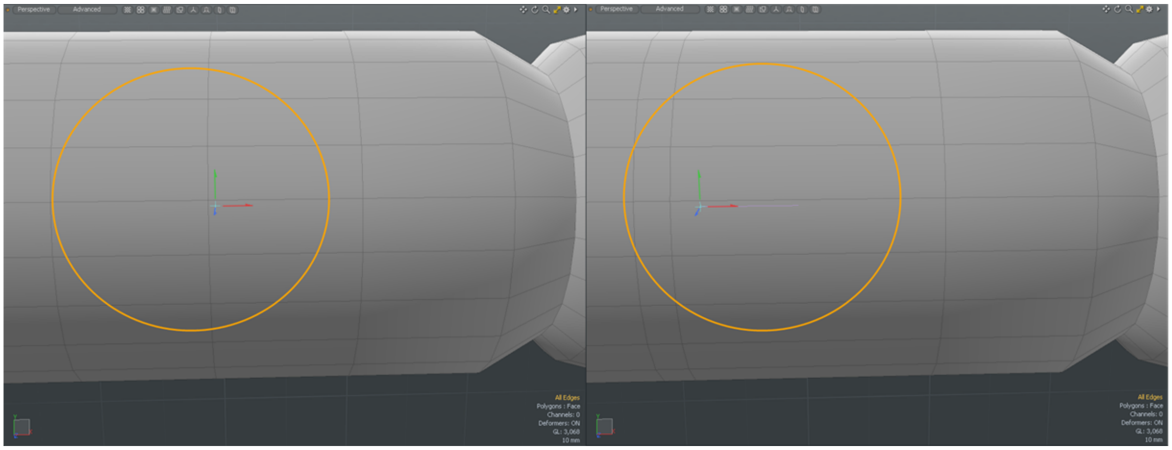

Work Plane
Render F9
Selection Sets
Quick Select
Set: armX
Select
Add to select
Remove from
Replace selec
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group
✓ armX
armY
armZ
loopSlide
preserve

5. On the right panel, open the **Mesh Ops** tab, and click **Add Operator**.
6. Double-click **Mesh Operations** > **Edge** > **Edge Slide**.
7. On the right panel, open the **Edge Slide** tab, and set the **Mode** to **Linear**.

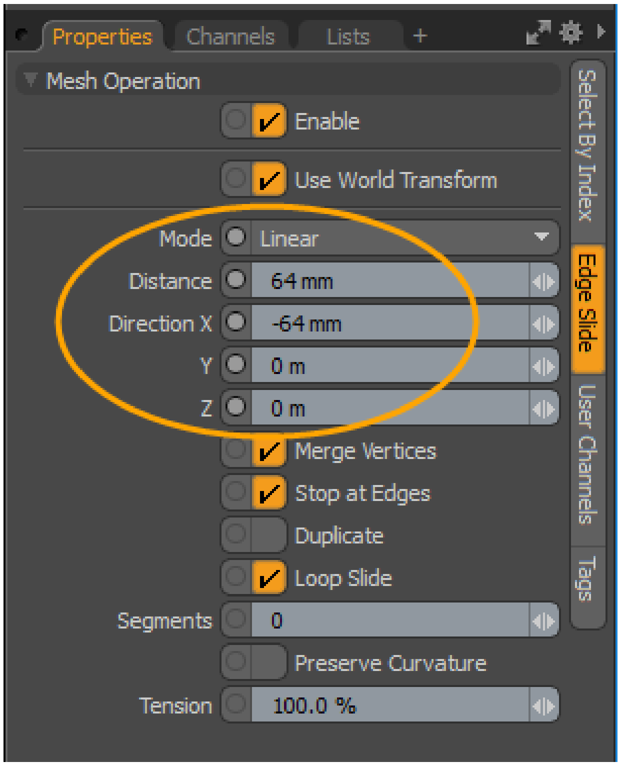
   Tool handles appear in the 3D viewport.



8. Drag the red (X) handle to slide your selection.


Perspective
Advanced
All Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,068
10 mm

The **Distance** and **Direction** of the selected X, Y, or Z handles displays the absolute value in the **Properties** tab.


Properties
Channels
Lists
+
Select By Index
Edge Slide
User Channels
Tags

Mesh Operation

☑ Enable

☑ Use World Transform

Mode: Linear

Distance: 64 mm

Direction X: -64 mm

Y: 0 m

Z: 0 m

☑ Merge Vertices

☑ Stop at Edges

☐ Duplicate

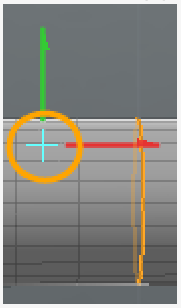
☑ Loop Slide

Segments: 0

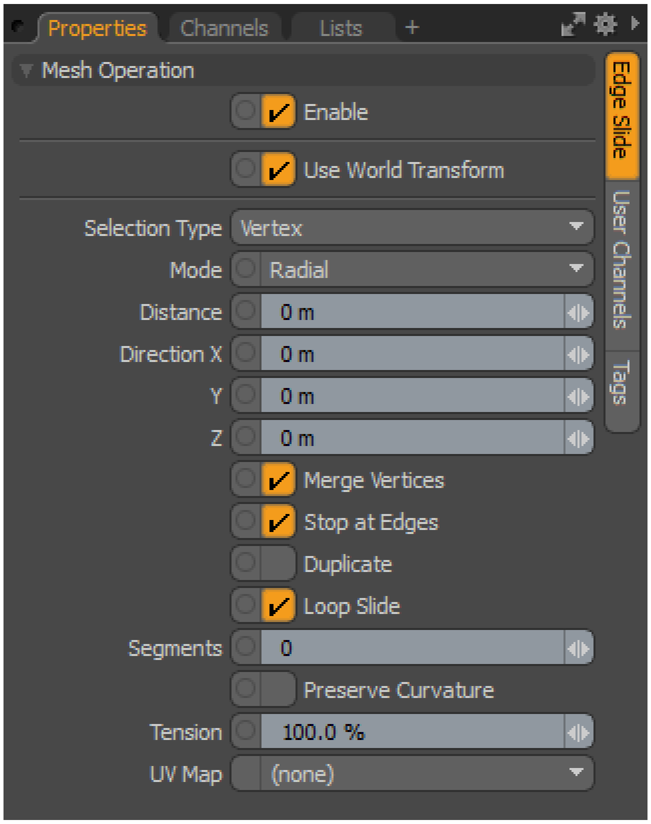
☐ Preserve Curvature

Tension: 100.0 %

Note: Dragging the cyan center tool handle updates the **Distance** and **Direction** for **X** and **Z** only.



Procedural Edge Slide Properties



Mesh Operation

☑ Enable

☑ Use World Transform

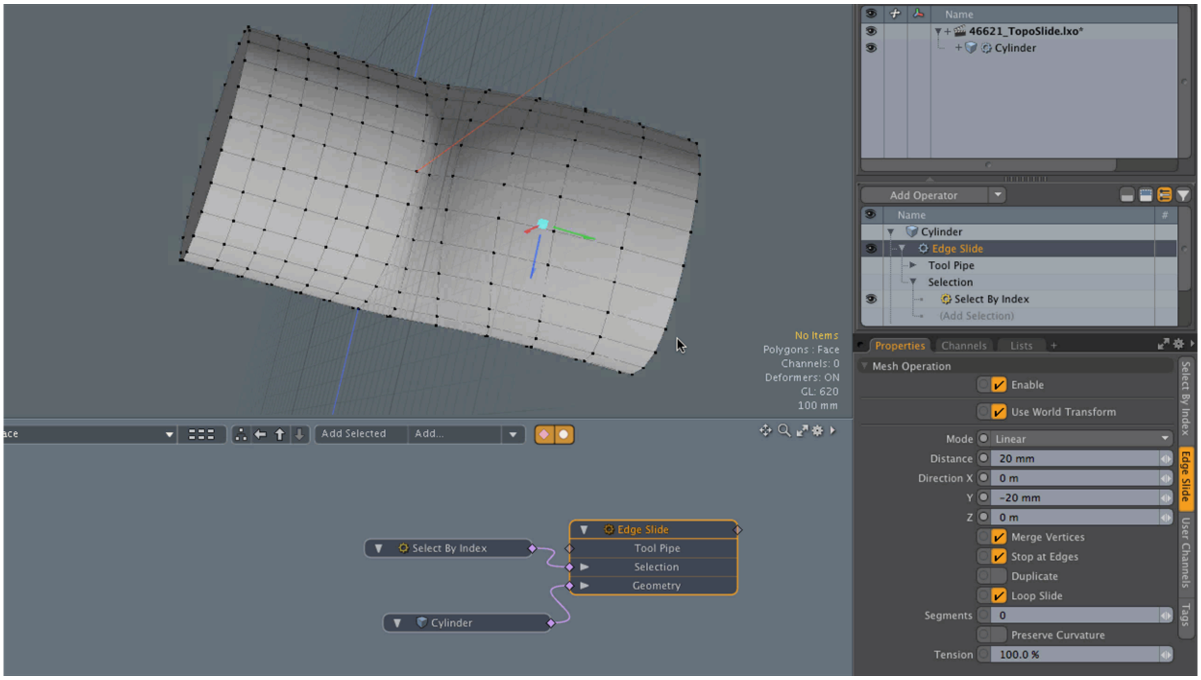
Selection Type: Vertex
Mode: Radial
Distance: 0 m
Direction X: 0 m
Y: 0 m
Z: 0 m
☑ Merge Vertices
☑ Stop at Edges
☐ Duplicate
☑ Loop Slide
Segments: 0
☐ Preserve Curvature
Tension: 100.0 %
UV Map: (none)

Mode	• Allows you to specify one of the following modes: • **Radial Mode** - This is most useful for sliding entire edge loops such as widening or narrowing a selected edge loop around an eye. When this mode is active, the **DirectionX**, **Y**, and **Z** options are disabled as the only relevant input is **Distance**, which indicates how far to slide the selected loop between the next and previous loops. • **Linear Mode** - Provides a method to slide elements along intersecting edges. When this mode is enabled, the **Direction X**, **Y**, and **Z** options are activated. When interacting with the tool in a 3D viewport the direction of the handle determines the edge to slide along. • **Face** - This mode is not available in Procedural modeling.
Distance	Allows the tool's numeric entry to be a specific distance based on the current unit system, so that geometry can be moved a specific number of feet, inches, meters, and so on.

Merge Vertices	When this option is active, vertices are merged if the slide tool results in vertices sharing the exact same coordinates. For example, sliding an edge 100% with this option enabled results in that edge being merged into its neighbor. This can be particularly convenient when combined with the **Stop at Edges** option.
Stop at Edges	With this option active, the selected tool stops the slide when the selected mesh elements reach the surrounding edges.
Duplicate	When active, the **Duplicate** option creates a new edge rather than simply sliding the selected edge. With this option enabled, it is simple to create new edges or edge loops, and place them along the existing curve.
Loop Slide	When sliding an edge selection, Loop Slide constrains the movement along the existing polygonal bounds.
Segments	Adds extra edge loops between the original edge loop and duplicated edge loop. This option works with **Duplicate** option.
Preserve Curvature	When enabled, the new position is computed along a curve through the edges to maintain the overall curvature or surface continuity. Once enabled, the **Tension** setting becomes available, controlling the tightness of the curvature preservation.
Tension	Determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately (a value of 0% disables the **Preserve Curvature** option). Negative values inset the curvature and higher values further offset the vertex out from the curve's position.
UV Map	Select a UV map to use Slide on a UV map. When set to **(none)**, the tool works on 3D geometry.

Applying the Slide Tool using the Schematic Viewport

Using the procedural **Edge Slide** tool in the **Schematic** viewport allows you to push the selected mesh elements along their connected edges.


Name
46621_TopoSlide.lxo*
Cylinder
Add Operator
Name
Cylinder
Edge Slide
Tool Pipe
Selection
Select By Index
(Add Selection)
No Items
Polygons : Face
Channels: 0
Deformers: ON
GL: 620
100 mm
Add Selected
Add...
Properties
Channels
Lists
Mesh Operation
Select By Index
Edge Slide
User Channels
Tags
Edge Slide
Tool Pipe
Selection
Geometry
Select By Index
Cylinder

☑ Enable

☑ Use World Transform

Mode: Linear

Distance: 20 mm

Direction X: 0 m

Y: -20 mm

Z: 0 m

☑ Merge Vertices

☑ Stop at Edges

☐ Duplicate

☑ Loop Slide

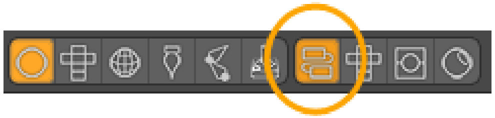
Segments: 0

☐ Preserve Curvature

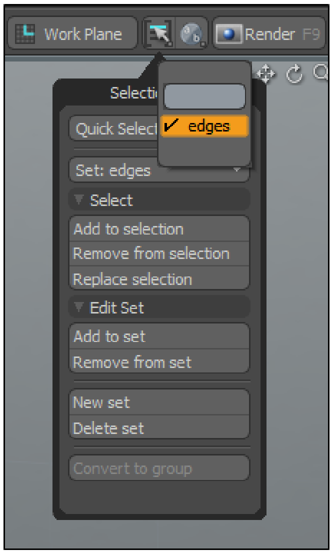
Tension: 100.0 %

Example

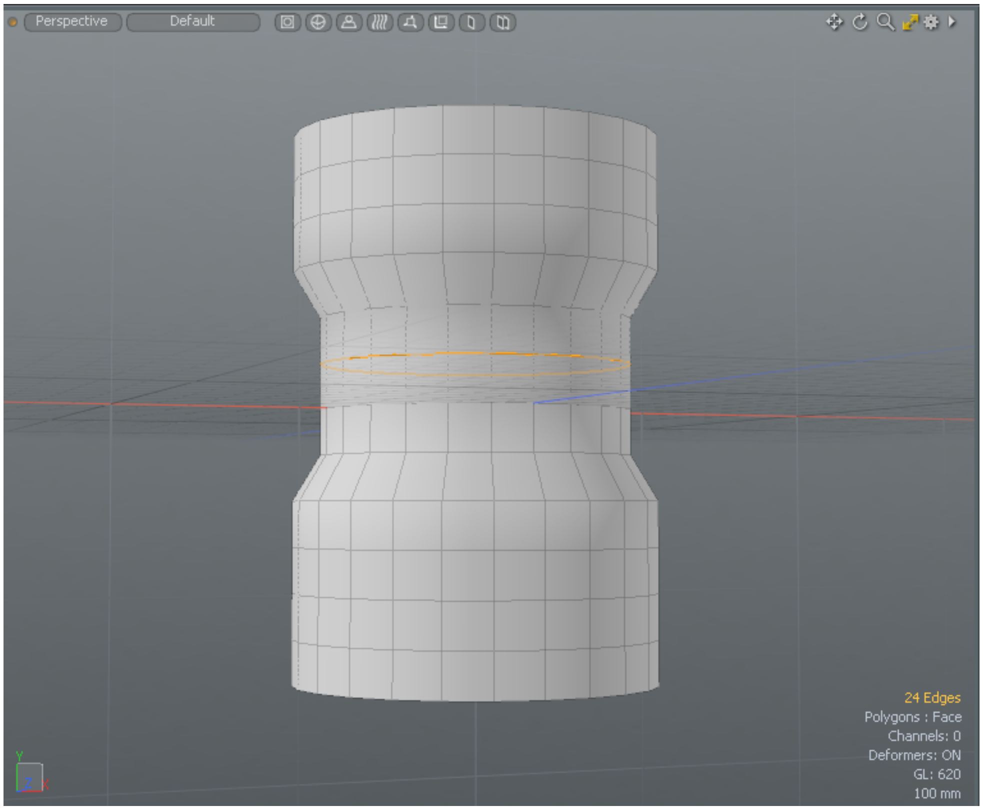
1. Download our example scenes and open **ModelPush.lxo** in the **Model** layout.
2. On the top-left corner of the interface, click the schematic palette icon.



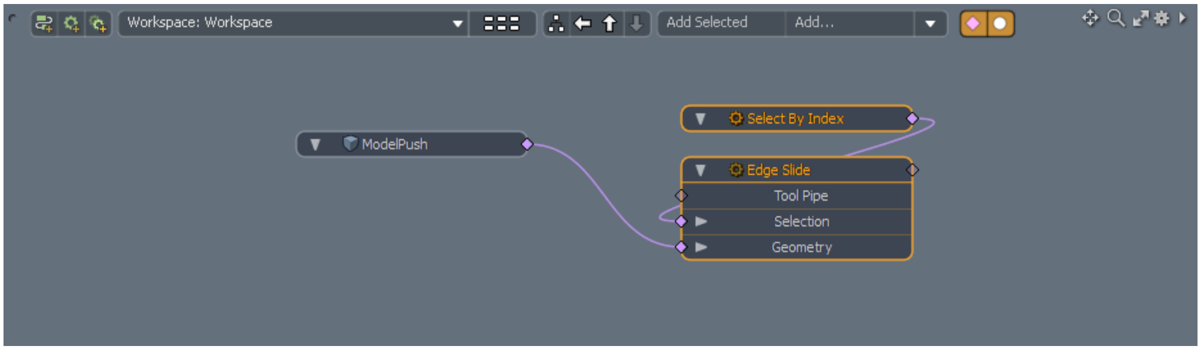
3. On the right panel, drag-and-drop **ModelPush** onto the **Schematic** viewport.
4. Under the layout menu bar, click **Edges**.
5. On the top-right of the interface, click the **Selection Sets** icon, and then click **Quick Select** > **edges**.


Work Plane
Render F9
Selecti
Quick Select
✓ edges
Set: edges
Select
Add to selection
Remove from selection
Replace selection
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group

An edge selection is highlighted on the model in the 3D viewport.

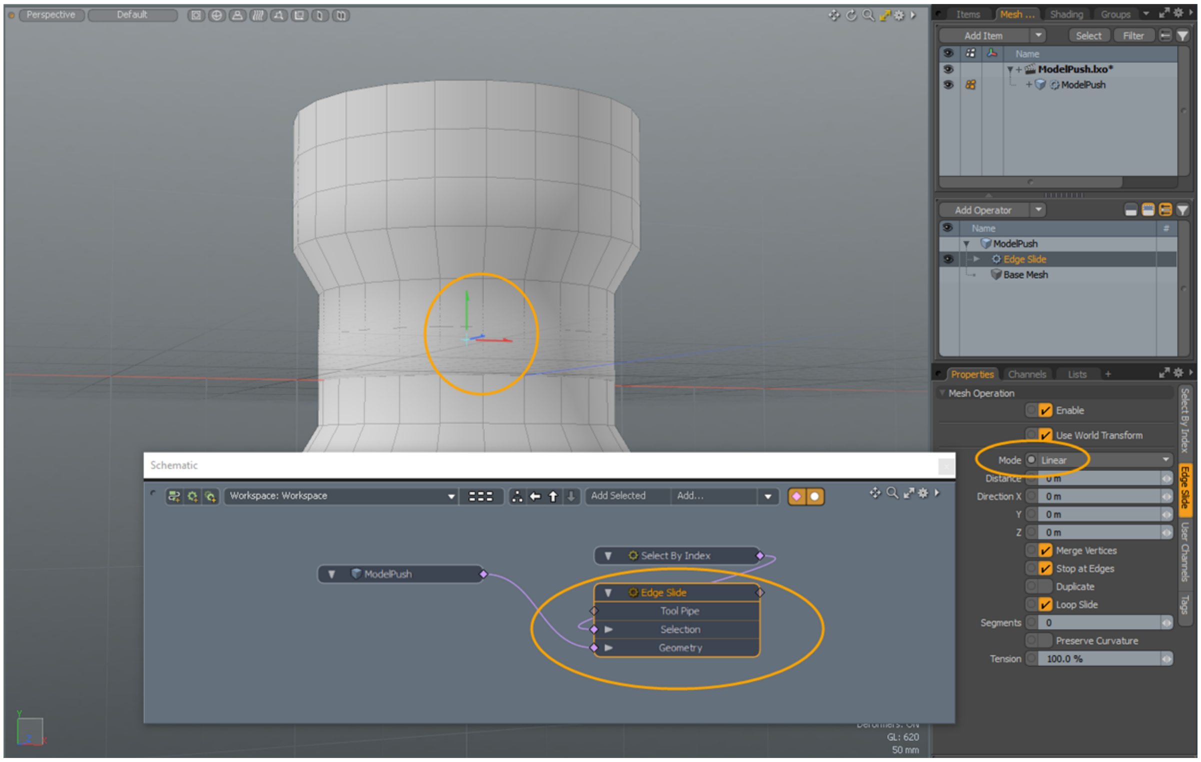

Perspective
Default
24 Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 620
100 mm

6. In the **Schematic** viewport, click **Add...**, and double-click **Mesh Operations** > **Edge** > **Edge Slide**.


Workspace: Workspace
Add Selected
Add...
Select By Index
ModelPush
Edge Slide
Tool Pipe
Selection
Geometry

7. In the **Schematic** viewport, click in the view area, and then select **Edge Slide**.
8. On the right panel, open the **Edge Slide** tab, and set **Mode** to **Linear**.

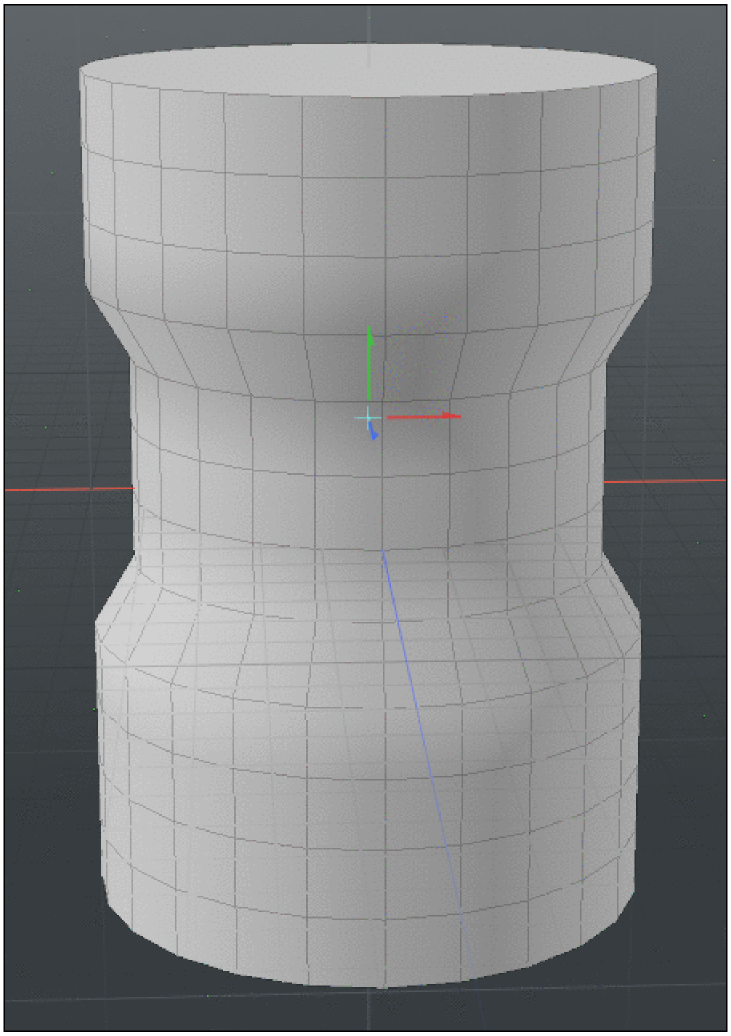
   The tool handles display in the 3D viewport.


Perspective
Default
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
ModelPush.lxo*
ModelPush
Add Operator
Name
ModelPush
Edge Slide
Base Mesh
Properties
Channels
Lists
Mesh Operation
Select By Index
Edge Slide
User Channels
Tags
Schematic
Workspace: Workspace
Add Selected
Add...
Select By Index
ModelPush
Edge Slide
Tool Pipe
Selection
Geometry
GL: 620
50 mm

☑ Enable
☑ Use World Transform
Mode: Linear
Distance: 0 m
Direction X: 0 m
Y: 0 m
Z: 0 m
☑ Merge Vertices
☑ Stop at Edges
☐ Duplicate
☑ Loop Slide
Segments: 0
☐ Preserve Curvature
Tension: 100.0 %

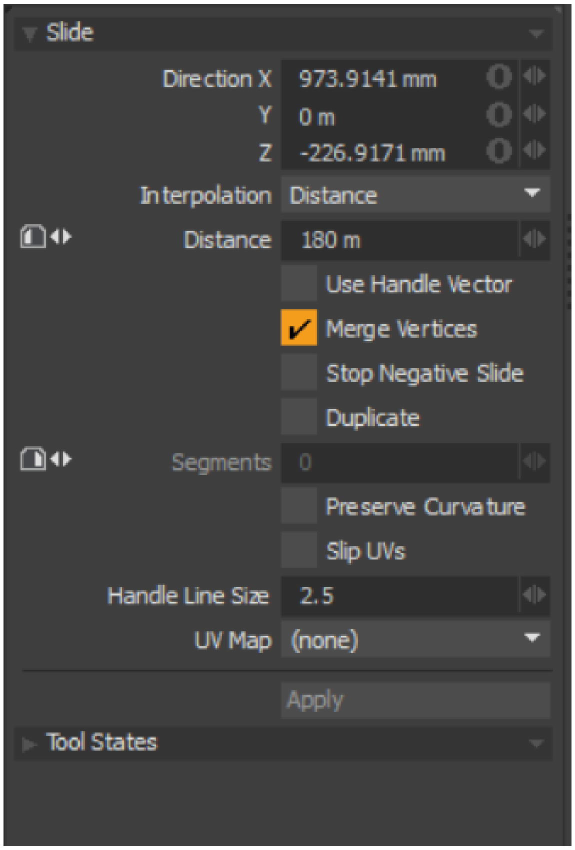
9. In the 3D viewport, click-and-drag the green handle upwards.

   The selection set moves upwards.



Direct Modeling - Slide Tool Properties

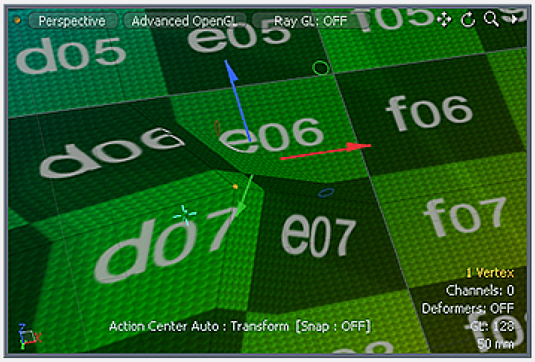
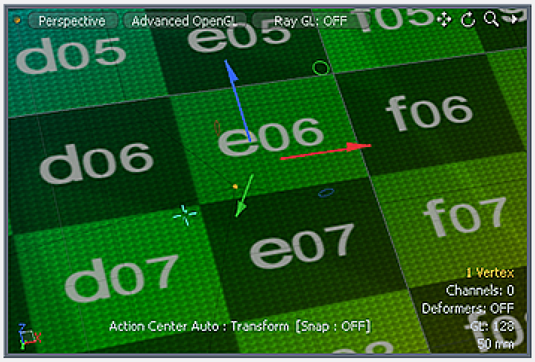
The Direct Modeling **Slide** tool is found in the toolbox located on the left panel of the interface.



Direction X 973.9141 mm
Y 0 m
Z -226.9171 mm
Interpolation Distance
Distance 180 m
☐ Use Handle Vector
☑ Merge Vertices
☐ Stop Negative Slide
☐ Duplicate
Segments 0
☐ Preserve Curvature
☐ Slip UVs
Handle Line Size 2.5
UV Map (none)

Direction X/Y/Z	Sets the directional position for **X**, **Y**, and **Z**.
Interpolation	Allows you to choose between **Distance** and **Percent** methods for sliding the geometry. • When **Distance** is enabled, the **Distance** option is enabled. • When **Percent** is enabled, the **Percentage** option is enabled.
Distance	When **Interpolation** is set to **Distance**, set the numeric value and unit system used to move the geometry.
Percentage	When **Interpolation** is set to **Percent**, set the percentage value to slide the selected geometry. You can set the percentage using its initial position (0%) and the neighboring edge or edge loop position (-100% to 100%).
Merge Vertices	When enabled, vertices are merged when vertices share the same coordinates. For example, sliding an edge 100% results in that edge being merged into its neighbor.

	Tip: This can be particularly useful when combined with the Stop at Edges option to avoid over deforming geometry.
Stop at Edges	When enabled, the Slide tool stops the slide when the selected mesh elements reach the surrounding edges.
Duplicate	When enabled, creates a new element, rather than sliding to the selected element. **Tip:** Use this option to create new edges or edge loops and place them along the existing curve.
Loop Slide	**Note:** Loop Slide is only available for edge selections. When enabled and sliding on an edge selection, constrains the movement along the existing polygonal bounds.
Segments	Specifies the number of segments for profiles. If the value is **0** and a profile is selected, the **Slide** tool automatically decides the best adaptive segments for the profile. **Note:** **Duplicate** must be enabled for **Profiles** to work. To specify a profile, see Profiles Options.
Preserve Curvature	When enabled, the new position is computed along a curve through the edges to maintain the overall curvature or surface continuity. Once enabled, the Tension option becomes available, controlling the tightness of the curvature preservation.
Tension	Determines the strength of the curvature preservation. The default value of **100%** calculates the curvature, placing the new vertex appropriately. **Note:** Setting the value to 0% is the same as disabling the **Preserve Curvature** option.

	Negative values inset the curvature and higher values further offset the vertex out from the curve's position.
Slip UVs	When enabled, existing UV mappings applied to the geometry are not disturbed. UV values are generally fixed to specific vertices, so further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or redo it altogether. To avoid this, enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. For more information about UV Mapping, see Working with UV Maps.  Slip UVs function disabled, note texture warping.  Slip UVs function enabled, texture remains even.
Handle Line Size	Specifies the line width of vertex handle.
UV Map	Select a UV map to use Slide on a UV map. When set to **(none)**, the tool works on 3D geometry.
Apply	Applies property settings.

Profile Attributes Options

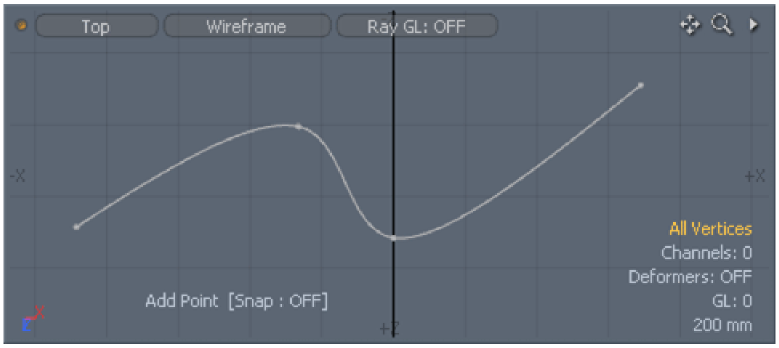
Inset	This specifies the offset amount of **X** direction of the profile.
Reverse Direction	If this option is enabled, the 1D profile is evaluated from top to bottom.
Reverse Inset	If this option is enabled, Inset is reversed.
Keep Aspect	Automatically sets Inset value based on the aspect ratio of the profile.

Profiles Options

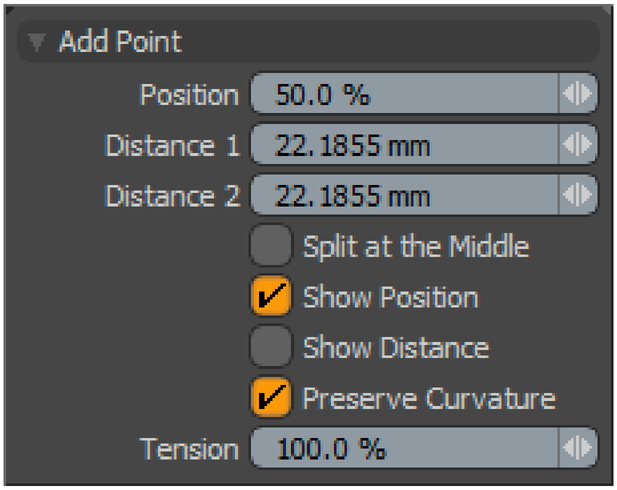
Mini Preset browser for viewing various profiles. Works the same as the standard Preset Browser.

Add Point

The **Add Point** tool provides a means to easily add a new vertex point along an existing polygon edge, splitting the edge in two, or along a curve wherever you click.



To use the **Add Point** tool, locate the tool button in the **Edges** sub-tab of the default modeling toolbox and click it. This activates the tool. Next, in the 3D viewport, locate the target edge for the operation. The mouse pointer must be close enough to the edge to trigger the pre-highlighting (turning the edge a light blue color). Clicking on the edge adds the new vertex. Subsequent clicks repositions the vertex, while clicking and dragging slides the vertex along the target edge. Clicking away from the model removes the current active vertex point. Pressing and holding the **Shift** key adds a new vertex with each click. When the tool is activated the **Add Point** options are available in the tool properties panel.



Add Point

Position 50.0 %

Distance 1 22.1855 mm

Distance 2 22.1855 mm

☐ Split at the Middle

☑ Show Position

☐ Show Distance

☑ Preserve Curvature

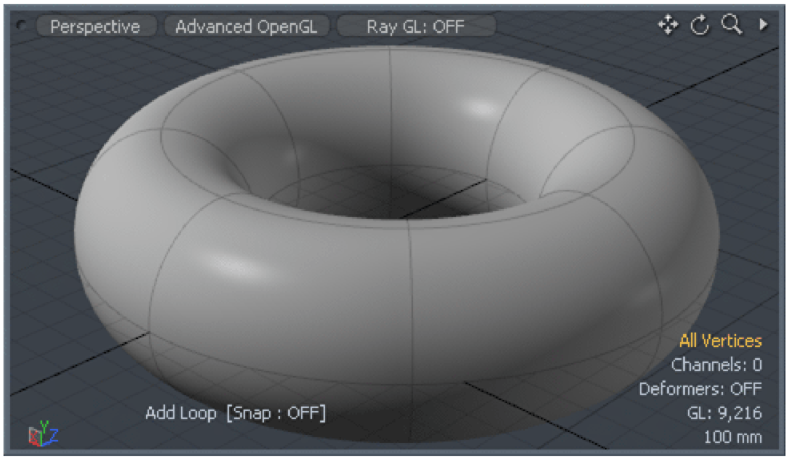
Tension 100.0 %

The following **Add Point** options are available for the **Add Point** tool:

- **Position** - determines the position of the new point across the entire edge length, displayed as a percentage, with 50% being the center. This interactively changes as you adjust the position in the 3D viewport. A new value can also be typed here manually, adjusting the currently-active vertex.
- **Distance 1/2** - the new vertex position displayed as a numeric distance value from the start (1) to the end (2) of the edge.
- **Split at the Middle** - when enabled, Modo automatically slices at the middle position of the target edge, regardless of where the edge was clicked. This option can also be temporarily enabled by holding down the **Ctrl** key prior to clicking.
- **Show Position** - when enabled, displays a text overlay of the new position as a percentage.
- **Show Distance** - when enabled, displays the distances between the edge positions and the new position as a numeric value.
- **Preserve Curvature** - when enabled, the **Preserve Curvature** option attempts to maintain the flow of surrounding geometry. The new position is computed along a curve through the surrounding edges to maintain the overall curvature of the surface. Once enabled, the **Tension** setting becomes available.
- **Tension** - determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately, where a value of 0% would be like disabling the **Preserve Curvature** option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.

Add Loop

The Add Loop tool is the faster, interactive version of the Loop Slice tool. With it, you can easily add a continuous edge through a series of polygons (typically quad polygons). Loops are generally a continuous band of polygons encircling a surface or terminating at an edge, but the Add Loop tool only needs two or more polygons that share and edge to add the loop.



To use the tool, follow these steps.

1. In the **Model** layout, open the **Edges** sub-tab in the Toolbox, and click **Add Loop**.

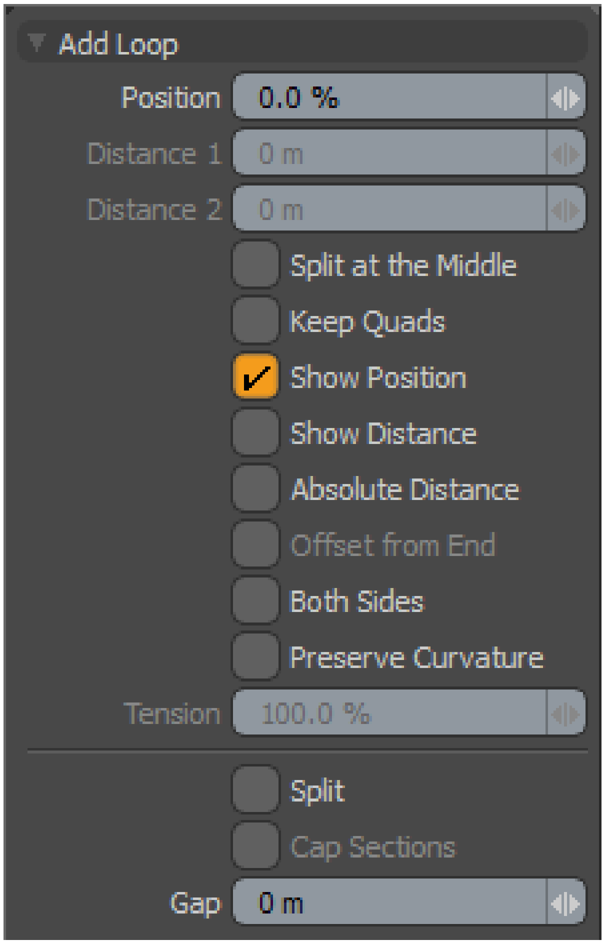
   Alternatively, this tool is available in the **Topology** layout, under the **Tools** sub-tab as well in the menu**Geometry** > **Edges** > **Add Loop**.

   This activates the tool.

2. Next, in the 3D viewport, locate the target edge for the operation. The mouse pointer must be close enough to the edge to trigger the pre-highlighting, turning the edge a light blue color. Click it to add a new edge perpendicular to the target, that cuts though the polygon band at the intersection of the mouse pointer and the target edge.

   The tool is now in interactive mode, subsequent clicks reposition the edge, clicking and dragging slide the edge loop. Click away from the active edge to remove it. Press and hold the **Shift** key and click, or click on a new highlighted edge to trigger a new edge loop. When the newly generated edge encounters a triangle polygon, the edge terminates. When a polygon is encountered with more than four sides, the tool attempts to cut through the center of the polygon, but may produce undesirable results, especially in polygons with odd numbers of edge.

When the tool is activated, the following options are available in the tool's **Properties** panel.



Add Loop

Position 0.0 %

Distance 1 0 m

Distance 2 0 m

☐ Split at the Middle

☐ Keep Quads

☑ Show Position

☐ Show Distance

☐ Absolute Distance

☐ Offset from End

☐ Both Sides

☐ Preserve Curvature

Tension 100.0 %

☐ Split

☐ Cap Sections

Gap 0 m

Add Loop	
Position	This value determines the slice position across the entire edge length displayed as a percentage, with 50% being the center. This interactively changes as you adjust the edge in the 3D viewport. You can also type in a new value manually to adjust the active edge in the viewport. **Note:** An edge must be active in order to manually enter its position.
Distance 1/2	The slice position displayed as a numeric distance value from the start (1) to the end (2) of the edge.
Split at the Middle	When enabled, Modo automatically slices at the middle (50%) position regardless of where the edge was clicked. You can enable this option temporarily by holding **Ctrl** before clicking.
Keep Quads	Forces the Add Loop tool to produce quad polygons if possible, such as when meeting a polygon with more than 4 edges. This is achieved by adding additional connecting edges.
Show Position	When enabled, displays a text overlay of the new position as a percentage.
Show Distance	When enabled, displays the distances between the edge positions and the new position as a numeric value.
Absolute Distance	When enabled, the offset of the new loop is defined by an absolute distance instead of as a percentage, maintaining a fixed distance from the closest edge when created.
Offset from End	When enabled, allows the offset of the loop from the opposite end of edge. This option is only available when **Absolute Distance** is enabled.
Both Sides	When enabled, simultaneously slices loop from both sides of the existing loop providing two edge loops.
Preserve Curvature	When enabled, the **Preserve Curvature** option attempts to maintain the flow of surrounding geometry. The new position is computed along a curve through the surrounding edges to maintain the overall curvature of the surface. Once enabled, the **Tension** setting becomes available.
Tension	Determines the strength of the curvature preservation. The default value of 100%

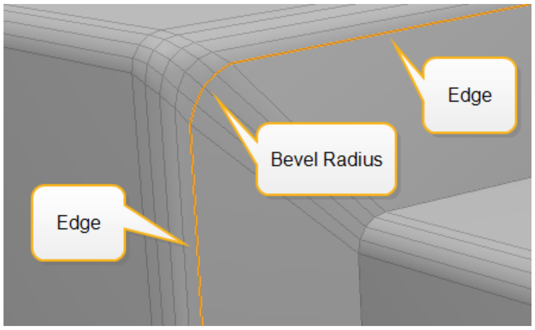
Add Loop	
	calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately, where a value of 0% is like disabling the **Preserve Curvature** option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.
Split	When enabled, the resulting loop splits the surface, producing two distinct overlapping edges, rather than a single connected edge.
Cap Sections	When using the **Split** option, the **Cap Sections** option can be added to create a single polygon cap connected to the two resulting edge loops, in effect closing off the two new edges.
Gap	When splitting surfaces, this option defines a width around the split line, in effect thickening the width of the resulting cut by the defined value.

Edge Relax

With the **Edge Relax** tool you can change the radius of beveled edges after the fact. To learn more about beveling edges, see Edge Bevel.

To use Edge Relax:

1. Select the beveled edges. The selection should include the radius of the bevel, and an edge on either side.

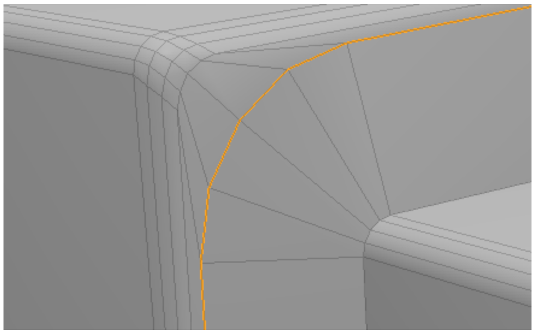


2. In the Model toolbar's **Edge** sub-tab, click **Edge Relax**.

   This activates the Edge Relax tool.

3. Click and drag in the viewport to adjust the **Offset** value, or edit it in the **Tool Properties** panel on the lower left.

> **Note:** If you're working in the **Modo** layout, click the **Tool Properties** button, or press the **K** key to reveal the panel.



You can also propagate the offset to any parallel edges. To do this, increase the **Propagate** value in the **Tool Properties**. For more information, see Edge Relax Properties.

Procedural Edge Relax

Edge Relax is also available as a mesh operation.

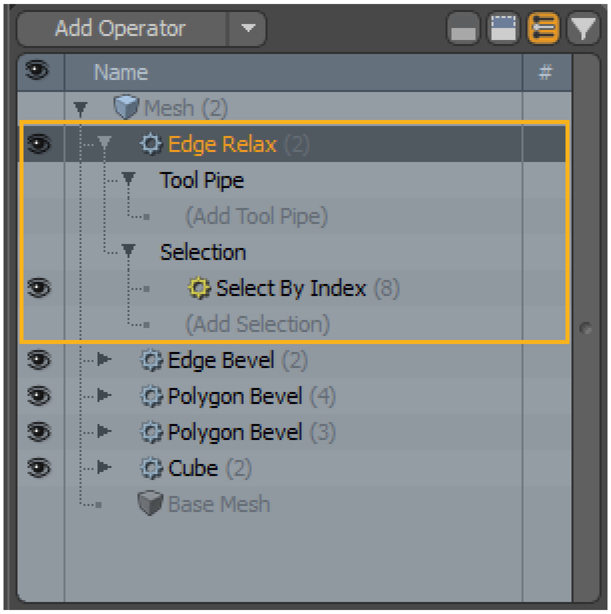
> **Tip:** For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

1. In **Edges** selection mode, select the edges to relax on your mesh in the 3D viewport.
2. On the right panel, click the **Mesh Ops** tab.

> **Note:** If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

3. Click the **Add Operator** button, and under **Mesh Operations** > **Edge**, double-click **Edge Relax**.

   This adds Edge Relax to the Mesh Operations list and opens its properties on the lower right pane.


Add Operator
Name
#
Mesh (2)
Edge Relax (2)
Tool Pipe
(Add Tool Pipe)
Selection
Select By Index (8)
(Add Selection)
Edge Bevel (2)
Polygon Bevel (4)
Polygon Bevel (3)
Cube (2)
Base Mesh

You can expand the **Edge Relax** operation in the list by clicking the arrow in front of it. This reveals the inputs the operation uses:

- **Tool Pipe** - Add falloffs and sub-tools to the operation.
- **Selection** - Select edges to relax or modify your existing selection. For more information on procedural selection, see Procedural Selection.

4. In the Edge Relax **Properties**, adjust the **Offset** value and other properties as needed. For more information on what each setting does, see Edge Relax Properties.

   The bevel radius is updated in the 3D viewport.

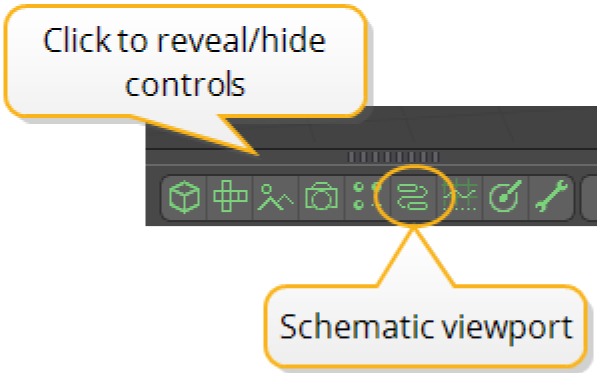
Edge Relax in the Schematic

You can also use the Edge Relax tool when working in the Schematic viewport.

Note: For more information on working with Schematic viewport in general, see Schematic Viewport.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.

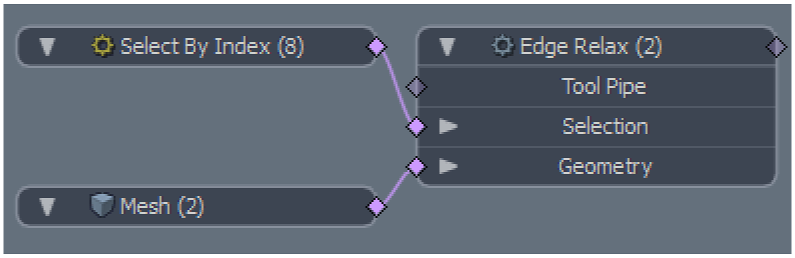

Click to reveal/hide controls
Schematic viewport

>

MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

To add the Edge Relax node, click **Add...**, and under **Mesh Operations** > **Edge**, double-click **Edge Relax**.


Select By Index (8)
Edge Relax (2)
Tool Pipe
Selection
Geometry
Mesh (2)

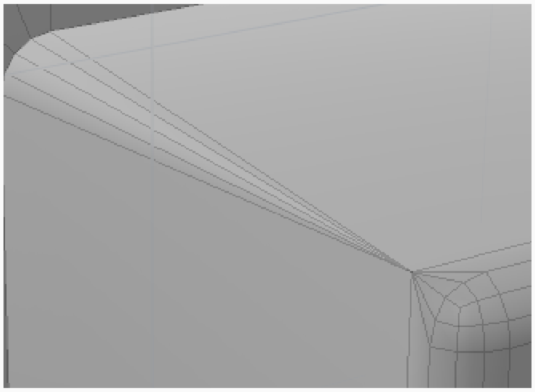
The node has the following inputs:

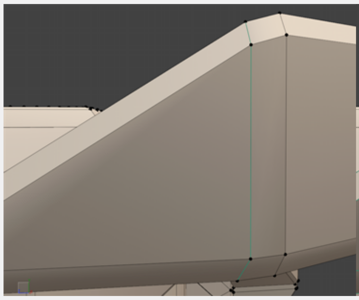
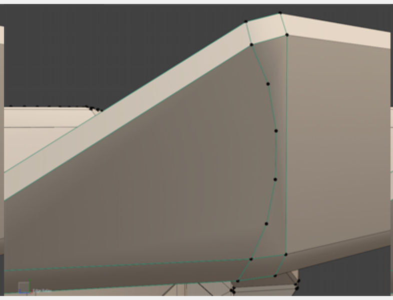
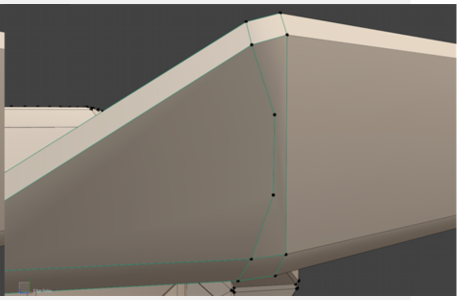
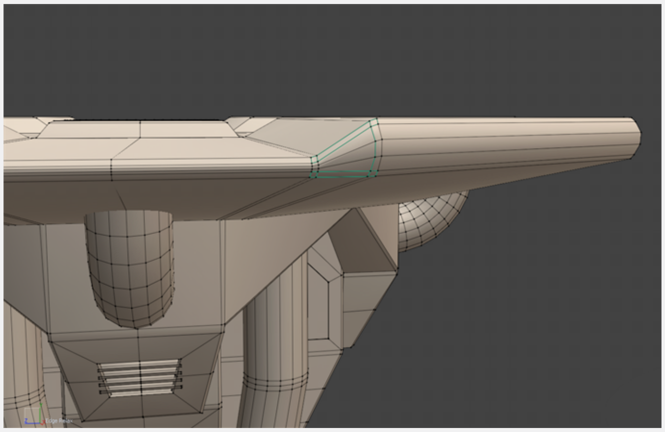
- **Tool Pipe** - Add falloffs and sub-tools to the operation.
- **Selection** - Select edges to relax or modify your existing selection.
- **Geometry** - Any geometry that is affected by the tool.

Edge Relax Properties

You can find the Edge Relax Tool Properties on the left panel.

Offset	The value by which the bevel radius is modified.
Tension	Changes the profile of the radius.
Depth	Adjusts the roundness of relaxed edges. 100% is the default value for round edges. 0% makes round positions flat, and -100% produces a negative radius.
Propagate	Specifies the number of parallel edges that are affected by the **Offset** value.

Propagate Scale	When **Propagate** is higher than **0**, determines the effect of the falloff on the propagation. The default is **100%** and decreasing the value decreases the effect of the falloff.
Propagate Side	By default, edges are relaxed on either side of the selection, but you can choose to propagate to parallel edges on only the **Positive** or the **Negative** side.
Shape Preset	When **Propagate** is higher than **0**, specifies the falloff type affecting the propagation. You have the following options: • **Linear** • **Ease-In** • **Ease-Out** • **Smooth** • **Custom**
Convergence	When enabled, instead of relaxing the edges, Modo welds the points of the radius together. 
Add Point	Inserts new points along the selected edges, which helps you increase the rounding of edges. • **Relative** - Increases the number of points by the specified amount. So if you had two points and set **Number of Points** to '4', you will have six points altogether. • **Absolute** - Sets the number of points to the specified amount. So if you had two points, and set **Number of Points** to '4', Modo adds two points to the existing two points.

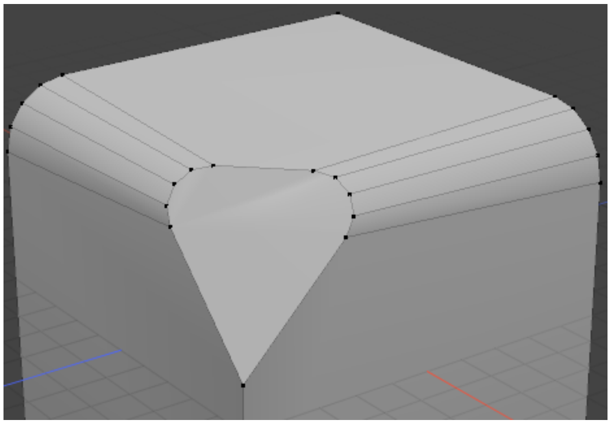
	 No added points.  Four points added with **Relative** mode  Four points added with **Absolute** mode **Tip:** You can use the **Propagate** property to copy the new point settings to parallel edges.  Four points added with **Relative** mode and Propagate set to 4
Number of Points	The number of points inserted by the **Add Point** option. When this value is negative, the tool removes existing points.

Edge Chamfer

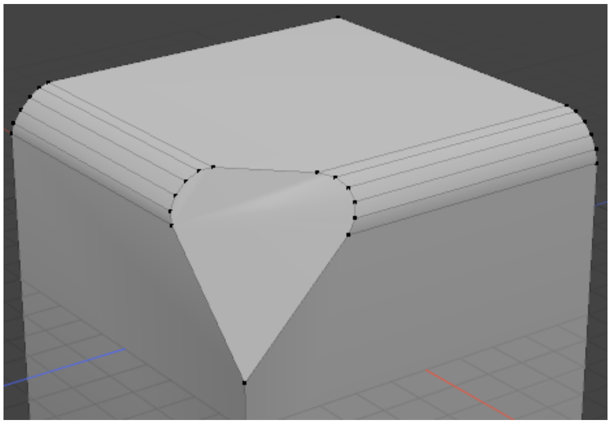
Edge Chamfer allows you to round the edges of a mesh, similarly to Edge Bevel. You can customize the effect by fine-tuning the tool's properties, use Profile shapes to determine the shape of the chamfer, and modulate the effect using Falloffs.

Edge Chamfer has a few advantages over the Edge Bevel tool:

- Edges are offset from the source in parallel, producing better geometry. You can toggle this behavior with the **Offset in Parallel** option in the Edge Chamfer Properties.

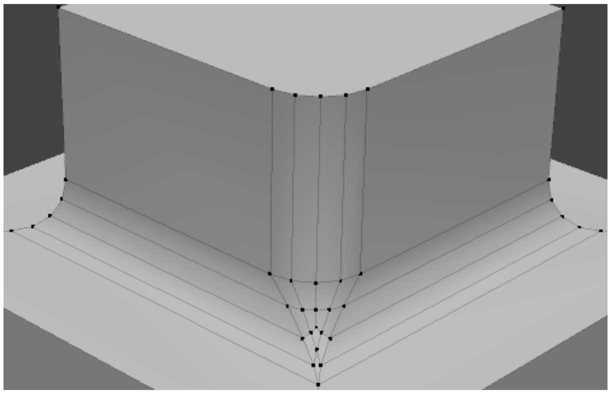


Edge Bevel

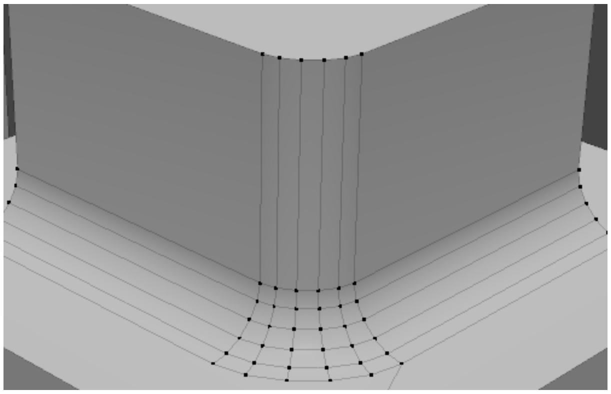


Edge Chamfer

- Cleaner corner geometry compared to Edge bevel.

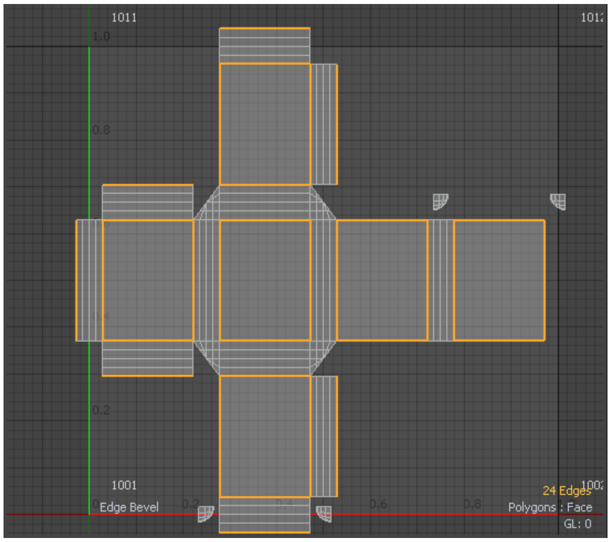


Edge Bevel

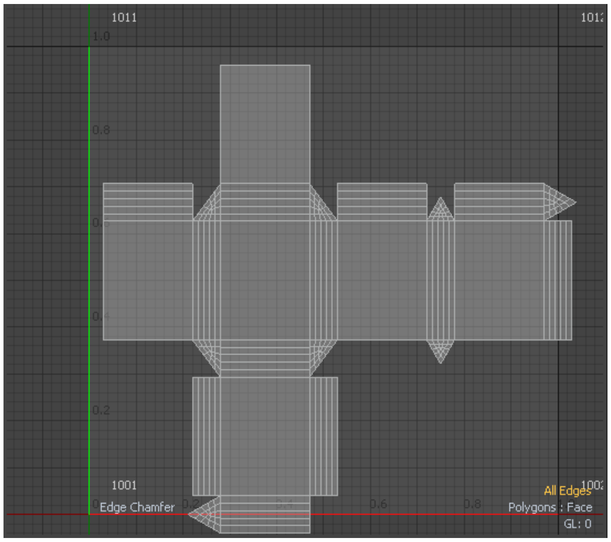


Edge Chamfer

- Produces cleaner UVs by handling discontiguous edges better.


1011
1.0
0.8
0.2
1001
Edge Bevel
24 Edges
Polygons : Face
GL: 0

Edge Bevel


1011
1.0
0.8
0.2
1001
Edge Chamfer
All Edges
Polygons : Face
GL: 0

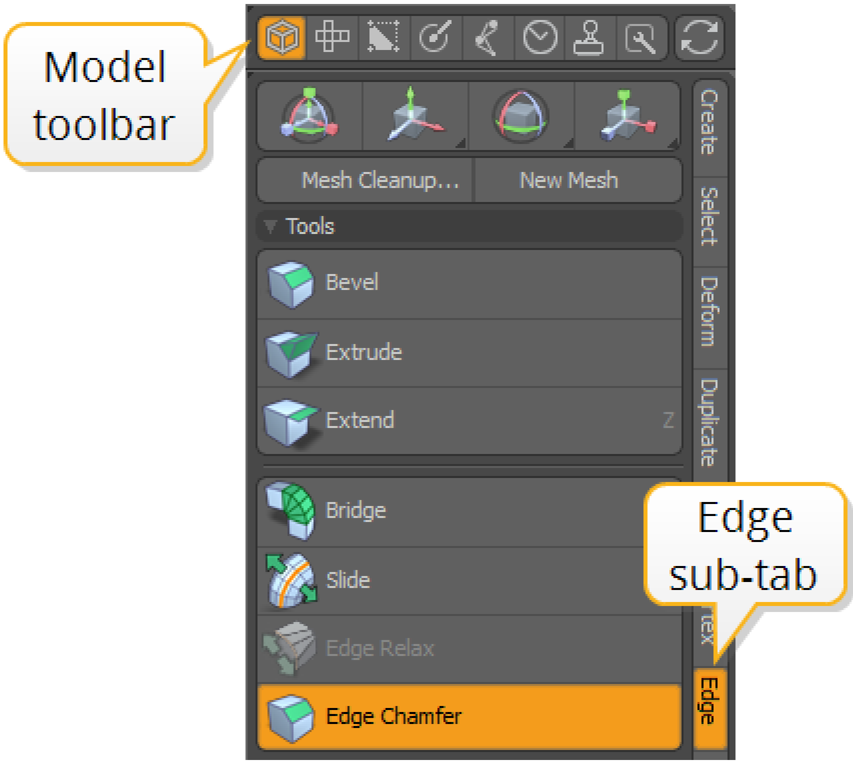
Edge Chamfer

Using Edge Chamfer

Direct Modeling

It's best to use the **Modo** or **Model** layouts when working with Edge Chamfer. This workflow example uses the **Modo** layout.

1. In **Edges** selection mode, select the edges to round.
2. Open the **Model** toolbar on the left panel and click the **Edge** sub-tab.


Model toolbar
Mesh Cleanup...
New Mesh
Tools
Bevel
Extrude
Extend
Z
Bridge
Slide
Edge Relax
Edge Chamfer
Create
Select
Deform
Duplicate
Edge
Edge sub-tab

3. Click **Edge Chamfer**, then click in the viewport to activate the tool.

   A blue tool handle appears in the 3D viewport, indicating that the tool is active.
4. Drag on the handle to adjust the inset, or click the **Tool Properties** button on the left panel to adjust the tool's settings.

   For more information on the tool properties, see Edge Chamfer Properties.

You can use Falloffs to modulate the strength of the tool on different parts of the geometry. For more information on working with Falloffs, see Using Falloffs.

Procedural Edge Chamfer

The Edge Chamfer mesh operation allows you to round edges on a mesh procedurally.

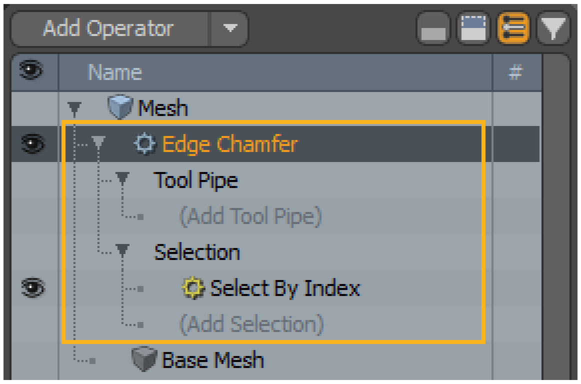
1. In **Edges** selection mode, select the edges to chamfer.
2. On the right panel, click the **Mesh Ops** tab.

> **Note:** If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

3. Click the **Add Operator** button, and under **Mesh Operations**, click **Edge** , then double-click **Edge Chamfer**.

   A blue tool handle appears in the 3D viewport, indicating that the tool is active.

   You can expand the **Edge Chamfer** mesh operation in the list by clicking the arrow ▼ in front of it. This reveals the inputs the operation uses:



   - **Tool Pipe** - Add falloffs and sub-tools to the operation.
   - **Selection** - Select edges or modify your existing selection. For more information on procedural selection, see Procedural Selection.

4. Drag on the handle to adjust the inset, or open the **Properties** tab and adjust the tool's settings. To do this, select the Edge Chamfer mesh operation in the list. This opens the **Properties** tab on the right panel.

   For more information on the mesh operation's properties, see Edge Chamfer Properties.

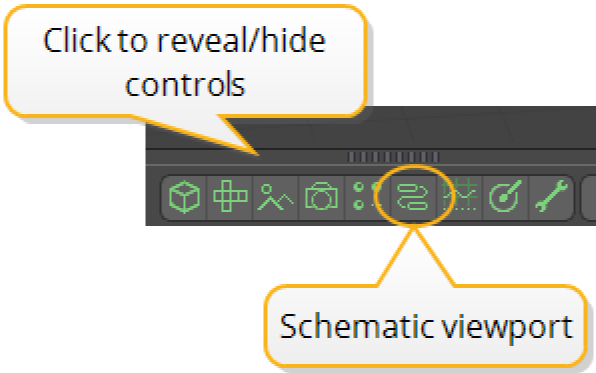
Edge Chamfer in the Schematic

You can also use Edge Chamfer when working in the Schematic viewport.

Note: For more information on working with Schematic viewport in general, see Schematic Viewport.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.



>

MadCap:conditions="Default.NoTranslate">OR

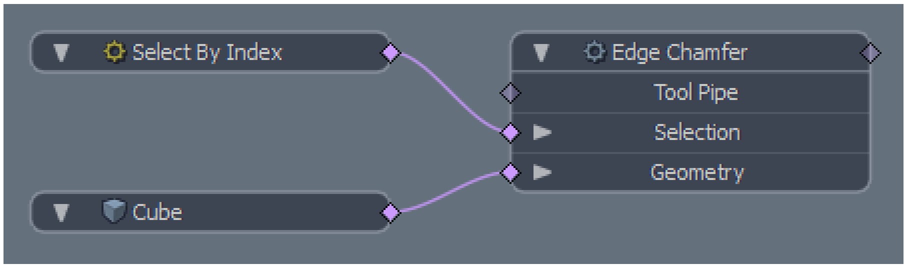
- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

To use the Edge Chamfer node:

1. Click **Add...**, and under **Mesh Operations** > **Edge**, double-click **Edge Chamfer**.

   This adds the Edge Chamfer node to the viewport and opens its properties on the lower right panel. You can also see a blue tool handle appear in the 3D viewport, indicating that the tool is active.

Tip: Selecting the edges in the viewport in advance adds the **Select By Index** selection operator containing your selection. You can edit the selection through the node by selecting it. This opens its properties in the lower right panel. For more information see Select by Index.



The node has the following inputs:

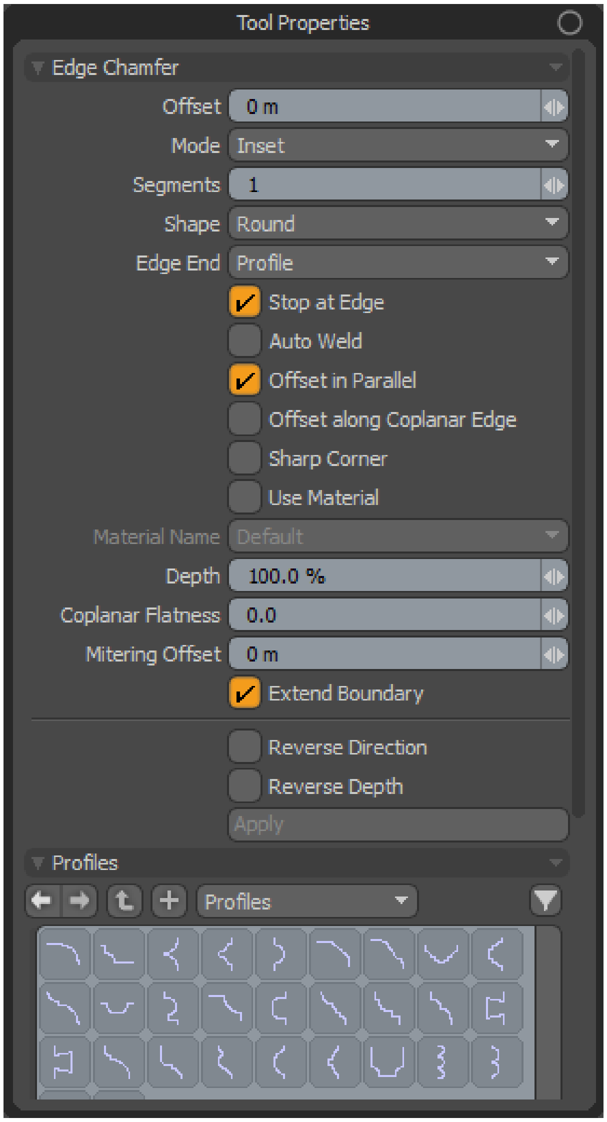
- **Tool Pipe** - Add falloffs and sub-tools to the operation.

  - **Selection** - Select polygons or modify your existing selection.
  - **Geometry** - Any geometry that is affected by the tool.

2. Drag the blue tool handle in the viewport or adjust the properties in the **Properties** tab to get the result you need.

Edge Chamfer Properties

Edge Chamfer has the following properties.



Edge Chamfer

Offset: 0 m

Mode: Inset

Segments: 1

Shape: Round

Edge End: Profile

☑ Stop at Edge

☐ Auto Weld

☑ Offset in Parallel

☐ Offset along Coplanar Edge

☐ Sharp Corner

☐ Use Material

Material Name: Default

Depth: 100.0 %

Coplanar Flatness: 0.0

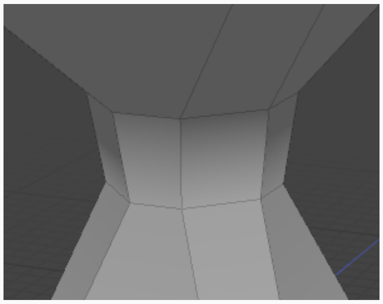
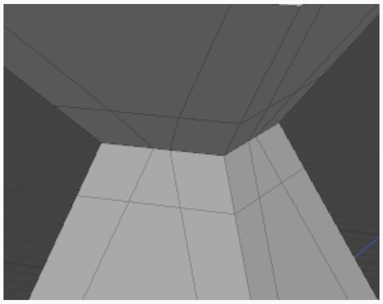
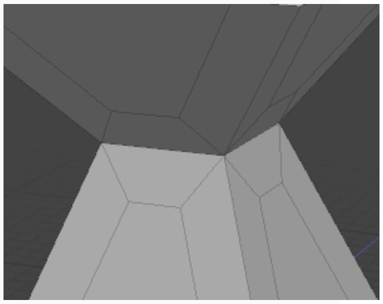
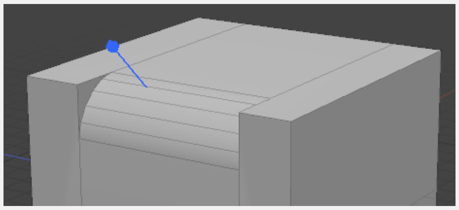
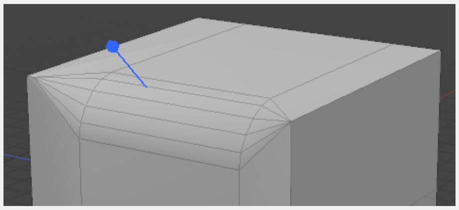
Mitering Offset: 0 m

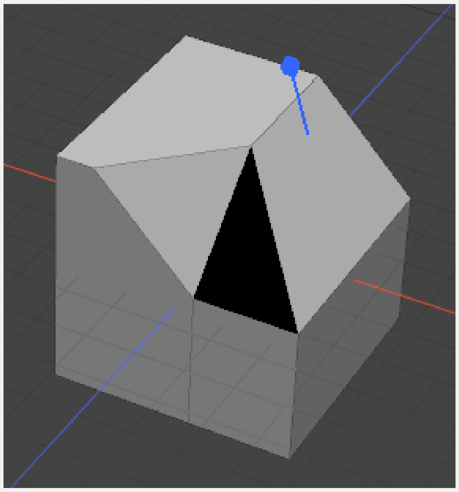
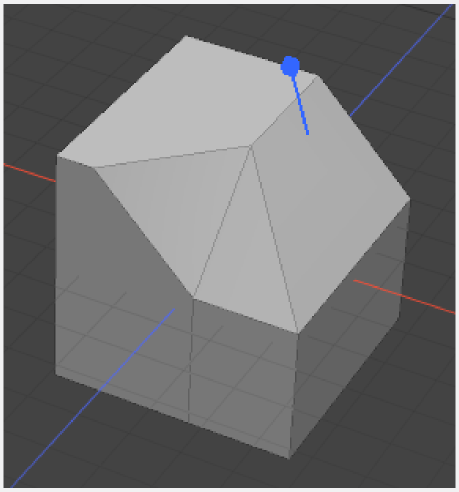
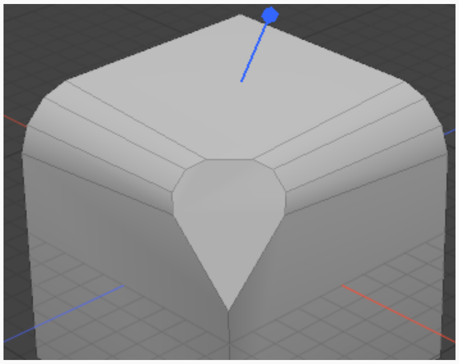
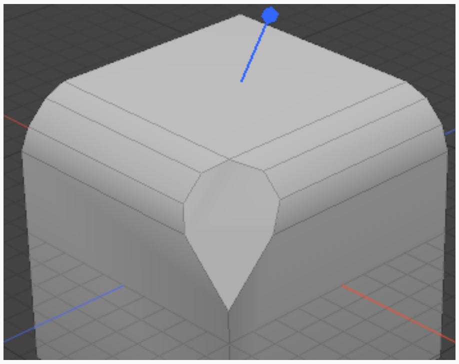
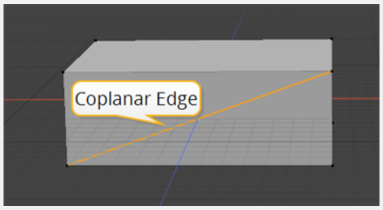
☑ Extend Boundary

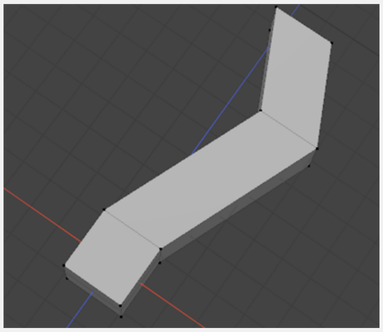
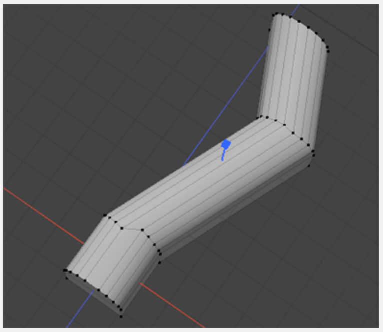
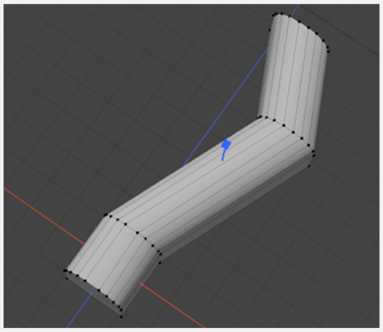
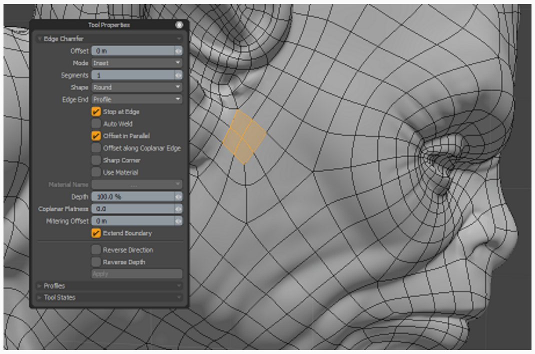
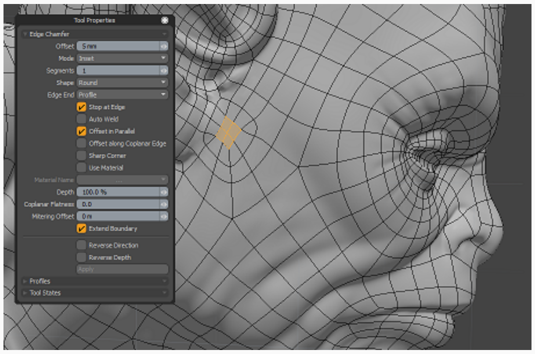
☐ Reverse Direction

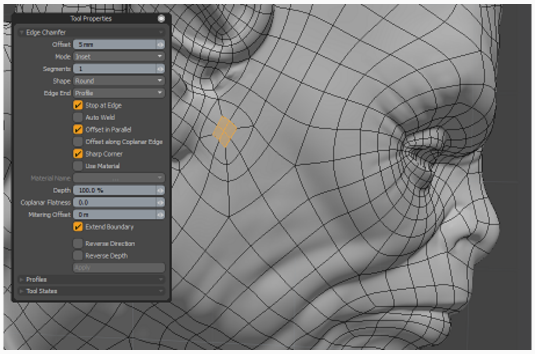
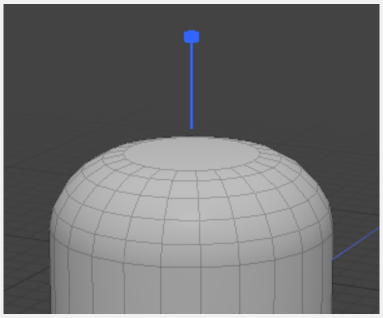
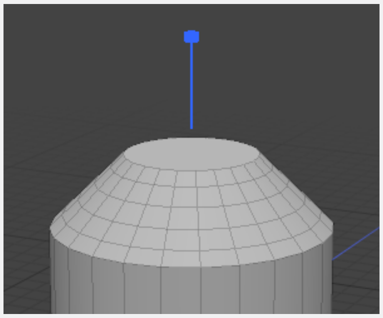
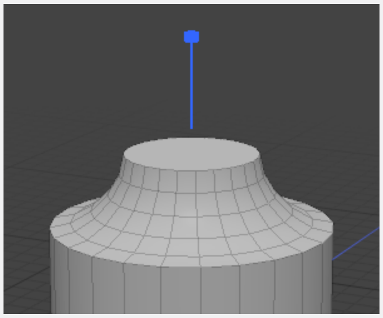
☐ Reverse Depth

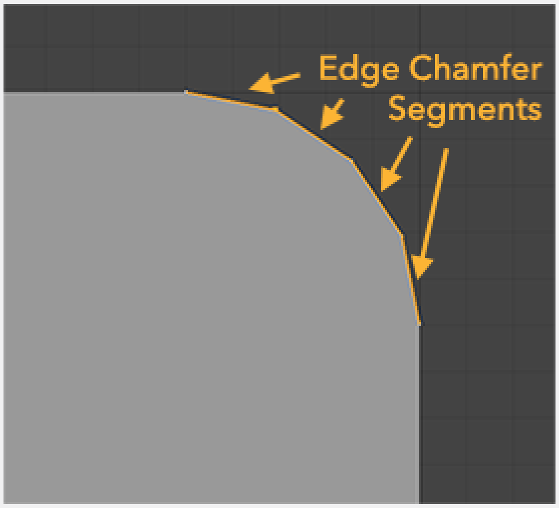
Offset	Specifies the chamfer inset or width, depending on what **Mode** option is used.

Mode	There are two ways you can use the tool: • **Inset** - Determines the distance away form the original edge. The inset is parallel to the source edge. • **Width** - Determines the actual width of the new chamfer.
Segments	The number of divisions for rounding the edge. This is similar to the Edge Bevel tool's **Level** property, but allows you to set an odd number of segments. The more segments you have, the smoother the chamfer.
Shape	Specifies the chamfer shape. The options are:  **Round** The default shape.  **Square** Creates parallel edges horizontally and vertically.  **Sharp** Creates evenly inset edges.
Edge End	Specifies the shape created at the end of the selected edges. The following options are available.  Profile  Triangle Fan
Stop at Edge	When enabled, it prevents the geometry from overrunning when using a large offset size.
Auto Weld	When enabled, the new inset vertices on the same position are merged into one.

	 Disabled  Enabled
Offset in Parallel	Edge Chamfer offsets edges from source position in parallel. When **Offset in Parallel** is disabled, new vertices are placed along the edge at the specific offset distance from the source vertex, similar to the Edge Bevel tool.  Disabled  Enabled
Offset along Coplanar Edge	Coplanar edges are edges that connect two polygons on the same plane.  Usually, coplanar edges are unnecessary edges in beveling. Edge Chamfer hides coplanar edges before chamfering and computes the inset vector from incoming and outgoing edges at the vertex on the face.

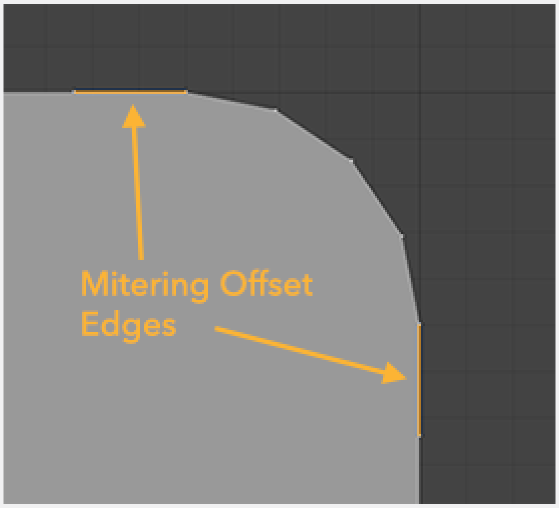
	However, sometimes you need to inset vertex positions along coplanar edges. **Offset along Coplanar Edge** allows you to offset vertex positions along connecting coplanar edges.  Source Mesh  Disabled  Enabled
Sharp Corner	When enabled, keeps chamfered corners as sharp as possible. For edges shared by three or more polygons, the polygons are grouped into pairs and new faces are created between each pair, ignoring polygons not part of a pair. At junctions between three or more edges, extra polygons are inserted to bridge any gaps created by the extrusion. Extra polygons are also inserted at the bare ends of edges to keep the shape closed and planar as the edge is chamfered.  Tool Properties Edge Chamfer Offset 0 m Mode Inset Segments 1 Shape Round Edge End Profile ☑ Stop at Edge ☐ Auto Weld ☑ Offset in Parallel ☐ Offset along Coplanar Edge ☐ Sharp Corner ☐ Use Material Material Name ... Depth 100.0 % Coplanar Flatness 0.0 Mitering Offset 0 m ☑ Extend Boundary ☐ Reverse Direction ☐ Reverse Depth Apply Profiles Tool States Four polygons selected on a mesh.  Tool Properties Edge Chamfer Offset 5 mm Mode Inset Segments 1 Shape Round Edge End Profile ☑ Stop at Edge ☐ Auto Weld ☑ Offset in Parallel ☐ Offset along Coplanar Edge ☐ Sharp Corner ☐ Use Material Material Name ... Depth 100.0 % Coplanar Flatness 0.0 Mitering Offset 0 m ☑ Extend Boundary ☐ Reverse Direction ☐ Reverse Depth Apply Profiles Tool States An edge chamfer applied to the four selected polygons.

	 Tool Properties Edge Chamfer Offset: 5 mm Mode: Inset Segments: 1 Shape: Round Edge End: Profile ☑ Stop at Edge ☐ Auto Weld ☑ Offset in Parallel ☐ Offset along Coplanar Edge ☑ Sharp Corner ☐ Use Material Material Name: ... Depth: 100.0 % Coplanar Flatness: 0.0 Mitering Offset: 0 m ☑ Extend Boundary ☐ Reverse Direction ☐ Reverse Depth **Sharp Corners** enabled, keeping the corners of the chamfer sharp and removing unnecessary triangles.
Use Material	When enabled, you can choose an existing material to apply to all newly created geometry. When disabled, the tool uses the neighboring material.
Material Name	When **Use Material** is enabled, you can select the material to be used from this dropdown.
Depth	Adjusts the roundness of chamfered edges. 100% is the default value for round bevel. 0% makes round positions flat and -100% produces a negative radius.  Depth 100%  Depth 0%  Depth -100%
Coplanar Flatness	Edge Chamfer hides all coplanar edges before modifying the edge, to remove unnecessary complexity. This is the threshold value to detect coplanar edges.
Mitering Offset	The **Mitering Offset** option creates extra co-planar polygons beside the outer chamfered edges. Increasing the value increases the distance between the outer chamfered edge and the edge of the added polygon.


Edge Chamfer
Segments

Segments = **4**

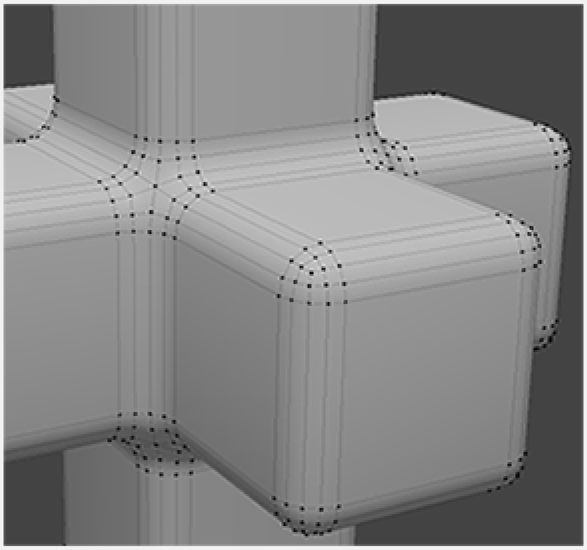
Mitering Offset = **0mm**


Mitering Offset
Edges

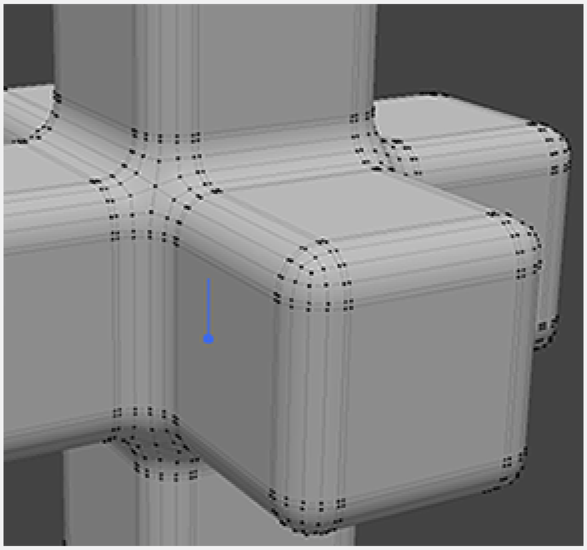
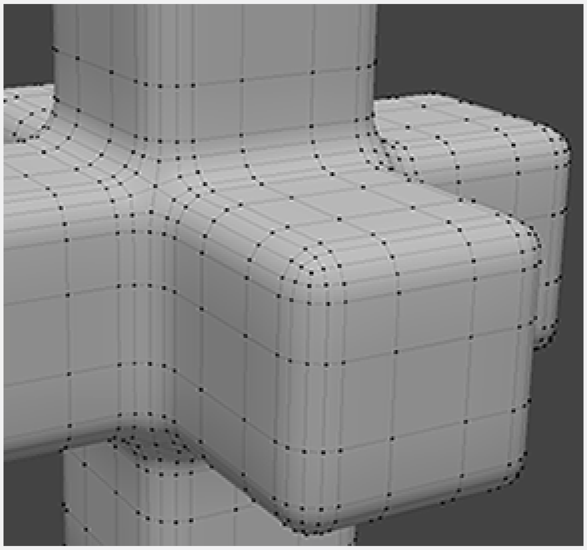
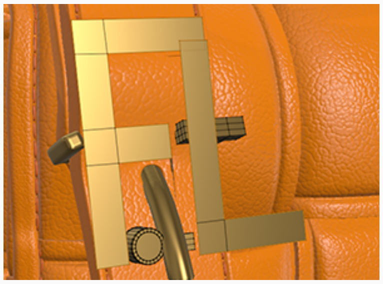
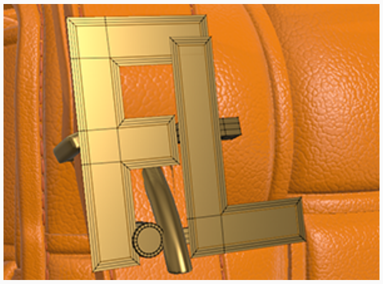
Segments = **4**

Mitering Offset = **120mm**

This option is useful to smooth artifacts on connecting polygons, allowing for a better transition between a chamfered edge and its connecting polygon.

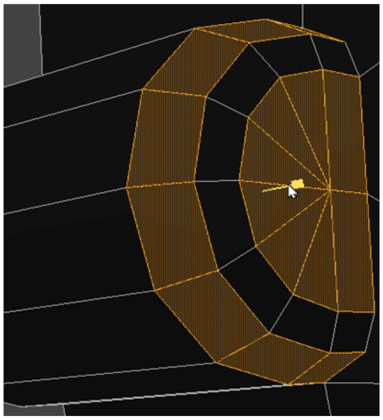


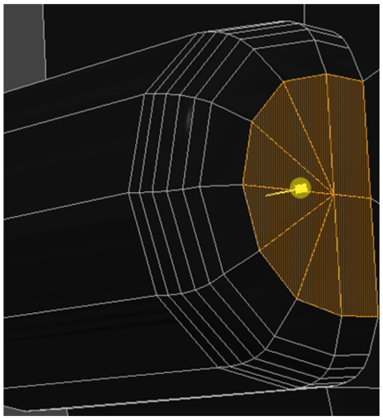
Offset = **150mm**
Segments = **4**
Mitering Offset = **0mm**

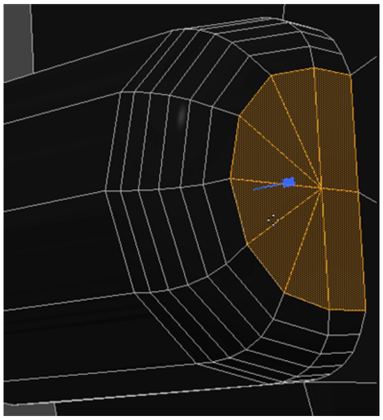
	 **Offset** = **150mm** **Segments** = **4** **Mitering Offset** = **20mm**  **Offset** = **150mm** **Segments** = **4** **Mitering Offset** = **200mm**
Extend Boundary	When enabled, the boundary of selected edges can be extended outward. This option is available both for procedural and direct Edge Chamfer.  Edge selected  Boundary extended
Reverse Direction	When enabled, the start and end of the profile curve is reversed.
Reverse Depth	When enabled, the depth direction of the profile shape is reversed. You can use **Depth** to control the depth amount.
Profiles	Select a 1D profile shape to use on the chamfered edge.

Chamfer Edit

Chamfer Edit allows you to refine your chamfered and beveled faces after a Bevel or Chamfer operation has been applied. The tool increases the number of segments on chamfered faces and also changes the width of faces on chamfered edges.







You can also assign profiles to chamfered edges using the Chamfer Edit tool, which apply a curve of your choice to an edge to create interesting forms for your edges.

Note: The Chamfer Edit tool does not currently support editing rounded beveled faces.

Using Chamfer Edit

The Chamfer Edit tool is available in both direct modeling and procedural modeling workflows. To understand the difference between these workflows, see Modeling Techniques.

Direct Modeling

The direct modeling Chamfer Edit tool can be found in the **Polygon** tab on the **Model** layout. To apply a chamfer edit to an existing chamfered or beveled edge using direct modeling tools:

1. Click **Polygons** to activate Polygon Selection mode.
2. Select the two outer polygons of a beveled or chamfered edge.
3. Click **Chamfer Edit** in the **Polygon** tab to activate the tool, and drag the blue widget in the viewport to change the offset of the chamfer.

4. Click **Tool Properties** or press **K** to open the Tool Properties popover, and change **Segments** to **5**.

   The selected faces have five new segments added to them, causing the corners to become rounder.

Note: See Chamfer Edit Tool Properties to learn more about the different properties available on the Chamfer Edit tool.

Procedural Modeling

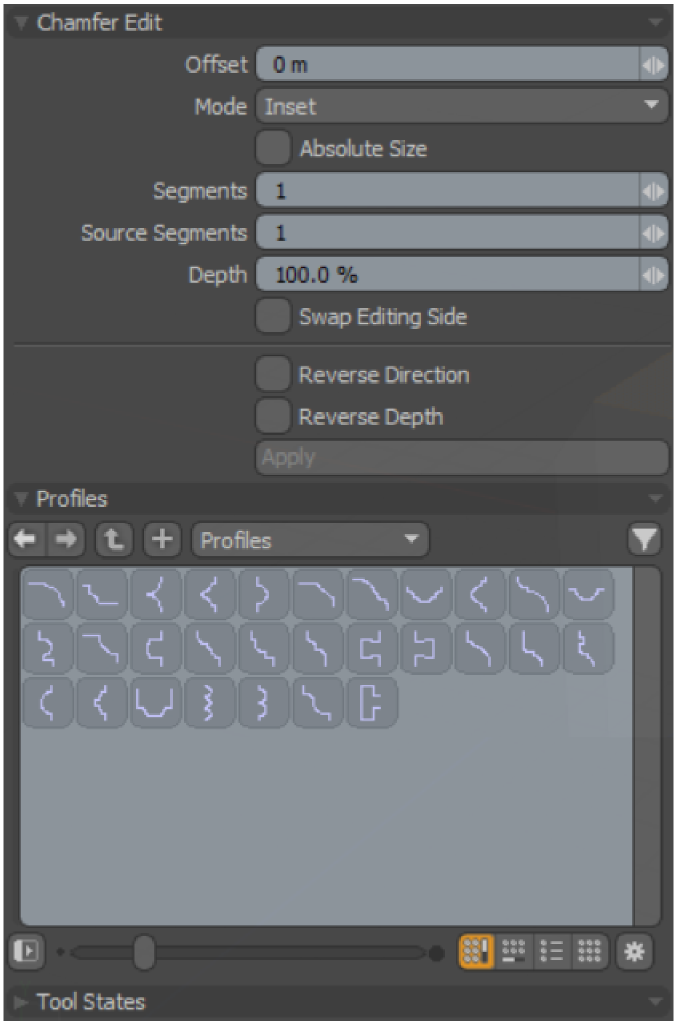
Procedural mesh operations can be found on the **Model** layout in the **Mesh Ops** tab. To apply a chamfer edit to an existing chamfered or bevel edge using procedural mesh operations:

1. Click **Polygons** to activate Polygon Selection mode.
2. Select the two outer polygons of a beveled or chamfered edge.
3. Click the **Mesh Ops** tab in the **Model** layout.
4. Click **Add Operator** and type **Chamfer Edit** into the search field.
5. Double-click **Chamfer Edit** to add it to the mesh operations stack.
6. Drag **Offset** in the **Chamfer Edit** properties to adjust the inset of the chamfer.
7. Change **Segments** to **5** to add more segments to each selected face.

Tip: To use profiles with the procedural chamfer edit mesh operation, add a **Content Preset** to the **Tool Pipe** menu under the **Chamfer Edit** in the **Mesh Ops** tab.

Chamfer Edit Tool Properties

With the Chamfer Edit tool selected, click **Tool Properties** or press **K** to open the **Tool Properties** panel.



Offset: 0 m

Mode: Inset

☐ Absolute Size

Segments: 1

Source Segments: 1

Depth: 100.0 %

☐ Swap Editing Side

☐ Reverse Direction

☐ Reverse Depth

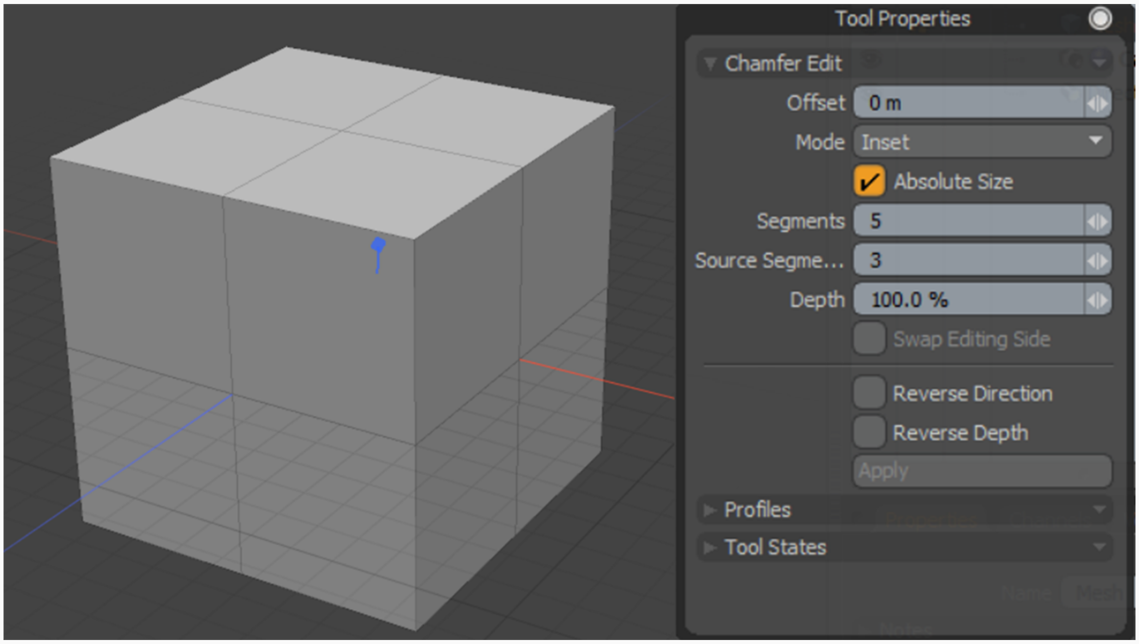
The Chamfer Edit **Tool Properties**.

Enable	Enables or disables the chamfer edit operation on the mesh. **Note:** The **Enable** option is only available in the Chamfer Edit mesh operation.
Offset	Specifies the chamfer inset or width, depending on what **Mode** option is used.
Mode	There are two ways you can use the tool: • **Inset** - Determines the distance away from the original edge. The inset is parallel to the source edge. • **Width** - Determines the width of the new chamfer.
Absolute	Enabling **Absolute Size** allows you to edit the offset of a chamfered edge assuming that a

Size

zero offset is a straight edge and any positive number above zero will be a curved edge.

When **Absolute Size** is not enabled, an offset value of zero represents how curved the edge was before Edit Chamfer was activated.



Tool Properties

Chamfer Edit

Offset 0 m

Mode Inset

☑ Absolute Size

Segments 5

Source Segme… 3

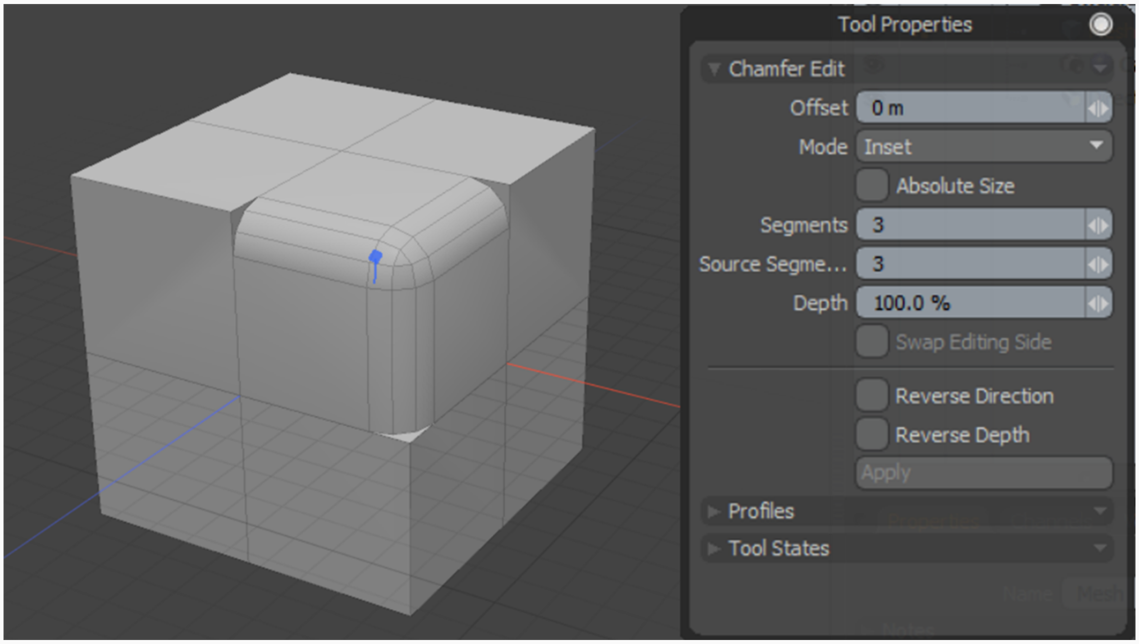
Depth 100.0 %

☐ Swap Editing Side

☐ Reverse Direction

☐ Reverse Depth

Offset 0 with **Absolute Size**



Tool Properties

Chamfer Edit

Offset 0 m

Mode Inset

☐ Absolute Size

Segments 3

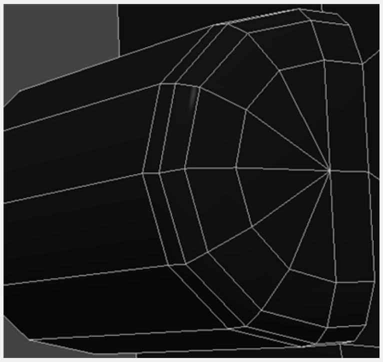
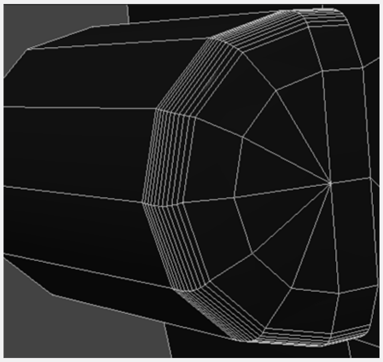
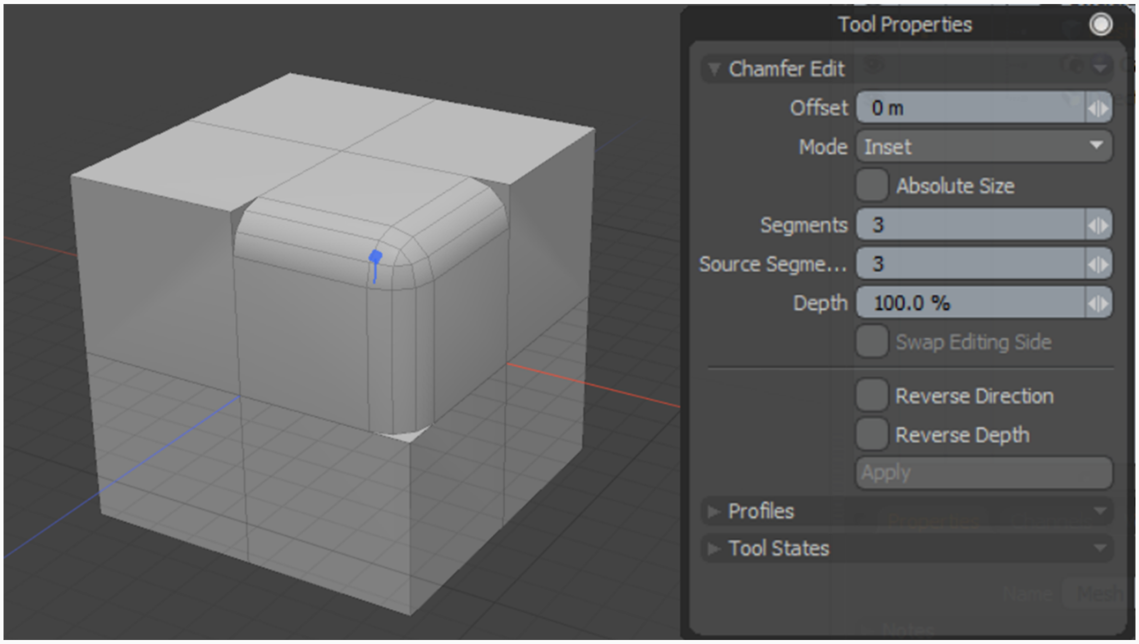
Source Segme… 3

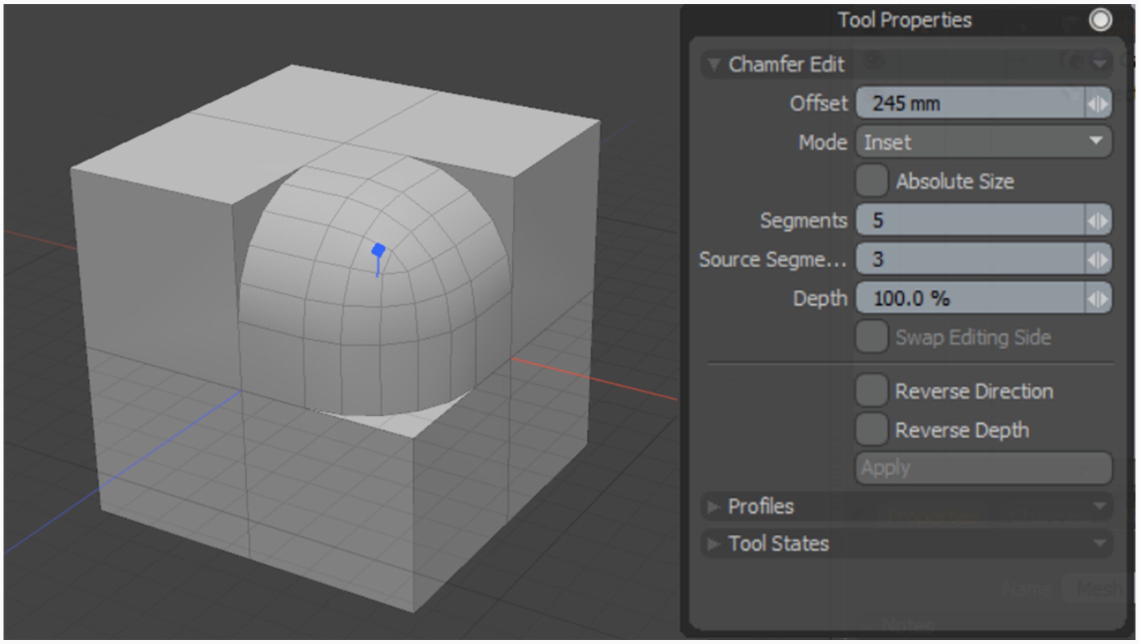
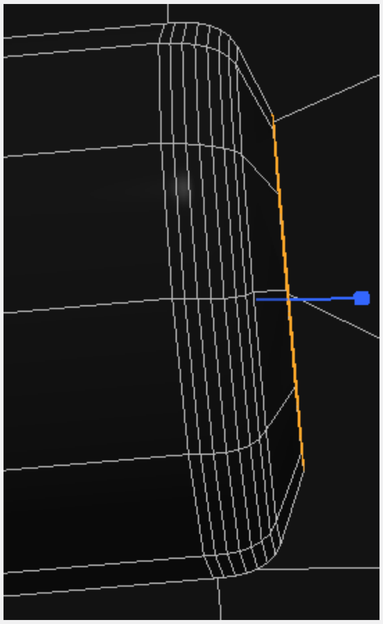
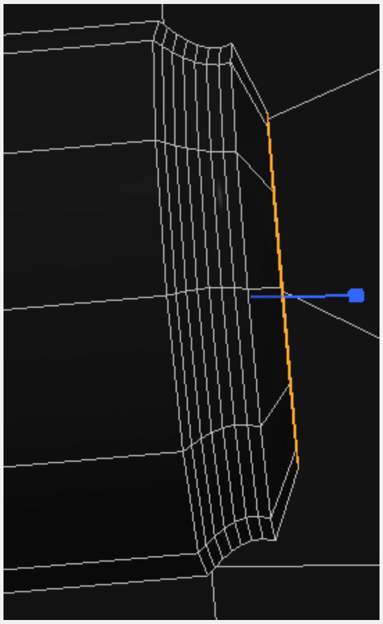
Depth 100.0 %

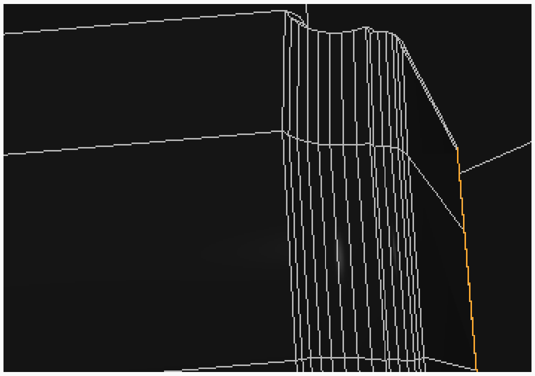
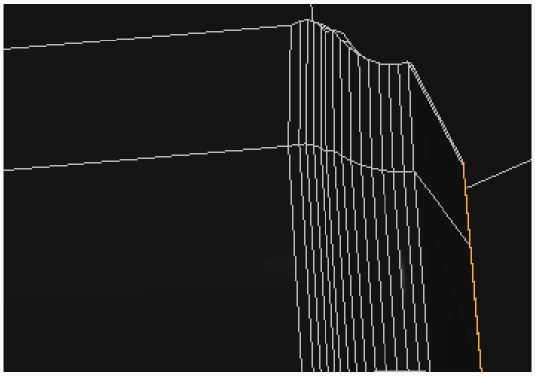
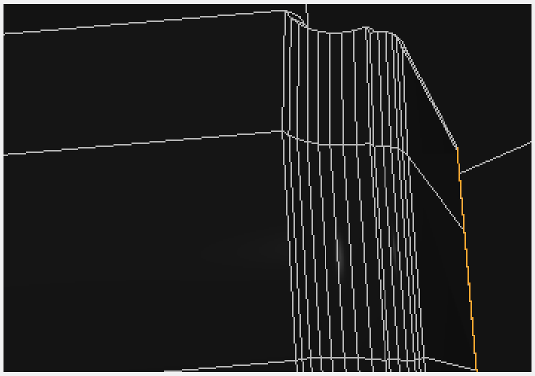
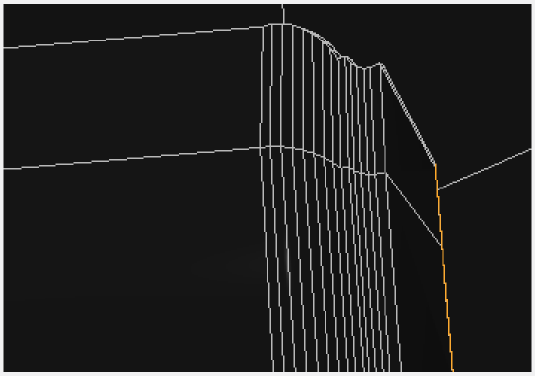
☐ Swap Editing Side

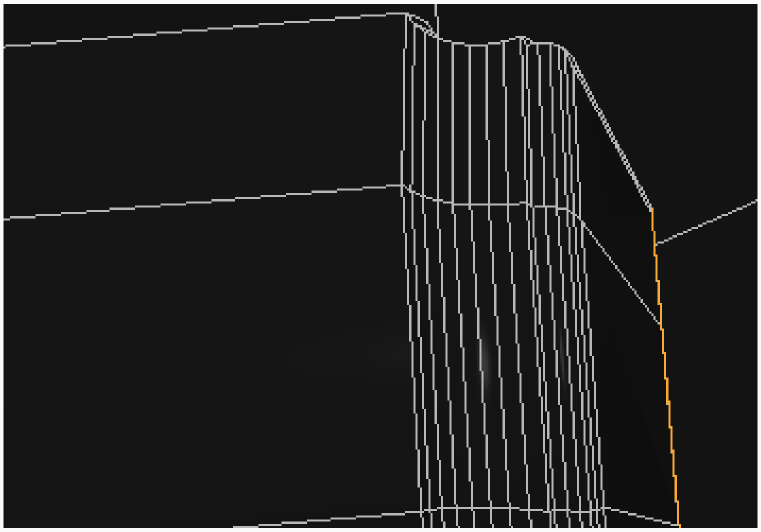
☐ Reverse Direction

☐ Reverse Depth

	Offset 0 without **Absolute Size**
Segments	The number of divisions for rounding the edge. The more segments you have, the smoother the chamfer.  A chamfer edit with a **Segment** value of **2**.  A chamfer edit with a **Segment** value of **6**.
Source Segments	Source Segments allows you to edit the number of segments and offset of a chamfered edge. This tool is useful when making edits to larger models that already have rounded edges. In order for the tool to work, the Source Segments parameter must match the amount of segments selected.  Tool Properties Chamfer Edit Offset 0 m Mode Inset ☐ Absolute Size Segments 3 Source Segme… 3 Depth 100.0 % ☐ Swap Editing Side ☐ Reverse Direction ☐ Reverse Depth Apply Profiles Tool States

	 Tool Properties Chamfer Edit Offset 245 mm Mode Inset ☐ Absolute Size Segments 5 Source Segme... 3 Depth 100.0 % ☐ Swap Editing Side ☐ Reverse Direction ☐ Reverse Depth Apply Profiles Tool States Using the Chamfer Edit tool to increase the **Offset** and **Segment** amounts of a chamfered edge.
Depth	Adjusts the roundness of chamfered edges. 100% is the default value for round bevel. 0% makes round positions flat and -100% produces a negative radius.  A chamfer edit with a Depth value of **100%**.  A chamfer edit with a Depth value of **-100%**.

Reverse Direction	When enabled, the start and end of the profile curve is reversed.  A chamfered edge with a **Profile** assigned.  A chamfered edge with a profile assigned and **Reverse Direction** enabled.
Reverse Depth	When enabled, the depth direction of the profile shape is reversed. You can use **Depth** to control the depth amount.  A chamfered edge with a profile assigned.  A chamfered edge with a profile assigned and **Reverse Depth** enabled.
Profiles	Select a 1D profile shape to use on the chamfered edge.



A chamfered edge with a profile assigned.

Note: The **Profiles** option is only available on the direct modeling Chamfer Edit tool.

Edge Tools

The following are useful actions you might want to do using **Edge Tools**:

Spin Edge

The **Spin Edge** tool turns the current edge or edges to re-organize the way they separate the two adjoining polygons. For example, if you select an edge, this command spins it so that it attaches to different vertices while leaving the two bisected polygons in place. It changes the flow of your polygons while maintaining the surrounding mesh.

Collapse

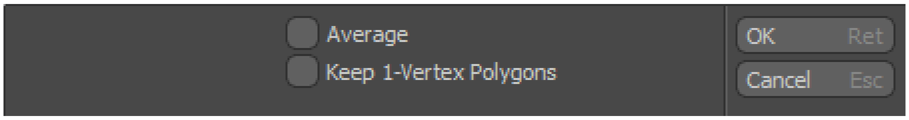
The **Collapse** command removes the selected element without destroying the integrity of the geometry. Any selected polygon, edge, or vertex is deleted but no hole is left behind. Instead, the mesh heals, closing any gaps by merging the neighboring elements together.

Remove

The **Remove Edge** command can be used to delete edges from the mesh geometry. This differs slightly from the **Geometry** > **Remove** tool in that it doesn't remove attached vertices, it only removes the selected edges. The vertices remain behind. To remove the edges and vertices, select the edges and then on the menu bar choose **Geometry** > **Remove**. To remove only the edges, select the edges, then on the **Modo Tools** toolbar, click the **Edge** tab, then **Remove Edges** on the **Commands** section. The edges are removed and the vertices remain.

Join

Join Edges allows you to weld two edges together. Simply select two edges and run the tool. The first selected edge moves to the position of the second selected edge and the vertices are joined.



Split

Split edge works like the **Split** vertex command. Instead of splitting vertices around a selected vertex, it separates mesh polygons along selected edges. In polygon selection mode, the command splits boundary edges of the selected polygons. This is useful when separating polygons that are selected by the close loop selection command. This is also useful for opening up a "hinge" section on a mesh.

Activating the Edge Split Tool

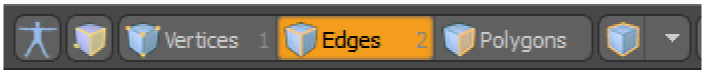
- In the **Model** layout, on the left panel, open the **Edge** tab, and click **Split.**
- In the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Edge** > **Edge Split**.
- In the **Schematic** view, click **Add...**, and double-click **Mesh Operations** > **Edge** > **Edge Split**.

Applying the Procedural Edge Split Tool

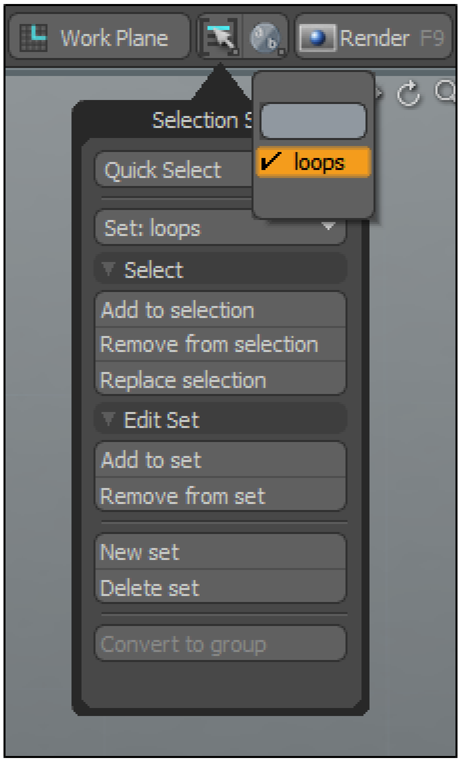
The procedural **Edge Split** tool allows you to select a set of edges and split the mesh in two or more patches. You can interactively update the **Gap** distance by changing the value.

Example

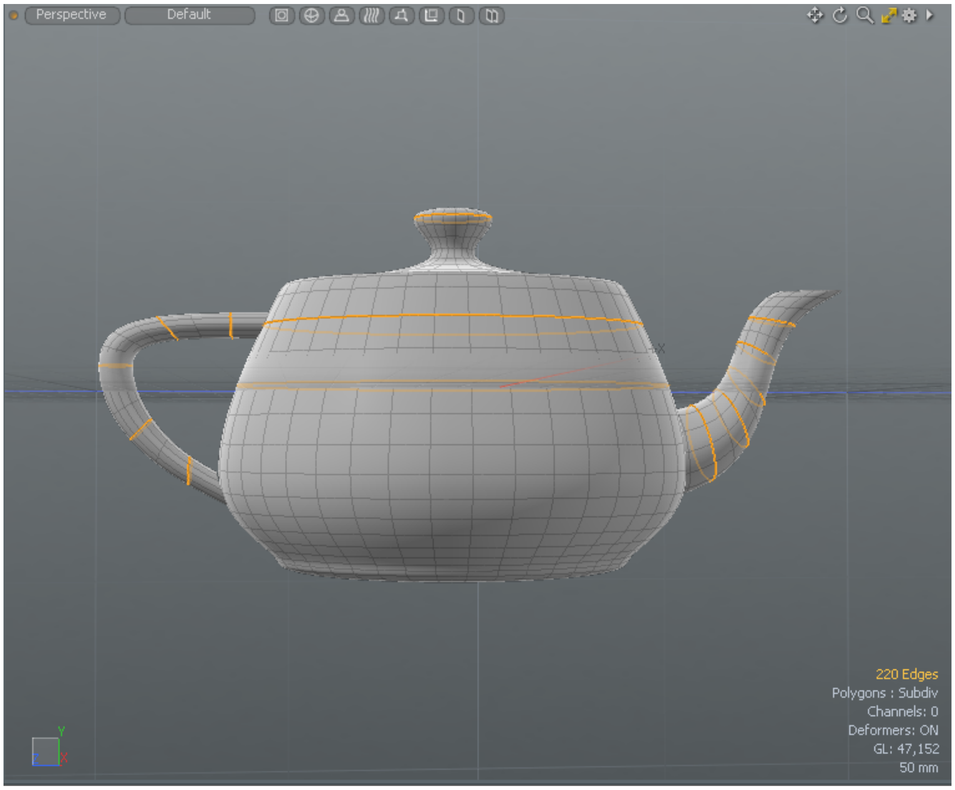
1. Download our example scenes and unpack the files.
2. In the **Model** layout, load **assign_edge_split_tool_procedural.lxo**.
3. On the right panel, ensure that the **Teapot** item is selected.
4. Under the layout menu, select **Edges**.


Vertices 1
Edges 2
Polygons

5. On the top-right of the interface, click the **Selection Sets** icon, and then click **Quick Select** > **loops**.

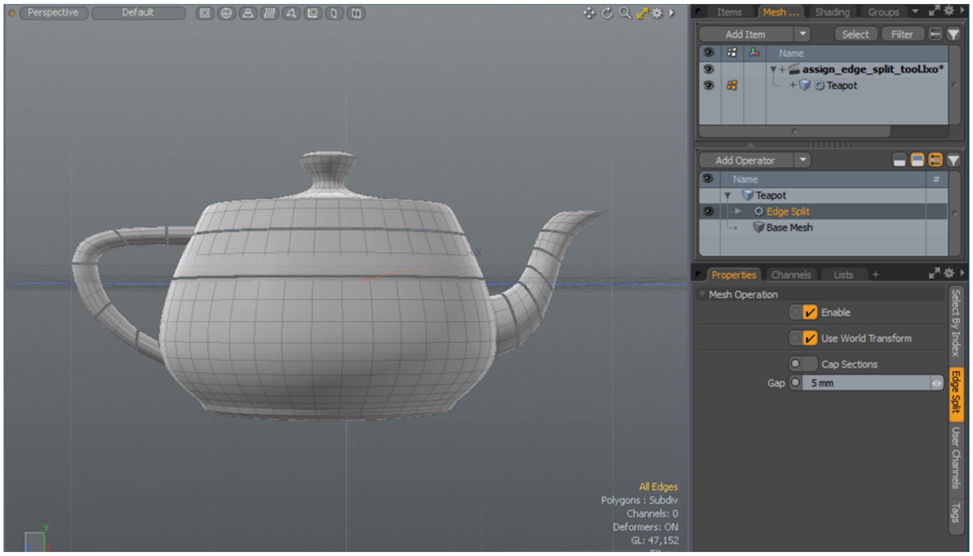

Work Plane
Render F9
Selection S
Quick Select
✓ loops
Set: loops
Select
Add to selection
Remove from selection
Replace selection
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group

The edge selection set is displayed in the 3D viewport.


Perspective
Default
220 Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
50 mm

On the right panel, open the **Mesh Ops** tab, and click **Add Operator**.

6. Double-click **Mesh Operations** > **Edges** > **Edge Split**.
7. On the right panel, open the **Edge Split** tab and set **Gap** to **5mm** and click **Return**.


Perspective
Default
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
assign_edge_split_tool.lxo*
Teapot
Add Operator
Name
#
Teapot
Edge Split
Base Mesh
Properties
Channels
Lists
+
Select By Index
Edge Split
User Channels
Tags
All Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152

Mesh Operation

☑ Enable

☑ Use World Transform

☐ Cap Sections

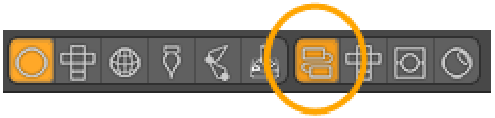
Gap 5 mm

Applying the Edge Split Tool using the Schematic Viewport

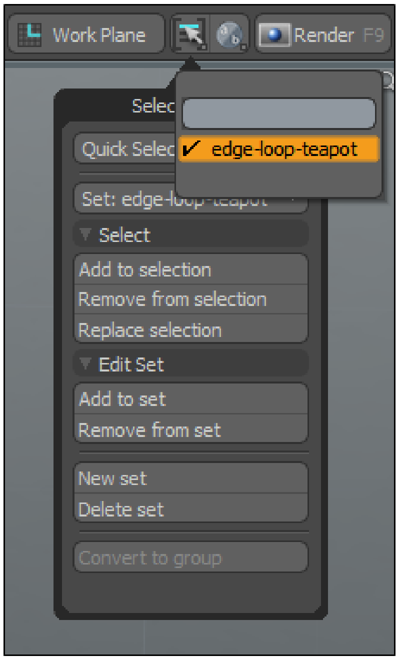
Using the procedural **Edge Split** tool in the **Schematic** viewport allows you to select a set of edges and split the mesh in two or more patches. You can interactively update the **Gap** distance by changing the value.

Example

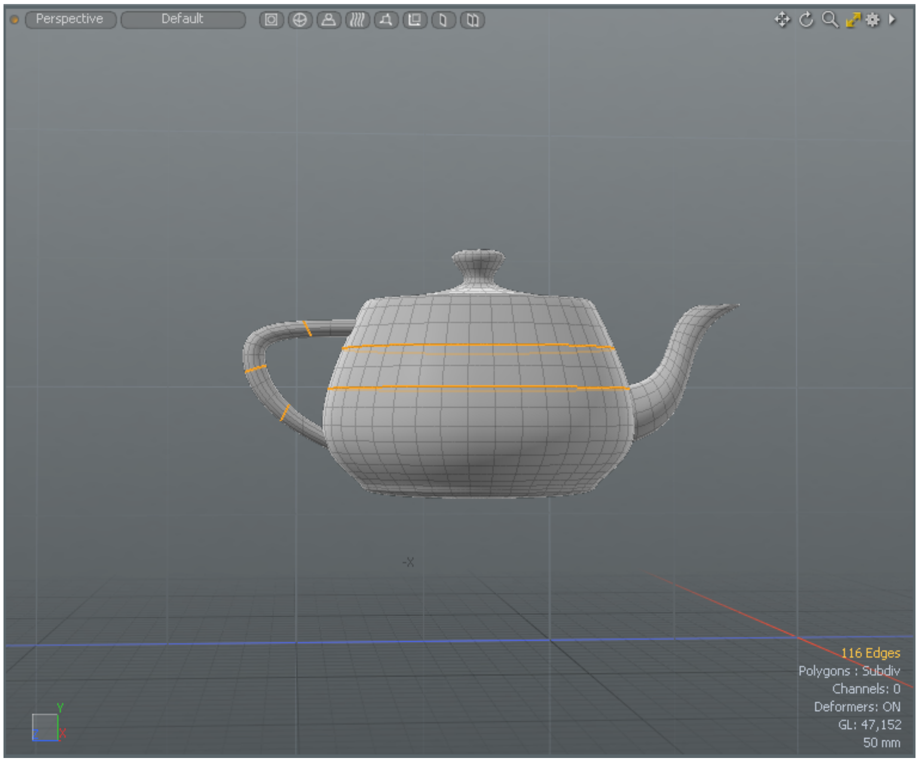
1. Download our example scenes and unpack the files.
2. In **Model** layout, load **assign_edge_split_tool_schematic.lxo**.
3. On the top-left corner of the interface, click the schematic palette icon.



4. On the right panel, drag-and-drop **Teapot** onto the **Schematic** viewport.
5. On the top-right of the interface, click the **Selection Sets** icon, and then click **Quick Select** > **edge-loop-teapot**.

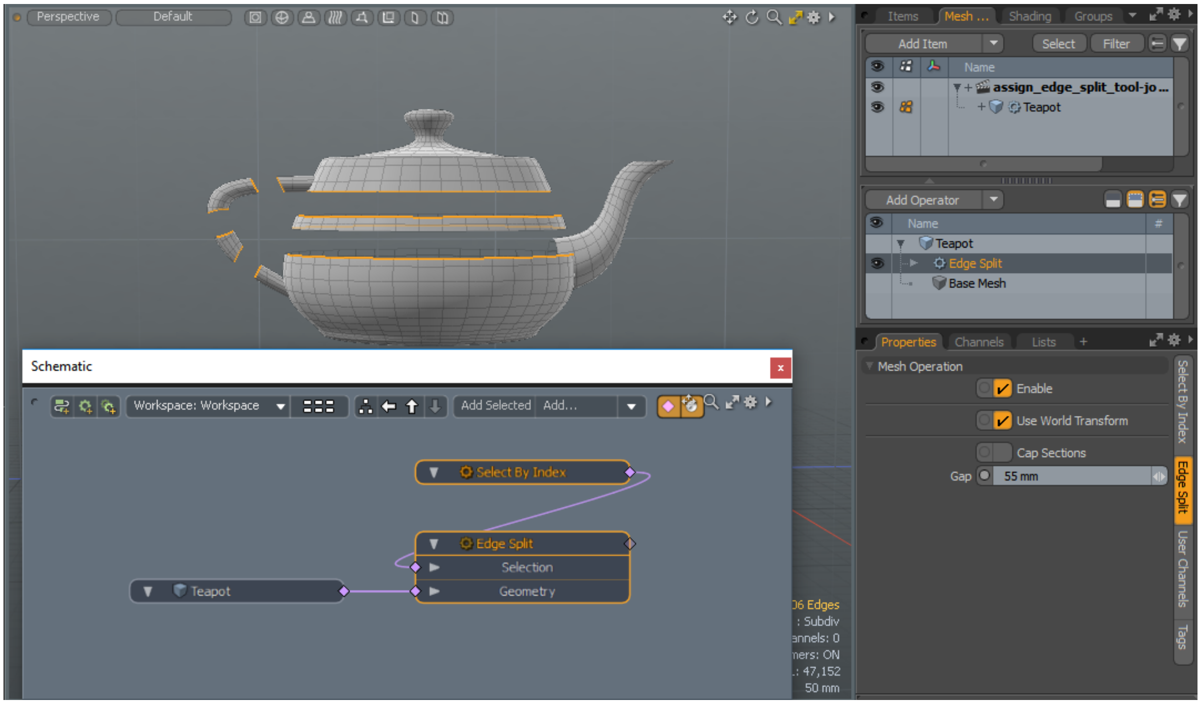

Work Plane
Render F9
Selec
Quick Selec
✓ edge-loop-teapot
Set: edge-loop-teapot
Select
Add to selection
Remove from selection
Replace selection
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group

The edge selection set is displayed in the 3D viewport.


Perspective
Default
116 Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
50 mm

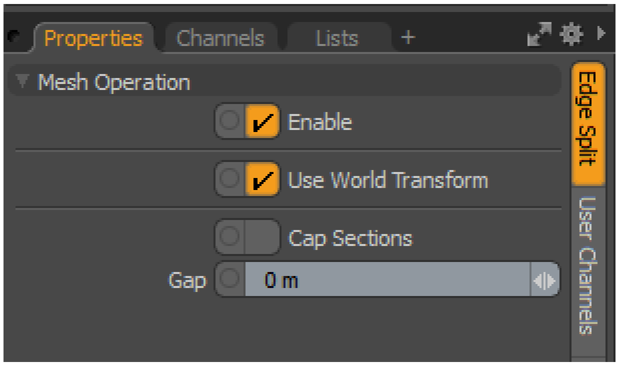
6. In the **Schematic** viewport, click **Add...**, and double-click **Mesh Operations** > **Edges** > **Edge Split**.
7. In the **Schematic** viewport, click in the view area, and then select **Edge Split**.
8. On the right panel, open the **Edge Split** tab, set **Gap** to **55mm** and press **Return**.

   The selection set is split at the edges.


Perspective
Default
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
assign_edge_split_tool-jo...
Teapot
Add Operator
Name
Teapot
Edge Split
Base Mesh
Properties
Channels
Lists
Schematic
Workspace: Workspace
Add Selected
Add...
Select By Index
Edge Split
Selection
Geometry
Teapot
Select By Index
Edge Split
User Channels
Tags

Mesh Operation
☑ Enable
☑ Use World Transform
☐ Cap Sections
Gap 55 mm

Procedural Edge Split Properties



Mesh Operation

☑ Enable

☑ Use World Transform

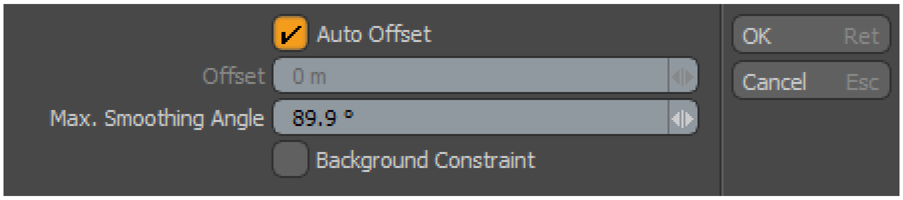
☐ Cap Sections

Gap: 0 m

- **Enable** - Enables or disables the Edge Split tool
- **Use World Transform** - Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
- **Caps Sections**- Creates extremities when the mesh is totally cut in two (or more) parts.
- **Gap**- Specifies the distance to split the mesh at the selected edges.

Grow Quads

The **Grow Quads** command extends a row of quad polygon along a selected edge border.



☑ Auto Offset

Offset: 0 m

Max. Smoothing Angle: 89.9 °

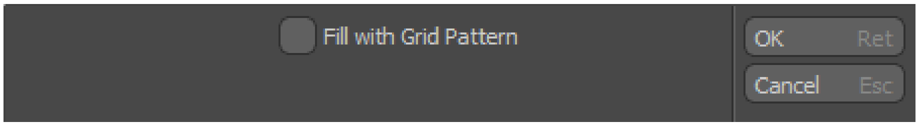
☐ Background Constraint

- **Auto Offset** - Attempts to match preceding row's size, or disable to set a specific offset value.
- **Offset** - Specifies distance of offset.
- **Max. Smoothing Angle** - Specifies maximum smoothing for newly created polygons across angular surfaces.
- **Background Constraint** - When enabled, the tool uses a background mesh to constrain the operation.

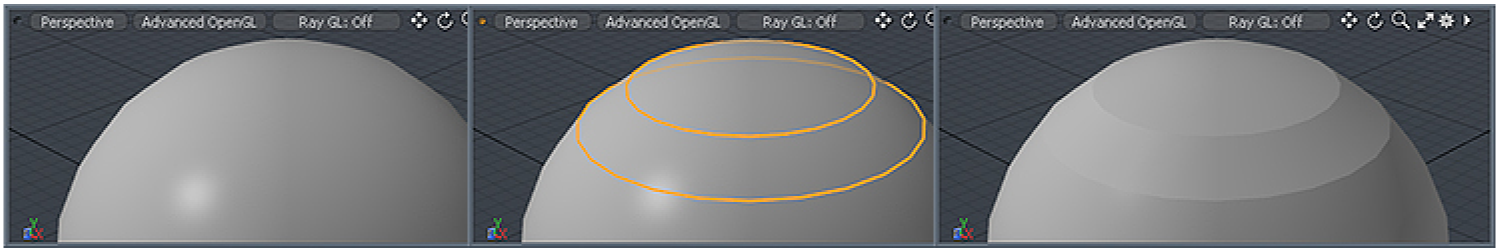
Note: For more information on using Background Constraints, see Constrain to Background.

Fill Quads

The **Fill Quads** command attempts to fill any open holes within a mesh with only quad polygons. You first need to define the area to fill with an edge selection that outlines the target. When the command is invoked the dialog below. There are two possible options, pressing **OK** and accepting the default uses the least number of polygons to fill the area. Enabling the **Fill with Grid Pattern** option fills the target area with an even grid of polygons. However, this option also requires the edge selection to have an even number of edges for both its width and height. This option detects corners and properly fills the area with a grid of quad polygons, even when an entire side of the selection area is left open, such as near an outer edge of a plane.



Split Normals



The initial vertex normal map applied.

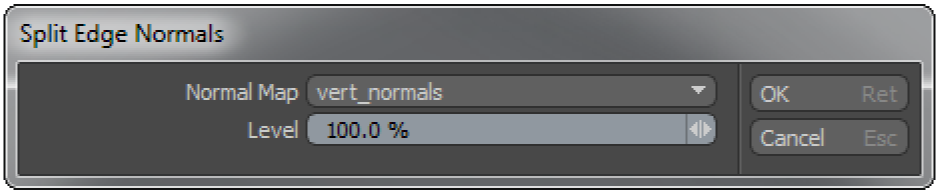
An edge selection defined.

The resulting split vertex normals.

With the Split Vertex Normals command you can modify a vertex normal map to create hard or soft transition edges between polygons based on edge selections. First create a vertex normal map if one isn't already present. You do this by choosing **Vertex Map** > **Set Vertex Normals**. Doing this freezes the smoothing values for the item as it appears in the 3D viewport. Prior to assigning the map, you can adjust the **Smoothing Angle** in the **Material Item** of the **Shader Tree**. Once you assign the map to a surface (by being created), its values supersede those of the **Material Item** and need to be edited manually. The vertex normal map is in the **Other Maps** section of the **Lists** palette.

Usage

With a vertex normal map present, you can select a series of edges meant to define the hard surface transition and then click the **Split Normals** button in the **Edges** sub-tab of the toolbox.


Split Edge Normals

Normal Map vert_normals

Level 100.0 %

You can set the following properties:

- **Normal Map**: Lists the target vertex normal map, which is named **vert_normals** by default.
- **Level**: Determines the amount of hardness added to the edge. **100%** makes a fully hard edge; **0%** does not adjust the edge at all; and **-100%** turns a hard edge into a soft edge.

Edges to Curves

The **Edges to Curves** mesh operation lets you create multiple Spline curves, Polylines, Bezier curves, or B-spline curves from edges on an existing mesh, selected through selection operators.

Using Edges to Curves

Before adding the Edges to Curves mesh operation, add a mesh to your scene. You can do this in the **Mesh Ops** tab, in the **Mesh Operations** list, or add a mesh with the traditional modeling tools.

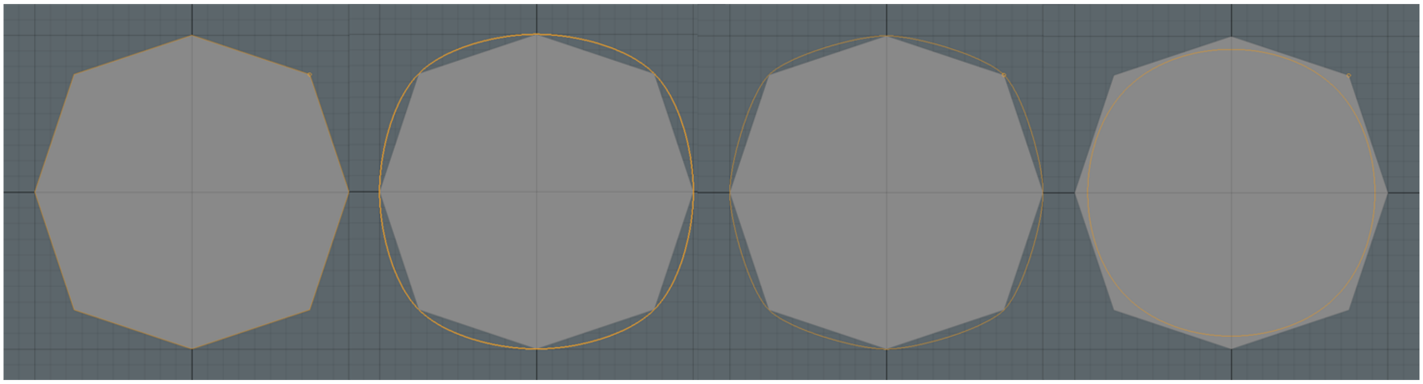
Once you've added a mesh, follow these steps:

1. Under the layout menu, click **Edges**, and select multiple edges on your mesh item.
2. On the right panel, open the **Mesh Ops** tab, click **Add Operator**, and under **Mesh Operations** > **Edge**, double-click **Edges to Curves**.
3. Expand the **Edges to Curves** item in the stack to reveal its **Selection** input.
4. Click **(Add Selection)**, and in the **Preset Browser**, under **New**, double-click **Select by Border**.

   This selects the border edges of your mesh and applies the Edges to Curves operation to them.

Note: Alternatively, you can choose the **Select by Index** selection type, and in **Edges** selection mode, define your selection in the 3D viewport, or edit the **Properties** of the selection operation. For more information on selection types, see Procedural Selection.

The Edges to Curves operation uses a polyline curve type by default, but you can change this in the **Properties** panel. In addition to **Polyline**, you can select **Spline**, **Bezier**, or **B-Spline** curve types:



Polyline Curve Type

Spline Curve Type

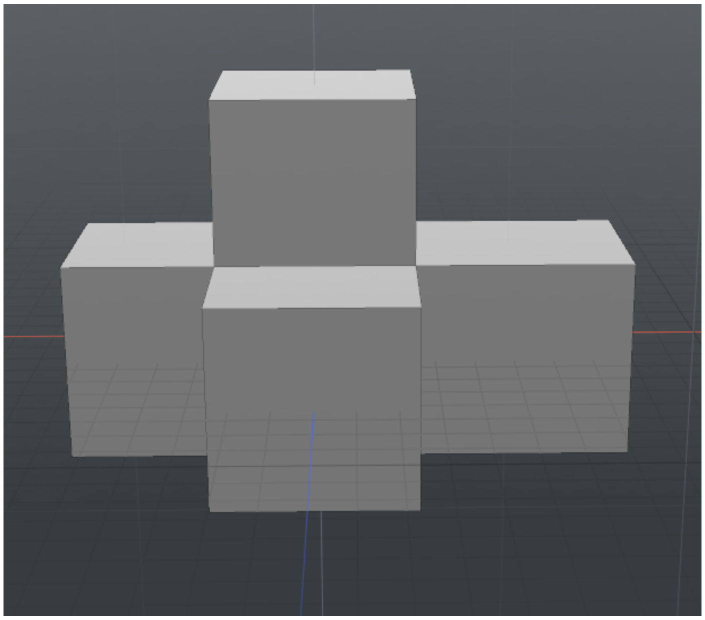
Bezier Curve Type

B-Spline Curve Type

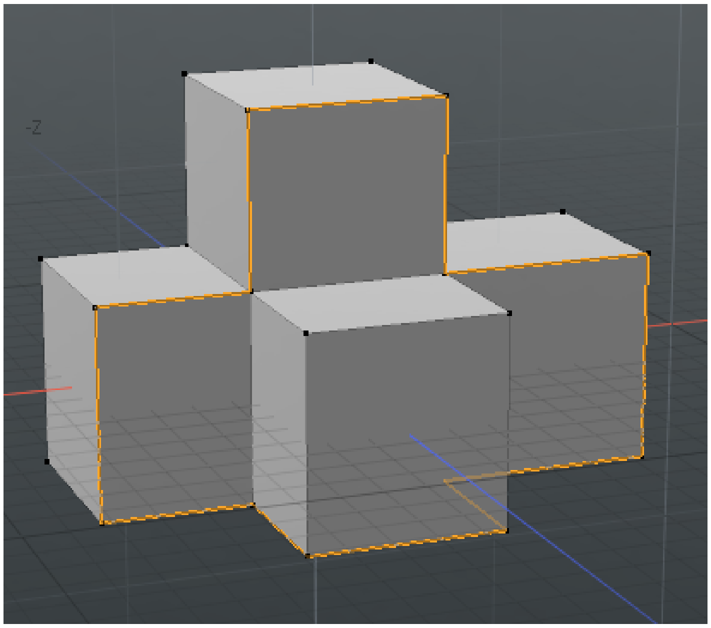
Creating Edges from Multiple Curves

The Edges to Curves tool supports multiple curves. In this simple example, we will walk through the process of applying this tool to a set of edges and rendering the output.

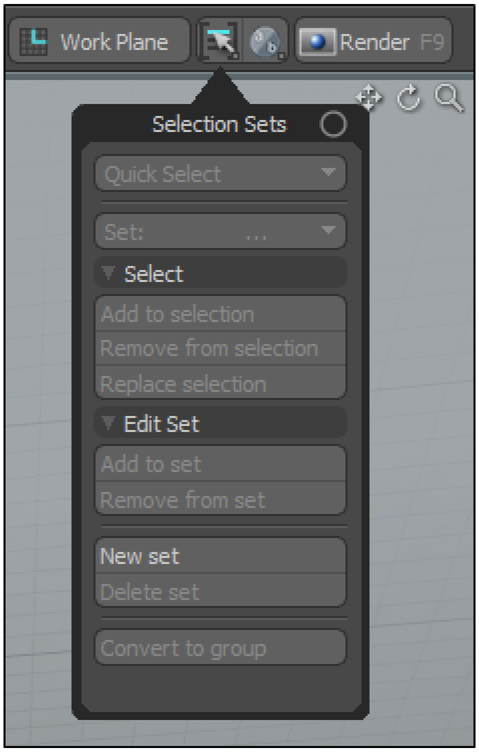
1. Using the **Model** layout, **Ctrl**/**Cmd** + click the cube icon.
2. Under the layout menu, click **Polygons**, and select one polygon on the cube.
3. On the left panel, open the **Mesh Edit** tab, and click **Bevel**.
4. Click-and-drag the bevel handle to extend the polygon outwards.
5. Press the **Spacebar** and repeat the previous step to create a mesh item similar to this one.



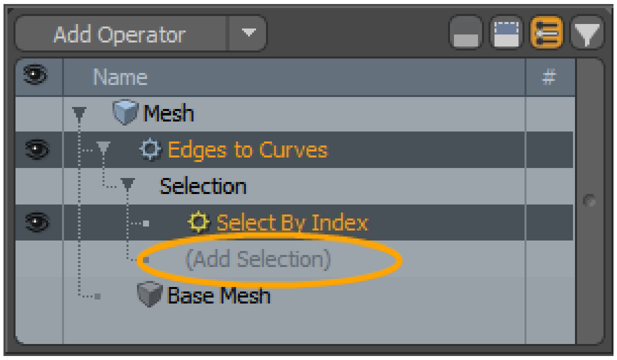
6. Under the layout menu, click **Edges**, hold **Shift** , and select multiple edges on the mesh item.



7. Under the layout menu, click the selection set icon, click **New set**, and name it **edges**.


Work Plane
Render F9
Selection Sets
Quick Select
Set: ...
Select
Add to selection
Remove from selection
Replace selection
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group

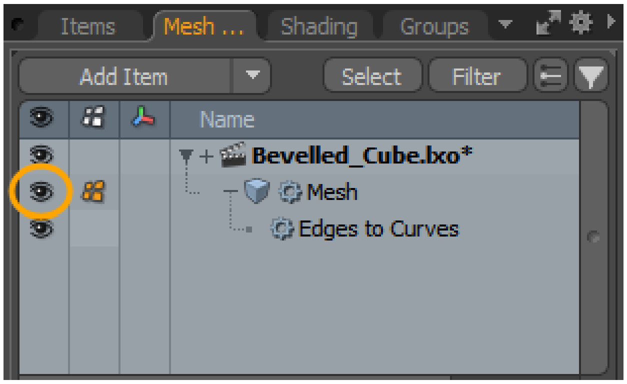
8. On the right panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Edge** > **Edges to Curves**.
9. On the right panel, expand **Edges to Curves**, and click **(Add Selection)**.


Add Operator
Name
#
Mesh
Edges to Curves
Selection
Select By Index
(Add Selection)
Base Mesh

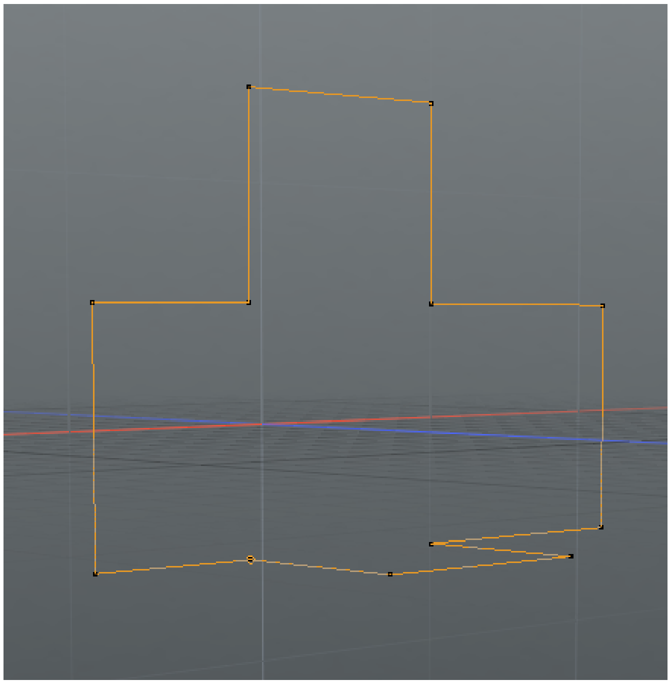
10. Double-click **Mesh Operations** > **Selection** > **Select By Selection Set**.
11. In the bottom right panel, on the **Section By Selection Set** tab, type **edges** into the **Name** field, and press **Enter**.

    Your selected edges are converted into curves.
12. Under the layout menu, click the selection set icon, click **Quick Select**, and select **edges**.
13. Press **Ctrl/Cmd** + **C** to copy the curves.
14. On the **Mesh Ops** tab click **Add Item**, and click **Mesh**.
15. Press **Ctrl/Cmd V** to paste the curves into your new mesh layer.

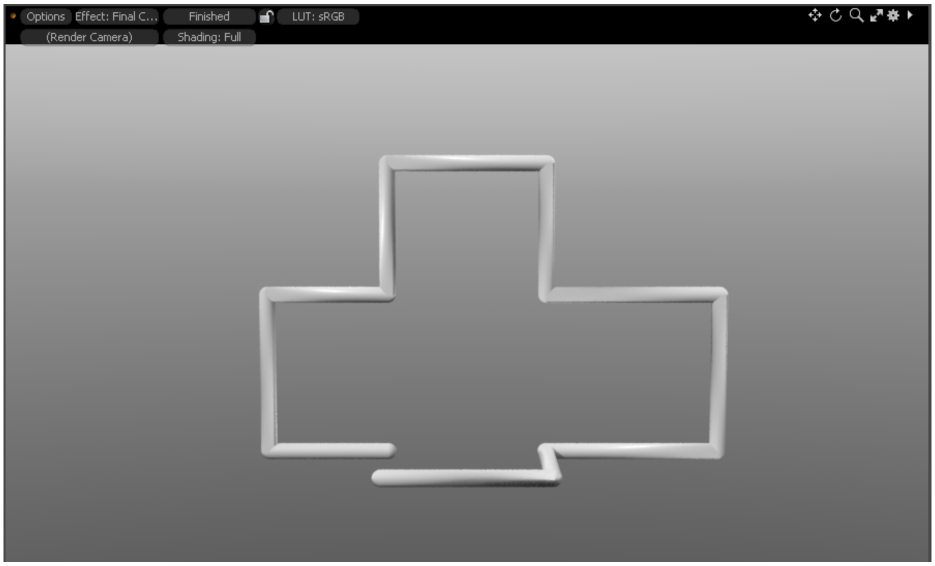
16. On the **Mesh Ops** tab, expand **Mesh** and click the visibility 👁 icon to hide the initial mesh item which contains your deformed cube.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Bevelled_Cube.lxo*
Mesh
Edges to Curves

Tip: Notice that you see a point for every joint position on the curve. You can use these points to reshape your curve. Activate the **Vertices** selection mode, select a vertex, press **W**, and move the handle to reposition.



17. On the right panel, select the mesh item, open the **Curves** tab, and enable **Render Curves**.



Edges To Curves Properties

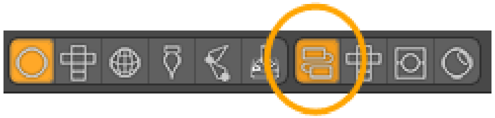
Delete Geometry	Delete all non-curve geometry from the mesh.
Curve Type	Select the type of curve you would like to use. The following options are available: • **Polyline** - This option works well with the Pen Generator tool. • **Spline Curve** - Smoothed curve that works well with the Path Generator tool. Also matches subdivision surfaces more closely. • **Bezier Curve** - Linear curve that works with the Path Generator, but matches cage edges more closely. • **B-Spline Curve** - This type of curve works well with Catmull-Clark subdivision surfaces.
Curve Mode	Select the curve mode. The following options are available: • **Single Curve** - Creates a single curve running through the edges you've selected. This only works with edge selections that have a distinct start and end point, or closed loops. Overlapping edge selections, branching, and multiple edge loops are currently not supported. • **Curve Per Edge** - Creates a two point curve along each selected edge. This has no limitations about what can be selected, but it does not create a continuous curve, so it doesn't work well with things

	like the Sweep Effector or Path Constraints.
Bezier Tangent Weight	This option is only available if the **Curve Type** is set to **Bezier Curve**. In addition to having points for each point of the edges you've selected, a Bezier Curve has two surrounding control points, which define how the curve enters and exits the point. The further the control points are from that middle point where the curve goes through, the more the curve heads in the direction of the control point before it bends back towards the next point in its path. The **Bezier Tangent Weight** option controls how close or far those control points are from the middle point that the curve is passing through. At 0%, you get straight lines between points, and at 100% you get a more rounded curve.

Using Edges to Curves in the Schematic

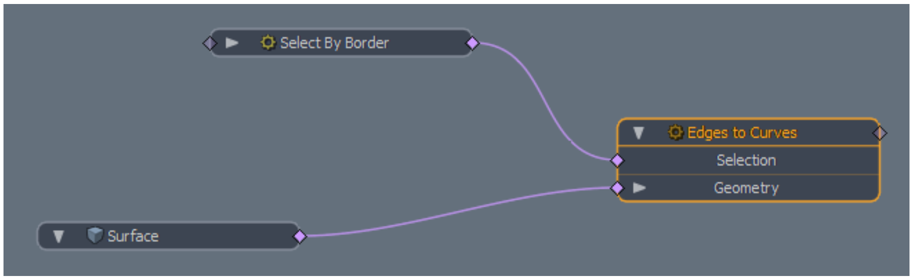
Once you have a mesh in your scene, you can add the Edges to Curves mesh operation to it, using the **Schematic** viewport.

1. On the top left corner of the interface, click the Schematic palette icon.



2. In the **Schematic** viewport, click **Add...**, and double-click **Mesh Operations** > **Edge** > **Edges to Curves**.

   The mesh operation is added to your scene, and automatically gets connected to your mesh item in the **Schematic**.
3. To define the selection for the Edges to Curves operation, click **Add...**, and select **Mesh Operations** > **Selection** > **Select By Border**.
4. Connect the **Select By Border** operation to the **Selection** input of the **Edges to Curves** node.

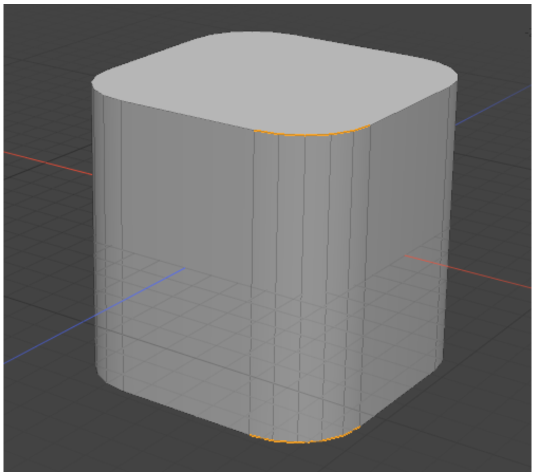


Unbevel

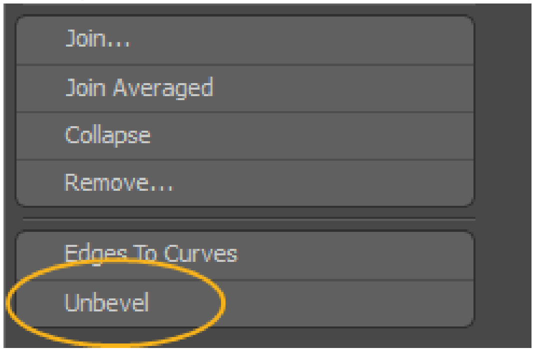
Unbevel collapses beveled faces, created by Edge Bevel or Edge Chamfer, between two selected edge loops.

To use the tool:

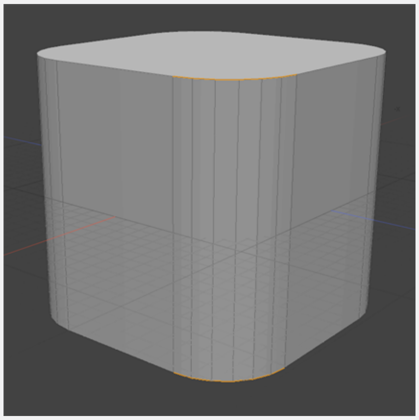
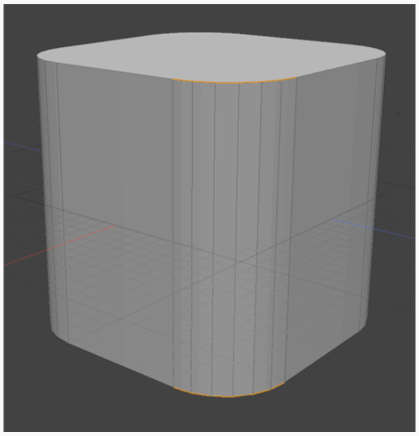
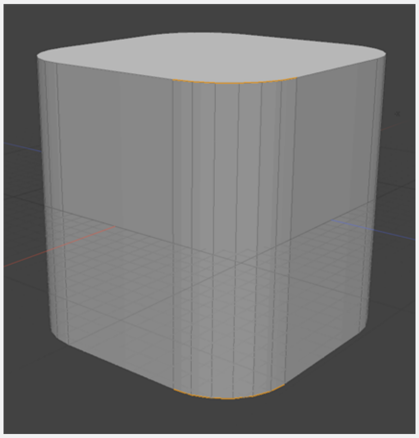
1. In **Edges** selection mode, select the two edge loops that surround the faces you want to unbevel:



2. In the modeling toolbar, click the **Edge** sub-tab, and click **Unbevel**.


Join...
Join Averaged
Collapse
Remove...
Edges To Curves
Unbevel

3. A dialog prompts you to select the **Unbevel** Mode. You have the following three options:

Convergence	Collapses beveled polygons into sharp edges.	
Flat	Turns beveled polygons into a single flat polygon.	
Square	Removes beveled polygons and creates square ones.	

Using Polygon Component Tools

Some of Modo's tools are specific to the various component types: vertices, edges, and polygons. For example, there are three component beveling tools, and each has its own settings. The Edge Bevel tool has a rounding option, but the Polygon Bevel tool does not. Some tools are interactive tools, but others are one-shot commands that produce a specific result. You may use some regularly and others only rarely; however, all could be considered indispensable. For information on using a specific tool, click any link below to see its topic.

Polygons		
	Detriangulate	Converts triangle pairs to quadrangles by deleting edges to share triangle pairs.
	Triangulate	Converts polygons to triangles.
	Polygon Bevel	Performs a shift and inset on the selected polygons.
	Thicken	Adds dimension to flat geometry.
	Triple	Subdivides any selected polygon with more than 3 vertices so that it consists only of triangle polygons.
	Align	Makes all polygons face the same direction.
	Flip Polygon	Inverts the normal of surface polygons.
	Polygon Extrude	Repositions the selected polygon in space based on the tool's three handles.
	Set Polygon Type	Converts polygon type to Face, Subdivision and Catmull Clark.
	Spikey	Takes selected polygons and adds a vertex to the center of the polygon.

Polygons		
	Split	Draws an edge between two selected vertices of a polygon.
	Merge	Combines selected polygons into a single polygon of n-number of sides.
	Smooth Shift	Smooths out sharp edges in polygonal geometry based on a smoothing angle.
	Inset	Subdivides selected polygons to create new geometry within the original selection.
	Spin Quads	Changes where your edges are attached within the geometry.
	Collapse	Removes the selected element without destroying the integrity of the geometry.
	Shift	Extrudes selected polygons along their individual normal axes.
	Polygon Reduce	Iteratively removes edges from the mesh, while trying to maintain as much of the original contouring as possible.
	Convert to Quadrangles	Converts a series of vertices from an N-gon into a row of regular four sided (quadrangle) polygons.
	Remove	Deletes selected polygons.
	Flip	Reverses the direction of selected polygons.
	PolyScale Offset	Scales a selected polygon on a mesh item.
	PolyScale Offset Detach	Scales a selected polygon on a mesh item and detaches it.
	Flatten Ngons	Flattens Ngons in a model that are not planar.

Polygons		
🔧	Triple Fan	Splits a selected polygon on a selected vertex point.
🔧	Flip It	Flips a selected polygon on a specified axis.
🔧	Unify	Merges two polygons into a single polygon when you have two polygons that are in exactly the same space, and the polygons share the same vertices.

Detriangulate

The **Detriangulate** tool converts triangle pairs to quadrangles by deleting edges to share triangle pairs. This conversion tool evaluates the flatness of the geometry.

Tip: rted from other software products. For more information about file formats, see Importing and Exporting File Formats.

This tools works as procedural modeling operator and direct modeling tool.

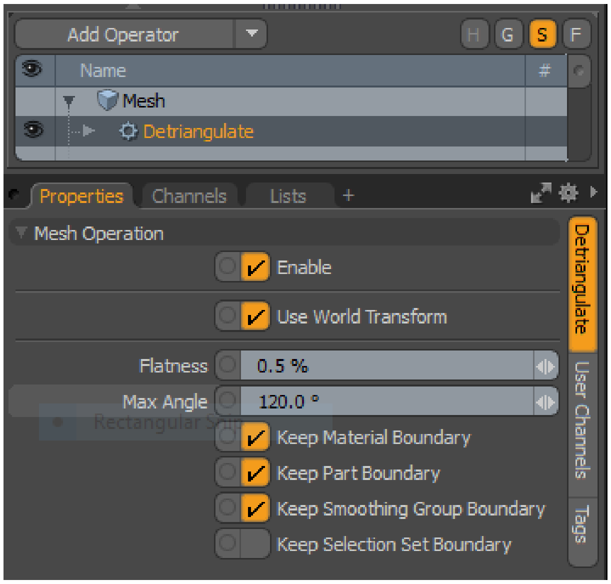
- For Direct Modeling, the **Detriangulate** tool is found under the **Geometry** > **Polygon** menu. To enable this tool, first apply the Automatic Retopology Tool.
- For Procedural Modeling, the **Detriangulate** tool is found under the **Mesh Ops** tab. Click **Add Operator** and select **Mesh Operations** > **Polygons** > **Detriangulate**.

Applying the Detriangulate Tool

1. Open the **Topology** layout and select a mesh in your scene.
2. In the **Toolbox**, open the **Tools** sub-tab, click **Automatic Retopology** and apply the default setting.

   For detailed information about the **Automatic Retopology** tool, see Automatic Retopology Tool.
3. In the **Mesh Ops** tab, click **Add Operator** and select **Mesh Operations** > **Polygons** > **Detriangulate**.
4. Set the properties for **Detriangulate**.

Detriangulate Properties



☑ Enable

☑ Use World Transform

Flatness 0.5 %

Max Angle 120.0 °

☑ Keep Material Boundary

☑ Keep Part Boundary

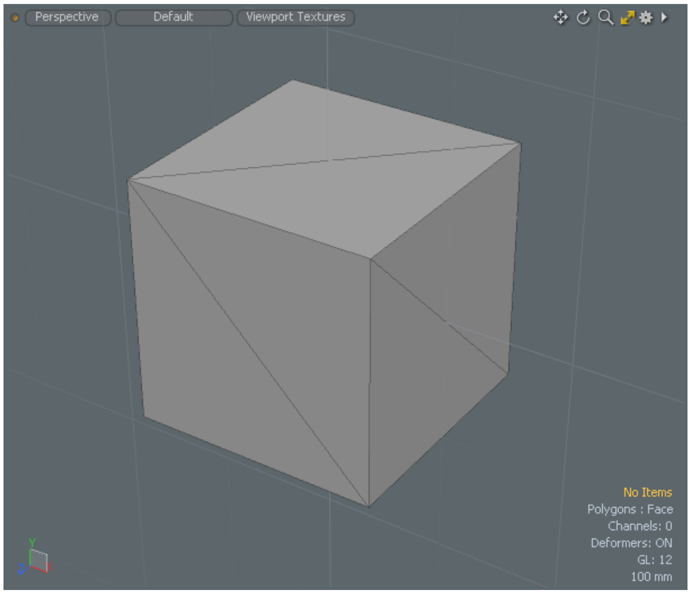
☑ Keep Smoothing Group Boundary

☐ Keep Selection Set Boundary

Flatness	Set the percent of the coplanar flatness threshold between opposing triangle pairs, designating whether the edge is selected or not.
Max Angle	Set the maximum angle for the corners of a quadrangle from the triangle pair.
Keep Material Boundary	When enabled, edges that fall along the material boundary are not.
Keep Part Boundary	When enabled, edges that fall along a part boundary are not selected.
Keep Smoothing Group Boundary	When enabled, edges that fall along a smoothing group boundary are not selected.
Keep Selection Set Boundary	When enabled, edges that fall along a selection set boundary are not selected.

Triangulate

The **Triangulate** Mesh Operations tool converts polygons into triangle pairs. You can use this tool to triangulated polygon models and export them to other software products. For more information about file formats, see Importing and Exporting File Formats.



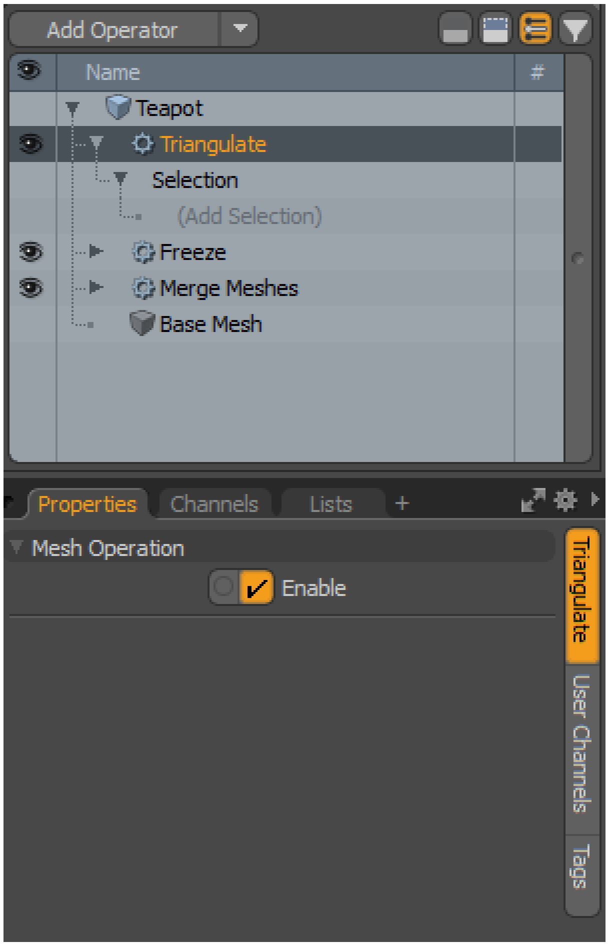
Tip: When working with complex models with multiple layers, apply Merge Meshes to create a single procedural mesh item.
In addition, you can apply a **Freeze** mesh operation to rasterizes non-face type polygons into face polygons or specific shapes. For more information, see Freeze for Procedural Modeling.

The **Triangulate** tool is found under the **Mesh Ops** tab. Click **Add Operator**, and select **Mesh Operations** > **Polygons** > **Triangulate**.

Applying the Triangulate tool

1. Open the **Model** layout and select a mesh in your scene.
2. In the right panel, open the **Mesh Ops** tab.
3. Optional - Click **Add Operator** and select **Mesh Operations** > **Edit** > **Merge Meshes**.
4. Optional - Click **Add Operator** and select **Mesh Operations** > **Edit** > **Freeze**.
5. Click **Add Operator** and select **Mesh Operations** > **Polygon** > **Triangulate**.
6. Click **File** > **Export**, specify the name and format, and click **Save**.

Triangulate Properties


Add Operator
Name
#
Teapot
Triangulate
Selection
(Add Selection)
Freeze
Merge Meshes
Base Mesh
Properties
Channels
Lists
+
Mesh Operation
Enable
Triangulate
User Channels
Tags

Triangulate	
Enable	Toggles triangulated polygons on and off.

Polygon Bevel

The **Bevel** tool performs a shift and inset on the selected polygons. **Shift** pushes the polygons away from their starting positions, and **Inset** scales the polygons. New polygons are created to bridge the gaps that would be left by moving the original polygons from their starting positions.

For direct modeling, the **Bevel** tool is available in the modeling Toolbox. Open the **Mesh Edit** tab and click **Bevel**. The **Bevel** tool is also available as a procedural in the **Mesh Operations** list. Open the **Mesh Ops** tab, click **Add Operator**, and select **Mesh Operations** > **Polygon** > **Polygon Bevel**.

Polygon Bevel in Direct Modeling

Modo has a direct modeling **Polygon Bevel** tool available in the **Toolbox** while working in the **Model** layout. The following is a simple workflow example using one of the **Polygon Bevel** options to demonstrate the outcome. For more information about the properties available, see Polygon Bevel Options.

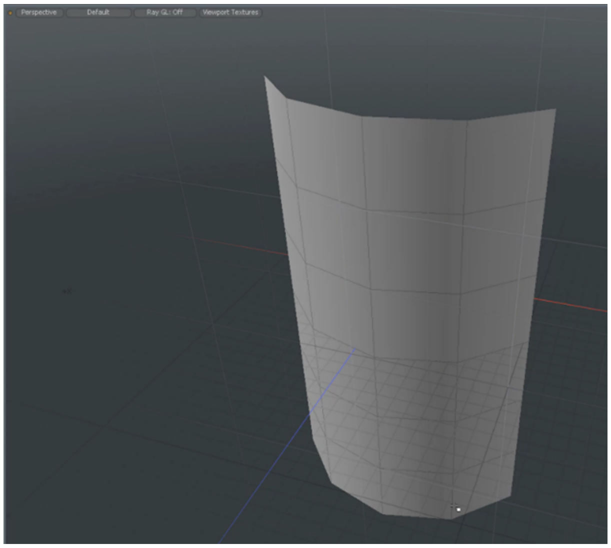
Creating Square Corners

The Polygon Bevel Square Corner option can be easily applied to create a thicken surface on your model surface without having to apply edge weights or add edge loops to the corners. In the following example, we are using this option to quickly create a label for a bottle.

To create square corners and thicken the surface:

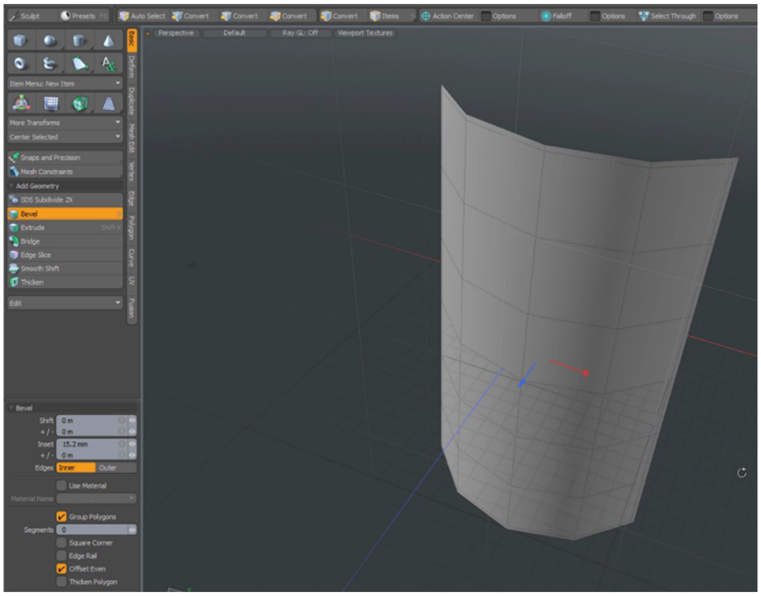
1. Create the geometry for the bottle label.

> **Note:** You can download our example scenes and check out **toolpanel_polybevel_bottle_label.lxo**. In this example, we've used the **Cylinder** primitive and removed the top, bottom, and side polygons using the Polygon selection mode.


Perspective
Default
Ray GL: Off
Viewport Textures

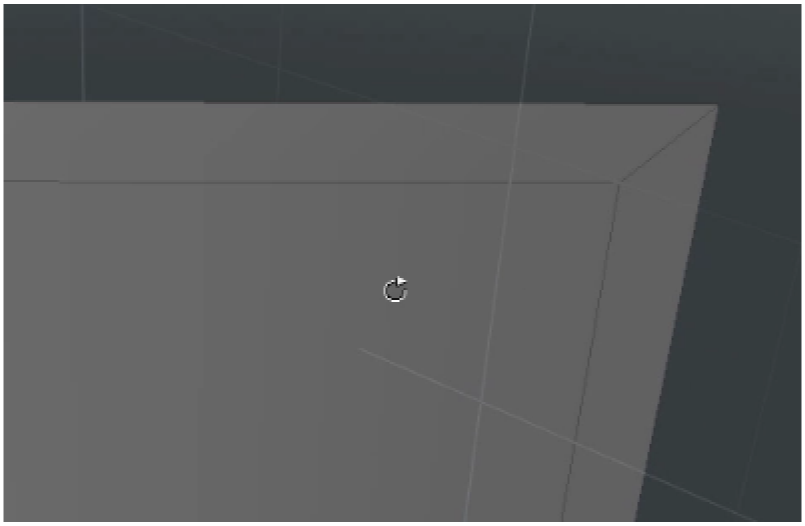
2. In the Toolbox, on the left panel, open the **Basic** tab and select **Bevel**.

   The Bevel handles display in the 3D Viewport.
3. Adjust the **Inset** value by either entering in a value in the Toolbox Bevel options or click-and-drag the handle to set the value. In this example we have set the value to **15.2 mm**.


Sculpt
Presets
Auto Select
Convert
Items
Action Center
Options
Falloff
Select Through
Perspective
Default
Ray GL: Off
Viewport Textures
Item Menu: New Item
More Transforms
Center Selected
Snaps and Precision
Mesh Constraints
Add Geometry
SDS Subdivide 2X
Bevel
Extrude
Bridge
Edge Slice
Smooth Shift
Thicken
Edit
Basic
Deform
Duplicate
Mesh Edit
Vertex
Edge
Polygon
Curve
UV
Fusion

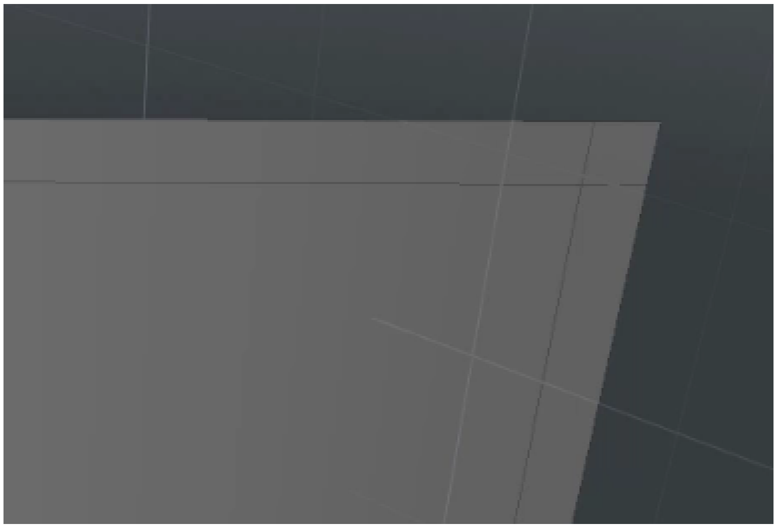
Bevel
Shift 0 m
+/- 0 m
Inset 15.2 mm
+/- 0 m
Edges Inner Outer
☐ Use Material
Material Name
☑ Group Polygons
Segments 0
☐ Square Corner
☐ Edge Rail
☑ Offset Even
☐ Thicken Polygon

An extra edge is created on the corner of the surface. By default, it creates an extra edge along the outer rim that runs to the corner. This creates a loop of polygons that goes around the entire surface.



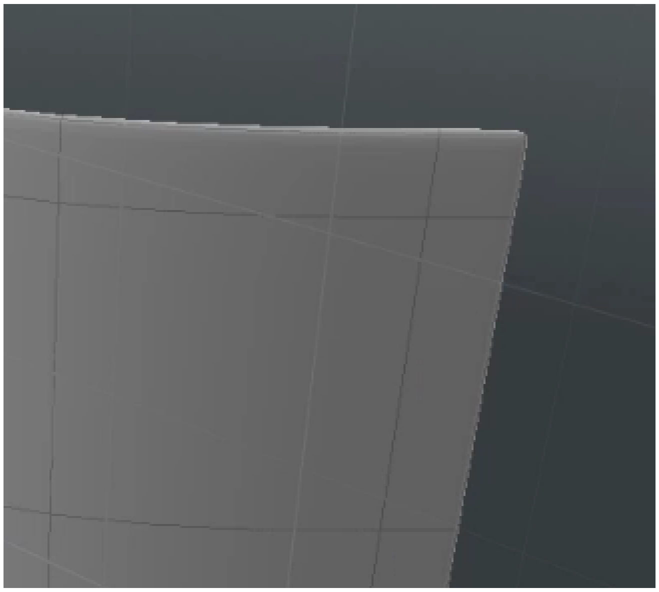
4. In the Toolbox on the left panel, enable the **Square Corner** option.

   The corner edge is modified to produce a square edge.



5. In the Toolbox on the left panel, open the **Basic** tab, and click **Thicken**.
6. Set the **Offset** value to **7 mm**.
7. Press **Shift**+ **Tab**.

   The corners are rounded and the surface is thicken.



Procedural Polygon Bevel

Modo has a procedural Polygon Bevel mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the list in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Polygon**, double-click **Polygon Bevel**.

You can apply the bevel to a selection by making the selection before adding the bevel operation. This adds a **Select by Index** to the **Selection** input of the bevel. Once added, the selection can be edited using the **Properties** panel for the **Select by Index**. For more information, see Procedural Selection.

You can expand the Polygon Bevel item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses. Polygon Bevel requires the following inputs:

- **Selection** - Allows you to select the polygons you want to bevel.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

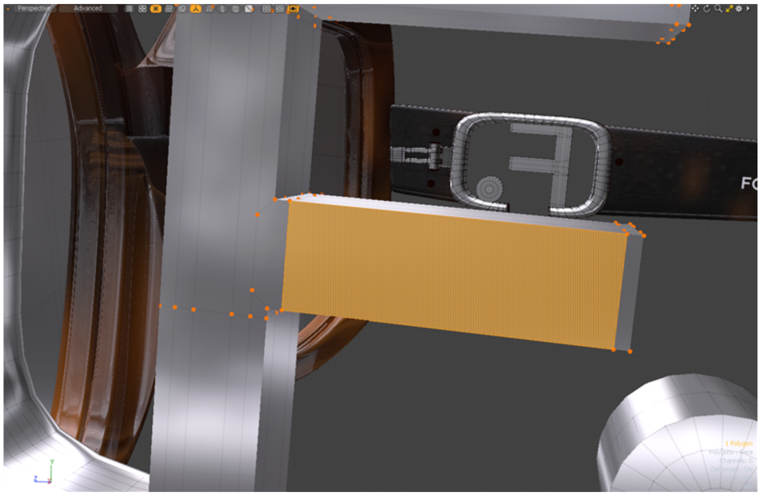
Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Using Auto Weld to Manage Overlap

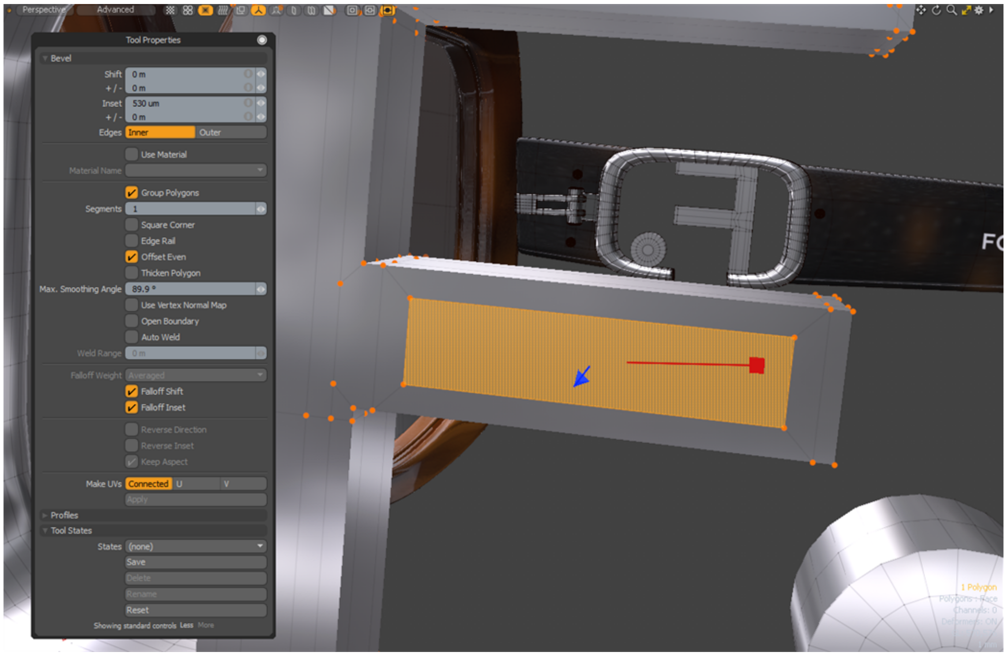
Performing a Polygon Bevel may sometimes cause the inner vertices to intersect, which results in overlapping geometry. It is good practice when modeling to avoid overlapping geometry as it can cause issues when rendering, or if you were to later animate or deform the geometry.

The **Auto Weld** option can be used to join the inner vertices and avoid the overlap.

1. Select the polygon you want to apply the Polygon Bevel to.



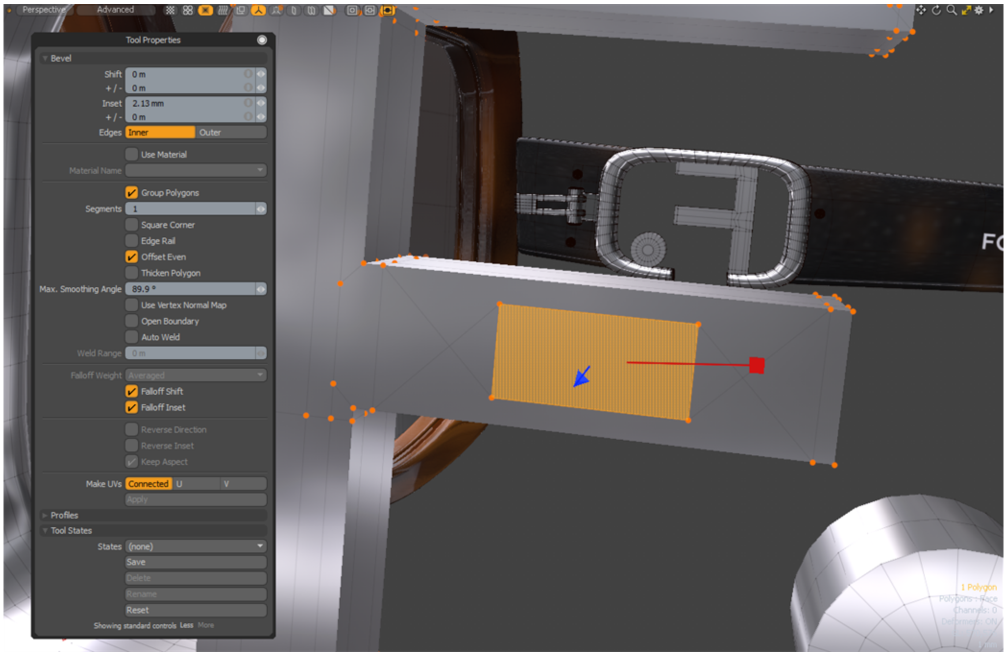
2. Click the **Bevel** option from the **Model** toolbar in the **Polygon** tab.
   Press **K** on the keyboard to open the **Tool Properties** window.
3. Drag the red handle or increase the **Inset** value in the **Tool Properties** to apply the bevel effect to the selected polygon.
   Increasing the **Inset** value too much may result in intersecting vertices.


Perspective
Advanced
Tool Properties
Bevel

Shift: 0 m
+ / -: 0 m
Inset: 530 um
+ / -: 0 m
Edges: Inner | Outer
☐ Use Material
Material Name:
☑ Group Polygons
Segments: 1
☐ Square Corner
☐ Edge Rail
☑ Offset Even
☐ Thicken Polygon
Max. Smoothing Angle: 89.9 °
☐ Use Vertex Normal Map
☐ Open Boundary
☐ Auto Weld
Weld Range: 0 m
Falloff Weight: Averaged
☑ Falloff Shift
☑ Falloff Inset
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
Make UVs: Connected | U | V

Apply
Profiles
Tool States
States (none)
Save
Delete
Rename
Reset
Showing standard controls Less More

No intersecting vertices

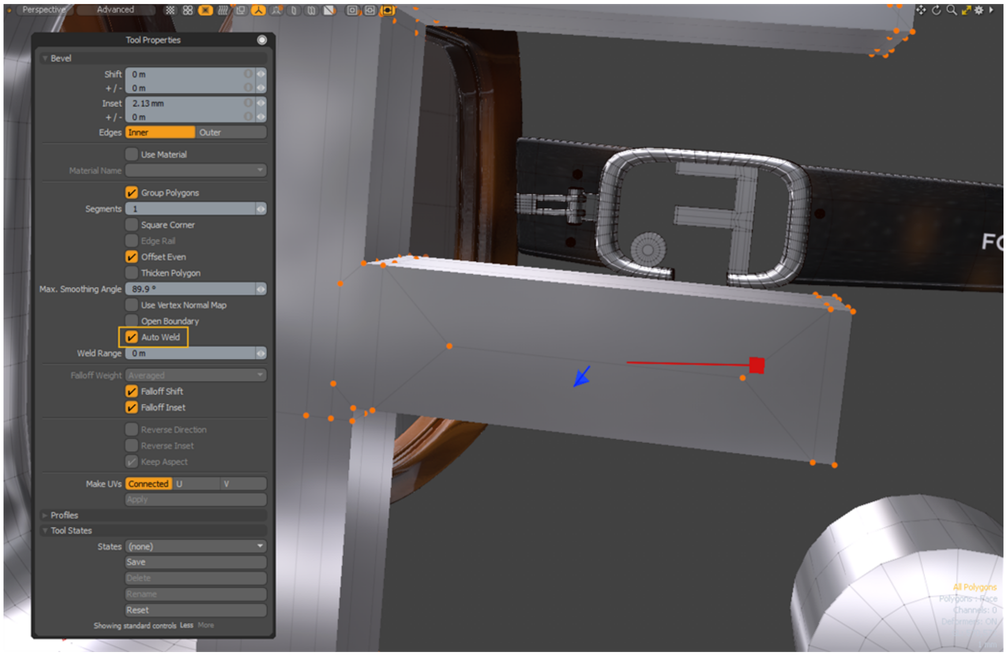

Perspective
Advanced
Tool Properties
Bevel

Shift: 0 m
+ / -: 0 m
Inset: 2.13 mm
+ / -: 0 m
Edges: Inner | Outer
☐ Use Material
Material Name:
☑ Group Polygons
Segments: 1
☐ Square Corner
☐ Edge Rail
☑ Offset Even
☐ Thicken Polygon
Max. Smoothing Angle: 89.9 °
☐ Use Vertex Normal Map
☐ Open Boundary
☐ Auto Weld
Weld Range: 0 m
Falloff Weight: Averaged
☑ Falloff Shift
☑ Falloff Inset
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
Make UVs: Connected | U | V

Apply
Profiles
Tool States
States (none)
Save
Delete
Rename
Reset
Showing standard controls Less More

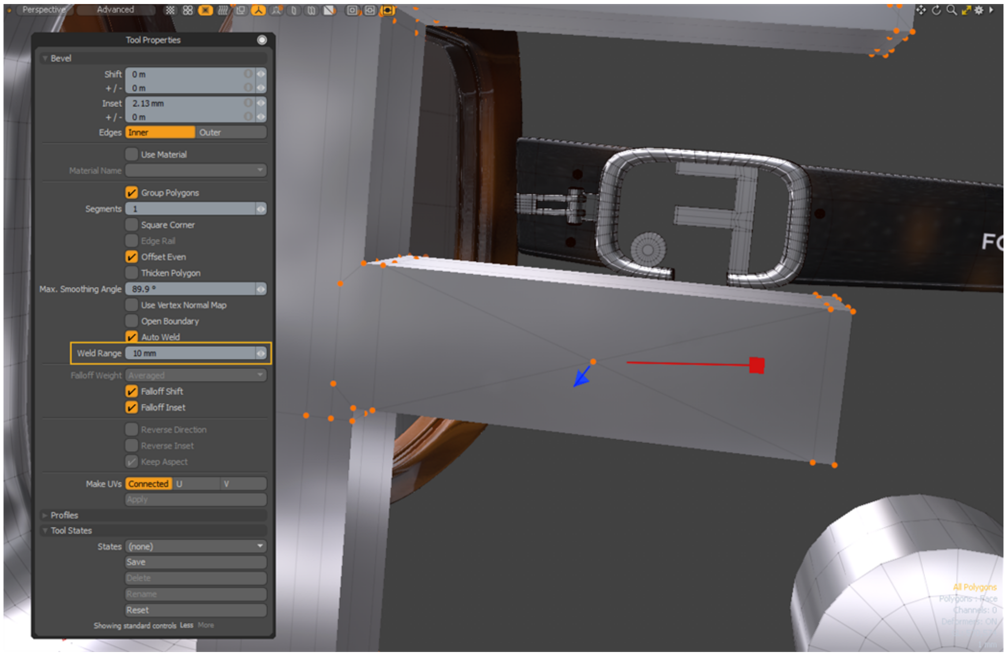
Intersecting vertices

4. Enable **Auto Weld** in the **Tool Properties** to weld the vertices together.


Perspective
Advanced
Tool Properties
Bevel
Profiles
Tool States
Apply
Save
Delete
Rename
Reset
Showing standard controls Less More

Shift 0 m
+ / - 0 m
Inset 2.13 mm
+ / - 0 m
Edges [✓ Inner] Outer
☐ Use Material
Material Name
☑ Group Polygons
Segments 1
☐ Square Corner
☐ Edge Rail
☑ Offset Even
☐ Thicken Polygon
Max. Smoothing Angle 89.9 °
☐ Use Vertex Normal Map
☐ Open Boundary
☑ Auto Weld
Weld Range 0 m
Falloff Weight Averaged
☑ Falloff Shift
☑ Falloff Inset
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
Make UVs [✓ Connected] U V
States (none)

5. Increase or decrease the **Weld Range** value in the **Tool Properties** to adjust the range that the **Auto Weld** function is applied to.


Perspective
Advanced
Tool Properties
Bevel
Profiles
Tool States
Apply
Save
Delete
Rename
Reset
Showing standard controls Less More

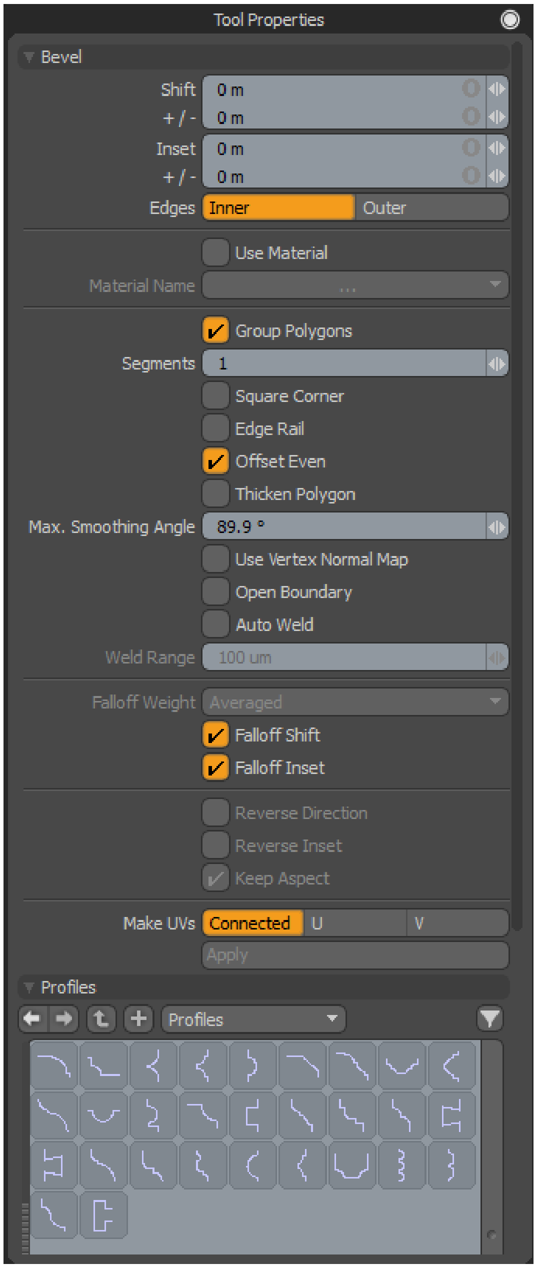
Shift 0 m
+ / - 0 m
Inset 2.13 mm
+ / - 0 m
Edges [✓ Inner] Outer
☐ Use Material
Material Name
☑ Group Polygons
Segments 1
☐ Square Corner
☐ Edge Rail
☑ Offset Even
☐ Thicken Polygon
Max. Smoothing Angle 89.9 °
☐ Use Vertex Normal Map
☐ Open Boundary
☑ Auto Weld
Weld Range 10 mm
Falloff Weight Averaged
☑ Falloff Shift
☑ Falloff Inset
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
Make UVs [✓ Connected] U V
States (none)

Weld Range increased to 10mm

Quickclip...

Polygon Bevel Options

The following options are available:



Tool Properties

Bevel

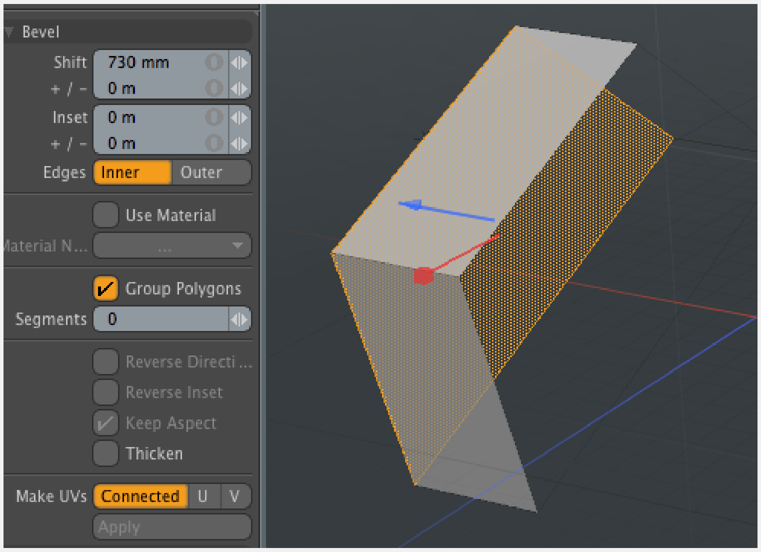
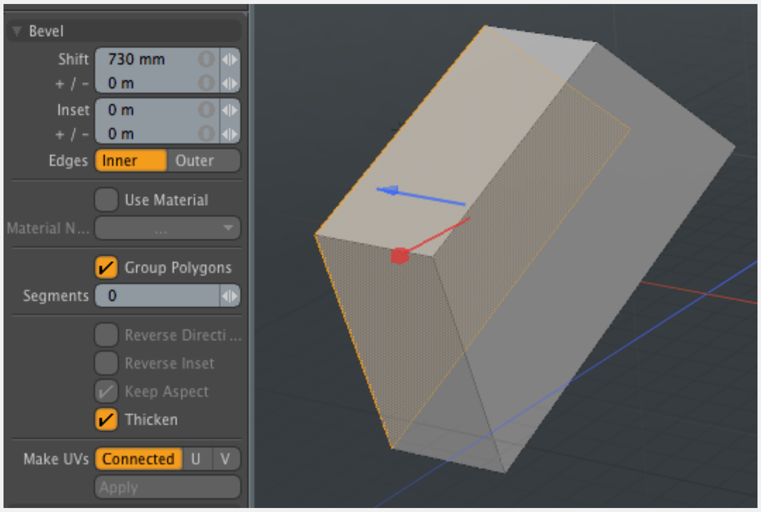
Shift 0 m
+ / - 0 m
Inset 0 m
+ / - 0 m
Edges Inner Outer
☐ Use Material
Material Name ...
☑ Group Polygons
Segments 1
☐ Square Corner
☐ Edge Rail
☑ Offset Even
☐ Thicken Polygon
Max. Smoothing Angle 89.9 °
☐ Use Vertex Normal Map
☐ Open Boundary
☐ Auto Weld
Weld Range 100 um
Falloff Weight Averaged
☑ Falloff Shift
☑ Falloff Inset
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
Make UVs Connected U V
Apply

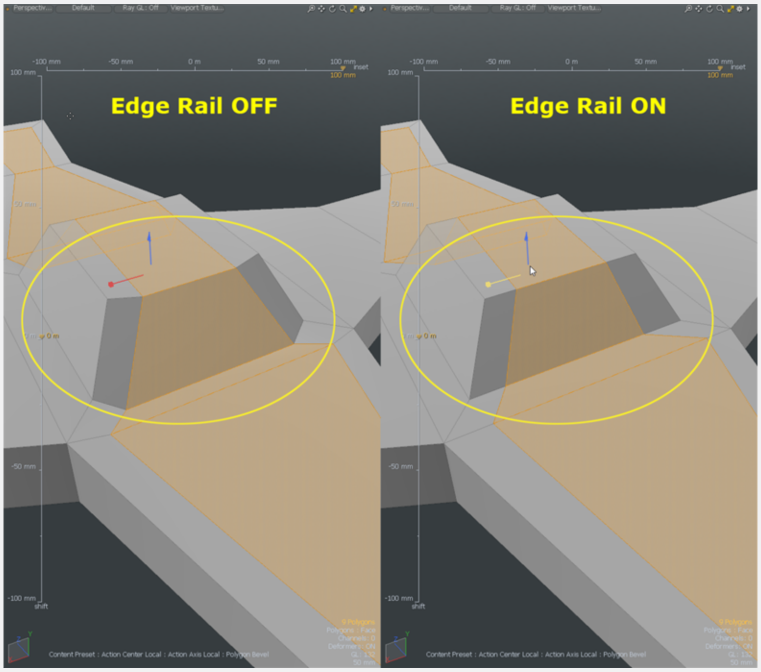
Profiles

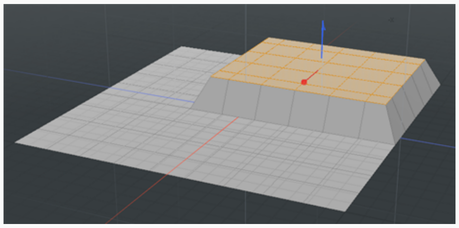
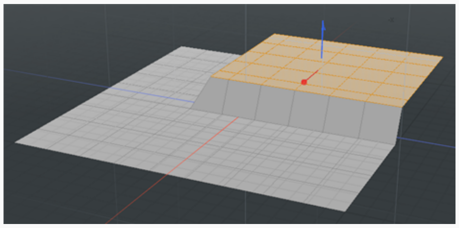
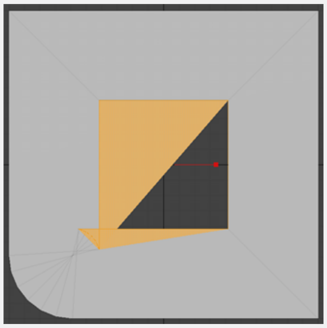
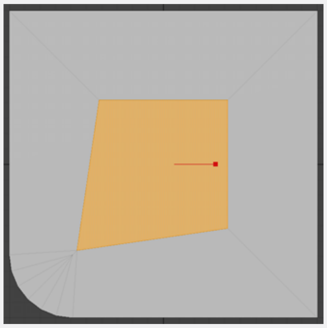
Profiles

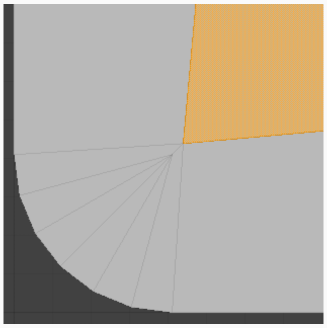
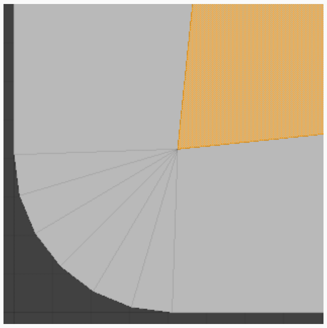
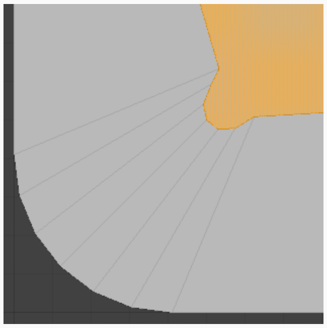
Bevel	
Enable	Enables or disables the Polygon Bevel tool.

Bevel	
Use World Transform	Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Shift/Inset	Both **Shift** and **Inset** accept positive or negative values. When the values are positive **Shift** and **Inset** work as described above. When the values are negative, **Shift** moves the original polygons back behind their original location and **Inset** scales the polygons to be larger than their original size. Each of the **Shift** and **Inset** values can be edited by hauling in the 3D viewport, dragging tool handles, or by using the HUD sliders (HUD sliders require advanced handle viewing to be active). You can also edit these values numerically on the tool properties sheet.
+/-	Under **Shift** and **Inset** this field allows you to set a random value range for the **Shift** and **Inset** value. If you have a **Shift** value of 1 meter and a +/1 value of .5 meters, the resulting bevel(s) are between .5 and 1.5 meters. This can be useful when beveling many polygons simultaneously.
Edges (Inner/Outer)	This toggle option determines whether the resulting polygons are created in such a way as to allow them to fall inside the original polygons boundaries, or so that they are outside the original polygon. The default, **Inner**, is the preferred choice for most bevel scenarios. The **Outer** option is very useful when beveling polygons that are not attached to other polygons. The classic example of using the **Outer** option is for beveling text. In this case you may want the original text polygons to maintain their scale, but would like to bevel an edge outside the original polygons.
Use Material	This option also enables the **Material Name** field. When this option is active, any new polygons created by the bevel tool have this **Material Name** assigned to them.
Material Name	When Use Material is enabled, this option is then enabled, allowing you to select which material in your scene to use.
Group Polygons	Provides a quick method for beveling contiguous polygon selections essentially treating the entire selection as a single polygon. This is incredibly useful for adding fine fillets around selections by creating a series of bevels.
Thicken Polygon	Adds flipped copies of the source polygons to add thickness to the operation.

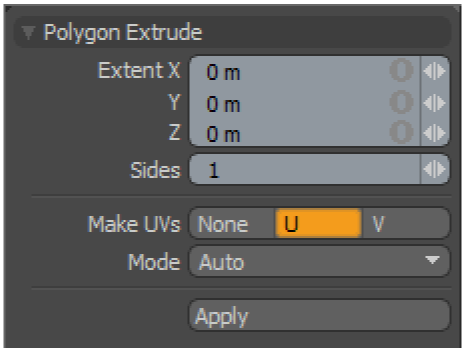
Bevel	
	 Bevel Shift 730 mm + / - 0 m Inset 0 m + / - 0 m Edges Inner Outer ☐ Use Material Material N... ... ☑ Group Polygons Segments 0 ☐ Reverse Directi... ☐ Reverse Inset ☑ Keep Aspect ☐ Thicken Make UVs Connected U V Apply  Bevel Shift 730 mm + / - 0 m Inset 0 m + / - 0 m Edges Inner Outer ☐ Use Material Material N... ... ☑ Group Polygons Segments 0 ☐ Reverse Directi... ☐ Reverse Inset ☑ Keep Aspect ☑ Thicken Make UVs Connected U V Apply
Segments	Specifies the number of edge segment along the bevels shift direction. If it is **0** and a 1D profile is selected, the bevel tool automatically decides the best adaptive segments for the profile.
Square Corner	Adds quadrangles at each sharp corner of the boundary polygons. When Group Polygons is enabled, squares are created at the corner vertices connected to the selected polygon(s).
Offset Even	Makes the vertex distance of the inserted face even. This option is enabled by default. When this option is disabled, the edge length between source vertex to the new source vertex is the same as set **Inset** value. See Shift/Inset.

Bevel	
Edge Rail	This option works with the Group Polygons option. When this option is enabled, the new vertex from the source vertex is created along an inner edge, which is connected to the contour source vertex. If there are multiple inner edges, the most perpendicular to the contour edge is selected. 
Max Smoothing Angle	Shifts vertices along the averaged vertex normal vectors computed from connecting polygon face normals. When the angle between faces is less than the **Max Smoothing Angle**, the averaged normals are computed per smoothing face group and then combine the separated normal vectors as the shift vector. This is useful when beveling polygon groups with very sharp edges.
Use Vertex Normal Map	The group Polygon Bevel uses vectors in vertex normal map for the shift vector when **Use Vertex Normal Map** is enabled and the mesh has vertex normal map.
Open Boundary	When enabled, boundary edges are not inset.

Bevel	
	 **Open Boundary** disabled  **Open Boundary** Enabled
Auto Weld	When enabled, **Auto Weld** joins overlapping vertices to make a flat polygon without overlapping edges. Specify the range in which vertices are welded in the **Weld Range** field. This option is available for both direct and procedural versions of the Polygon Bevel tool.  **Auto Weld** disabled  **Auto Weld** enabled **Note: Auto Weld** computes the splines on flat 2D space. This option does not work when selected polygons with Group Polygons are not on planar plane.
Weld Range	Defines the range to weld continuous beveling points. This allows you to weld closed points in the specified range.

Bevel	
	 **Weld Range** set to 0  **Weld Range** set to 200mm  **Weld Range** set to 400mm
Reverse Direction	Reverses the profile direction.
Reverse Inset	Reverses the inset of the profile. If this option is enabled, the value specified for the **Inset** is reversed. See Shift/Inset.
Keep Aspect	When enabled, **Keep Aspect** automatically sets the **Inset** value based on the aspect ratio of the profile. See Shift/Inset.
Make UVs	This control allows you to choose three UV creation methods. The default, **None**, creates the UVs for the new faces inside the existing UV coordinates attempting to bevel the UV data along with the mesh. The U and V options create a completely new set of UV data in the negative UV coordinates with the newly created polygons fanned out in the user selected axis (U or V). These options can be tremendously useful when beveling the geometry results in overlapping UV data, as it creates a more manageable starting point for the UV set.
Profiles	
Mini Preset browser for viewing various profiles. Works the same as the standard Preset Browser.	

Polygon Extrude



Extent X 0 m
Y 0 m
Z 0 m
Sides 1
Make UVs U
Mode Auto

Found on the **Mesh Edit** tab of the Modo Tools toolbar, the Polygon Extrusion tool repositions the selected polygon in space based on the tool's three handles. Modo creates additional polygon surfaces between the new location of the polygon and the original location of the selected faces.

Polygon Extrude	
Extent X/Y/Z	Displays the position of the extruded polygon. You can specify values to reposition the extruded polygon precisely.
Sides	Controls the number of loops created with the Polygon Extrusion tool.
Make UVs	Determines whether Modo generates UV maps automatically for the new polygons in the existing space or fans them out on the U or V axis. • **None** - Generates UV maps in the existing space. • **U/V** - Fans UV maps out on the **U** or **V** axis.
Mode	Provides options to override Modo's automatic setting (**Auto** mode). • **Auto** - Evaluates the geometry to determine if Modo should remove the original selected polygon after the extrusion. This is the default setting. The tool automatically changes to **Shift** or **Thick** mode based on the evaluation unless you specify a behavior. • **Shift** - Removes the original polygon when that polygon would be left inside the mesh. • **Thick** - Leaves the original polygon in place if the selected polygon(s) are single-sided polygons and are not encased by existing geometry.

Polygon Extrude in the Procedural

Modo has a procedural Polygon Extrude mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the list in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Polygon**, double-click **Polygon Extrude**.

You can expand the Polygon Extrude item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses. Polygon Extrude requires the following inputs:

- **Selection** - Allows you to select the polygons you want to extrude.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Flip Polygon

The Flip Polygons tool enables you to quickly reverses the normals of polygons to assist you in correcting UVs while working in the **UV Viewport**. You can select a mesh item or a selection of surface polygons and toggle the **Flip Polygon** tool off and on to see the results. The Procedural Mesh Operation, **Flip Polygons**, is found by clicking **Add Operator** and expanding **Mesh Operations** > **Polygons**.

Flipping UVs using the Procedural Modeling Method

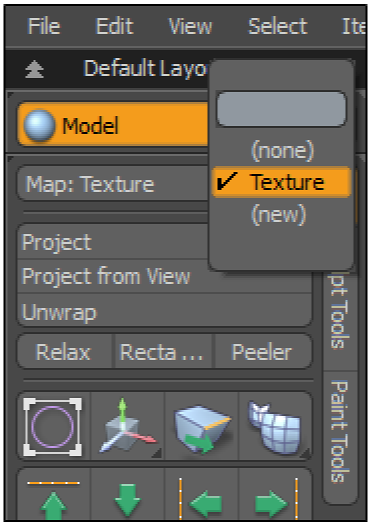
The following is a workflow example, which demonstrates how to us the tool.

To apply the Flip Polygon tool using procedural tools:

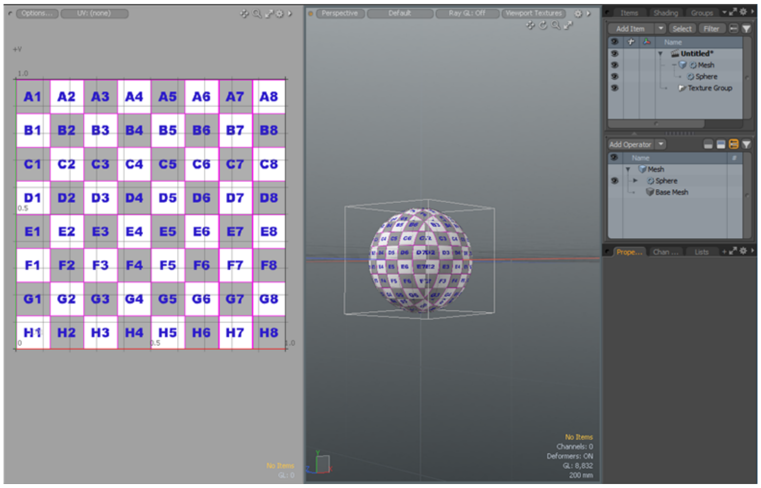
1. In the **UV** layout, click **Viewport Textures** in the 3D Viewport, and select **UV Textures** > **Grid512.png**.

   The **UV Viewport** displays the texture.

2. Open the **Mesh Ops** tab on the right panel.
3. Click the dropdown arrow beside the **Add Operator** button, select **Procedural Modeling** > **Create** > **Primitives** > **Sphere**.
4. On the Toolbox, located in the left panel, click **Map:Texture > Texture**.


File
Edit
View
Select
Ite
Default Layo
Model
Map: Texture
(none)
✓ Texture
(new)
Project
Project from View
Unwrap
Relax
Recta ...
Peeler
pt Tools
Paint Tools

The Sphere is displayed in the 3D Viewport with the UV Texture.

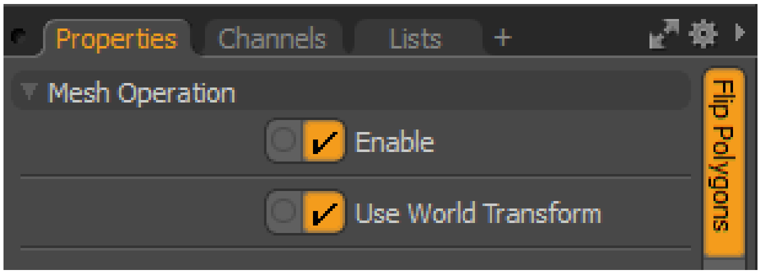

A1 A2 A3 A4 A5 A6 A7 A8
B1 B2 B3 B4 B5 B6 B7 B8
C1 C2 C3 C4 C5 C6 C7 C8
D1 D2 D3 D4 D5 D6 D7 D8
E1 E2 E3 E4 E5 E6 E7 E8
F1 F2 F3 F4 F5 F6 F7 F8
G1 G2 G3 G4 G5 G6 G7 G8
H1 H2 H3 H4 H5 H6 H7 H8
Perspective
Default
Viewport Textures
Items
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Mesh
Sphere
Texture Group
Add Operator
Mesh
Sphere
Base Mesh

5. In the 3D Viewport, with **Items** selection mode enabled, select the **Sphere**.

> **Tip:** You can also select surface polygons to flip by enabling the **Polygon** selection mode and selecting a number of polygons.

6. Click **Add Operator** and select **Mesh Operations** > **Polygons** > **Flip Polygons**.

   The properties display on the bottom right panel with **Flip Polygons** enabled.


Properties
Channels
Lists
+
Flip Polygons

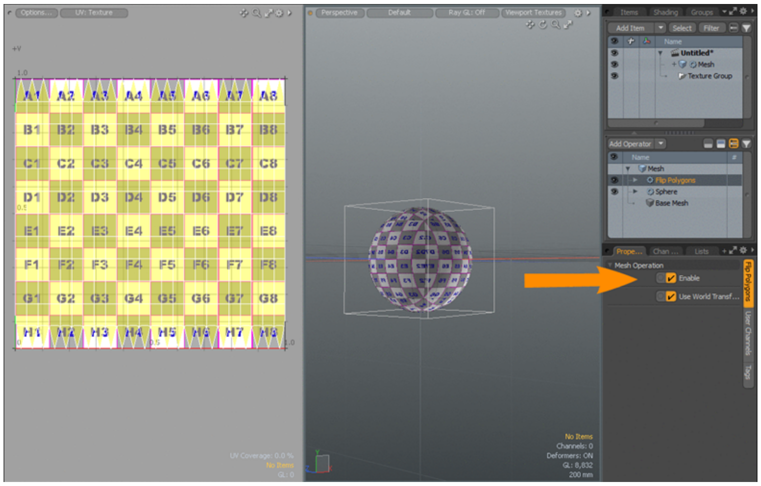
Mesh Operation

☑ Enable

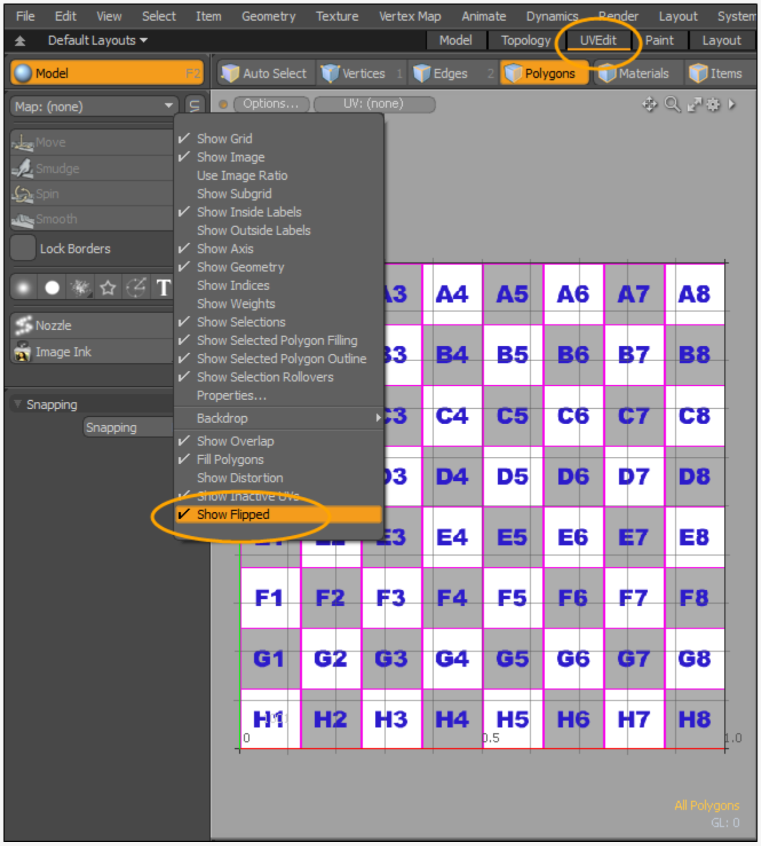
☑ Use World Transform

7. Toggle the **Enable** option to view flipped UVs in the UV Viewport displayed in yellow.

   Click on the image below to view an animation.



Tip: To display flipped UVs in the UV Viewport, open the **UV** layout, click the **Options** button, and enabling **Show Flipped**. This option is independent, meaning you can **Show Flipped** and other menu options, such as **Show Overlap**, at the same time.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Default Layouts
Model
Topology
UVEdit
Paint
Layout
Model
F2
Auto Select
Vertices
Edges
Polygons
Materials
Items
Map: (none)
Options...
UV: (none)
Move
Smudge
Spin
Smooth
Lock Borders
Nozzle
Image Ink
Snapping
Snapping
A3 A4 A5 A6 A7 A8
B3 B4 B5 B6 B7 B8
C3 C4 C5 C6 C7 C8
D3 D4 D5 D6 D7 D8
E3 E4 E5 E6 E7 E8
F1 F2 F3 F4 F5 F6 F7 F8
G1 G2 G3 G4 G5 G6 G7 G8
H1 H2 H3 H4 H5 H6 H7 H8
0
0.5
1.0
All Polygons
GL: 0

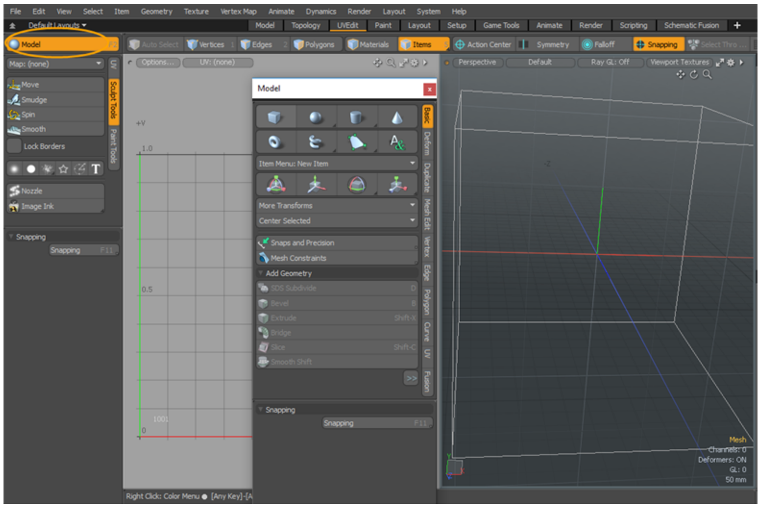
✓ Show Grid
✓ Show Image
Use Image Ratio
Show Subgrid
✓ Show Inside Labels
Show Outside Labels
✓ Show Axis
✓ Show Geometry
Show Indices
Show Weights
✓ Show Selections
✓ Show Selected Polygon Filling
✓ Show Selected Polygon Outline
✓ Show Selection Rollovers
Properties...
Backdrop
✓ Show Overlap
✓ Fill Polygons
Show Distortion
✓ Show Inactive UVs
✓ Show Flipped

Direct Modeling Flipped UV Example

The following example walks you through the process of flipping UVs for a duplicated Mesh item, created by using the Mirror tool.

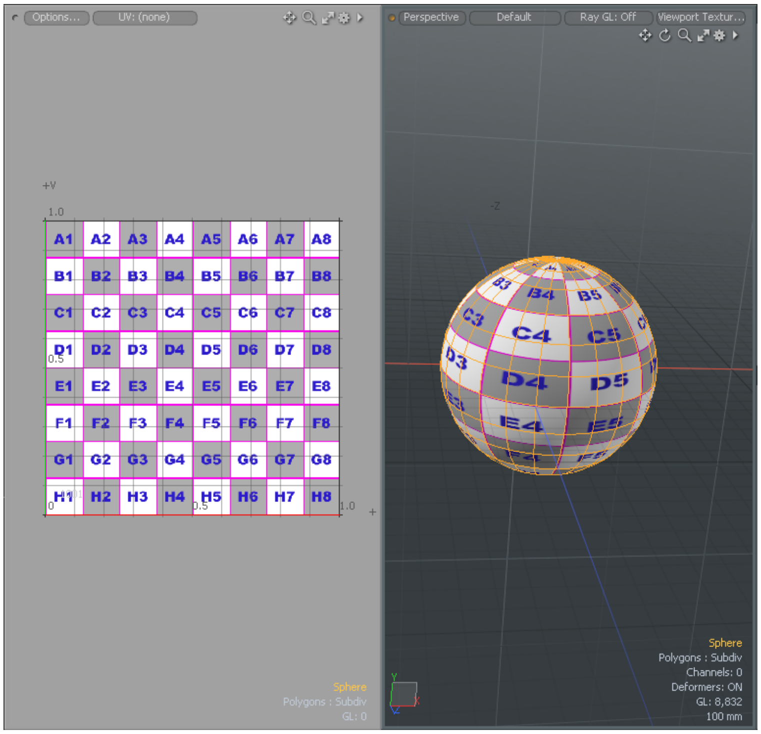
To flip UVs using direct modeling tools:

1. Open the **UV**layout.
2. Click the **Model** tab on the top left corner to open the Modeling toolbox. Alternatively, press **F2**.

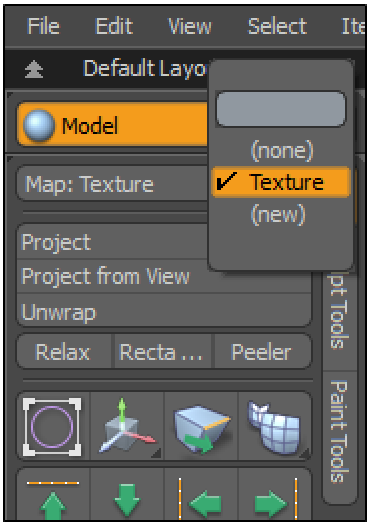

File Edit View Select Item Geometry Texture Vertex Map Animate Dynamics Render Layout System Help
Default Layouts
Model Topology UVEdit Paint Layout Setup Game Tools Animate Render Scripting Schematic Fusion
Model
Auto Select Vertices Edges Polygons Materials Items Action Center Symmetry Falloff Snapping
Map: (none)
Move
Smudge
Spin
Smooth
Lock Borders
Nozzle
Image Ink
Snapping
Options... UV: (none)
Item Menu: New Item
More Transforms
Center Selected
Snaps and Precision
Mesh Constraints
Add Geometry
SDS Subdivide
Bevel
Extrude
Bridge
Slice
Smooth Shift
Perspective Default Ray GL: Off Viewport Textures
Mesh
Channels: 0
Deformers: ON
GL: 0
50 mm
Right Click: Color Menu

3. Press **Shift** and click **Sphere** to add to your scene.
4. In the 3D Viewport, click **Viewport Textures** and select **UV Textures** > **Grid512.png**.

   The Sphere displays with the numbered texture.

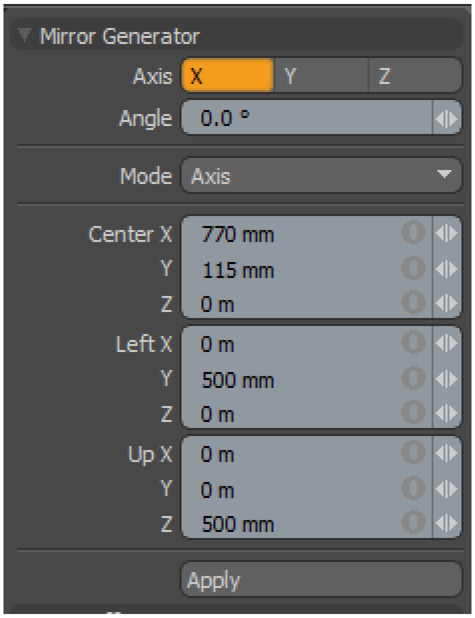

Options... UV: (none)
Perspective Default Ray GL: Off Viewport Textur...
+V
1.0
A1 A2 A3 A4 A5 A6 A7 A8
B1 B2 B3 B4 B5 B6 B7 B8
C1 C2 C3 C4 C5 C6 C7 C8
D1 D2 D3 D4 D5 D6 D7 D8
E1 E2 E3 E4 E5 E6 E7 E8
F1 F2 F3 F4 F5 F6 F7 F8
G1 G2 G3 G4 G5 G6 G7 G8
H1 H2 H3 H4 H5 H6 H7 H8
0.5
0
1.0
Sphere
Polygons : Subdiv
GL: 0
Sphere
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

5. From the Toolbox on the left panel, click **Map:Texture** > **Texture**.


File
Edit
View
Select
Ite
Default Layo
Model
Map: Texture
(none)
✓ Texture
(new)
Project
Project from View
Unwrap
Relax
Recta ...
Peeler
pt Tools
Paint Tools

The **UV Viewport** displays the UVs in the correct orientation.

6. From the **Model** Toolbox, open the **Duplicate** sub-tab, and click **Mirror**.
7. Set the following options and click **Apply**.



Mirror Generator

Axis: X

Angle: 0.0 °

Mode: Axis

Center X: 770 mm
Y: 115 mm
Z: 0 m

Left X: 0 m
Y: 500 mm
Z: 0 m

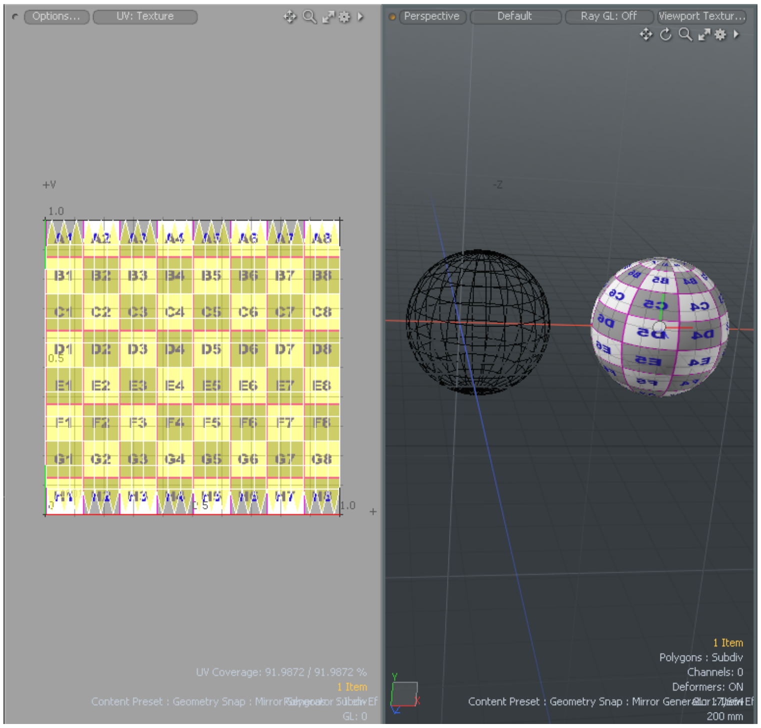
Up X: 0 m
Y: 0 m
Z: 500 mm

Apply

A duplicate Sphere is added to your scene.

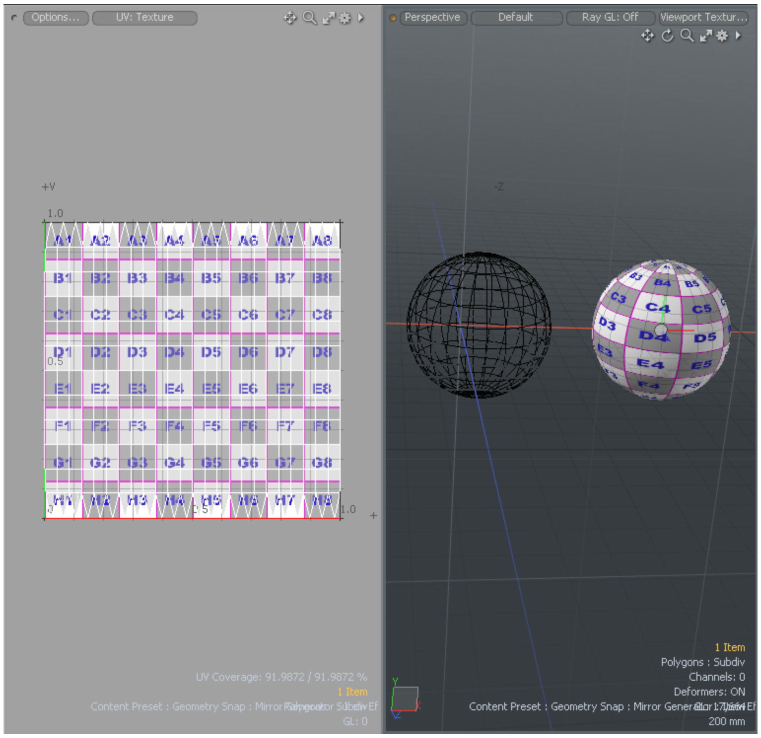
8. From the application menu, enable the **Items** selection mode, and then select the UVs of the mirrored Sphere in the **UV Viewport**.

The inverted UVs display in yellow.


Options...
UV: Texture
Perspective
Default
Ray GL: Off
Viewport Textur...
+V
1.0
0.5
UV Coverage: 91.9872 / 91.9872 %
1 Item
Polygons : Subdiv
Channels: 0
Deformers: ON
200 mm

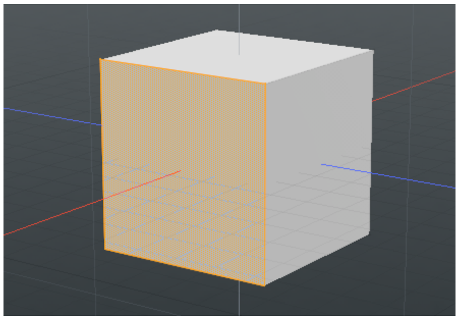
9. With the **UV** layout open, in the left panel, open the **UV** sub-tab, and click **Flip** > **Flip U**.

   The UVs flip back to the correct orientation.

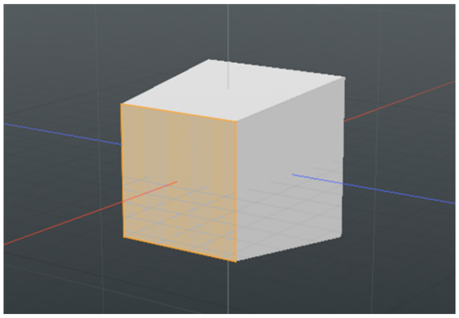

Options...
UV: Texture
Perspective
Default
Ray GL: Off
Viewport Textur...
+V
1.0
0.5
UV Coverage: 91.9872 / 91.9872 %
1 Item
Polygons : Subdiv
Channels: 0
Deformers: ON
200 mm

PolyScale Offset

Use the **PolyScale Offset** tool to scale a selected polygon on a mesh item.



Before with a polygon selected

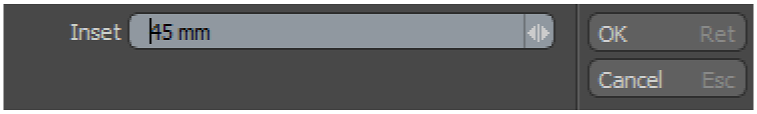


After applying an Inset value

When applying the PolyScale Offset command, setting the Inset value re-sizes a selected polygon. In this example, the selected polygon is scaled onto the original Cube item.

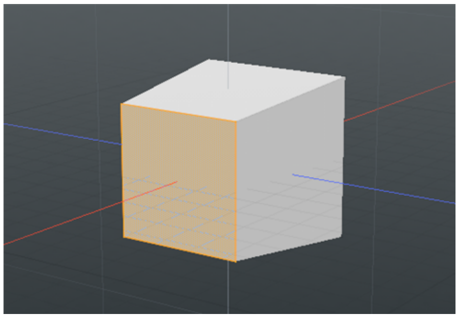
1. Add a Cube, and select two polygons.
2. On the left panel, open the **Polygon** tab, and click **PolyScale Offset**.
3. Enter an **Inset** value and click **OK**.

   The Inset value determines contraction distance of the newly generated polygons, creating a smaller or larger quad polygon framed within the original. The Inset value is applied to all edges.

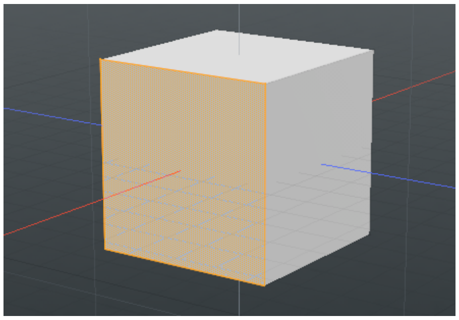
   Inset 45 mm

   The selected polygon face decreases in size.

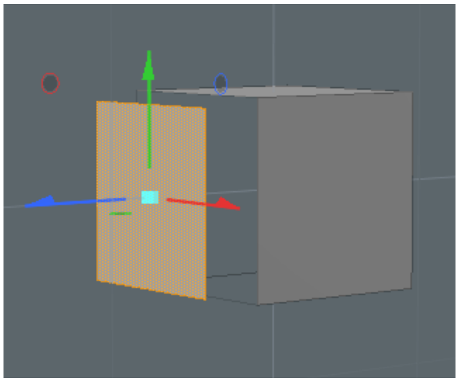
   

PolyScale Offset Detach

Use the **PolyScale Offset Detach** tool to scale a selected polygon on a mesh item and detach it.



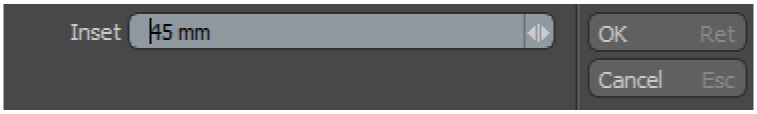
Before with a polygon selected



After applying an Inset value and moving it

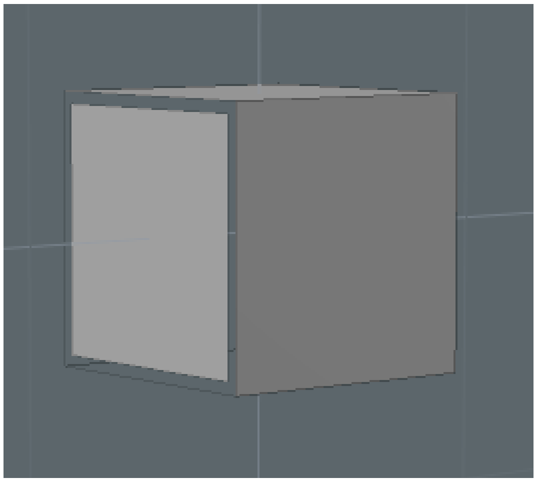
When applying the PolyScale Offset Detach command, setting the Inset value re-sizes a selected polygon and detaches from the original mesh. In this example, the selected polygon is scaled and removed from the original Cube item.

1. Add a Cube, and select a polygon.
2. On the left panel, open the **Polygon** tab, click and hold **PolyScale Offset**, and select **PolyScale Offset Detach**.
3. Enter an **Inset** value and click **OK**. In this example, type **45 mm**.

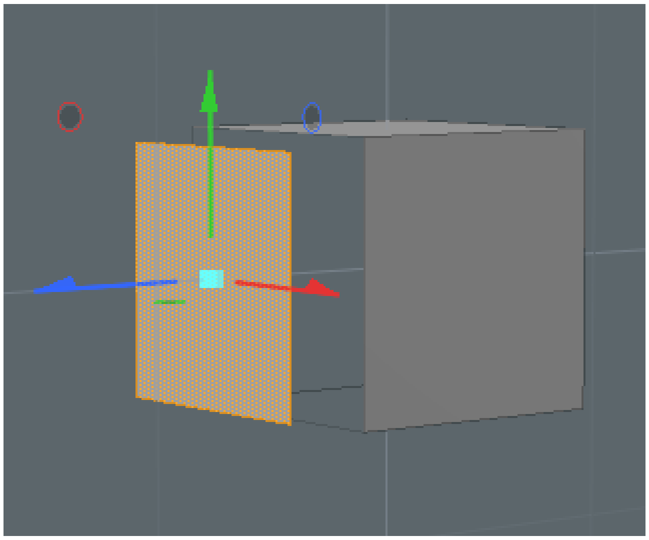
   

   Inset 45 mm

   The Inset value determines the contraction distance of the newly generated polygons, creating a smaller or larger quad polygon framed within the original. The Inset value is applied to all edges.



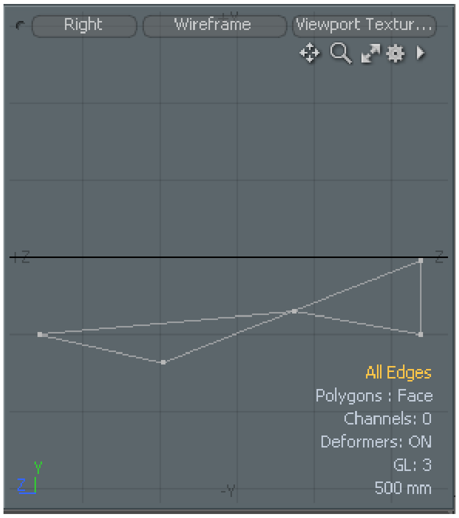
4. Press **W** and move the selected polygon.



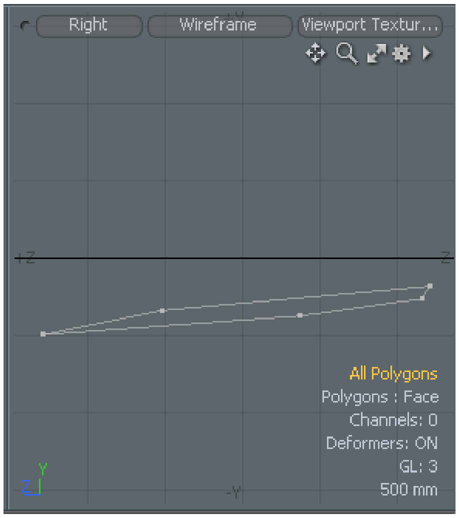
The selected polygon is detached from the mesh item in the viewport.

Flatten Ngons

Use the **Flatten Ngons** tool to flatten Ngons in a model that are not planar. Applying this command makes the selected mesh item a two dimensional form and produces smoother properties when rendered. For example, use this tool when you have a hard surface model that isn't subdivided.


Right
Wireframe
Viewport Textur...
+Z
Z
All Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 3
500 mm
Y
Z
-Y

Before - non planar mesh

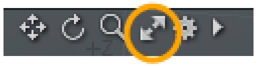

Right
Wireframe
Viewport Textur...
+Z
Z
All Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3
500 mm
Y
Z
-Y

After - planar mesh

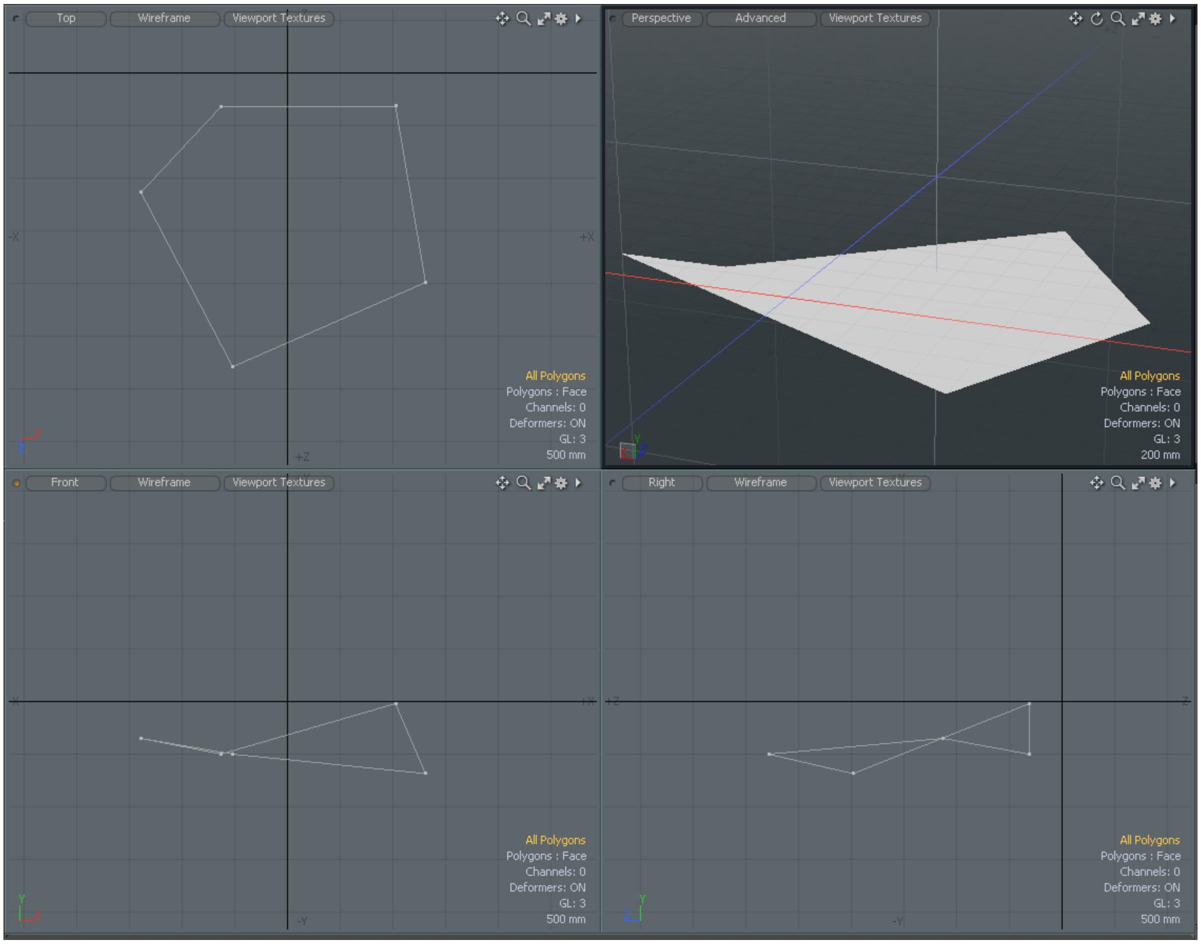
Tip: A n-gon is a polygon with *n* sides; for example, a triangle is a 3-gon.

For more information, see Morph Influence.

1. Download our example scenes and unpack the files.
1. Open **Flatten Ngons.lxo**, which has a morph map applied.
2. In the viewport, click the **View Type** button on the top corner.

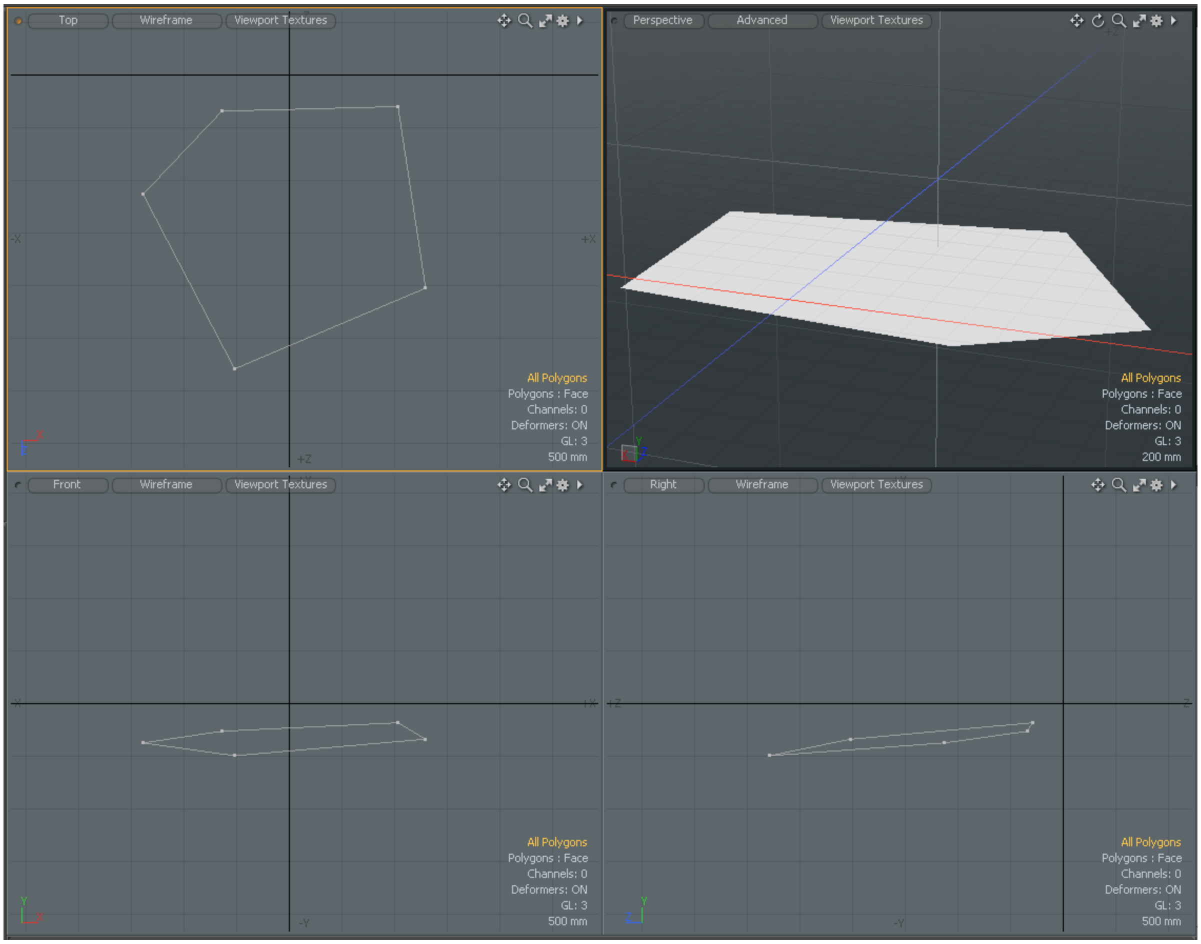


All viewport views are displayed and you will notice the model is not planar.


Top
Wireframe
Viewport Textures
Perspective
Advanced
Front
Right
All Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3
500 mm
200 mm

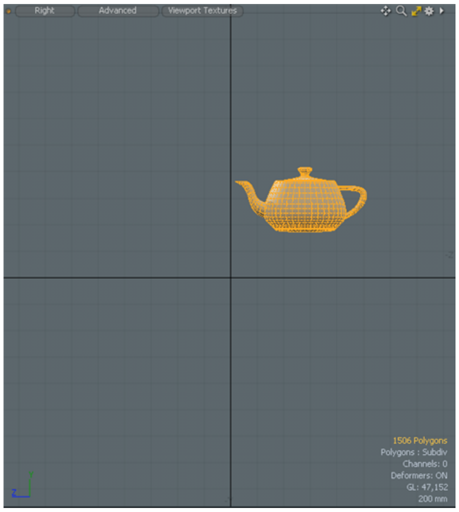
3. In **Model** layout, on the left panel, open the **Polygon** tab, and select **Flatten Ngons**.

   The Ngons are now flattened.

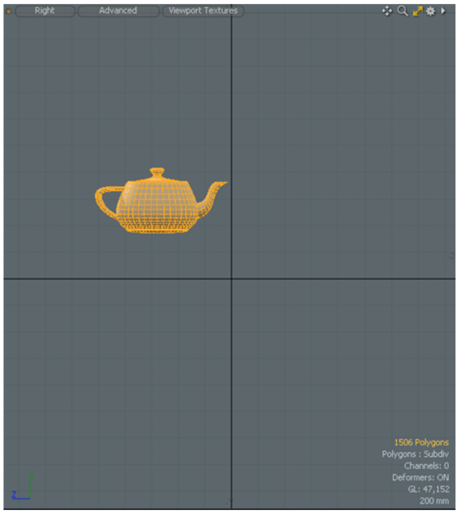

Top
Wireframe
Viewport Textures
Perspective
Advanced
Front
Right
All Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3
500 mm
200 mm

Flip It

Use the **Flip It** tool to flip a selection of polygons to a specified axis.


Right
Advanced
Viewport Textures
1506 Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
200 mm

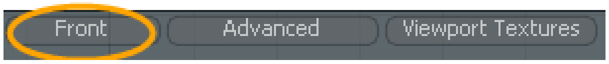
Before with all polygons selected


Right
Advanced
Viewport Textures
1506 Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
200 mm

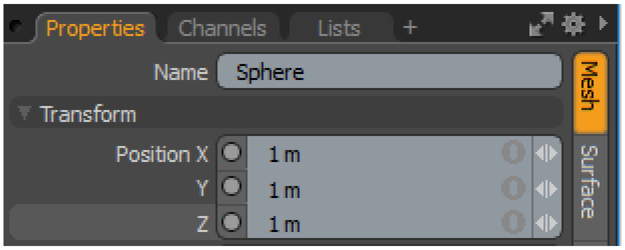
After setting the flip to the X axis

The Flip It tool allows you to specify which axis to flip your polygon selection to. In this example, selected polygons for a sphere are flipped to the X axis.

1. Add a Sphere.
2. On the top of the viewport, set the camera to **Front**.

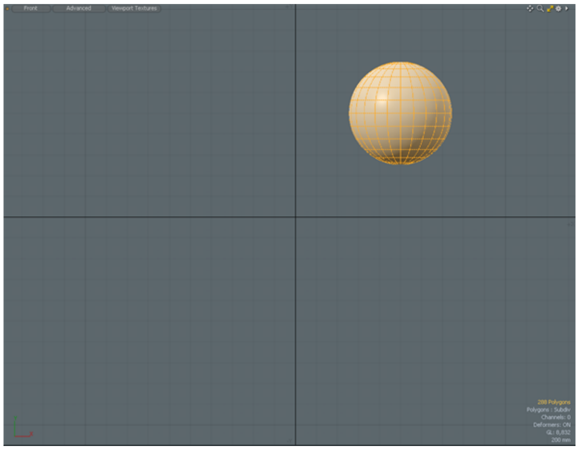

Front
Advanced
Viewport Textures

3. On the right panel, on the **Properties** tab, set the **Position** values for**X**, **Y**, and **Z** to **1 m**.

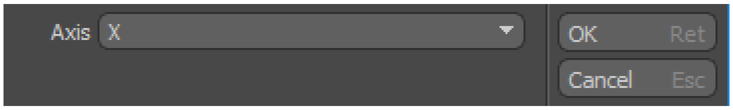

Properties
Channels
Lists
+
Transform
Mesh
Surface

Name Sphere
Position X 1 m
Y 1 m
Z 1 m

4. Activate the **Polygon** selection mode and double-click on the sphere to select all of the polygons.
5. On the left panel, open the **Polygon** tab, and click **Flip It**.

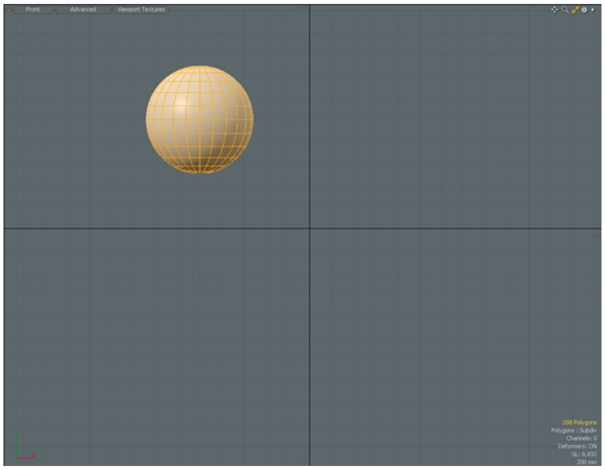


6. Change the Axis value to **X** and click **OK**.



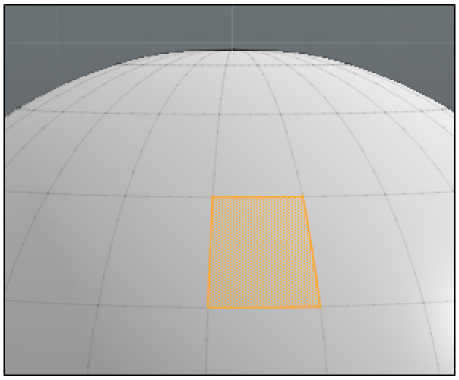
Axis X

The selection moves to the X axis position.

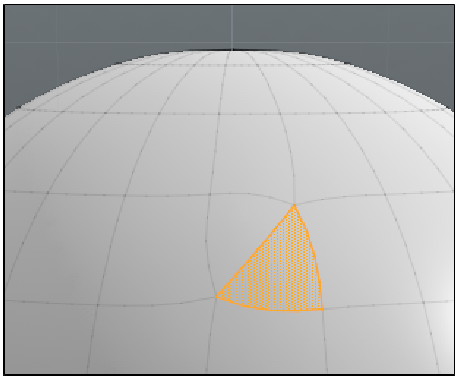


Triple Fan

Use the **Triple Fan** tool to split a selected polygon at a selected vertex point.

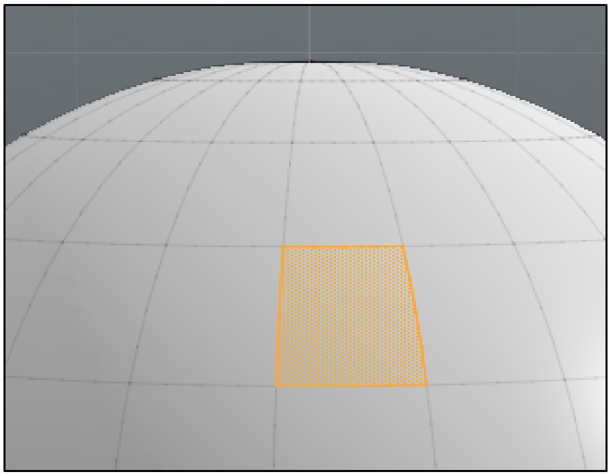


Before with a polygon selected

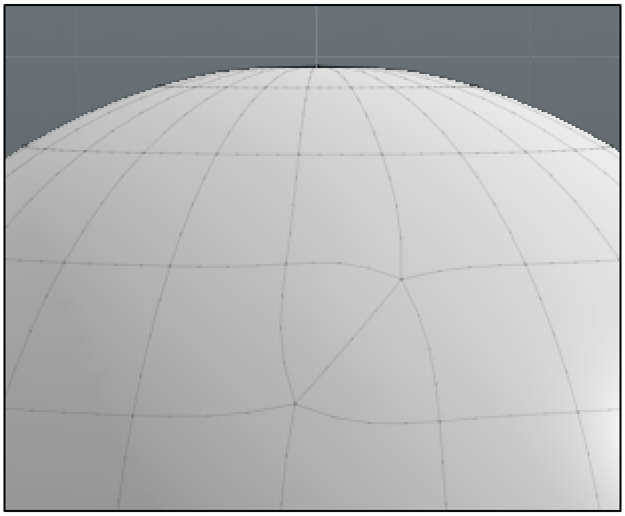


After applying the Triple Fan tool

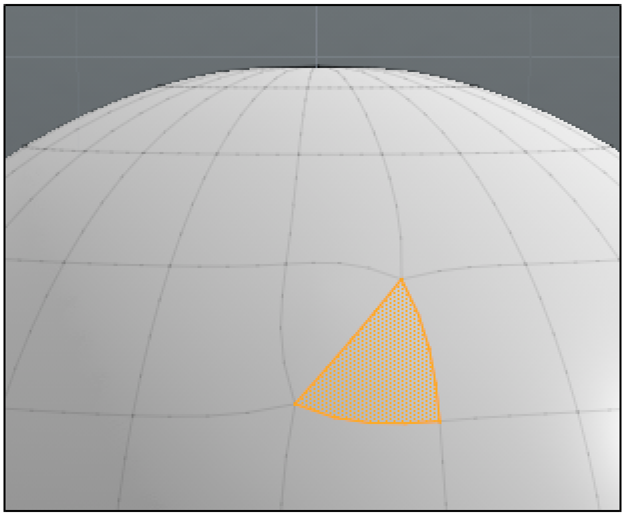
1. In this example, add aSphere, and select one face on the sphere.



2. Activate the **Vertices** selection mode and select a vertex on the previously selected polygon.
3. On the left panel, open the **Polygon** tab, and select **Triple Fan**.



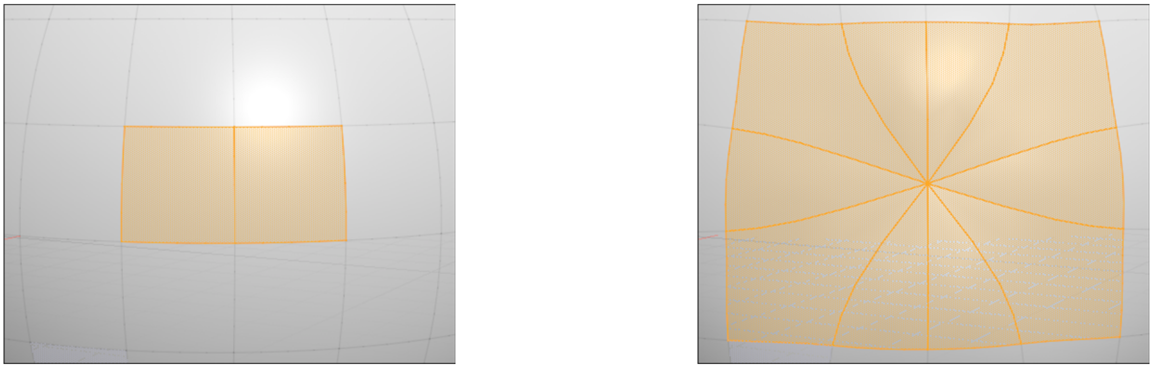
4. Activate the **Polygon** selection mode again and select one of the split polygons.



Reduce Collapse Tools

Use the **Reduce Collapse** and **Reduce Collapse Selection** tools to collapse selected polygons and merge the surrounding vertices together. They remove the selected polygon without destroying the integrity of the geometry. Selected polygons are deleted but no hole is left behind. The gap is closed by merging the neighboring polygons together.

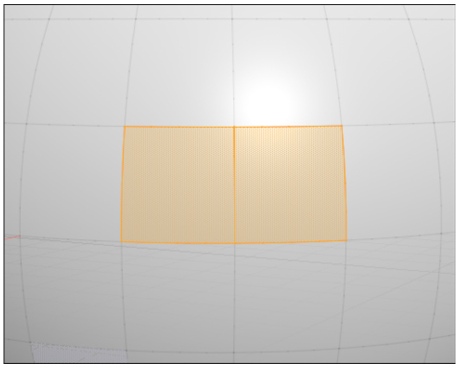
- **Reduce Collapse** - Selected polygon faces collapse and the surrounding vertices are merged together.



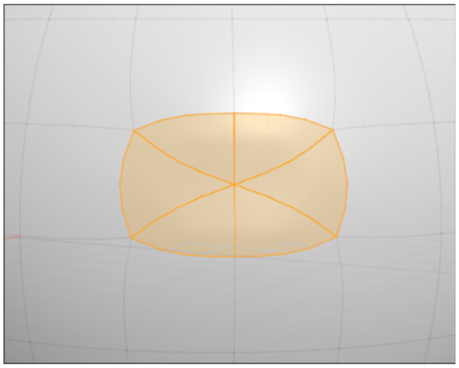
Before

After

- **Reduce Collapse Selection** - Selected polygons split into six polygons and merge together at the center.



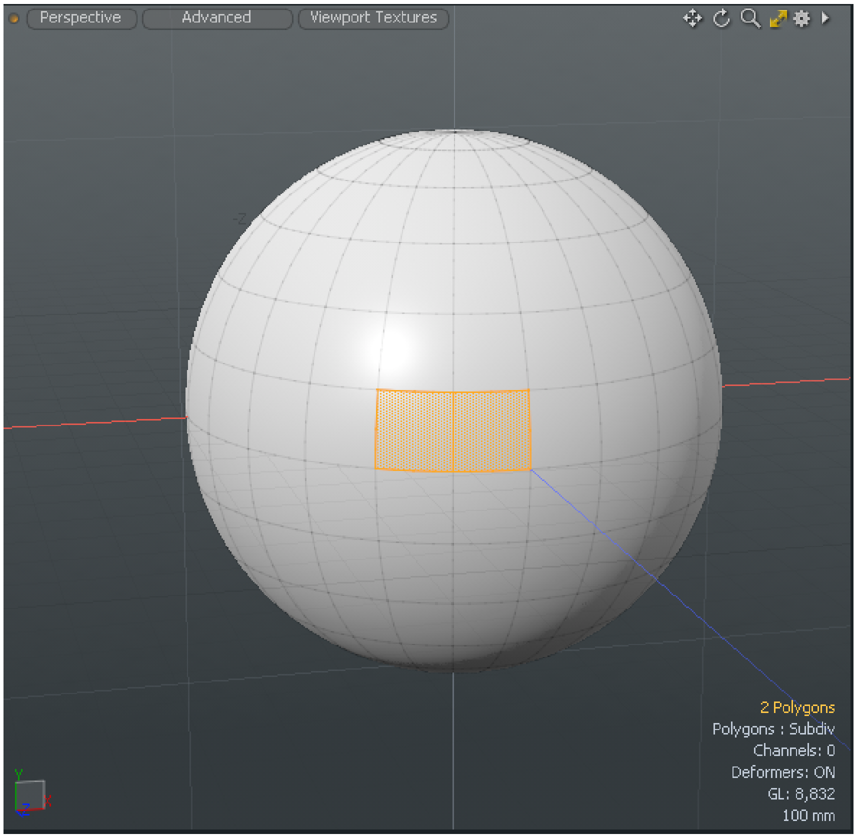
Before



After

Note: For more information about other Reduce tools, see Merge, Remove, Detriangulate, and Unify

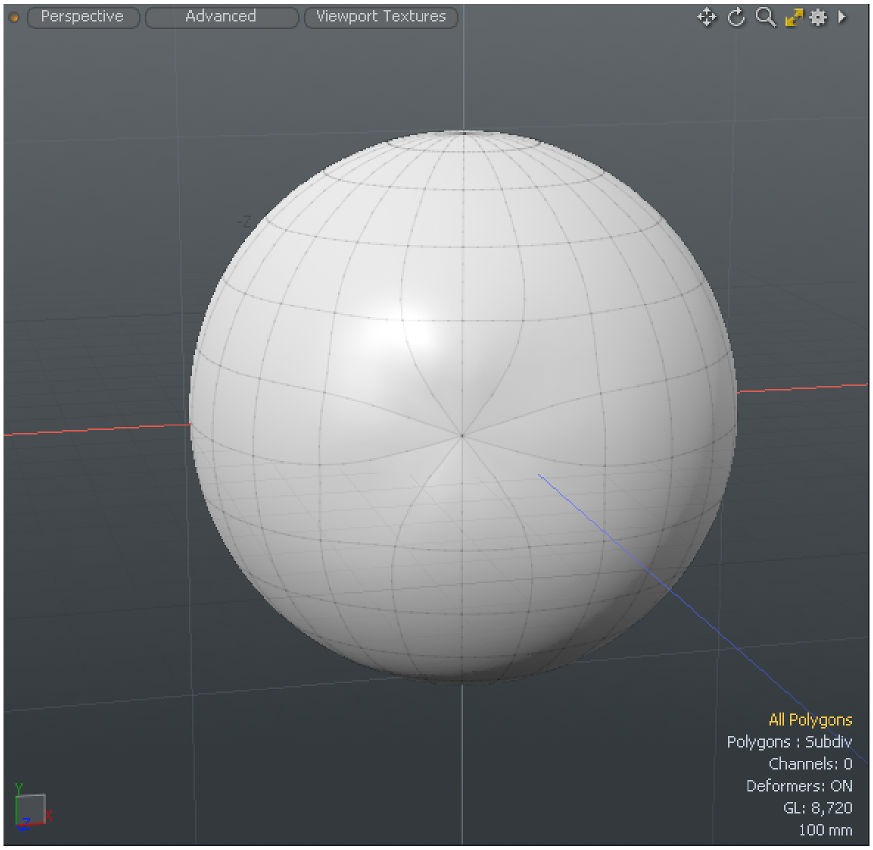
1. In this example, add a Sphere, and select two polygons.



2 Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

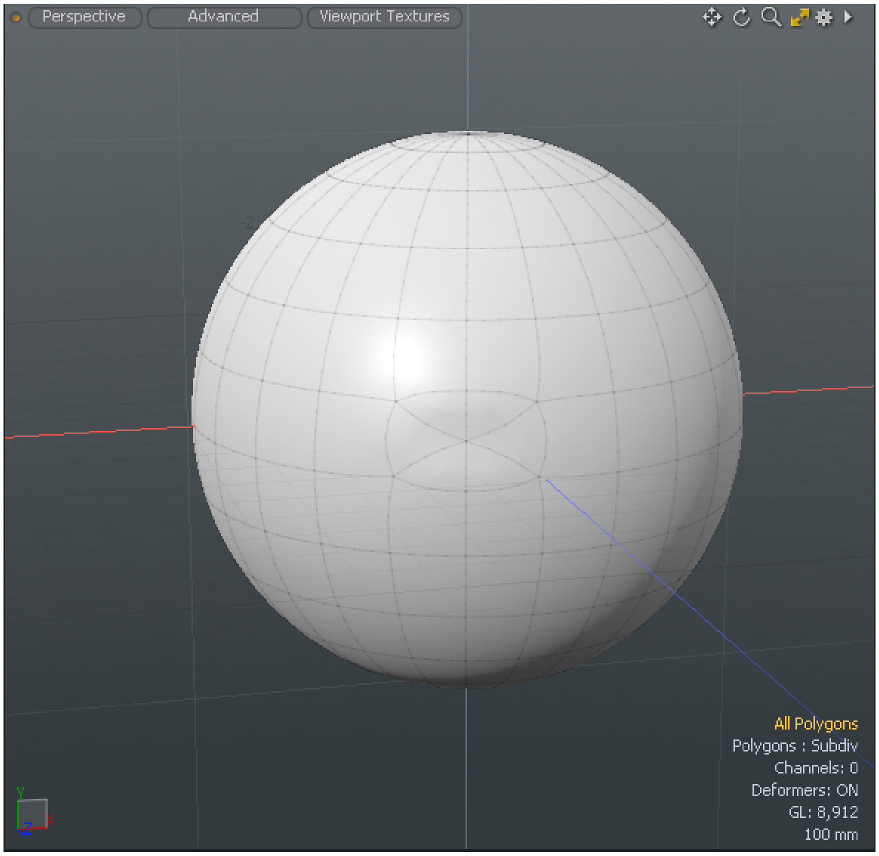
2. On the left panel, open the **Polygon** tab, click and hold **Reduce** and select **Collapse**.

   The selected polygon faces collapse and the surrounding vertices are merged together.

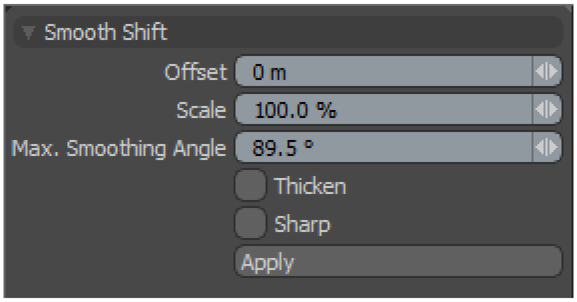

Perspective
Advanced
Viewport Textures
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,720
100 mm

3. Press **Ctrl/Cmd** + **Z** to undo.
4. With the original polygons selected and click and hold **Reduce**, and select **Collapse Selection**.

   The selected polygons split into six polygons and merge together at the center.


Perspective
Advanced
Viewport Textures
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,912
100 mm

Smooth Shift


Smooth Shift
Apply

Offset: 0 m
Scale: 100.0 %
Max. Smoothing Angle: 89.5 °
☐ Thicken
☐ Sharp

The Smooth Shift tool is a deformation tool that smooths out sharp edges in polygonal geometry based on a smoothing angle. When the angle between polygons is greater than the assigned smoothing angle, Modo replaces the edge with two edges that are scaled and offset to round the sharp corners of the mesh. You can use its two handles in the viewport to control individually the offset and scale.

The Smooth Shift tool is on the **Polygon** tab of the Modo Tools toolbar in the **Tools** group. You can access this tool also by choosing **Geometry** > **Polygon** > **Smooth Shift**.

You can use the Smooth Shift tool interactively in the 3D viewport, or you can adjust options in the tool's **Properties** panel and then click **Apply**.

Smooth Shift in the Procedural

Modo has a procedural Smooth Shift mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the list in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

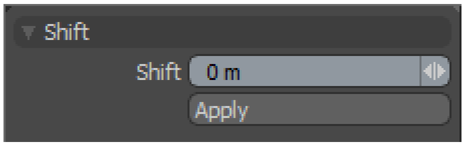
You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Polygon**, double-click **Smooth Shift**.

You can expand the Smooth Shift item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses. Smooth Shift requires the following inputs:

- **Selection** - Allows you to select the polygons you want to affect.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Shift



Shift: 0 m

The polygon Shift tool performs an extrusion on selected polygons along their individual normal axes. This tool is similar to the Extrude tool found on the **Mesh Edit** tab of the Modo Tools toolbar, except that you can extrude multiple polygons in different directions in a single operation.

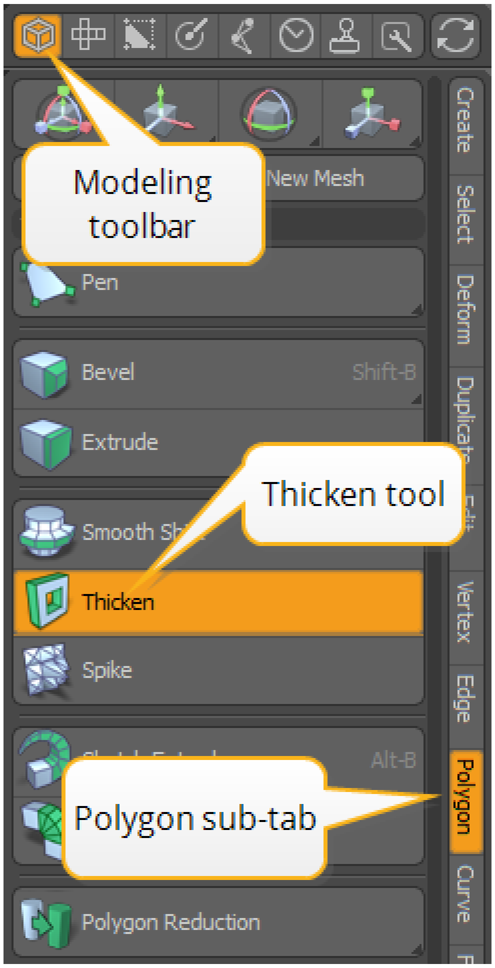
You access the polygon Shift tool from the **Polygon** tab of the Modo Tools toolbar in the **Tools** group, by clicking **Shift**.

Thicken

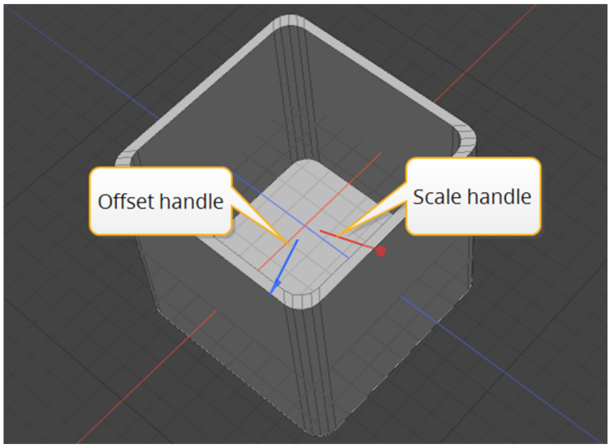
With the Thicken tool you can quickly add dimension to flat geometry. This functionality is often referred to as Shell. You can use the Thicken tool to add thickness to the outside of a volume to make the object larger or to the inside of the volume to create an inner surface.

To use the tool:

1. Make sure you're in the **Modo**, or **Model** layouts, as these are the best suited for working with the Thicken tool.
2. Select your geometry in the 3D viewport.
3. Open the Modeling toolbar on the left panel, and click the **Polygon** sub-tab.
4. Click the **Thicken** button.
5. With the tool active, click in the 3D viewport.


Modeling toolbar
New Mesh
Pen
Bevel
Shift-B
Extrude
Thicken tool
Smooth Sh
Thicken
Spike
Alt-B
Polygon sub-tab
Polygon Reduction
Create
Select
Deform
Duplicate
Vertex
Edge
Polygon
Curve

Modo creates a new set of polygons at the same location as the original polygons. You can see tool handles in the 3D viewport. The blue handle adjusts the offset and the red handle adjusts the scale.



6. Drag on the handles to thicken the geometry.

   You can also edit these properties in the Thicken panel.

Procedural Thicken

The Thicken mesh operation is the procedural equivalent of the Thicken tool.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

To add the mesh operation:

1. Click the **Add Operator** button above the **Mesh Operations** list.

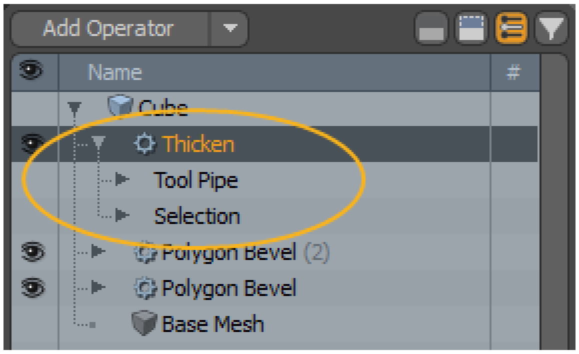
Note: If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

2. Under **Mesh Operations** > **Polygon**, double-click **Thicken**.

   This adds Thicken to the list and displays tool handles in the 3D viewport.
3. Drag on the handles to thicken the geometry.

   You can also edit these properties in the Thicken panel.

Expand the **Thicken** item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses.



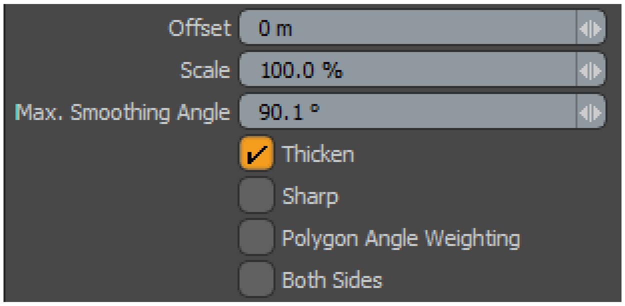
Thicken requires the following inputs:

- **Tool Pipe** - Add falloffs and sub-tools to the operation.
- **Selection** - Select the polygons you want to thicken.

Thicken Properties

You can find the direct modeling Thicken tool's properties on the lower left panel. In the Modo layout, click the **Tool Properties** button, or press **K** to open them.

When using the Thicken mesh operation, the properties can be found on the lower right panel.



Offset 0 m

Scale 100.0 %

Max. Smoothing Angle 90.1 °

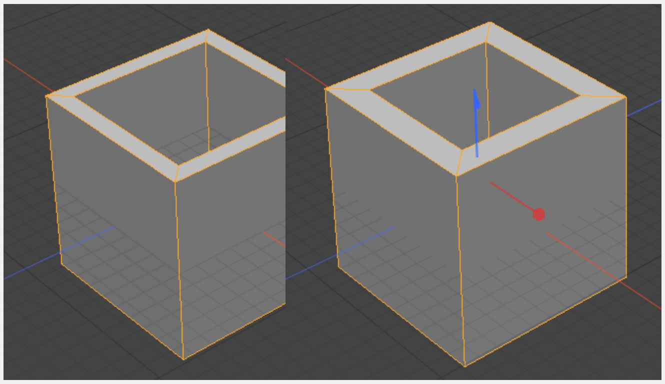
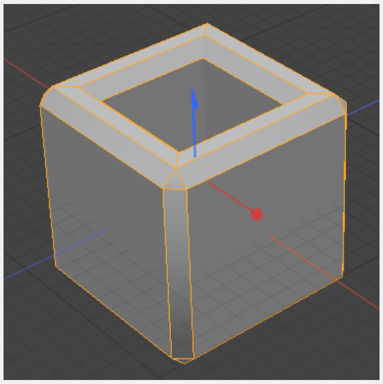
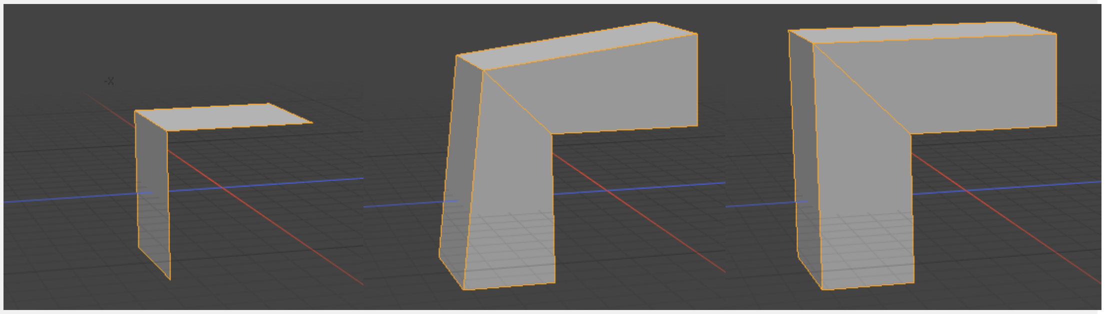
☑ Thicken

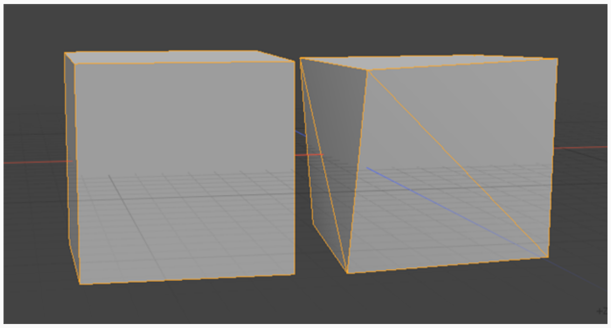
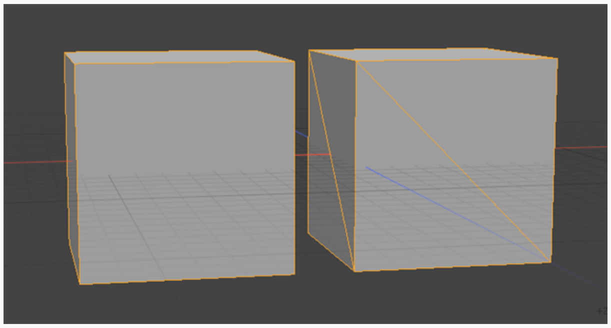
☐ Sharp

☐ Polygon Angle Weighting

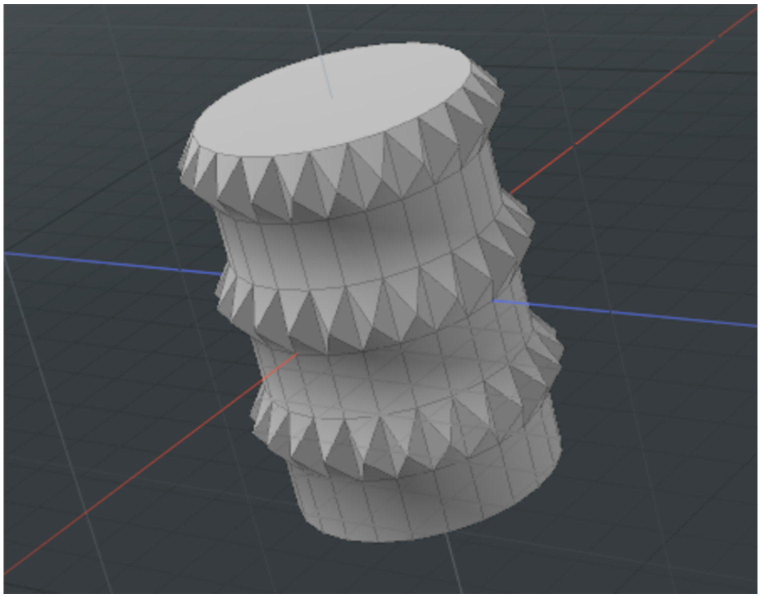
☐ Both Sides

Offset	Moves the selected or original polygons in the direction of their normals (with a positive offset). At the same time, Modo creates a set of polygons in the original location but facing the opposite direction. Modo connects these two sets of polygons with a row of polygons that are roughly perpendicular. If the offset is negative, Modo pushes the original polygons in the opposite direction of their original normals and flips them at the same time to avoid having the object turn inside out.
Scale	Specifies the scale of the operation.
Max. Smoothing	Shifts vertices along the averaged vertex normal vectors computed from connecting

Angle	polygon face normals. When the angle between faces is less than the **Max Smoothing Angle**, the averaged normals are computed per smoothing face group and then combine the separated normal vectors as the shift vector. This is useful when thickening polygon groups with very sharp edges.   Original mesh Thicken with a **Max. Smoothing Angle** of 90.1° Thicken with a **Max. Smoothing Angle** of 0°
Thicken	Adds thickness to flat geometry by creating a new set of polygons. Disabling this checkbox disables the tool.
Sharp	When enabled, it makes the angles sharp. For example, on a 90° mesh it keeps the middle edge in line with the others.  Original mesh Thicken with **Sharp** disabled Thicken with **Sharp** enabled
Polygon Angle Weighting /Weight by Polygon Angle	When enabled, the thicken amount is based on the average area of the polygon. Enabling this option produces better results with meshes that have triangle polygons.

	 **Polygon Angle Weighting** disabled on regular and triangled cube  **Polygon Angle Weighting** enabled on regular and triangled cube
Both Sides	When enabled, the tool thickens in both directions, rather than from the source polygon in one direction. Both outer and inner points are offset from the middle position by half of the **Offset** amount.

Spikey



The Spikey tool is a polygon tool that takes selected polygons and adds a vertex to the center of the polygon. It then repositions that vertex away from the original polygon based on the spike strength setting to create a spike. Modo creates spikes based on polygon face normals for each selected polygon.

You can access the Spikey tool by clicking the **Polygon** tab of the Modo Tools toolbar.

You can work with the Spikey tool interactively in the viewport, or you can adjust options in the tool's **Properties** panel and click **Apply**.

Procedural Spike

The Spike mesh operation is the procedural equivalent of the Spikey tool.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

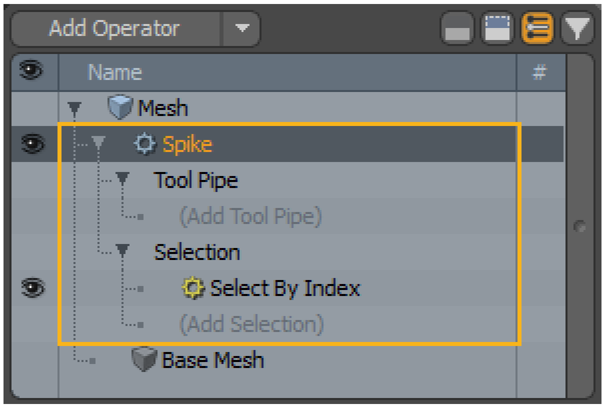
1. In **Polygon** selection mode, select the required polygons on your mesh in the 3D viewport.
2. On the right panel, click the **Mesh Ops** tab.

Note: If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

3. Click the **Add Operator** button, and under **Mesh Operations** > **Polygon**, double-click **Spike**.

   This adds vertices to the center of each selected polygon. The **Spike** mesh operation is added to the Mesh Operations list on the right, and its properties are displayed in the lower right panel.
4. Adjust the **Spike Strength** property to make the polygons protrude.

You can expand the **Spike** mesh operation in the list by clicking the arrow ▼ in front of it. This reveals the inputs the operation uses:



- **Tool Pipe** - Add falloffs and sub-tools to the operation.
- **Selection** - Select polygons or modify your existing selection. For more information on procedural selection, see Procedural Selection.

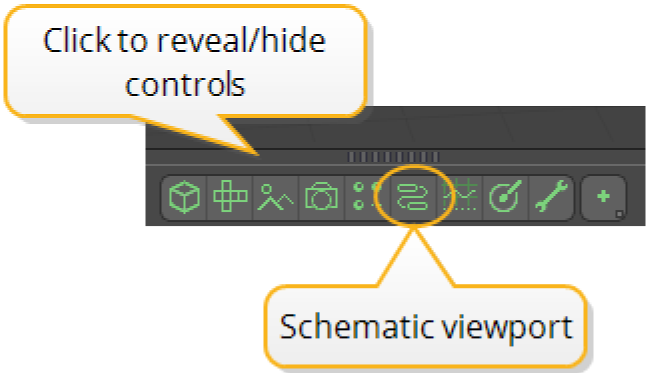
Spike in the Schematic

You can also use the Spike tool when working in the Schematic viewport.

Note: For more information on working with Schematic viewport in general, see Schematic Viewport.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.



>

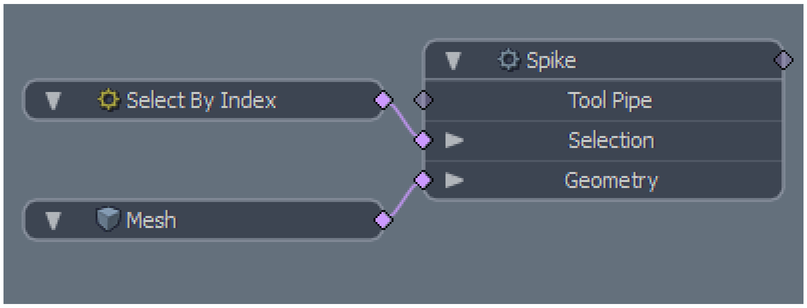
MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

To use the Spike node:

1. Click **Add...**, and under **Mesh Operations** > **Polygon**, double-click **Spike**.

   This adds the Spike node to the viewport and opens its properties on the lower right panel.

   The node has the following inputs:

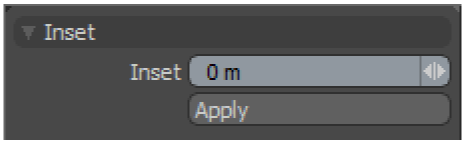
   - **Tool Pipe** - Add falloffs and sub-tools to the operation.
   - **Selection** - Select polygons or modify your existing selection.
   - **Geometry** - Any geometry that is affected by the tool.

2. Adjust the **Spike Strength** property to make the polygons protrude.

Inset

Inset is a polygon creation tool that takes selected polygons and subdivides them to create new geometry within the original selection. The new polygons are scaled duplicates of the original. For example, selecting a single polygon on the side of a cube, and then firing **Inset**, creates a smaller quad polygon framed within the original. Using negative values for inset forces the duplicated polygon to scale larger than the original.

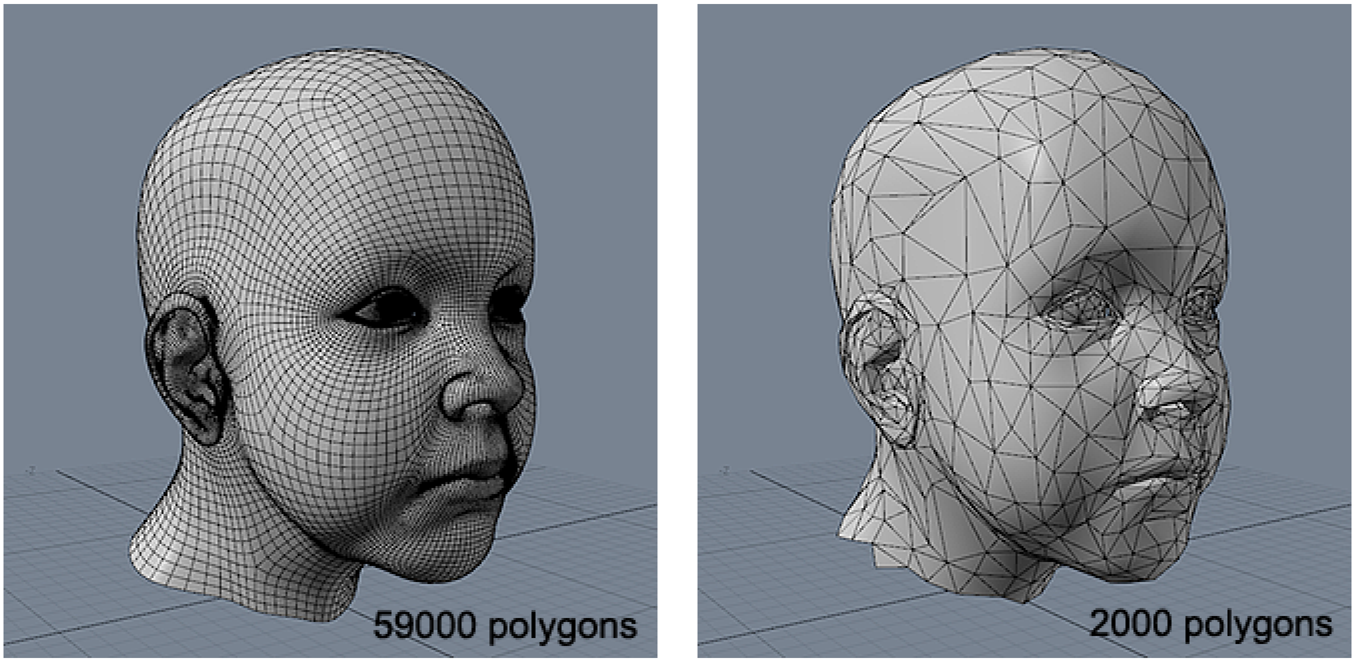
Inset can be found on the **Geometry** toolbar under **Polygon**.

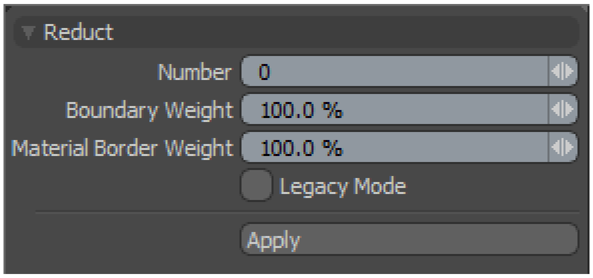

Inset
Inset
Apply

Inset: 0 m

Set the **Inset** value using the numeric field, then use the **Apply** button to fire the command with those values.

Polygon Reduce


59000 polygons
2000 polygons


Reduct
Legacy Mode
Apply

Number: 0
Boundary Weight: 100.0 %
Material Border Weight: 100.0 %
☐ Legacy Mode

The Polygon Reduce tool iteratively removes edges from the mesh while trying to maintain as much of the original contouring as possible.

Polygon Reduce	
Number	Displays the total number of polygons in the currently visible mesh layer. When you drag the tool in the viewport, Modo adjusts this number here and in the mesh. You can also type in an exact number of polygons.
Boundary Weight	Determines how much emphasis should be given to preserving the details around boundary areas of the mesh.
Material Border Weight	Determines how much emphasis should be given to using polygons to preserve the integrity and detail of the mesh along edges that define a boundary between materials groups.
Legacy Mode	When enabled, uses an alternative edge evaluation method available with older version of Modo. When disabled, Modo uses a polygon reduction method that is based on Quadric Error Matrices to produce better overall surface quality. You can enable and disable **Legacy Mode** to evaluate the differences between the two algorithms and determine which method you prefer.

Polygon Reduce in the Procedural

Modo has a procedural Polygon Reduce mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the stack in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Operator** button above the **Mesh Operations** list. Under **Mesh Operations** > **Polygon**, double-click **Polygon Reduce**.

You can expand the Polygon Reduce item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses. Polygon Reduce requires the following inputs:

- **Selection** - Allows you to select the elements to which you want to apply the tool.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

Set Polygon Type

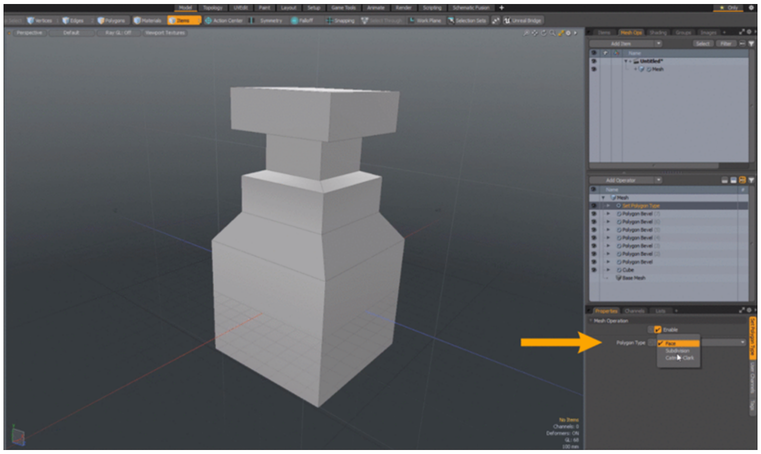
The **Set Polygon Type** tool is a procedural mesh operation that allows you to convert polygons at any point in your design workflow. It can be applied on any surface polygons or a selection of surface polygons by selecting the mesh item in your **Mesh Operations** list and changing the type of polygon used. Polygons can be quickly converted into **Faces**, **Subdivision**, or **Catmull-Clark** polygons. If the initial polygon type matches the selected type, no operation is performed. All parameters of the polygons, such as vertex maps and polygon tags, are maintained.

The **Set Polygon Type** tool is found in the **Mesh Ops** tab, located on the right side of the interface. Click **Add Operator**, expand **Mesh Operations** > **Polygon** and double-click **Set Polygon Type**.

To apply the Set Polygon Type tool:

1. Open the **Model** layout.
2. On the right panel of the interface, open the **Mesh Ops** tab.
3. From the **Mesh Operations** list, select a mesh item.
4. Click **Add Operator**, expand **Mesh Operations** > **Polygon**, and select **Set Polygon Type**.
5. In the **Properties** tab, select a **Polygon Type**.

   The updated mesh is displayed in the 3D viewport.


Set Polygon Type
Polygon Bevel
Cube
Base Mesh
Properties
Channels
Lists
Mesh Operation
Enable
Polygon Type
Face
Subdivision
Catmull-Clark

To disable the Set Polygon Type tool:

1. Select the **Set Polygon Type** item in the **Mesh Operations** list.
2. Click the **Enable** button.

Set Polygon Type Property Options

The following options are available:

Option	Description
Enable	Enables or disables the set Polygon Type. This gives you a similar workflow as Direct Modeling when pressing **Shift** + **Tab** or **Tab**.
Polygon Type	Allows you to specify the type of polygon to convert to. The following options are available: **Face** - Creates a faceted surface. This is the default value. **Tip:** An individual polygon is commonly called a face, which is bounded by three or more vertices and their associated edges. When many faces are connected together to create a network of it's faces called a polygon mesh. **Subdivision** - Creates a smoother surface. **Tip:** Subdivision smooths and adds extra resolution to curves and surfaces. **Catmull-Clark** - Converts any surface into a Catmull-Clark Subdivision Surface. **Tip:** The Catmull-Clark Subdivision Surfaces are superior to regular Subdivision Surfaces, especially in their handling of edge creasing. For more information about Catmull-Clark options, see Mesh Item.

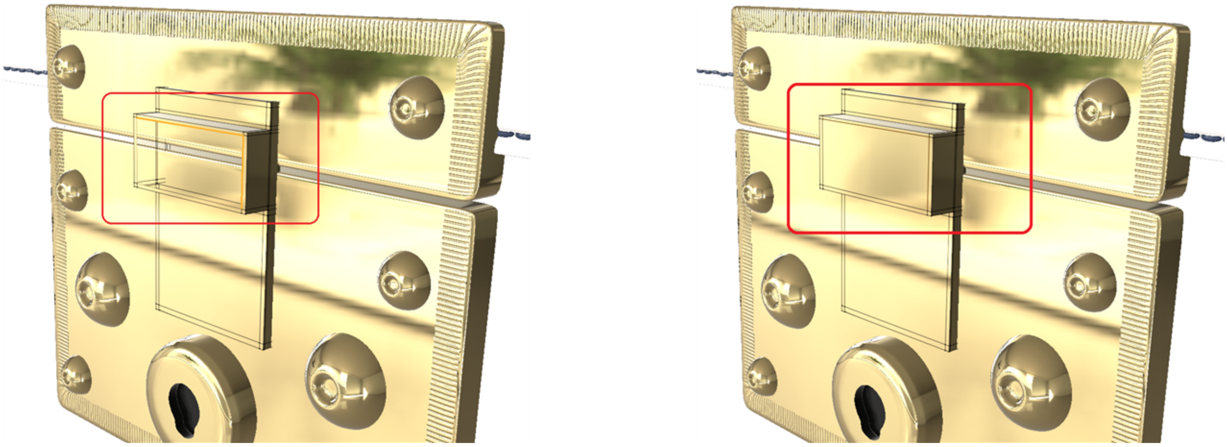
Polygon Tools

The following are useful actions you might want to do using **Polygon** tools.

Make

Automatically create a quadrangle polygon by selecting two edges on an object and clicking **Make** from the **Polygon** tab of the Modo **Tools**, or by pressing **P**. The **Make** command calculates where the endpoint vertex of the two selected edges is to create a new polygon, and merges all connecting edges together.

To access the **Make** tool, open the Polygon tab of the Modo **Tools** toolbar.

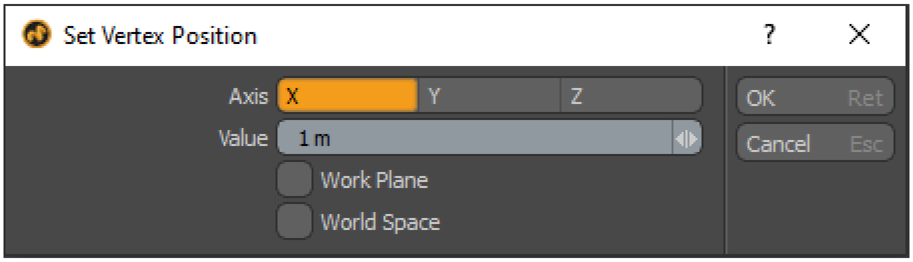


Selecting two edges and pressing **P** to create a merged quadrangle polygon.

Tip: Press and hold **Alt** in the **Polygon** tab of the Modo **Tools** toolbar to switch from **Make** to **Make Polygon**. The **Make Polygon** tool is similar to **Make**, except it provides some additional user preferences and can be performed on vertices. See Make Polygon for more information.

Set Position...

Offset the position of a selected polygon, vertex, or edge by an amount specified in **Value** along a specified **Axis**.



The **Set Vertex Position** dialog.

Tip: Enable **Work Plane** to align the offset along the work plane's axes.

Split

To split polygons, select two vertices on a polygon to force an edge to be drawn between them. Vertices can also be selected in succession across polygons.

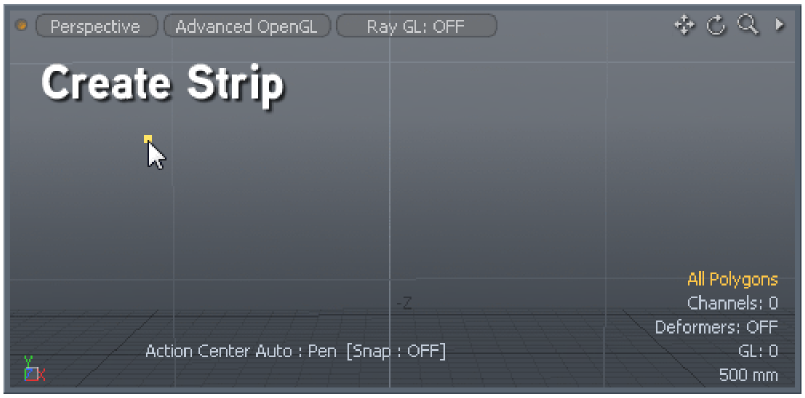
Triple

The **Triple** command subdivides any selected polygon with more than 3 vertices so that it consists only of triangle polygons. This can be very useful for preparing data to be saved in a format that does not support polygons with more than 3 vertices per polygon. It can also be useful for quickly breaking an n-gon down into triangles for similar export issues.

Alternatively, use the Mesh Operation, **Triangulate**, to subdivide polygons in your model. For more information, see Triangulate.

Convert to Quadrangles

Found in the menu bar under **Geometry** > **Polygon** > **Quadruple** or, within the **Topo** tab toolbox, the **Convert to Quadrangles** command takes a series of vertices from an N-gon and creates a row of regular four sided (quadrangle) polygons. This is useful in creating quad strips, especially when re-topologizing a model. Using the **Pen** tool, lay down two opposing rows of vertices, then invoke the **Convert to Quadrangles** command and Modo automatically adds edges spanning between the opposing vertices, making quad polygons.



Spin Quads

The **Spin Quads** command changes where your edges are attached within the geometry. For example, if you select two adjacent polys, this command spins them so that they attach to different points while leaving them in place. It changes the flow of your polygons while maintaining the surrounding mesh. **Spin Edges** does the same thing on an edge level, however, you can select one edge and spin it so that it bisects two polygons differently. **Spin Quads** only works with two polygons with the same edge number.

These tools are found on the **Edge** and **Polygon** tabs of the Modo **Tools** toolbar, under the **Commands** groups.

Flip

The **Flip** tool reverses the direction of selected polygons. Polygons are typically single sided and, as such, are only visible from one direction. The visible side is determined by the direction of the normal. The direction of a polygon is initially determined by the order in which its vertices were created, or selected, to make the polygon. The **Flip** tool effectively re-orders those vertices so that the polygon faces the opposite direction. For linear polygons, such as curves, the **Flip** tool reverses the order to force the curve to run the opposite direction.

Align

The **AlignPolygons** tool attempts to automatically make all polygons face the same direction. Use this when you have mesh geometry with polygons that have face normals facing both toward and away from the view. **Align Polygons** uses the first polygon you select as the model. It attempts to match the face normal direction of that polygon.

Make Curve Fill

Found under the **Commands** section of the **Polygons** toolbox, the **Make Curve Fill** command converts a closed curve, or a series of connected curves, into a renderable flat polygon's surface. It works with both Bezier or Spline curve types and is especially useful for rendering simple vector graphics.

Tip: Bezier curves for curve fill polygon types can be created by importing an **.eps** file, or by converting text to Beziers using the **Convert Text to Beziers** command. Curve fill polygons can also be created by turning on the **Fill** option on the **Curve** and **Bezier** tools.

Merge

The **Merge Polygons** option combines selected polygons into a single polygon of n-number of sides. Essentially, it removes all interior edges, so the multiple polygons can be treated as a single polygon.

Note: More complex selections merge as many polygons together as necessary to remove the target edges. If all the edges that use a vertex are deleted, then the vertex is deleted as well. It may be possible for you to specify selections that cannot all be consistently deleted. In that case the operation does the best that it can without leaving "spikes", which are edges entirely internal to a single polygon.

Polygon Merge is useful for cleaning up geometry where you have many coplanar edges.

To open the Merge Polygon tool:

- On the left panel, open the **Polygon** tab, under the **Commands** section, click on the **Reduce** pulldown, and select **Merge**.
- On the menu bar select **Geometry** > **Polygon** > **Merge**.
- On the right panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Polygons** > **Polygon Merge**.

Procedural Polygon Merge

Polygon Merge is also available as a mesh operation.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

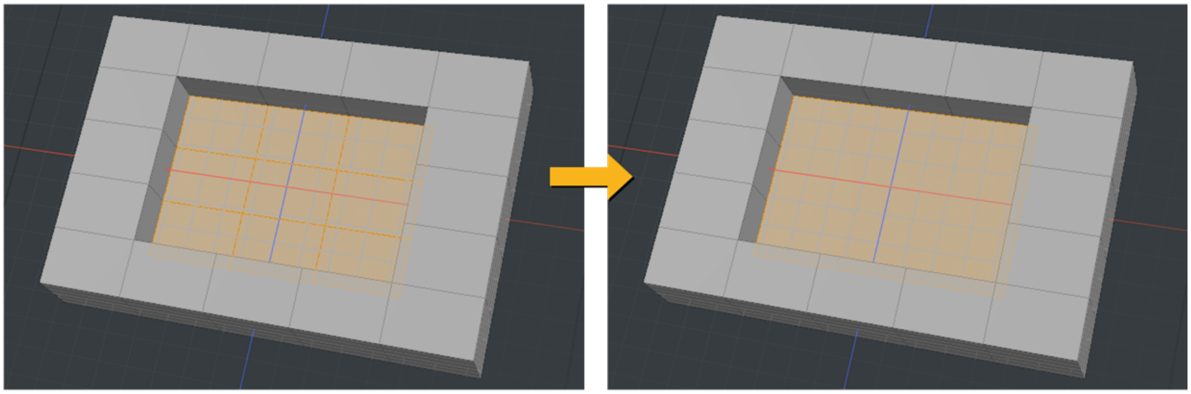
To use the tool:

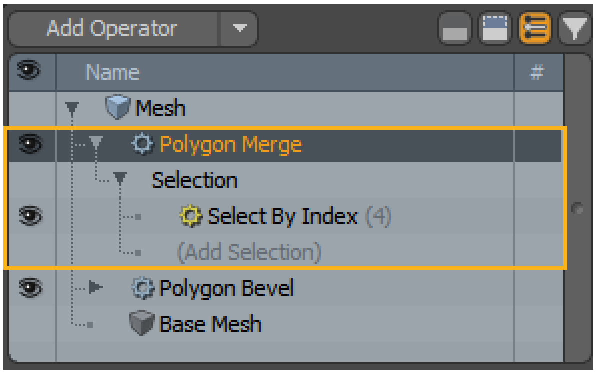
1. In **Polygons** selection mode, select the polygons you want to merge.
2. On the right panel, click the **Mesh Ops** tab.

Note: If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

3. Click the **Add Operator** button, and click **Mesh Operations** > **Polygon** > **Polygon Merge**.

   This adds Polygon Merge to the Mesh Operations list and opens its properties on the lower right pane. You can see the selected polygons merged in the 3D viewport.





You can expand the **Polygon Merge** mesh operation in the list by clicking the arrow ▼ in front of it. This reveals the inputs the operation uses:

- **Selection** - Select polygons to merge or modify your existing selection. For more information on procedural selection, see Procedural Selection.

Polygon Merge in the Schematic

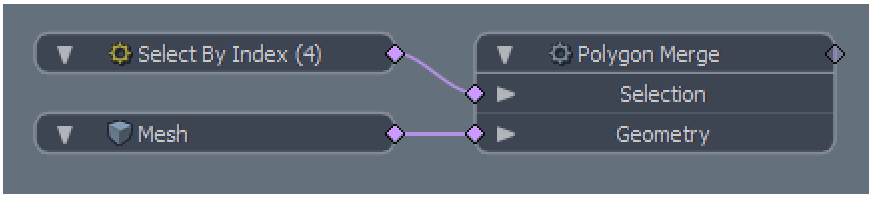
You can also use the Polygon Merge tool when working in the Schematic viewport.

Note: For more information on working with Schematic viewport in general, see Schematic Viewport.

To open the Schematic viewport, either:

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.
- Open only the viewport from the menu bar by clicking **Layout** > **Palettes** > **Schematic**.
- In the **Model** and **Modo** layouts, click the Schematic button.

To add the Polygon Merge node, click **Add...**, and click **Mesh Operations** > **Polygon** > **Polygon Merge**.



The node has the following inputs:

- **Selection** - Select polygons to merge or modify your existing selection.
- **Geometry** - Any geometry that is affected by the tool.

Collapse

The **Collapse** command removes the selected element without destroying the integrity of the geometry. Any select polygon, edge, or vertex is deleted but no hole is left behind. Instead, the mesh heals, closing any gaps by merging the neighboring elements together.

Polygons: On the **Polygons** tab of the **Modo Tools** toolbar > **Commands** > **Reduce** > **Collapse**.

You can also select whatever geometry you wish to collapse and then choose **Geometry** > **Collapse**.

For more examples, see Reduce Collapse Tools.

Collapse Selection

The **Collapse Selection** command removes the selected element without destroying the integrity of the geometry. Any select polygon, edge, or vertex is deleted but no hole is left behind. Instead, the mesh heals, closing any gaps by merging the neighboring elements together and merging together at the center.

Polygons: On the **Polygons** tab of the **Modo Tools** toolbar > **Commands** > **Reduce** > **Collapse Selection**.

For more examples and information about **Collapse Selection**, see Reduce Collapse Tools.

Remove

The **Remove Polygon** command deletes the selected polygons, completely removing them from the selected Mesh Item layer. The command is found within the **Modeling** tabs under the **Polygon** toolbox, or in the menu bar under **Geometry** > **Remove**. You can also simply press the **Backspace** key on the keyboard.

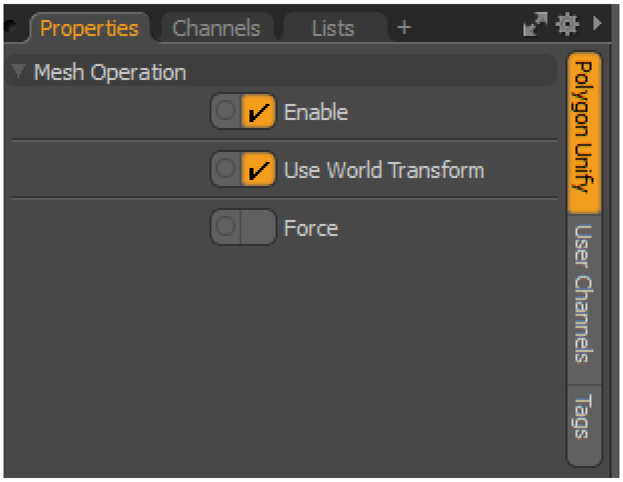
Unify

Unify Polygons lets you merge two polygons into a single polygon when you have two polygons that are in exactly the same position and the polygons share the same vertices.

To access the **Unify Polygons** tool:

- On the left panel of the Model layout, open the **Polygon** tab , click the dropdown arrow next to **Reduce**, and click **Unify**.
- Alternatively, on the right panel open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Polygon** > **Polygon Unify**.
- Alternatively, on the top left corner of the interface, click the **Schematic** palette icon , click **Add...**, and double-click **Mesh Operations** > **Deform** > **Polygon Unify**.

Polygon Unify Option



Mesh Operation

☑ Enable

☑ Use World Transform

☐ Force

- **Enable** - Enables or disabled the Push tool.
- **Use World Transform** - Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
- **Force unify** - Forces polygons to merge together to create one polygon.

Convert Text to Bezier

Precisely as the name implies, this command converts geometry created with the **Text** tool to Bezier curves. To use, type out a line of text using the **Text** tool, and then invoke the menu bar command **Geometry** > **Convert Text to Bezier**. Converted text becomes a collection of merged Bezier curves and can be further edited using the **Bezier Curve** tool. If desired, the **Make Curve Fill** command converts the curves back into render-able shapes.

Detriangulate

Converts triangle pairs to quadrangles by deleting edges to share triangle pairs. This conversion tool evaluates the flatness of the geometry.

- For Direct Modeling, the **Detriangulate** tool is found under the **Geometry** > **Polygon** menu. To enable this tool, first apply the Automatic Retopology Tool.
- For Procedural Modeling, the **Detriangulate** tool is found under the **Mesh Ops** tab. Click **Add Operator** and select **Mesh Operations** > **Polygons** > **Detriangulate**.

For more detailed information, see Detriangulate.

Using Selection Modifiers

The following tools make it easy to work with selections in Modo.

Convert Selection

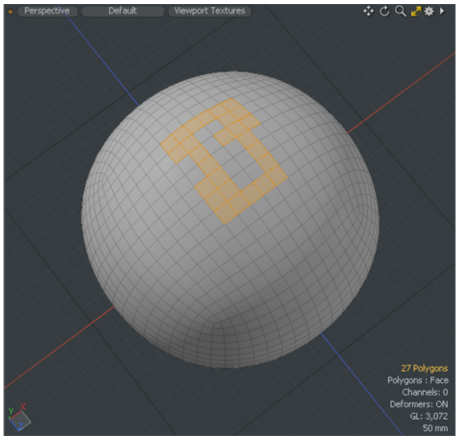
The Convert Selection tool sets a new selection mode specifically, using the content of the current selection to define the new one. You convert between selection types using the **Alt** key as a modifier. For example, if you have polygons selected in the viewport, holding the **Alt** key changes the option buttons on the Modo **Modes** toolbar to read **Convert**. Press the **Convert** button, where **Vertices** was previously displayed, and the current polygon selection is converted to a vertex selection. This way you can quickly access all the vertices that were part of the original polygon selection. The conversion types are:

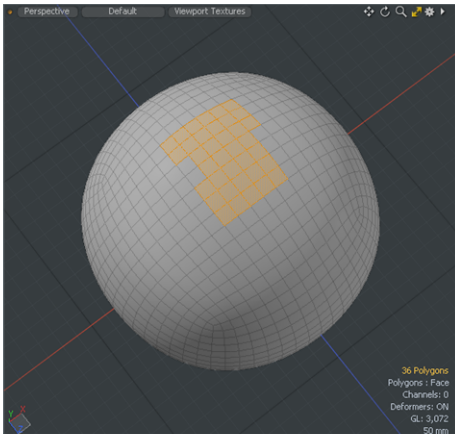
- **Vertex** : **edge** - selects edges defined by the selected points.
- **Vertex** : **polygon** - selects any polygons defined by the currently-selected points.
- **Edge** : **vertex** - selects points of all selected edges.
- **Edge** : **polygon** - selects polygons that are completely surrounded by selected edges.
- **Polygon** : **vertex** - selects points of all selected polygons.
- **Polygon** : **edge** - selects all the edges of the currently-selected polygons.

It's also possible to convert from a **Material** selection to any of the component level selections or the other way around.

Polygon Selection Fill

The **Polygon Selection Fill** tool selects the entire area within an existing enclosed selection. This tool is very similar to the Select Connected (] key) tool but fills in a selection outline instead of selecting the entire surface. To apply, make a selection that represents the outside perimeter of the desired selection area, and then press the **Ctrl**/**Cmd**+**Shift** modifiers and double-click inside the desired area to fill. The perimeter area must be fully connected, but it is not necessary that it be perfectly rectangular.


Perspective
Default
Viewport Textures
27 Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,072
50 mm


Perspective
Default
Viewport Textures
36 Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,072
50 mm

Grow Selection (Shift+Up Arrow)

The **Select** > **Grow** tool expands, or "grows", the selected elements by selecting all unselected geometry elements directly contiguous with the selection. Grow selection is mapped to **Shift**+up arrow so you can quickly expand your selection set. This tool works with all elements of geometry (Vertices, Edges, and Polygons) and is the functional opposite of the **Select** > **Shrink** tool.

Shrink Selection (Shift+Down Arrow)

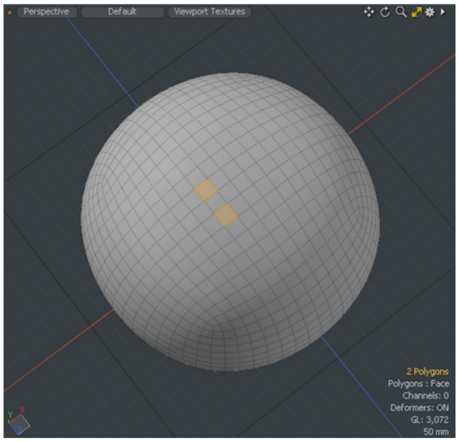
The **Select** > **Shrink** tool reduces the selection by deselecting elements that are at the "outside" of the currently-selected elements. This has the effect of "shrinking" the selection set to a smaller size. The keyboard equivalent of this tool is mapped to **Shift**+down arrow. This works with all geometry elements (Vertices, Edges, and Polygons).

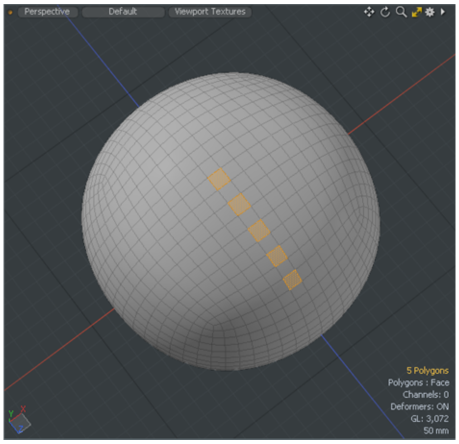
Note: It is possible to shrink your selection to nothing.

Select More/Less (Up/Down Arrows)

Selects the next plausible element. Select at least two polygons, edges, or vertices on your mesh item and then click the Up or down arrows to increase your selection. If a polygon was skipped then the next plausible element select would also skip a polygon.

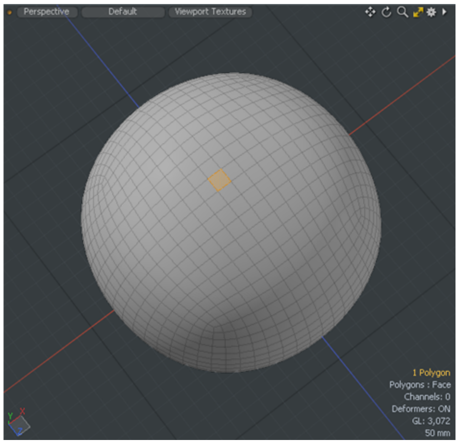
Note: It is possible to deselect everything.

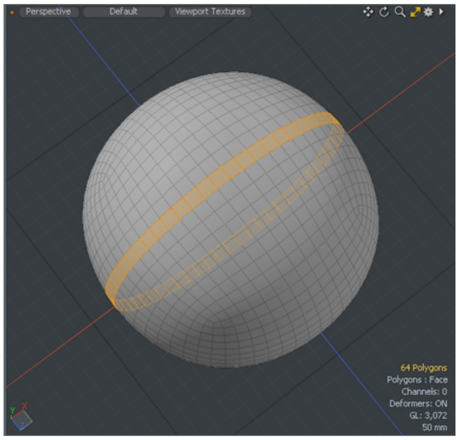

Perspective
Default
Viewport Textures
2 Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,072
50 mm


Perspective
Default
Viewport Textures
5 Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,072
50 mm

Select Loop (L key)

The **Select Loop** tool selects a continuous span of quadrangular polygons, edges, or vertices, depending on the selection mode. For polygons, select two adjoining polygons to define the loop direction and press the **L** key to select the entire loop. If not a full loop, selection terminates at non-standard geometry intersections.


Perspective
Default
Viewport Textures
1 Polygon
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,072
50 mm


Perspective
Default
Viewport Textures
64 Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 3,072
50 mm

For edge selections, a single edge can be selected and then press **L** to select the entire connected loop. If it is not a connected loop, the selection terminates at a multi-edged intersection. Edge loops can also be automatically selected by double-clicking on the target edge. Pressing **Ctrl**+**Shift**+double-click extends the loop beyond multi-edged intersections, based on the angle threshold, resolving toward the direction of the edge that is nearest to parallel with the adjoining edge.

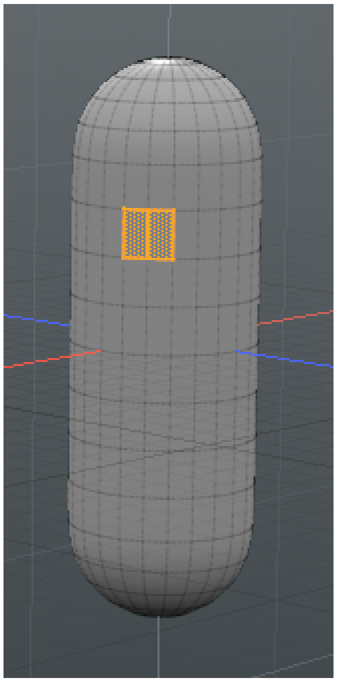
For vertices, a loop can be defined by selecting two consecutive vertices and pressing **L** to select the loop, terminating in the same way as an edge loop.

Procedural Select Loop

You can take a selection based on a previous selection operation and apply the mesh operation **Select Loop**. Once added, you can apply a previous selection operation in your **Mesh Operations** list to a polygon, edge, or vertices selection.

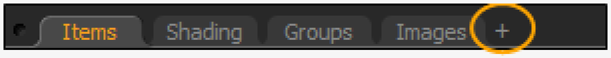
Example

1. In the **Model** layout, open the **Basic** tab on the left panel, and **Ctrl**/**Cmd** + click on the capsule icon.
2. On the right panel, click on your mesh item, and name it **Polygons**.
3. Under the layout menu bar, click **Polygons**, and select two polygons on the mesh item.



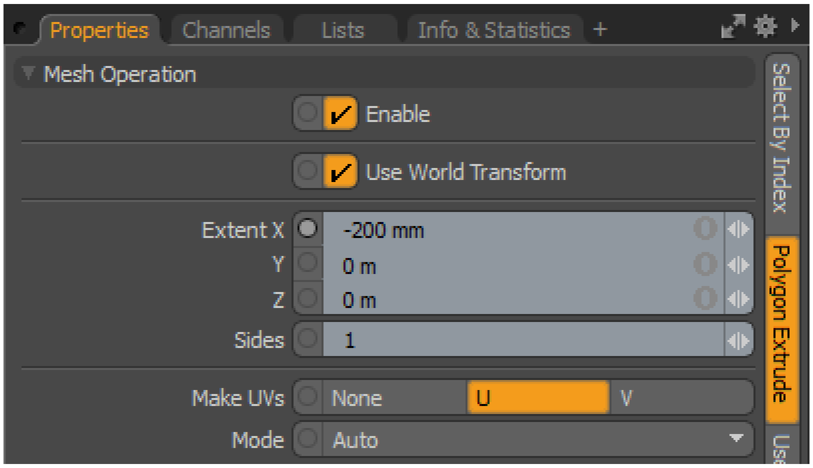
4. Open the **Mesh Ops** tab and click **Add Operator**.

Tip: If the Mesh Ops tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **New Tab** > **Data Lists** > **Mesh Ops**.



Double-click **Mesh Operations** > **Polygon** > **Polygon Extrude**.

Open the **Polygon Extrude** tab on the **Properties** tab and set **Extend X** to **-200 mm**.


Properties
Channels
Lists
Info & Statistics
+
Select By Index
Polygon Extrude
Use

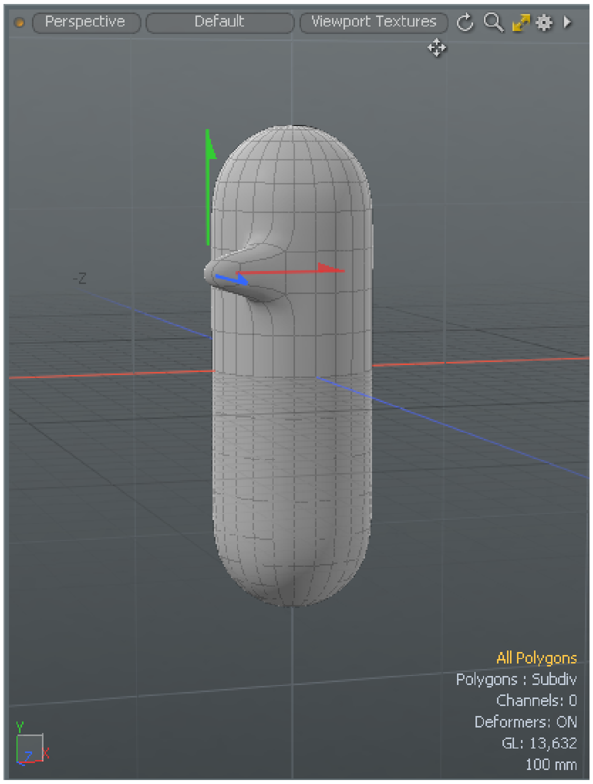
Mesh Operation

☑ Enable

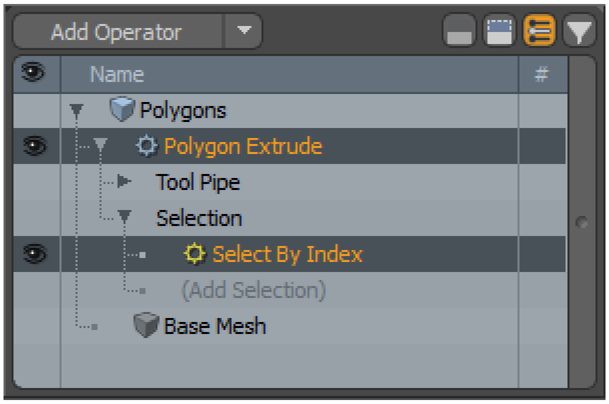
☑ Use World Transform

Extent X -200 mm
Y 0 m
Z 0 m
Sides 1
Make UVs None [✓ U] V
Mode Auto

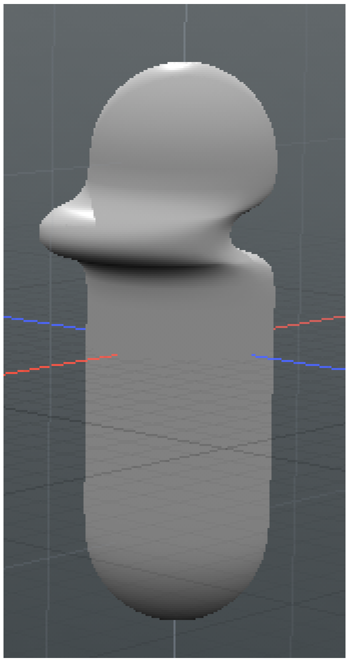
The selected polygon extrude from the cylinder.


Perspective
Default
Viewport Textures
-Z
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 13,632
100 mm
Y
Z
X

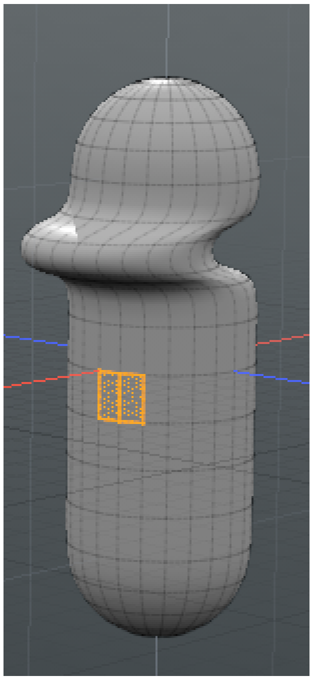
5. Expand the **Polygon Extrude** > **Selection**, click **(Add Selection)**, and double-click **Modifier Mesh Operations** > **New** > **Selection** > **Select Loop**. Alternatively, type **Loop** in the **Search All** field and double-click **Select Loop**.


Add Operator
Name
#
Polygons
Polygon Extrude
Tool Pipe
Selection
Select By Index
(Add Selection)
Base Mesh

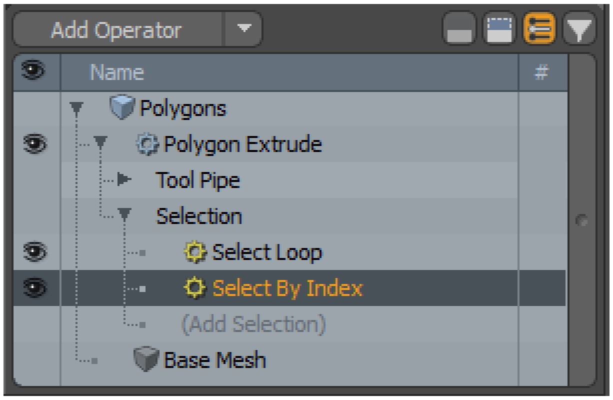
The loop selection is made and the polygons are extruded.



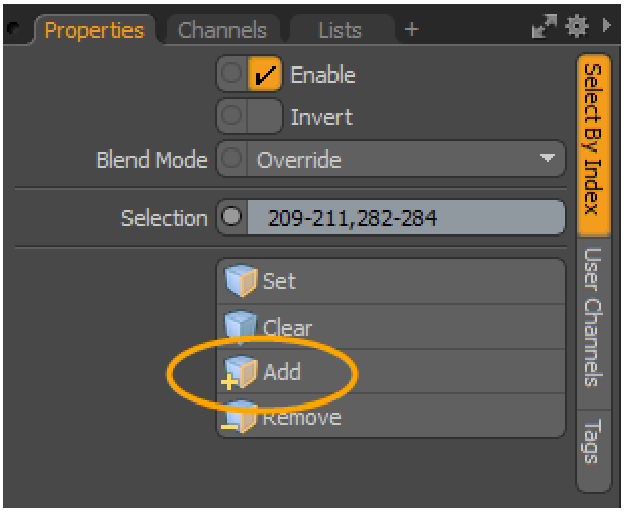
Ensure that Polygons selection mode is active and click + drag on the mesh item to select two other polygons.



On the right panel, expand **Polygons** > **Polygon Extrude** > **Selection**, and select the **Select by Index**.

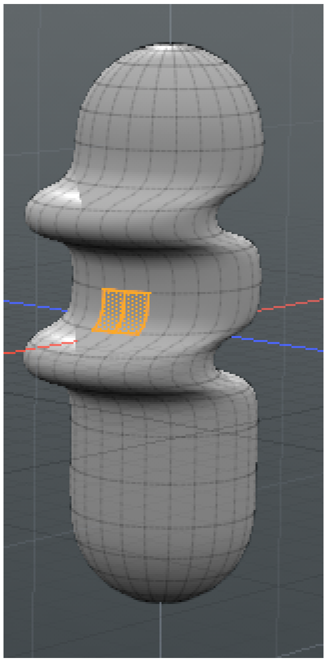

Add Operator
Name
#
Polygons
Polygon Extrude
Tool Pipe
Selection
Select Loop
Select By Index
(Add Selection)
Base Mesh

In the **Properties** panel, on the **Select By Index** tab, and click **Add**.


Properties
Channels
Lists
Select By Index
User Channels
Tags
Set
Clear
Add
Remove

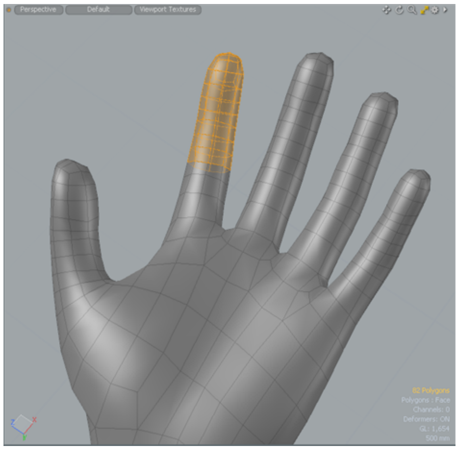
☑ Enable
☐ Invert
Blend Mode: Override
Selection: 209-211,282-284

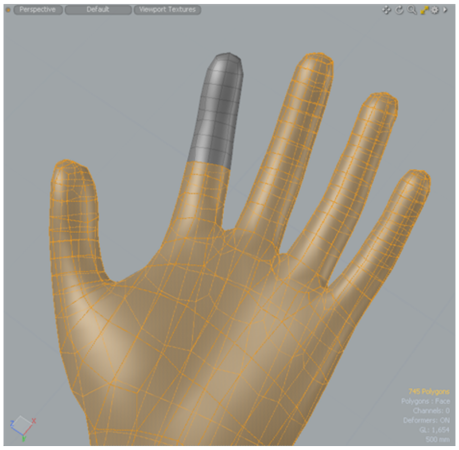
The previous selection operation is applied to your new selection.



Invert Selection ([key)

Accessed through the **Select** > **Invert** menu, this tool switches the state of all polygons, edges, or selected vertices. The selected elements become unselected and the unselected become the selected. Toggle the selection using the **Invert** shortcut key **[** (left square bracket).


Perspective
Default
Viewport Textures


Perspective
Default
Viewport Textures

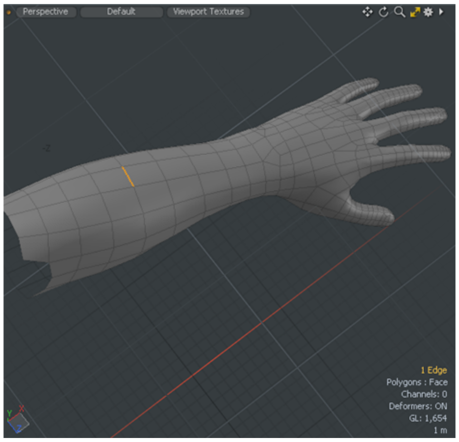
Alternatively, you can add the **Invert** tool as a procedural mesh operation. Open the **MeshOps** tab on the right panel, click **Add Operator** and double-click **Mesh Operations** > **Selection** > **Invert**.

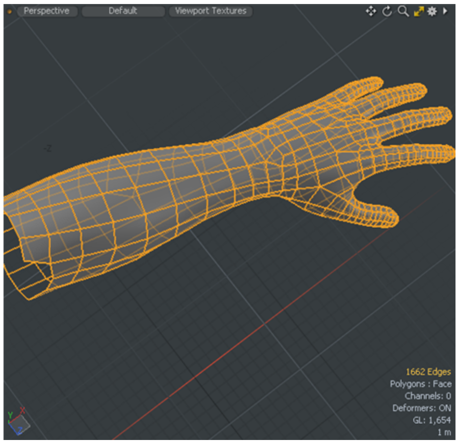
Select Connected (] key)

Accessed through the **Select** > **Connected**, this tool lets you quickly expand geometry selections to include all elements physically contiguous with the selection. This is a very handy workflow for selecting mesh geometry. You simply click to select a single polygon, edge, or vertex, then press **]** (right square bracket) and the entire piece of geometry is selected.

The **Invert Connected** tool (**Shift**+**[**) performs a similar function, but unlike **Select Connected**, the expanded geometry selection doesn't include your current selection.

Tip: While in **Polygon** or **Vertex** selection mode, you can also simply double-click on a vertex or polygon to select all connected elements.

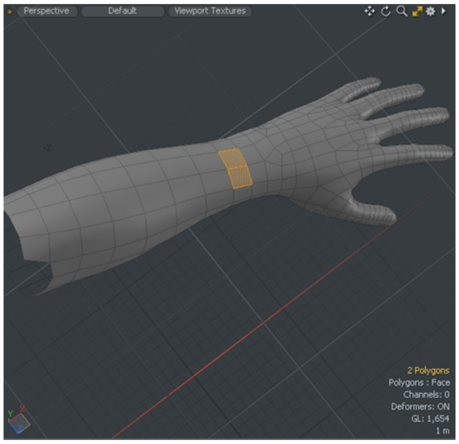

Perspective
Default
Viewport Textures
1 Edge
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,654
1 m

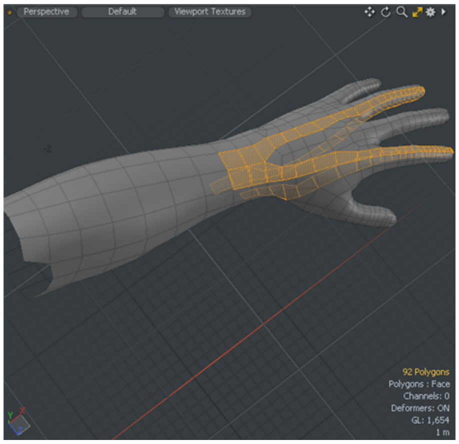

Perspective
Default
Viewport Textures
1662 Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,654
1 m

Alternatively, you can add the **Select Connected** tool as a procedural mesh operation. Open the **MeshOps** tab on the right panel, click **Add Operator** and double-click **Mesh Operations** > **Selection** > **Select Connected**.

Select Between (Shift+G)

The **Select Between** tool requires that two polygons or edges are selected prior to triggering the tool. Once the tool is triggered it completes the selection between the two based on the shortest distance between them. Only a single row is selected, if the initial polygons or edges were on the same loop. Polygons in adjoining rows or columns would create rectangular selection.


Perspective
Default
Viewport Textures
2 Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,654
1 m


Perspective
Default
Viewport Textures
92 Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,654
1 m

Boundary Edges

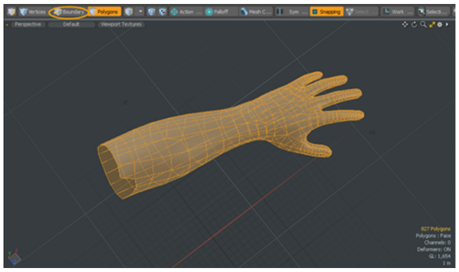
With polygons selected, this tool selects all the edges around the boundary edge of the polygon set. If no polygons are selected, it basically selects all the non-manifold edges of the mesh, which can be useful for finding holes. The mechanism is simply to select all edges bordered by an odd number of selected polygons. This tool works on both continuous and discontinuous selection sets.

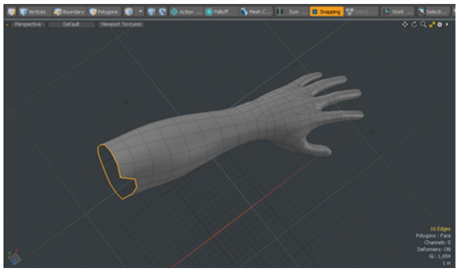
This tool is also available on the **Select** > **Boundary** menu or from the **Statistics** tab of the **Info and Statistics** viewport, by expanding the **Edges** > **By Boundary** entry. For more information, see Select by Info and Statistics.

To select a polygon boundary do the following:

1. Select the polygons on the mesh.
2. Press the **Ctrl**/**Cmd** key.

   Notice the **Edge** button now is labeled **Boundary**.
3. Click the **Boundary** button.

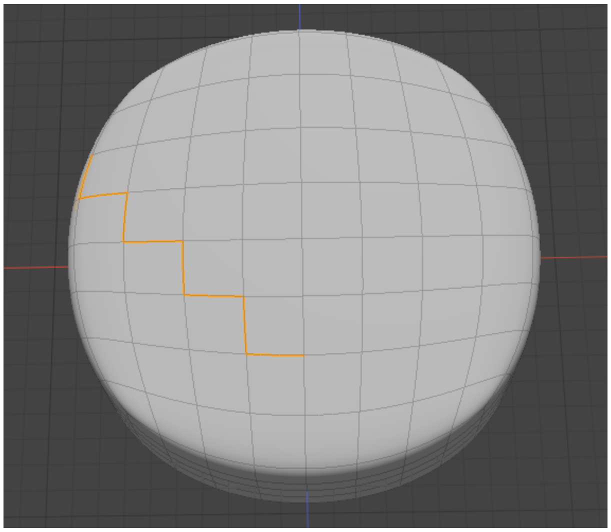




Select Symmetrical

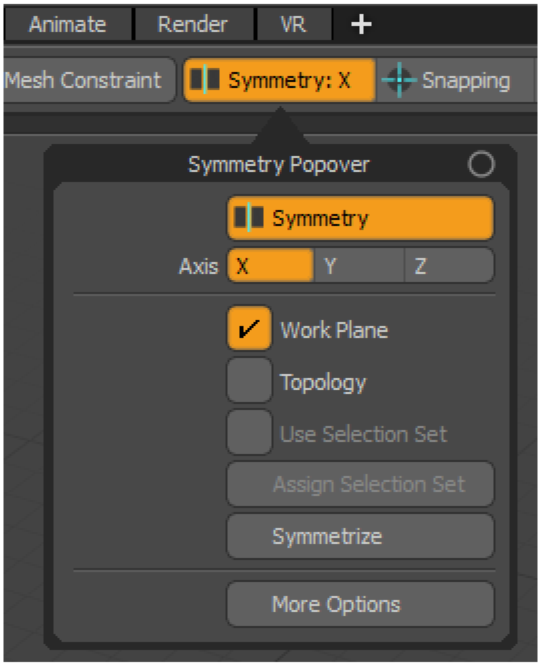
This command allows you to select any geometry that is symmetric to the current selection. To use the command:

1. Make a vertex, edge, or polygon selection on your mesh.



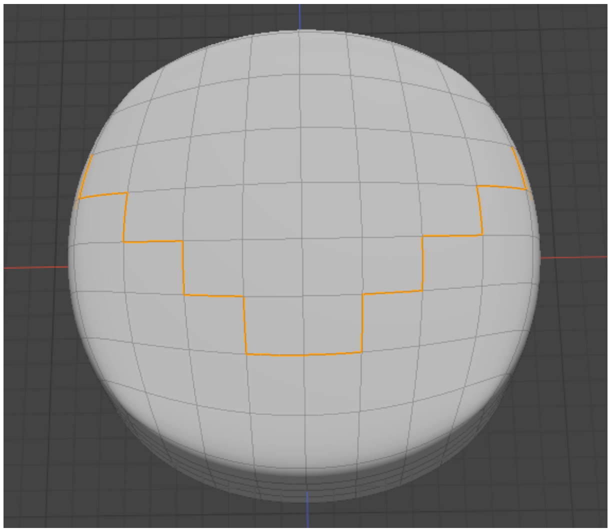
2. Enable Symmetry by clicking the **Symmetry** button above the 3D viewport.

   By default, the selected axis is X. To change the axis, hold down **Alt**. This changes the button to **Options**. Click it, then in the **Symmetry Popover**, select the required **Axis**.

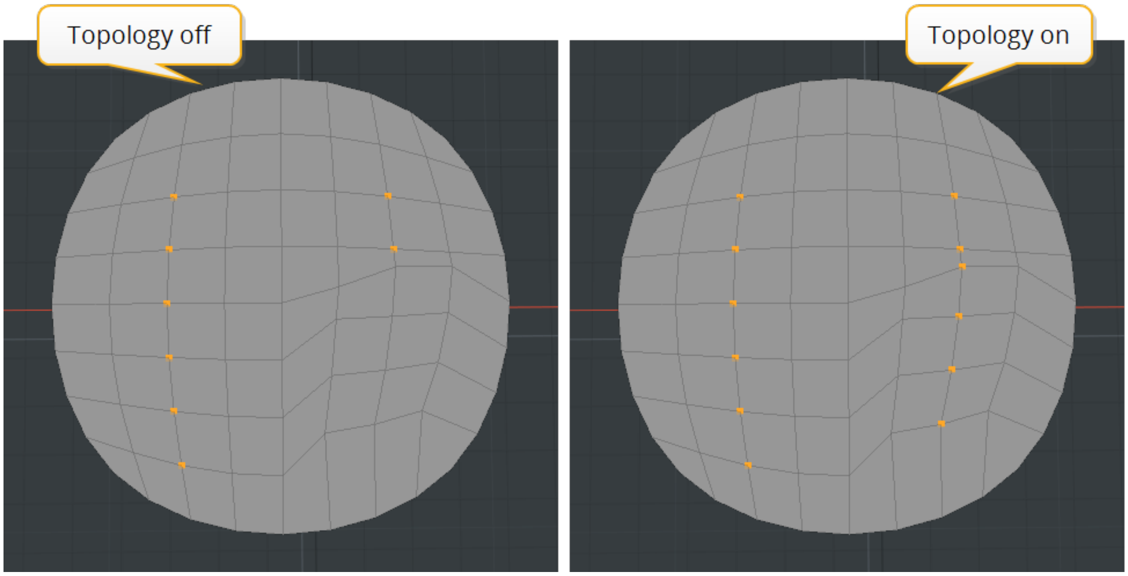

Animate
Render
VR
+
Mesh Constraint
Symmetry: X
Snapping
Symmetry Popover
Symmetry
Axis
X
Y
Z
Work Plane
Topology
Use Selection Set
Assign Selection Set
Symmetrize
More Options

3. In the menu bar, click **Select** > **Selection Commands** > **Symmetrical** to activate the tool.

   This adds symmetrically-corresponding vertices, edges, or polygons to your current selection.



When your mesh is mostly, but not entirely symmetrical, Modo doesn't always find all symmetrical elements. In such cases, enable **Topology** in the **Symmmetry Popover**. This uses a topological matching of components. For more information on working with Symmetry, see Symmetry.

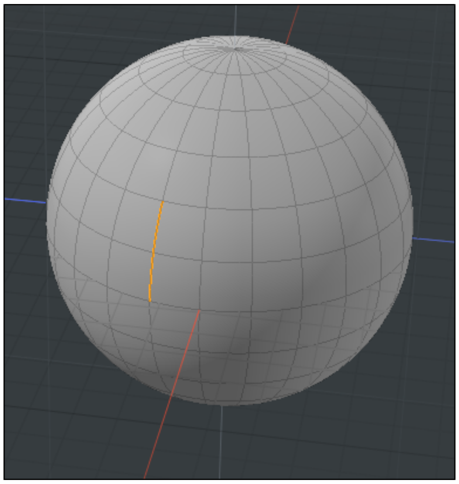


Select Ring (Alt+L)

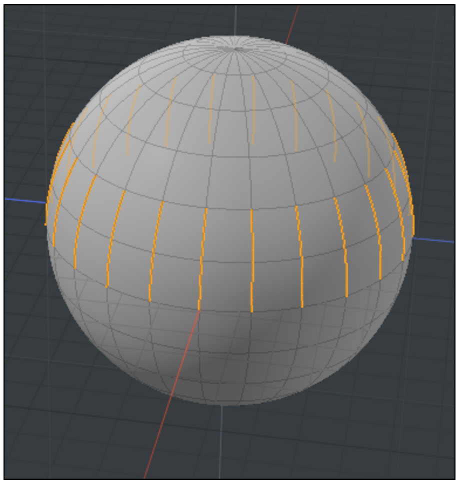
Geometric selections can be expanded to include all vertical or horizontal vertices or edges adjacent to the selection across quads. For example, in a sphere, when two vertical edges are selected.

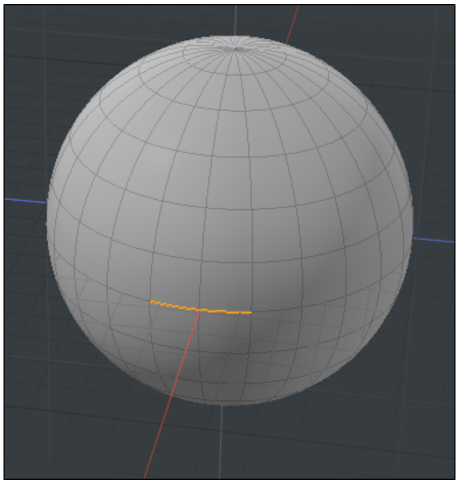
The **Ring** selection highlights all the vertical edges along the same selection direction around the geometry.

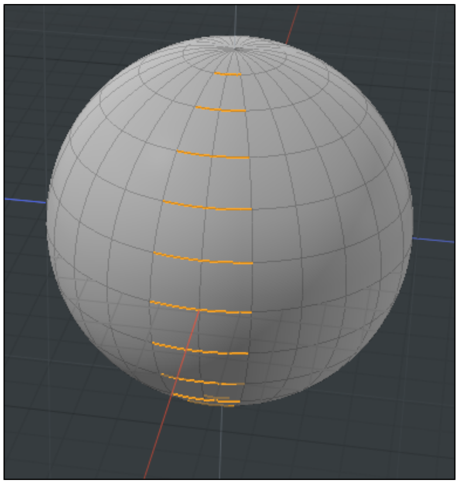
Before



After



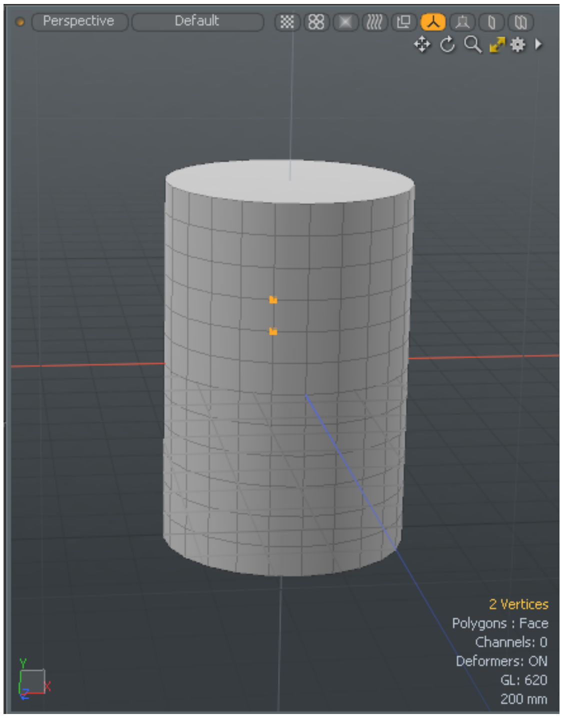




Direct Modeling Select Ring

1. In the 3D viewport, select two vertical vertices adjacent to each other on your mesh item.

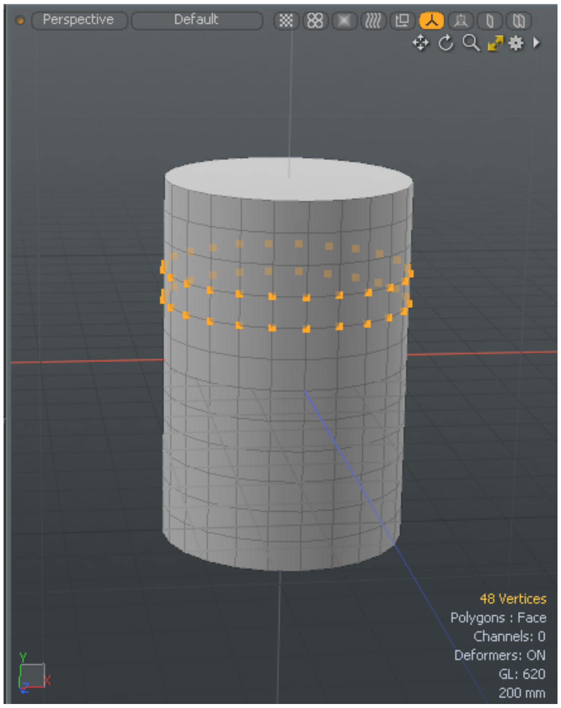
   For more information about selection modes, see Selecting Items.


Perspective
Default
2 Vertices
Polygons : Face
Channels: 0
Deformers: ON
GL: 620
200 mm
Y
X
Z

2. From the main menu, click **Select** > **Ring**.

   Alternatively, press **Alt**+**L**.

   The ring selection highlights all the vertices along the same selection direction around the geometry.

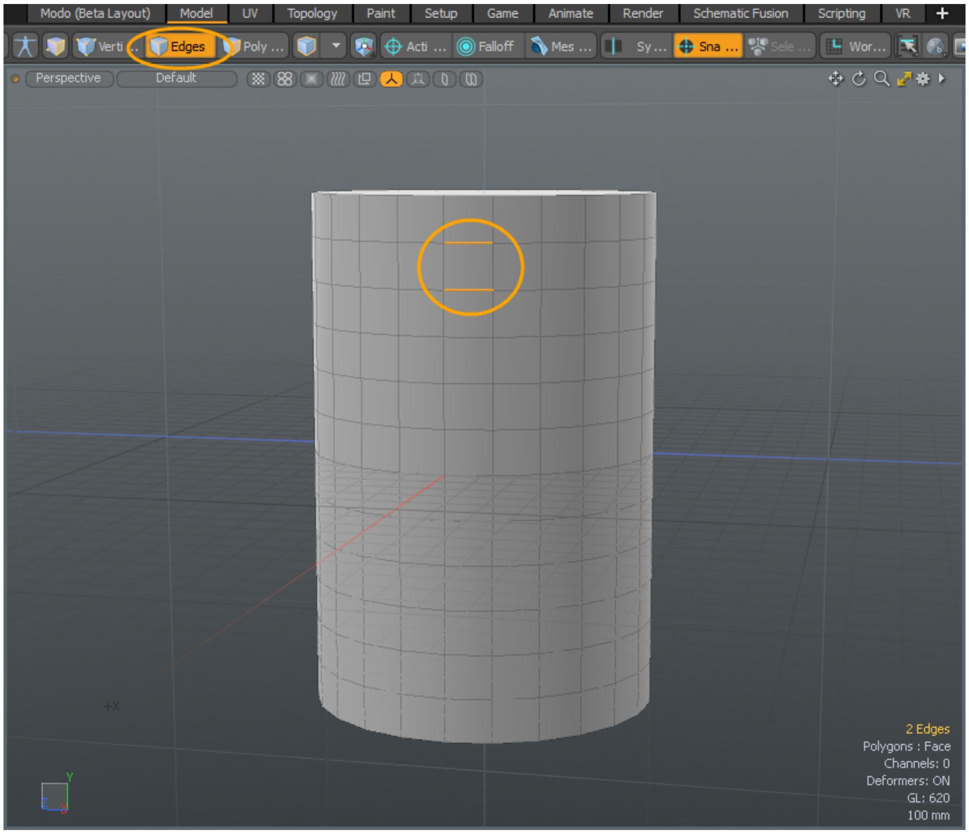

Perspective
Default
48 Vertices
Polygons : Face
Channels: 0
Deformers: ON
GL: 620
200 mm

Procedural Select Ring

We'll apply the Select Ring mesh operation to a previous Edge Bevel operation. You can see how easy it is to apply the mesh operation to the same selection direction around the geometry.

To apply the Select Ring tool:

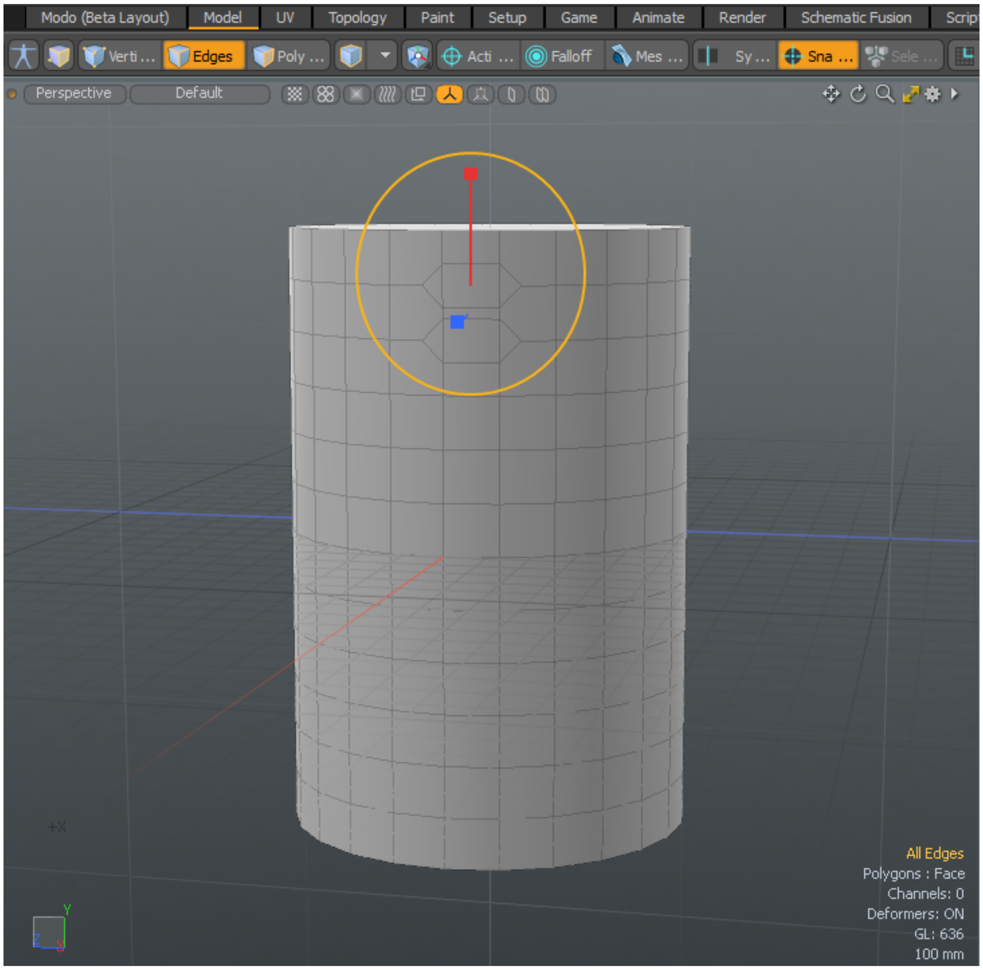
1. Add a Cylinder to your scene. Using the **Model** layout, open the **Basic** tab on the left panel, and **Ctrl/Cmd** + click Cylinder.
2. With **Edges** selection mode active, select two parallel edges.


Modo (Beta Layout)
Model
UV
Topology
Paint
Setup
Game
Animate
Render
Schematic Fusion
Scripting
VR
+
Verti ...
Edges
Poly ...
Acti ...
Falloff
Mes ...
Sy ...
Sna ...
Sele ...
Wor...
Perspective
Default
2 Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 620
100 mm

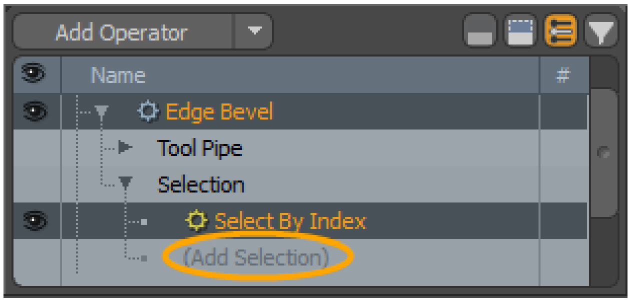
3. On the right panel, open the **MeshOps** tab, click **Add Operator**, and double-click **Mesh Operations** > **Edge** > **Edge Bevel**.

> **Tip:** If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

4. In the 3D viewport, move the handles to create a bevel, and press **Esc** to drop the tool.

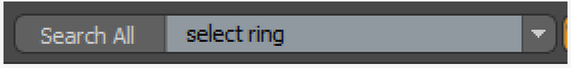

Modo (Beta Layout)
Model
UV
Topology
Paint
Setup
Game
Animate
Render
Schematic Fusion
Scrip
Verti ...
Edges
Poly ...
Acti ...
Falloff
Mes ...
Sy ...
Sna ...
Sele ...
Perspective
Default
+X
Y
All Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 636
100 mm

5. On the Mesh Ops panel, expand **Edge Bevel** > **Selection** > and click **(Add Selection)**.

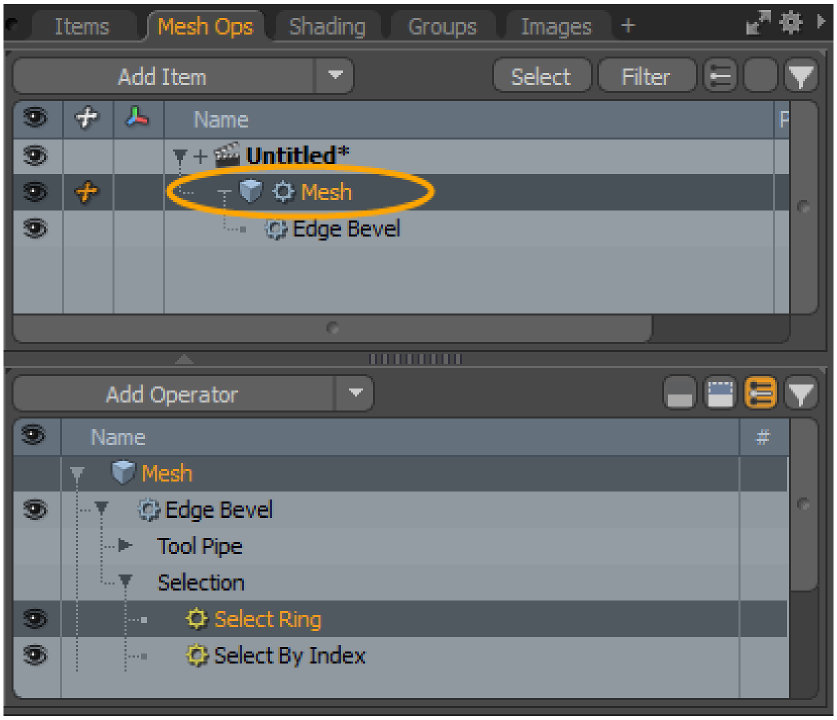

Add Operator
Name
#
Edge Bevel
Tool Pipe
Selection
Select By Index
(Add Selection)

6. In the **Add Operation** dialog, double-click **Modifier Mesh Operations** > **New** > **Selection** > **Select Ring**.

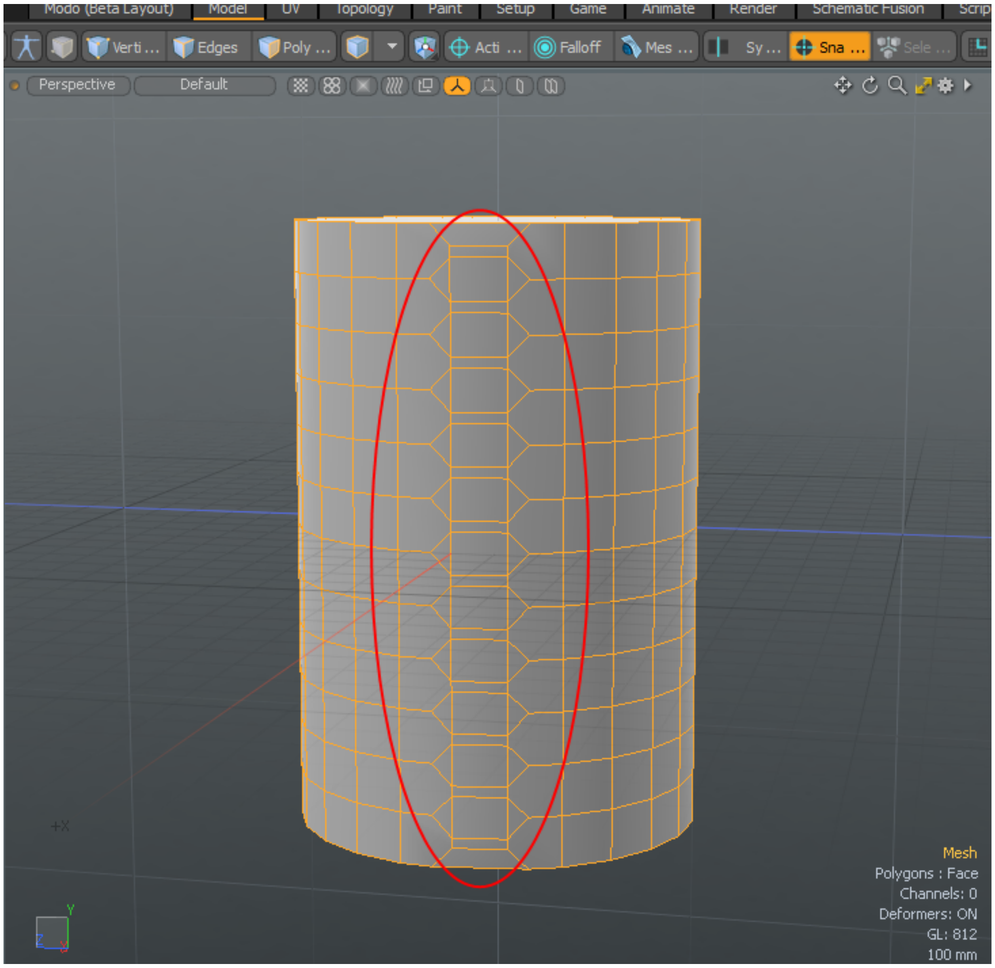
Tip: To quickly locate a mesh operator, type the name of the tool in the **Search All** field.


Search All
select ring

7. On the **Mesh Ops** panel, select the mesh item.

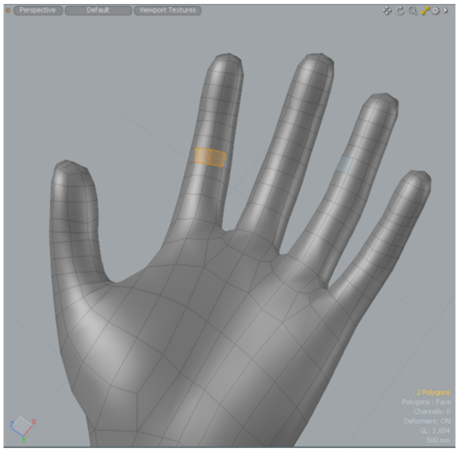

Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
Name
Untitled*
Mesh
Edge Bevel
Add Operator
Name
#
Mesh
Edge Bevel
Tool Pipe
Selection
Select Ring
Select By Index

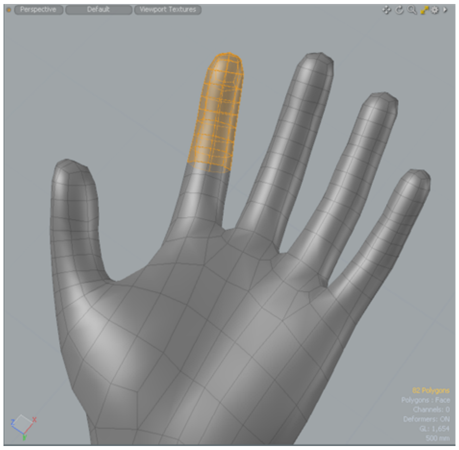
The Edge Bevel, originally applied only to the top two edges, is applied to the ring of edges along the same direction around the geometry.


Modo (Beta Layout)
Model
UV
Topology
Paint
Setup
Game
Animate
Render
Schematic Fusion
Scrip
Verti ...
Edges
Poly ...
Acti ...
Falloff
Mes ...
Sy ...
Sna ...
Sele ...
Perspective
Default
Mesh
Polygons : Face
Channels: 0
Deformers: ON
GL: 812
100 mm

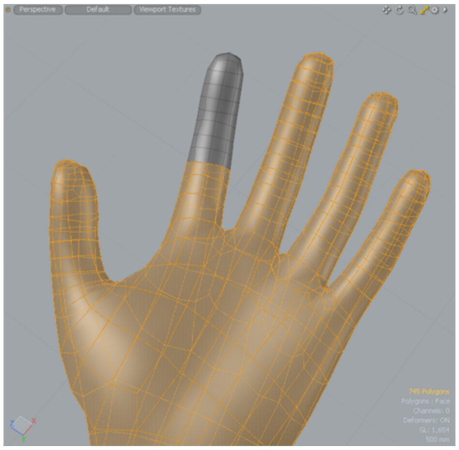
Select Close Loop (Shift+])

The **Select Close Loop** tool requires that two polygons are selected prior to triggering the tool. Once the tool is triggered it completes a loop selection using the two initial polygons to set direction and then selects all polygons on one side of the loop. The selection of the additional polygons is based on which side has fewer total polygons. For example, to select all polygons on a finger from the knuckle to the finger tip, you would select two polygons just above the knuckle, then apply the tool.


Perspective
Default
Viewport Textures

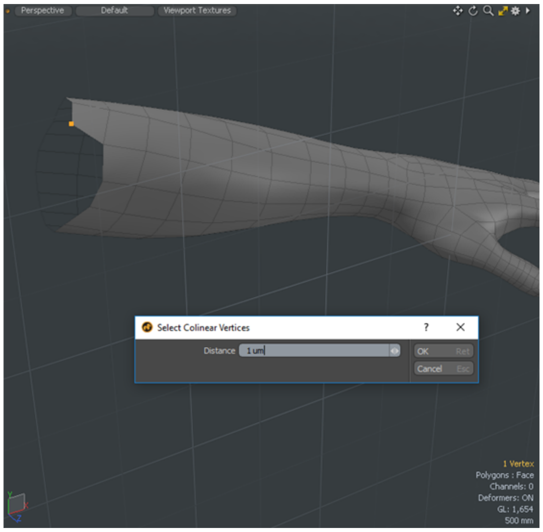

Perspective
Default
Viewport Textures

Since the finger itself has fewer polygons than the rest of the hand this is the side that is selected. This is generally the solution you are looking for, however, in the case that it's the opposite side of interest, the **Select Close Loop** tool can simply be followed by **Select Invert**.


Perspective
Default
Viewport Textures

Select Colinear

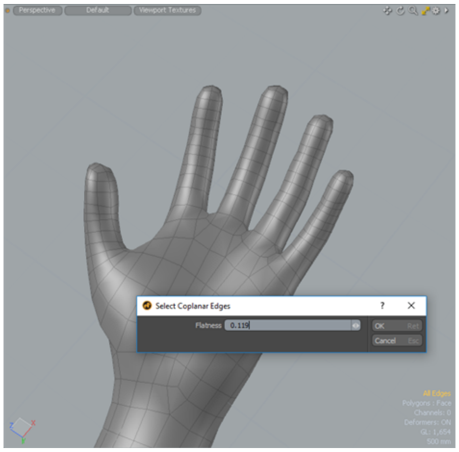
The **Select Colinear** tool selects any extraneous vertices across a polygon's boundary, making for easy removal. To use, select the tool from the menu bar **Select** > **Colinear** and a pop-up dialog asks you to define a distance. Click **OK** to execute the tool, resulting in the selection of all colinear vertices. In the example below, a vertex is selected in the area of the mesh which is not lying in the same straight line or linear sequence.


Perspective
Default
Viewport Textures
Select Colinear Vertices
?
OK Ret
Cancel Esc
1 Vertex
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,654
500 mm

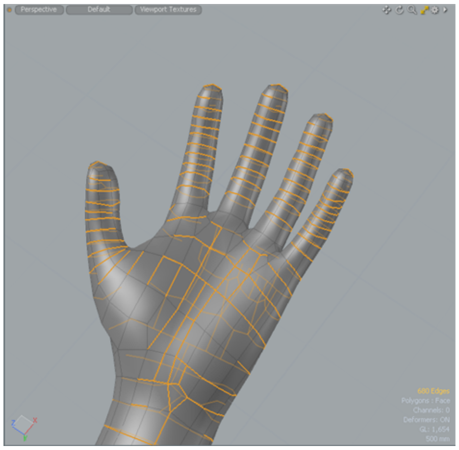
Distance 1 um

Select Coplanar

The **Select Coplanar** tool selects the edges between polygons that fall within a flatness range of a selected polygon. To use, select a polygon, and then select the tool from the menu bar **Select** > **Coplanar** and a dialog box appears, allowing you to specify a flatness limit (threshold between opposing triangle pairs). Click **OK** to execute the tool, resulting in an edge selection based on your input.


Perspective
Default
Viewport Textures
Select Coplanar Edges
?
OK Ret
Cancel Esc
All Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,654
500 mm

Flatness 0.119

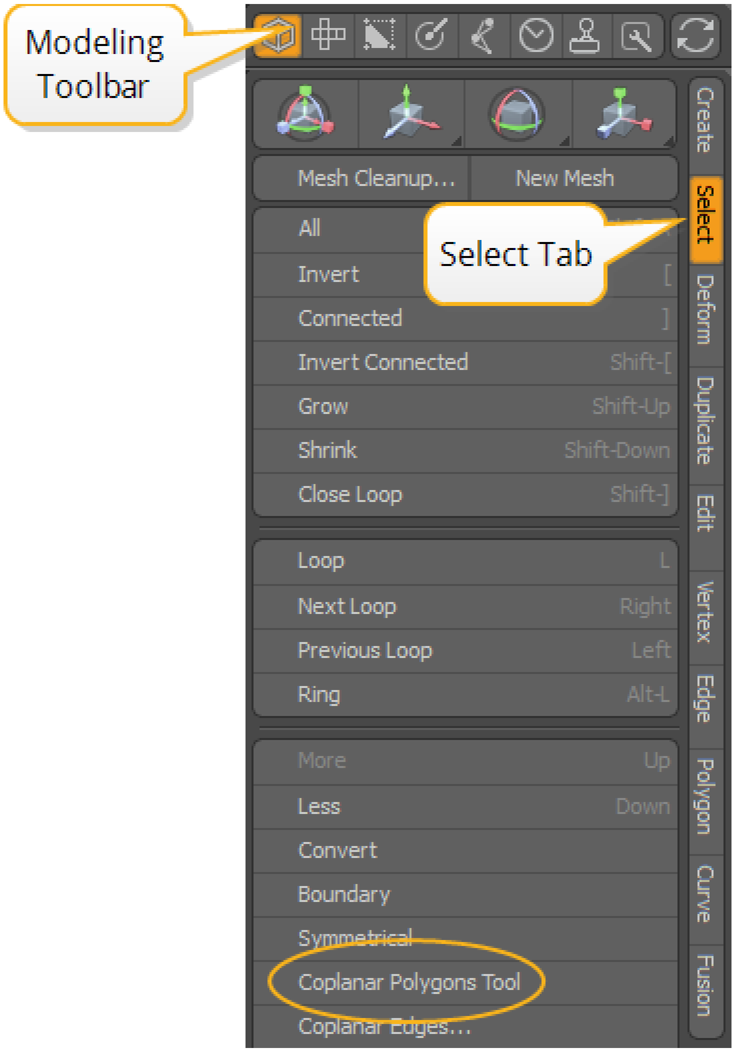

Perspective
Default
Viewport Textures
680 Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,654
500 mm

Coplanar Polygons Tool

The Coplanar Polygons tool allows you to select neighboring polygons on the same plane as the original selection. This tool works similarly to the Coplanar Falloff.

To use the tool:

1. With your scene open in the **Modo** layout, enable Polygons selection mode and select one or more polygons on your mesh.
2. In the Modeling toolbox's **Select** tab, click **Coplanar Polygons Tool**.


Modeling Toolbar
Select Tab
Mesh Cleanup...
New Mesh
All
Invert
Connected
Invert Connected Shift-[
Grow Shift-Up
Shrink Shift-Down
Close Loop Shift-]
Loop L
Next Loop Right
Previous Loop Left
Ring Alt-L
More Up
Less Down
Convert
Boundary
Symmetrical
Coplanar Polygons Tool
Coplanar Edges...
Create
Select
Deform
Duplicate
Edit
Vertex
Edge
Polygon
Curve
Fusion

Note: You can also access the tool from the menu bar. Click **Select** > **Selection Commands** > **Coplanar Polygons Tool**.

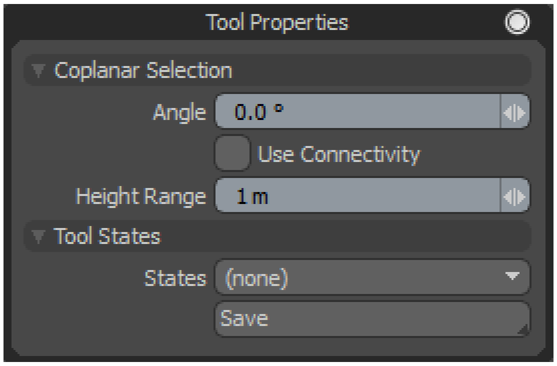
3. Click in the viewport to activate the tool.

   This selects all polygons that are coplanar with the original selection.

When the tool is active, you can adjust its settings to influence which polygons are selected.

1. Click **Tool Properties** on the left panel.

   This opens the tool properties.



Tool Properties

Coplanar Selection

Angle 0.0 °

☐ Use Connectivity

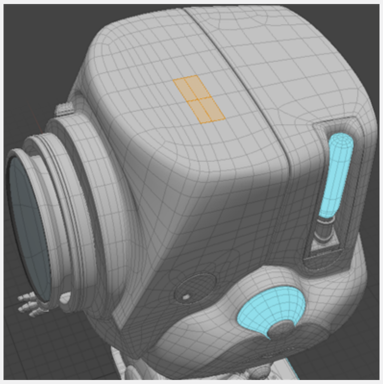
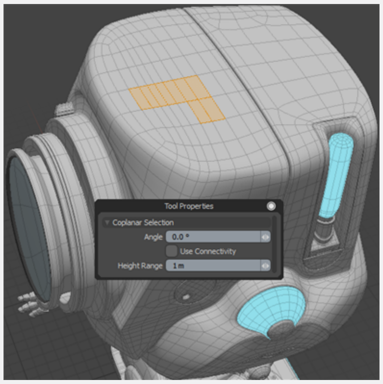
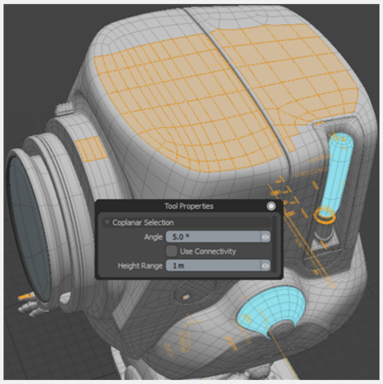
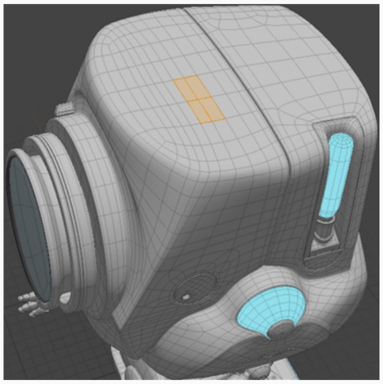
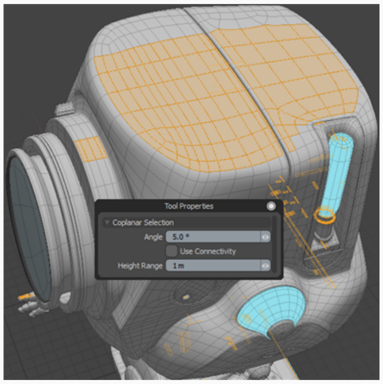
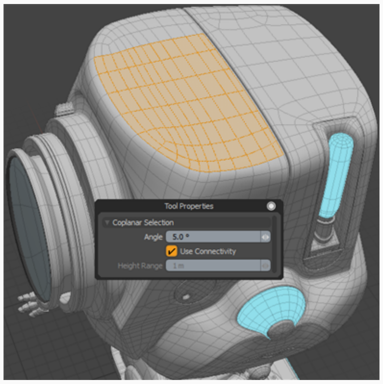
Height Range 1 m

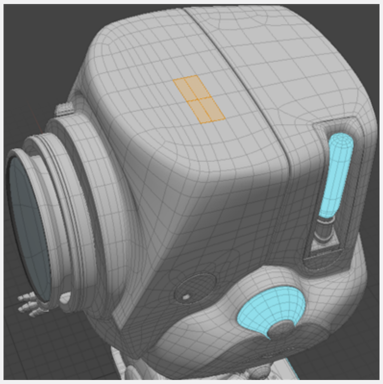
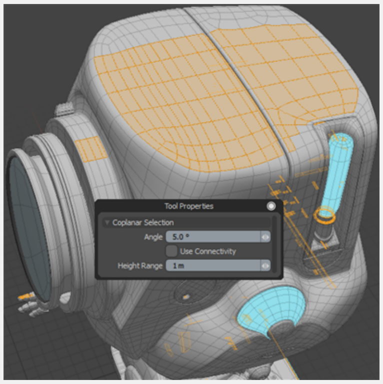
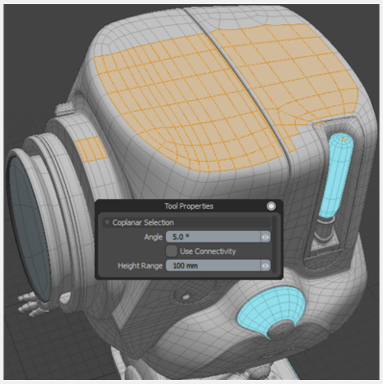
Tool States

States (none)

Save

You can adjust the following properties:

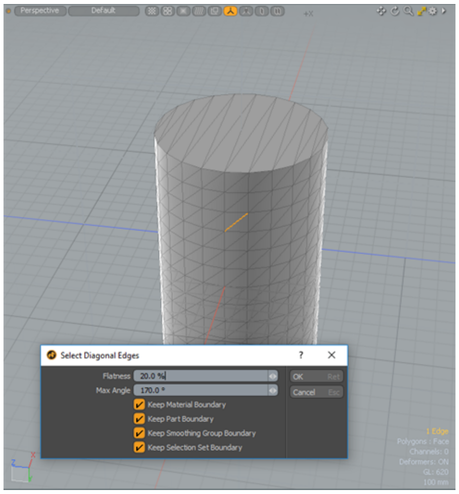
Angle	Determines the range outside the selected area to attenuate the effect of the tool. Increasing the value selects more polygons.  Original Polygon Selection  Coplanar polygons with default values  **Angle** at 5°
Use Connectivity	When enabled, only connected elements are selected. Unconnected elements within range are ignored.  Original Polygon Selection  **Use Connectivity** disabled  **Use Connectivity** enabled

Height Range	Determines the maximum threshold to include polygons with the same normal, but not on the same infinite plane. This is used when **Use Connectivity** is disabled. At 0 m, the tool selects only polygons on the infinite plane of the source polygons. When **Height Range** is higher than 0, the tool selects polygon with same normal direction with different height position in the specific range. The default value is 1 m.  Original Polygon Selection  Tool Properties Coplanar Selection Angle 5.0 ° ☐ Use Connectivity Height Range 1 m **Height Range** at 1 m  Tool Properties Coplanar Selection Angle 5.0 ° ☐ Use Connectivity Height Range 100 mm **Height Range** at 100 mm

The Coplanar Polygons Tool's procedural equivalent is the Select Coplanar Polygons mesh operation. For more information, see Select Coplanar Polygons.

Select Diagonal

The **Select Diagonal** tool selects edges to share triangle pairs. This is useful when you want to convert triangles to quadrangles. You can convert triangles to quadrangles by deleting the selected edges.



Select Diagonal Edges

Flatness 20.0 %

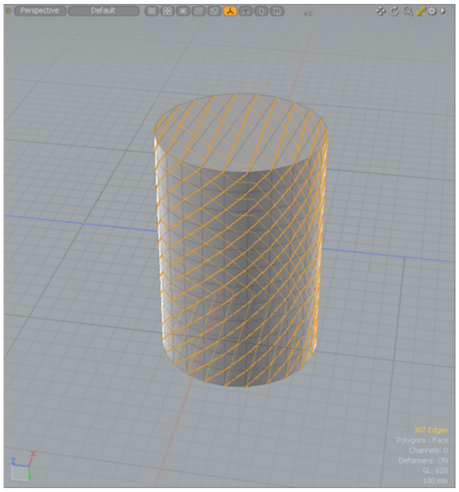
Max Angle 170.0 °

☑ Keep Material Boundary

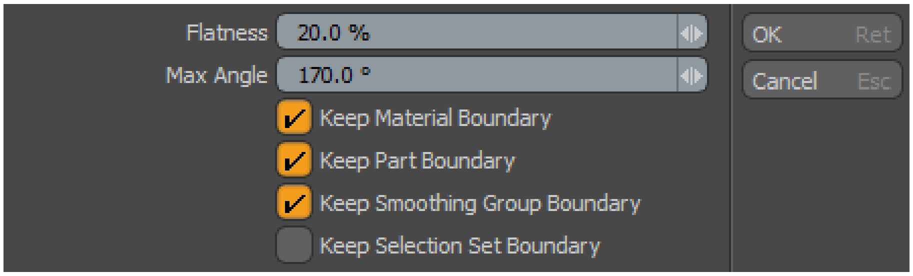
☑ Keep Part Boundary

☑ Keep Smoothing Group Boundary

☑ Keep Selection Set Boundary



To use, select an edge, and then select the tool from the menu bar **Select** > **Diagonal...** A dialog appears allowing you to specify options. Specify the options and click **OK**.



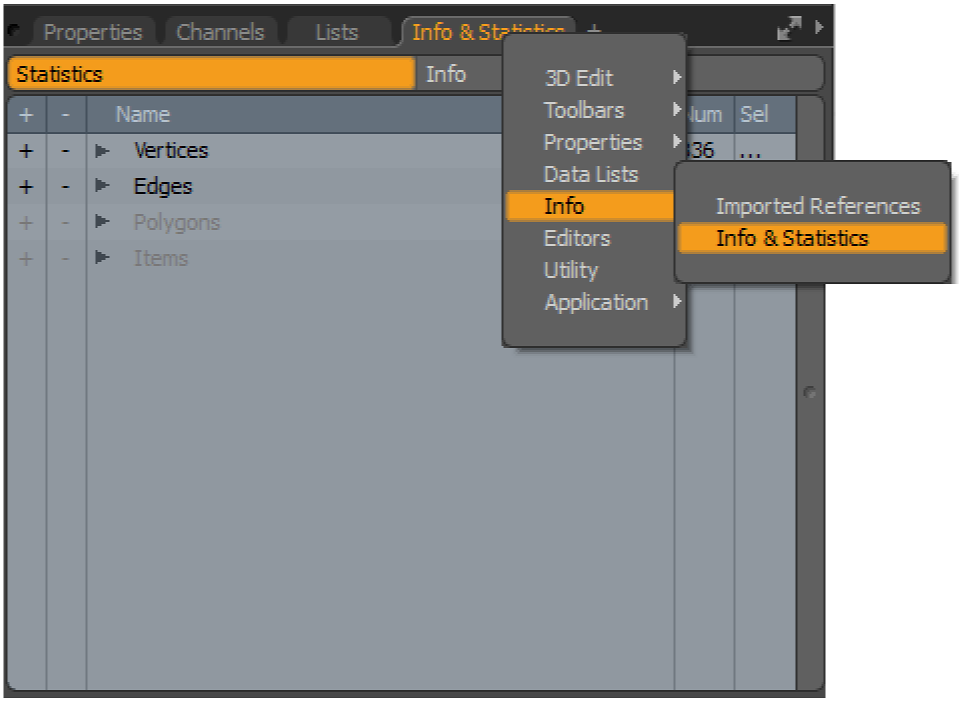
Flatness 20.0 %
Max Angle 170.0 °
☑ Keep Material Boundary
☑ Keep Part Boundary
☑ Keep Smoothing Group Boundary
☐ Keep Selection Set Boundary

OK Ret
Cancel Esc

- **Flatness** - this is the Coplanar flatness threshold between opposing triangle pairs, designating whether the edge is selected or not.
- **Max Angle** - this is the maximum angle for the corners of a quadrangle from the triangle pair.
- **Keep Material Boundary** - when enabled, edges that fall along material boundary are not selected.
- **Keep Part Boundary** - when enabled, edges that fall along a part boundary are not selected.
- **Keep Smoothing Group Boundary** - when enabled, edges that fall along a smoothing group boundary are not selected.
- **Keep Selection Set Boundary** - when enabled, selection set boundary are not selected.

Select by Info and Statistics

You can also select elements by using the Info and Statistics viewport. To open this viewport, on the right panel, click the plus button and select **Info** > **Info & Statistics**. For more information, see Info and Statistics.

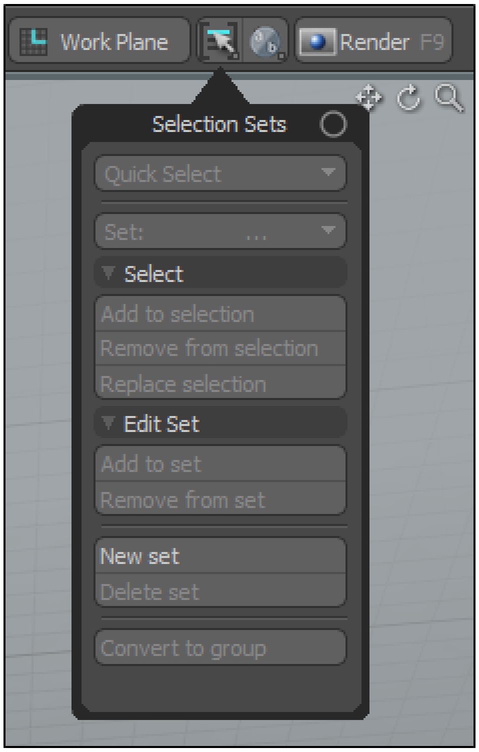


Properties
Channels
Lists
Info & Statistics
Statistics
Info
Name
Num
Sel
Vertices
Edges
Polygons
Items
3D Edit
Toolbars
Properties
Data Lists
Info
Editors
Utility
Application
Imported References
Info & Statistics

Using Selection Sets

Modo offers a range of tools for editing geometry to make it easy to generate accurate selections. If you have to make identical selections repeatedly for certain actions, you can create selection sets to save and edit component- and item-based selections to recall at a later time. You can assign selection sets to vertex, edge, polygon, and item selections. You can also use polygon selection sets to create additional mask layers in the Shader Tree for creative layering. Selection sets are flexible in how you can edit them. Unlike material or part tags, where you can only assign a single tag at a time, you can assign any component or item to as many sets as you wish.

You can access the commands for working with selection sets by clicking the **Selection Sets** button on the Modes toolbar. Click the button to open the **Selection Sets** panel.


Work Plane
Render F9
Selection Sets
Quick Select
Set: ...
Select
Add to selection
Remove from selection
Replace selection
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group

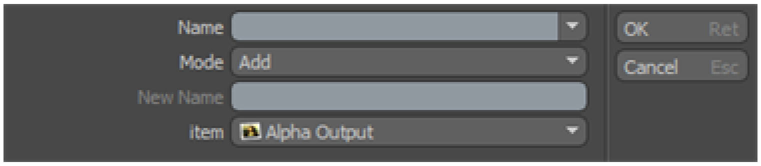
Alternatively, you can find the commands in the menu bar, under **Select**.

Defining Selection Sets

To create a selection set, you first need to define the selection to save. For information on creating selections, see Selecting Items. Modo indicates selected (active) elements with an orange color.

Once you have selected elements, assign them to a new selection set by doing either of the following:

- In the **Selection Sets** panel available from the Modes toolbar, click **New set**. Name your selection set, then click **OK**.
- From the menu bar, click **Select** > **Assign Selection Set**. In the **Edit Selection Set** dialog, specify a name for the set, set the mode to **Add**, then click **OK**.



Name

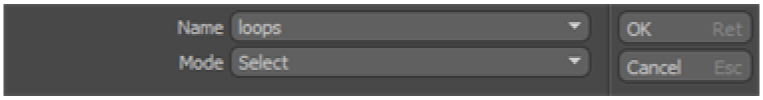
Mode Add

New Name

Item Alpha Output

To recall a selection set, do either of the following:

- In the **Selection Sets** panel, click **Quick Select**, and choose the appropriate selection set from the dropdown.
- From the menu bar, click **Select** > **Use Selection Set**. In the **Use Selection Set** dialog, specify a selection set and indicate whether to select or deselect the polygons in that grouping. Click **OK** to make the selection.



Name loops

Mode Select

Note: To recall or modify any existing selection set, you must be in the same mode as when you created the selection set. For example, you can only recall a vertex selection set while in **Vertices** mode.

Editing Selection Sets

You can add or remove selections from selection sets using either the **Selection Sets** button, or the **Select** menu.

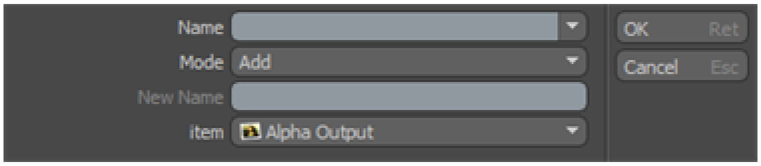
> **Note:** To recall or modify any existing selection set, you must be in the same mode as when you created the selection set. For example, you can only recall a vertex selection set while in **Vertices** mode.

To edit a selection set in the **Selection Sets** panel:

1. In the 3D viewport, select the elements you want to add or remove from the set.
2. In the **Selection Sets** panel, click the **Set** field and click the selection to modify.
3. Under **Edit Set**, click **Add to set** to add the new selection to the set, or **Remove from set** to remove the selected items from the set.

You can also use the menu bar to update existing selection sets.

1. In the 3D viewport, select the elements you want to add or remove from the set.
2. Click **Select** > **Assign Selection Set** to display the **Edit Selection Set** dialog.


Name
Mode Add
New Name
Item Alpha Output

3. Click the arrow on the right of the **Name** field to reveal a dropdown of your existing selection sets. Click the one to modify.
4. In the **Mode** field, select **Add** to add the active selection to the set, or **Remove** to delete it from the set, then click **OK**.

   Your selection set is updated.

The **Edit Selection Set** dialog contains the following fields.

Name	Specifies a name for a new set. By default, when creating a new set, the **Mode** option is **Add**. If the set already exists, you can specify the set to edit from the list. To rename an indicated selection set, select **Rename** as the **Mode** to be able to change the name.
Mode	Indicates options for modifying existing selection sets. • **Add** - Adds any active selections to an existing set. • **Remove** - Removes any active selections from an existing selection set. • **Rename** - Assigns a new name to the selection set. • **Set** - Replaces the elements in the selection set with the elements of the current selection.

New Name	Allows you to change the name of the specified selection set. To rename an existing selection set, select the existing set from the **Name** list, set the **Mode** to **Rename**, then type the new name in the **New Name** box, and click **OK**.

Using Selection Sets

You can recall selection sets using either the **Selection Sets** button, or the **Select** menu.

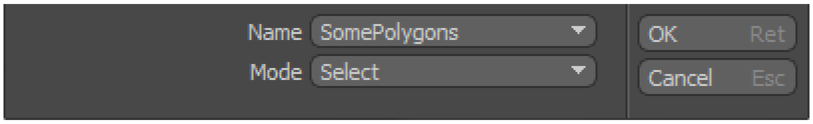
In the **Selection Sets** panel:

- To quickly activate a selection set, click **Quick Select**, and select the set you need.
- If you have elements selected in your scene, click the **Set** field and specify the selection set you need, then choose the appropriate action:
  - **Add to selection** - Adds the selection set to the current selection.
  - **Remove from selection** - Removes the elements in the selection set from the current selection.
  - **Replace selection** - Replaces the current selection with the selection set.

In the menu bar:

1. Click **Select** > **Use Selection Set**.

   A dialog displays, where you can recall existing selection sets.



Name SomePolygons

Mode Select

2. In the **Name** field, specify the set you want to use.

Note: Only sets created with the active selection mode are available in the list (**Vertices**, **Edges**, **Polygons**, or **Items**).

3. In the **Mode** field, click one of the options:
   - **Select** - Activates the specified selection set. Modo adds the set to any pre-exisiting active selections.
   - **Deselect** - Deactivates the specified selection set. Modo removes the set from any pre-exisiting active selections.
   - **Replace** - Replaces the active selection with the specified set.

Tip: You can also recall selection sets by using the Info and Statistics viewport. Each section has a listing of all the available named sets. You can select a named set by clicking the **+** (plus) button in the viewport. Conversely, you may deselect the geometry in a set by clicking the **-** (minus) button.

Deleting Selection Sets

You can add or remove selections from selection sets using either the **Selection Sets** button, or the **Select** menu.

Note: Only sets created with the active selection mode are available to delete (**Vertices**, **Edges**, **Polygons**, or **Items**).

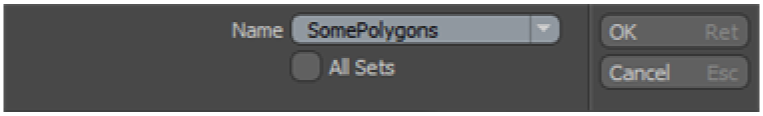
To delete a selection set in the **Selection Sets** panel:

1. Click the **Selection Sets** button in the Modes toolbar.
2. In the **Set** field, specify the set to remove.
3. Click **Delete set**.

To delete a selection set in the menu bar:

1. Click **Select** > **Delete Selection Set**.

   The **Delete Selection Set** dialog displays.



Name SomePolygons

☐ All Sets

2. In the **Name** field, select the set you want to delete, then click **OK**.

To remove all selection sets, enable the **All Sets** checkbox.

Selection Set Groups

You can organize items using a specific selection set into a selection set group. Existing selection sets can be converted into a selection set group. This allows you to use the selection set workflow to specify groups of items that you can use for operations, such as group-masking the **Shader Tree**.

Note: Selection set groups are only available in **Items** selection mode.

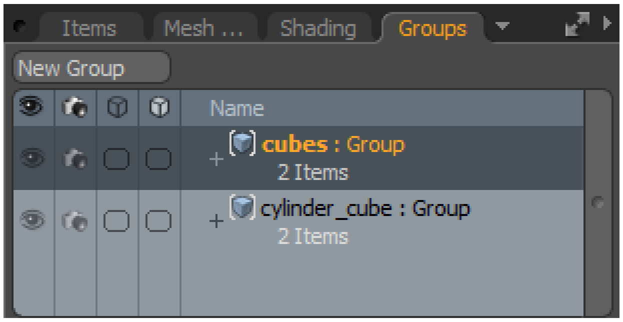
To create a selection set group:

1. Activate the selection set you want to convert to a group.
2. In the **Selection Sets** panel, click **Convert to group**.

   OR

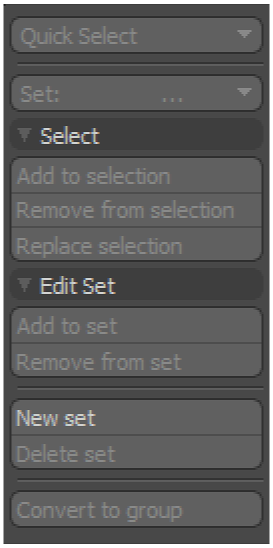
   From the menu bar, click **Select** > **Create Selection Set Group** and specify the name of the set in the **SSET** field.

   A group is created, containing the items that are part of the specified selection set. It also appears in the **Groups** tab.

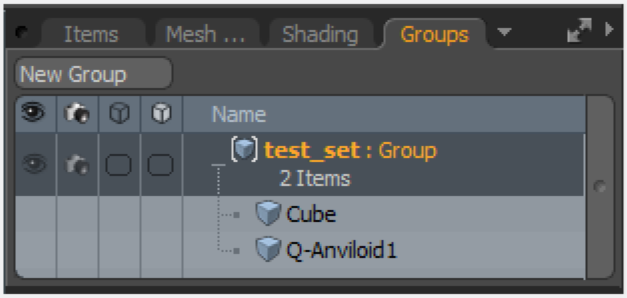

Items
Mesh ...
Shading
Groups
New Group
Name
cubes : Group
2 Items
cylinder_cube : Group
2 Items

Note: The **Groups** tab can be found on the right panel of the UI. If it's not visible, you may need to add it by clicking the arrow on the right side of the tabs and clicking **New Tab** > **Utility** > **Groups**. For more information, see Groups Viewport.

Selection Set Options


Quick Select
Set: ...
Select
Add to selection
Remove from selection
Replace selection
Edit Set
Add to set
Remove from set
New set
Delete set
Convert to group

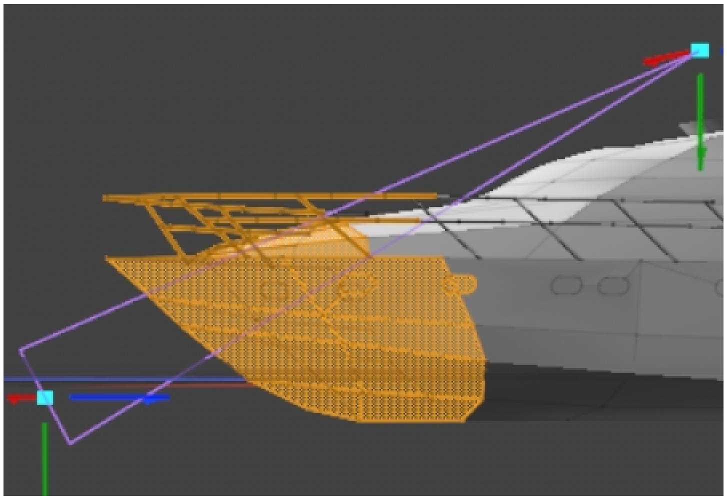
The following menu options are available:

Menu Option	Description
Quick Set	Restores a selection set, replacing the current selection with that defined in the Set.
Set	Allows you to specify the working set. The Select and Edit Set options applies to the selected Set.
Select	
Add to Selection	Adds the current selection to the specified Set.
Remove from selection	Removes the current selection from the specified Set.
Replace selection	Replaces the current selection specified in the Set.
Edit Set	
Add to set	Adds the current selection to the specified Set.
Remove from set	Removes the current selection from the specified Set.
New Set	Creates a new set containing the selected items in your scene.
Delete Set	Deletes the set specified in Set.
Convert to Group	Creates a new group item in the scene and links that to the selection set. Adding an item to the group also tags it with the selection set and adding the selection set to an item also adds it to the group. The selection set name is used for the group created and is displayed under the **Groups** tab.  Items Mesh ... Shading Groups New Group Name test_set : Group 2 Items Cube Q-Anviloid1 This allows you to, for example, to use selection sets on items to define shading in the Shader Tree by having a **Group** mask that filters on that selection set group.

Note: This only works in Items Selection Mode Selection Mode.

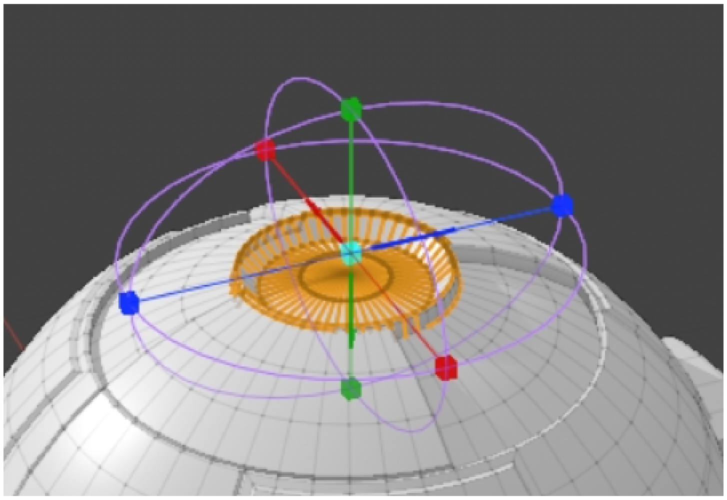
Make Selections Fast Using Falloffs

Modo includes an array of powerful falloff tools that you can use to simplify making selections in the viewport. The direct modeling tool **Select from Falloff** converts falloff weight to a component selection, allowing you to quickly add and remove vertices, edges, and polygons from your selection before applying other modeling tools to the updated selection. Falloff selection has four modes:



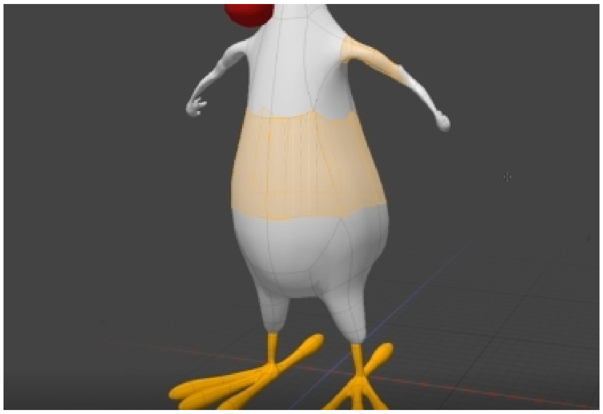
Linear - Linear falloff uses a straight line to define the falloff. The line itself determines the direction of the falloff's influence. All elements at the **Start** of the falloff are selected and none of the elements at the **End** of the falloff are selected. See Linear Falloff for more information.

The example shows using a **Linear** selection falloff in **Polygons** selection mode to select only the front polygons on a model ship. You could then easily assign a material to just those polygons.



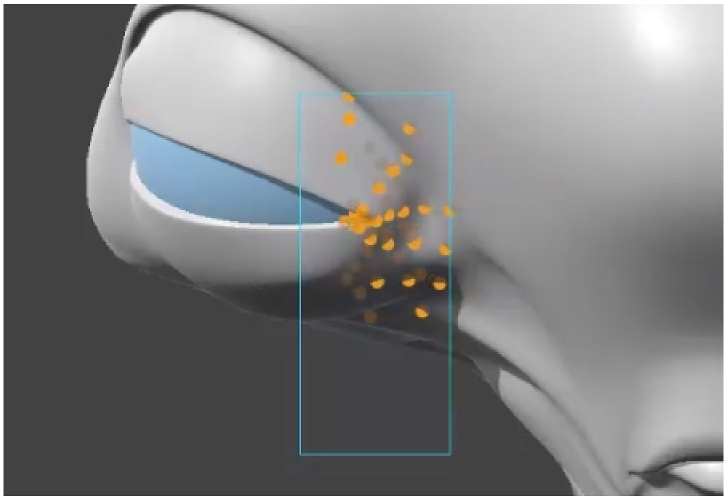
Radial - Radial falloff uses a sphere to define falloff where all elements at the center of the sphere are selected and no elements outside the sphere are selected. See Radial Falloff for more information.

The example shows using a **Radial** selection falloff in **Edge** selection mode to select only top loops on a dome. You could then extrude or bevel just those edges.



Vertex Map - Vertex Map falloff converts values of the currently selected Vertex Map to values representing falloff strength. This allows you to quickly create complex falloff selections. See Vertex Map Falloff for more information.

The example shows combining two **Vertex Map** falloffs in Polygon selection mode to select a number of polygons on a character automatically. You could then apply color maps or other operations supported by falloff to just those polygons.

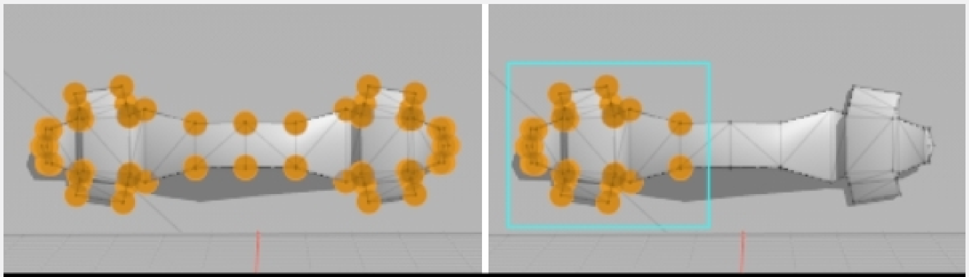


Lasso - Lasso selections drawn in screen space select all elements within the lasso area. Any elements outside the lasso area are ignored. See Lasso Falloff for more information.

The example shows using a **Lasso** selection falloff in **Vertices** selection mode to select only the vertices within the lasso area. You could then use **Deform** tools such as **Twist** or **Bend** to warp just those vertices to add texture to the mesh.

Select from Falloff's tool properties change depending on which mode you have selected, but there are several controls common to all modes.

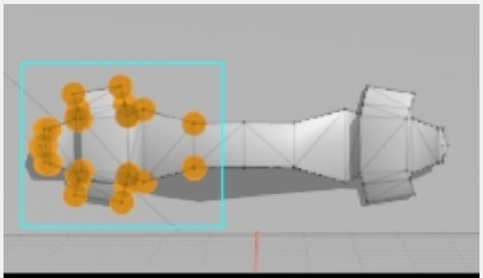
Mix Mode	Mix Mode controls how elements are affected by selections in the viewport: • **Multiply** - when applied to a subset of elements that are already selected, **Multiply** discards any selections that are not within the new selection. For example, using **Lasso** in **Multiply** mode narrows down existing selections by removing anything outside the Lasso area.



Multiplying an existing selection removes elements outside the Lasso area

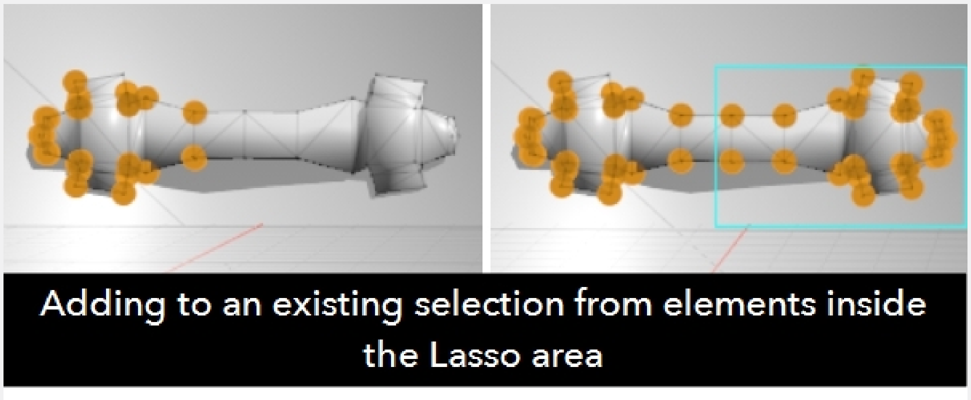
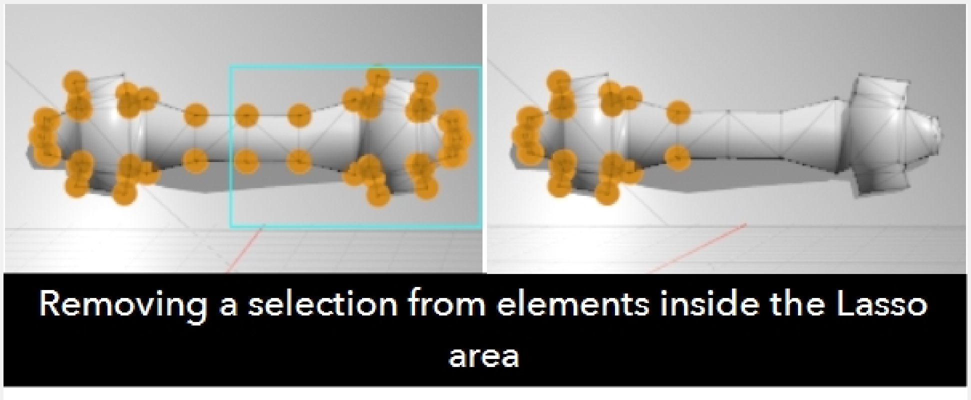
Using **Multiply** on a mesh with no existing selections acts like Set mode, creating a new selection.

- **Set** - the default mode, creates a new selection depending on the falloff mode selected. For example, using **Lasso** in **Add** mode adds the selections inside the Lasso area.



Adding selections using Set mode

- **Add** - when applied to a mesh with elements already selected, **Add** mode expands the existing selection by adding the new selection. For example, using **Lasso** in **Add** mode adds the selections inside the Lasso area.

	 Adding to an existing selection from elements inside the Lasso area • **Remove** - when applied to a mesh with elements already selected, **Remove** mode reduces the existing selection by removing the new selection. For example, using **Lasso** in **Remove** mode narrows down existing selections by removing anything inside the Lasso area.  Removing a selection from elements inside the Lasso area
Weight to Select	Sets the minimum falloff weight to select. This defaults to 0.5 for all modes, except for **Lasso** mode which defaults to 1.0 so that everything inside the Lasso area is selected.
Do Select	Click to apply the selection. Disabling this button allows you to edit the falloffs in the viewport.
Weighting Method	In **Edge** and **Polygon** selection mode, the weighting method controls when an element affected by the falloff is selected: • **All** - the element is only selected when all parts of it are within the falloff effect. • **Any** - the element is selected when any part of it is within the falloff effect. • **Average** - the element is selected when more than half of it is within the falloff effect.

World Transform	By default, falloffs calculate their influence over elements according to their relative position, using object or local space. When this option is enabled, all mesh surfaces are considered one surface and elements are selected or ignored accordingly. This option overrides the individual selection mode equivalent in the tool **Properties**.

Applying Precision

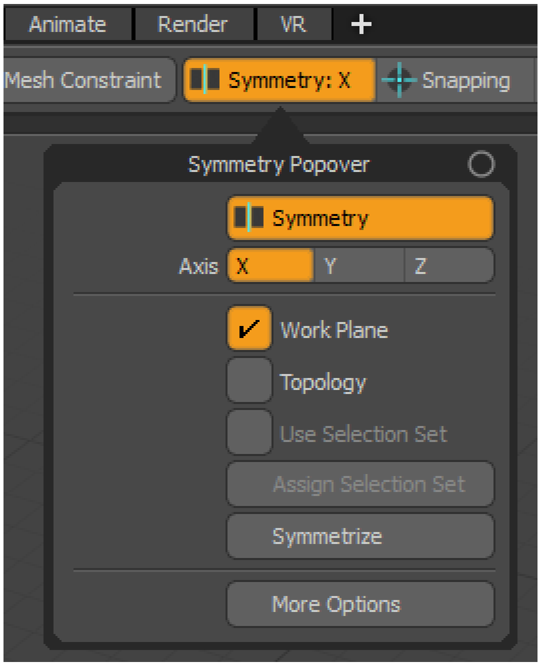
Learn about symmetry, snapping, constraints, and utilities.

Symmetry

Note: For the procedural Symmetry tool, see Procedural Symmetry.

The **Symmetry** function allows you to apply mirrored actions across a chosen axis. This allows for a simplified workflow on symmetrical objects, when corresponding sides of an object are exactly matched. To enable symmetry scene-wide, click the **Symmetry** button in the modes toolbar above any of the main 3D viewports. When clicked, it is highlighted in orange, denoting its active state, and the title changes to reflect the current symmetry axis. Alternatively, you can also enable **Symmetry** in the menu, by clicking **Edit** > **Symmetry** to choose the desired axis.

Once **Symmetry** is enabled, any operations performed in the component modes (vertices, edges, polygons), such as selecting polygons or deforming edges, is mirrored across the chosen axis to the corresponding geometry. For best results, make sure the model itself is perfectly symmetrical, as Modo generally applies symmetry operations to corresponding vertex positions. In some cases, it may prove useful to apply the Symmetry Tool to force a model's topology to match symmetrically across the desired axis. To open the **Symmetry** options hold the **Alt** key and press the **Symmetry** button, displaying the following panel.



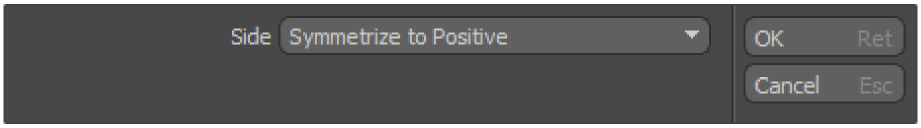
- **Symmetry** - toggles the symmetry function on or off; the button is highlighted in orange when it is on and remains gray when off.
- **Axis** - choose an axis to mirror actions across X, Y, or Z. Unless defined otherwise, the origin is always across the zero axis location.
- **Work Plane** - enables **Symmetry** when using the Work Plane.
- **Topology** - allows **Symmetry** to work in cases where regular symmetry fails by using a topological matching of components. To use this option, the target model should still be mostly symmetrical across the desired axis. Alternatively, you can use the **Assign Selection Set** option to define the symmetry axis for an object. To do this, select at least one edge of an edge loop that represents the centerline of the form, and then selecting the **Assign Selection Set** button to define the loop as the symmetry axis. Once defined, enable the **Use Selection Set** option and then topological symmetry mirrors across the defined edge loop.

  When using the **Topology** option, surfaces must still have a corresponding symmetrical element. In the case of asymmetric forms, the topological symmetry does not produce the expected result.
- **Use Selection Set** - aids **Symmetry** in finding the correct topological symmetry axis by defining an edge loop representing the centerline of the surface. To use it, first define the edge loop representing the center using the **Assign Selection Set** option.

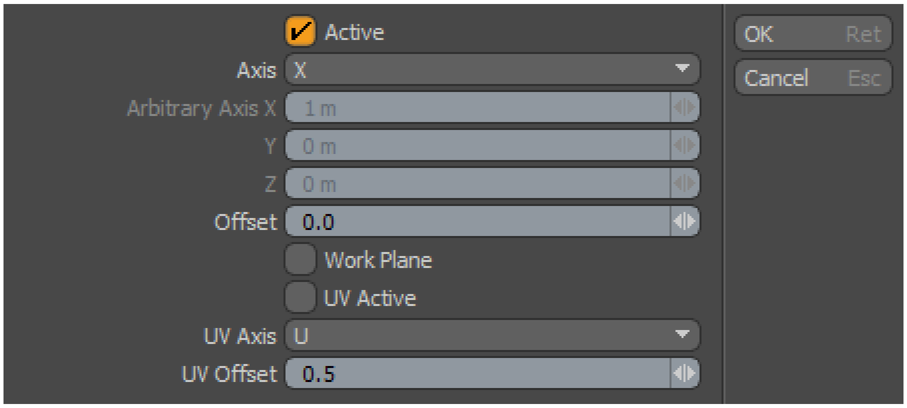
Note: This option does not provide for local symmetry; the symmetry plane is still always the axis center.

- **Assign Selection Set** - with an edge selected, representing the centerline on a symmetrical form, you can use this option to define an edge selection set specifically named for the purpose of enhancing the topological symmetry. Once defined, the **Use Selection Set** option must be enabled.

- **Symmetrize** - attempts to repair symmetry issues in a model, bringing models back into exact symmetrical form. Clicking **Symmetrize** opens the **Symmetrize Vertices** dialog, where you can choose the side to mirror across, either positive or negative, based on the values in the defined symmetry axis (values are based on either side of the axis center location, either positive or negative).


Side Symmetrize to Positive

- **More Options** - opens the **Symmetry Options** dialog window, containing some less used **Symmetry** options.


☑ Active
Axis X
Arbitrary Axis X 1 m
Y 0 m
Z 0 m
Offset 0.0
☐ Work Plane
☐ UV Active
UV Axis U
UV Offset 0.5

- **Active** - enables/disables the **Symmetry** function.
- **Axis** - choose which planar axis that **Symmetry** works over, or select **Arbitrary** to define your own custom axis.
- **Arbitrary Axis** - the symmetry axis is defined by a vector and center position. **Arbitrary Axis** allows you to set a custom axis vector.
- **Offset** - defines a symmetry position away from the origin, parallel to the base symmetry plane.
- **Workplane** - when enabled, **Symmetry** respects the **Work Plane**.

Note: If the **Work Plane** position is modified, the mesh must still have corresponding symmetrical geometry.

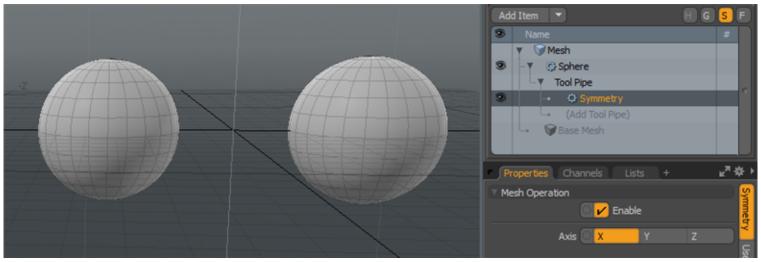
- **UV Active** - enables/disables the application of symmetry in the **UV** Viewport
- **UV Axis** - choose which axis that **Symmetry** works across. **U** is vertical, **V** is horizontal.
- **UV Offset** - defines a symmetry position away from the UV origin.

Tip: In certain instances of extreme asymmetry on a model, a quick and dirty way to make certain a model is symmetrical is to simply delete half of the model (making sure symmetry is disabled for this operation) and then use the **Mirror** command with the Specifying Action Centers and Falloffs set to **Origin**.

Procedural Symmetry

Modo provides a procedural Symmetry sub-tool that is compatible with a subset of the mesh operations in the procedural workflow including primitives and the Curve Extrude. The Symmetry tool can be added to the Tool Pipe using the **(Add Tool Pipe)** button in the Mesh Operations List. You can then set the **Axis** property to select the axis through which the tool is to be mirrored.

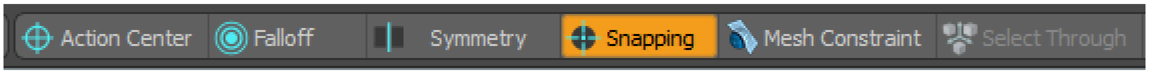
An example of the Symmetry tool with a procedural Primitive can be seen below. A single sphere has been added to the list, and a Symmetry tool has been added to the sphere's tool pipe. The **Axis** property is set to **X**. A translation of the sphere along X shows how the sphere is being mirrored in the **YZ** plane.


Add Item
Name
Mesh
Sphere
Tool Pipe
Symmetry
(Add Tool Pipe)
Base Mesh
Properties
Channels
Lists
Mesh Operation
Enable
Axis
X
Y
Z
Symmetry

Applying Snapping

Sometimes when positioning elements relative to each other, a high degree of precision is required, but zooming in to microscopic levels isn't accurate enough, nor is entering numeric values efficient (and neither is very user-friendly). In these cases, it is appropriate to use Snapping, a function that allows you to produce a highly accurate result, while maintaining a standard, interactive workflow.

Snapping in Modo is the action between two elements, one moving and the other static, where the moving element snaps to the static element's position, jumping away from the cursor when positioned within a user-defined range. This makes it easy to very accurately create and align elements in relation to one another, or in relation to 3D space.


Action Center
Falloff
Symmetry
Snapping
Mesh Constraint
Select Through

Usage

Snapping is disabled by default, but it can be easily enabled by clicking the **Snapping** button over the interface window (highlighted above). Snapping is active when the button is highlighted in orange. You can also activate snapping with a shortcut by pressing **X** on the keyboard. To temporarily enable snapping during an operation, press and hold **X** to snap the element, and then release the key to disable. The **X** key is a toggle so it can also be used to disable Snapping. Hold it down during an action to temporarily disable Snapping for that action.

When snapping is enabled, it conforms to the settings of the **Snapping Options** panel where you can enable and disable specific scene features that trigger the Snapping action. The panel can be accessed one of two ways. In the Properties Panel, there is a persistent **Snapping** button. Click it to open the panel, or press **F11** to open it under the mouse pointer. Click away from the panel to dismiss it. The various options of the Snapping panel are detailed further below.

To aid the snapping workflow, a number of pre-highlighting cues are employed for the cursor and target, when the cursor is close to the snapping range (the defined **Outer Range** control) the possible snapping targets highlight, making it easier to predict the possible snapping locations or as feedback that snapping has occurred.

Snapping Workflow

One of the most important things to keep in mind regarding snapping in Modo is that the Snapping function snaps a tool's handle, the widget that appears in the viewport when applying an action. The position and orientation of the active tool handle is controlled by the Action Center, so understanding these settings are important. Make sure to have an appropriate **Action Center** preset defined for the desired result.

By default, Modo has no defined Action Center (see Specifying Action Centers and Falloffs for more information). When you click in the viewport, Modo moves the active tool handle to the intersection point of the Work Plane and the mouse cursor's click. If you tried to click over a specific vertex, they might be exact (unlikely), they might be close, or they might not be close at all. Then, when moving the elements and snapping, the result would be very inaccurate and it would appear as if snapping failed. It actually didn't, It just wasn't told the correct thing to do.

To properly snap a single vertex to another single vertex, follow the steps below:

1. First, enable the snapping feature option for **Vertex** and then set the **Action Center** to **Selection Center Auto Axis**.
2. Then select a single vertex, activate the Move tool (**W**) and moving the vertex, press and hold down **X**, to enable snapping temporarily.

   The single vertex then snaps to whatever new vertex the mouse hovers over.

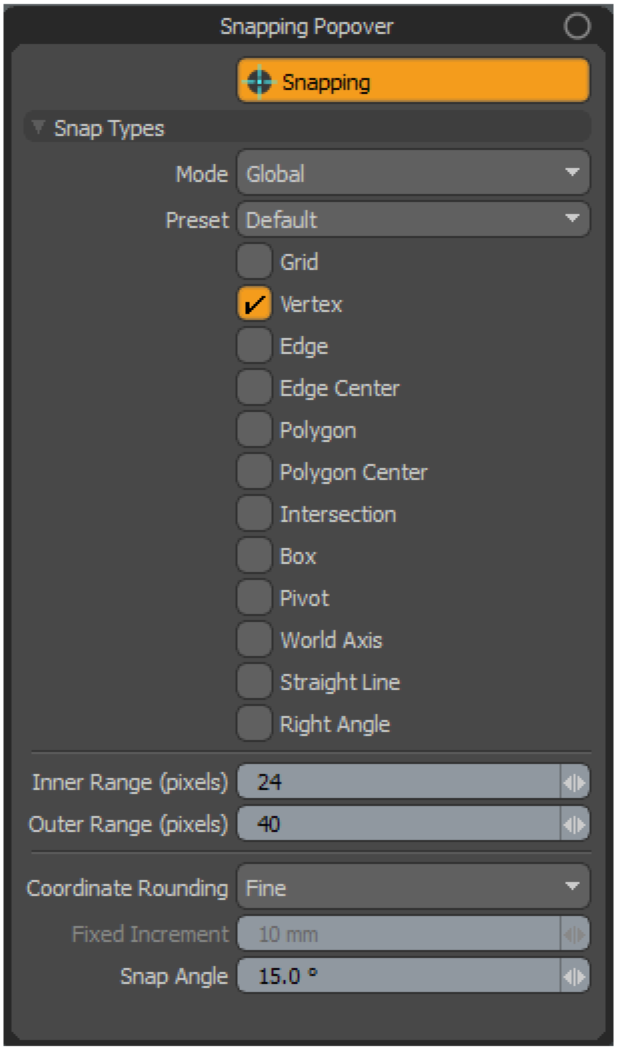
3. Release the mouse button and then press **X** to set the element at the snapped position.

You don't need to set all these options each time using Snapping, as Modo remembers what setting was applied the last time it was used.

Tip: You can also define Snapping presets to easily apply within the **Snapping Options** panel, or by creating custom snapping controls specific to a single tool with the power of the Tool Pipe.

Snapping Options

To open the Snapping options, click **Edit** > **Snapping** > **Snapping Options**, or press **F11** on the keyboard.



Snapping Popover

Snapping

Snap Types

Mode: Global

Preset: Default

☐ Grid
☑ Vertex
☐ Edge
☐ Edge Center
☐ Polygon
☐ Polygon Center
☐ Intersection
☐ Box
☐ Pivot
☐ World Axis
☐ Straight Line
☐ Right Angle

Inner Range (pixels): 24

Outer Range (pixels): 40

Coordinate Rounding: Fine

Fixed Increment: 10 mm

Snap Angle: 15.0 °

The **Snapping Options** menu allow you to control what scene features are snapped and when. You can select any number of controls simultaneously as well as the **Mode** that determines when the snapping occurs.

Snapping: This button enables and disables the Snapping functionality, same as the button above the modeling viewport.

Snap Types

Mode: This defines different snapping features (listed below) depending on the current selection mode, with each **Mode** retaining its own individual definition. Click on the field to open the available options.

- **Global**: Determines the Snapping features that work globally, regardless of selection mode.
- **Component**: Determines additional Snapping functionality that applies concurrently to the **Global** values when working with vertices, edges and polygons.
- **Items**: Determines additional Snapping functionality that applies concurrently to the **Global** values when working with items, centers and pivots.

For example, you could set a **Global** mode snap option for **Box** and then a **Component** mode option for **Vertex** and **Edge**, and finally an **Items** mode option for **Pivot**. Then, when editing a vertex, which is a component, snap would adhere to the **Global** setting of **Box**, plus the component settings for **Vertex** and **Edge**, but ignore the **Items** settings of **Pivot**. Feel free to mix them up as you see fit.

Presets: Combinations of Snapping features can be stored as presets, making it easy to switch between often used combinations. To create a preset:

1. Select the **New Preset** option from the panel.

   This opens a dialog box to enter a name for the preset.
2. Type in a unique name for the preset and click **OK** to create the preset.

   The new preset automatically gets selected in the **Preset** list.
3. While the named preset is selected, simply select any number of features in any **Mode**.

   They are saved into that preset automatically when the **Snapping Options** panel is closed.

To switch between different presets, select the named preset within the **Preset** list.

To remove a preset, select **Remove Preset** and then choose the named preset from the dialog and click **OK**.

Snapping Features: The following options all pertain to specific scene features that can be enabled/disabled. Multiple features can be selected at once and are stored independently for each **Mode**.

- **Grid**: When selected, you can define a fixed grid amount for snapping.
- **Vertex**: When selected, snaps to any individual vertex position.

- **Edge**: When selected, snaps to any position along a polygon's edge.
- **Edge Center**: When selected, snaps the center of the target edge.
- **Polygon**: When selected, snaps to any position of a polygon's surface.
- **Polygon Center**: When selected, snaps to the center of the target polygon. The center is the average position of all associated vertices.
- **Intersection**: When selected, snaps to the point of intersection between an edge and a polygon.
- **Box**: When selected, snaps to the cardinal positions of the target item's overall bounding box.
- **Pivot**: When selected, snaps to the pivot location of the target item.
- **World Axis**: When selected, tools align to XYZ axes. Only the Pen and Mirror tools support this snap type.
- **Straight Line**: When selected, snaps to a straight line. Only the Pen and any Pen generator type tools support this snap type.
- **Right Angle**: When selected, snaps to right (90°) angles. Only the Pen and any Pen generator type tools support this snap type.

Inner Range: Determines the range, in screen pixels, where a selected element snaps to the target element.

Outer Range: Determines the range, in screen pixels, where pre-highlighting occurs. When snap element and target element are within this proximity range, the target element highlights.

Coordinate Rounding: This option controls how mouse input is converted, through the Work Plane, into 3D coordinates.

- **None**: This means that no coordinate rounding is done. Every mouse move gives unclamped coordinates (typically with lots of decimals). This option is basically the raw 2D to 3D transform. Useful for working freehand.
- **Normal**: This option attempts to give clean, round coordinates based on your view transform. As the mouse moves, you can see values that are rounded in your current unit system display in the information tab. The step size gets smaller or larger as you zoom in or out. You may need to move the mouse cursor 2-3 pixels to see values update.
- **Fine**: This option is similar to **Normal**, but optimizes for closer to one step of coordinate rounding from one pixel of mouse movement. This gives you finer grained input, but can be difficult to hit exact values.
- **Fixed**: This option uses the **Fixed Increment** preference to put a lower limit on both coordinate rounding and the grid. If you set the fixed increment to 10 mm that means the grid never gets finer than 10 mm, and all input is rounded to the nearest 10 mm even at high zoom. When zoomed out, however, the grid shows larger values but the step size always remains a multiple of the fixed increment.
- **Forced Fixed**: This option is similar to **Fixed**, but forces the size of the grid and the input step to match the increment exactly no matter the zoom level.

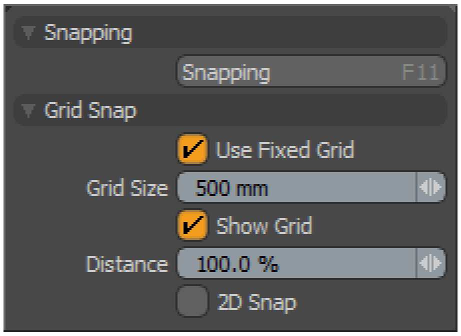
Fixed Increment: When the Coordinate Rounding is set to either fixed options, this value determines the fixed coordinate rounding grid.

Special Snapping Options

When certain snapping features are selected, additional options appear in both the **Snapping Options** panel and the tool **Properties** panel for added convenience. These allow you to define specific controls related to the selected Snapping feature.

Grid Snap

Snapping of selected element is fixed to a user-defined grid. This setting can be helpful in keeping values to whole numbers (eliminating numbers with decimals) and can make the construction of geometry that conforms to a grid far easier and more precise to create such as a blueprint or floor plan.



Snapping
Snapping F11
Grid Snap
☑ Use Fixed Grid
Grid Size 500 mm
☑ Show Grid
Distance 100.0 %
☐ 2D Snap

Use Fixed Grid: When this toggle is enabled, you can define a fixed grid with the **Grid Size** setting, which sets the snapping interval. When disabled, the grid snapping defaults to the dynamic grid displayed in the 3D Viewports.

Grid Size: When the **Use Fixed Grid** option is enabled, this value input filed defines the interval for the fixed grid, originating at the 0,0,0 World Origin position.

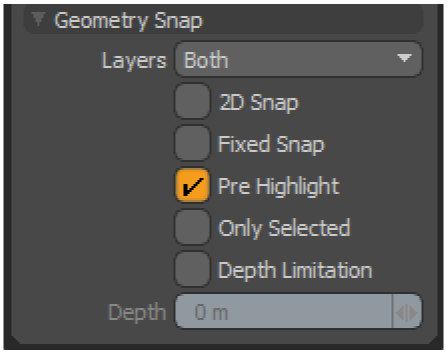
Show Grid: This toggle displays the grid as a light magenta overlay in the viewport, which dynamically updates similar to the Work Plane, in the center of the viewport as a visual reference to the **Grid Size** setting.

Distance: This option defines the snapping range, the distance when Modo snaps to the closest grid position. When set at the default 100%, Modo always keeps values fixed to the grid. Reducing this value would require the cursor to be closer to the grid positions before snapping occurs.

2D Snap: The 2D snap function relates to the orthographic viewports (top, front, side, and so on) that allow snapping on fixed planes parallel to the viewport window. This allows you to move points so they snap into position in the viewport, but they don't necessarily snap into a co-located position in 3D space.

Geometry Snap

Snapping to specific component elements. These snapping options allow you to very precisely position geometric elements in relation to one another.



Geometry Snap

Layers: Both

☐ 2D Snap

☐ Fixed Snap

☑ Pre Highlight

☐ Only Selected

☐ Depth Limitation

Depth: 0 m

Layers: Instructs Modo as to which layers it should consider for snapping. **Background** only takes into account layers that are visible but not selected, **Active** only takes into account selected visible layers (also known as foreground layers) and **Both** which takes into account all visible geometry in the Items (Scenes) List.

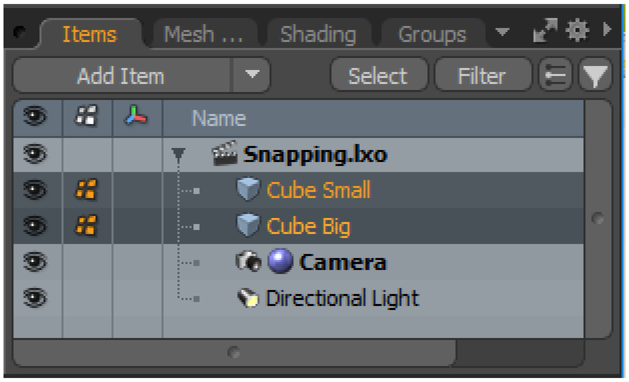
2D Snap: The 2D snap function relates to the orthographic viewports (top, front, side etc.) and allow snapping on fixed planes parallel to the viewport window, thus allows you to move points so they snap into position in the viewport, but they don't necessarily snap into a co-located position in 3D space.

Fixed Snap: This toggle, when enabled, forces Modo to only position the moving element to snapping positions, normally moving elements would move, and then snap as they neared an element, but **Fixed Snap** jumps the elements between snappable positions.
Only Selected: When enabled, geometry snapping finds the element to be snapped to from the selected polygons of the Foreground. This makes geometry snapping fast for heavy meshes. For more information see Geometry Snapping - Only Selection Example.

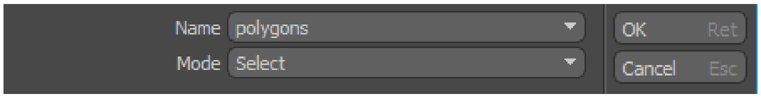
Geometry Snapping - Only Selection Example

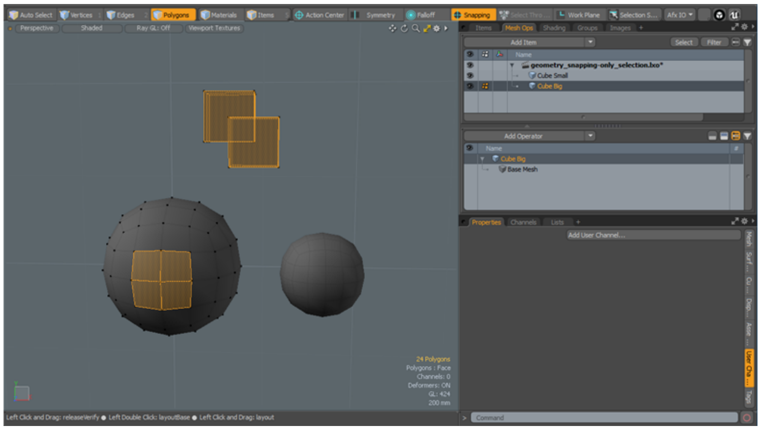
The following workflow example demonstrate how to apply the **Only Selection** option.

1. Download our example scenes and open **geometry_snapping-only_selection.lxo** in the **Model** layout.
2. On the right panel, in the **Items** list, select both mesh items; **Cube Small** and **Cube Big**.

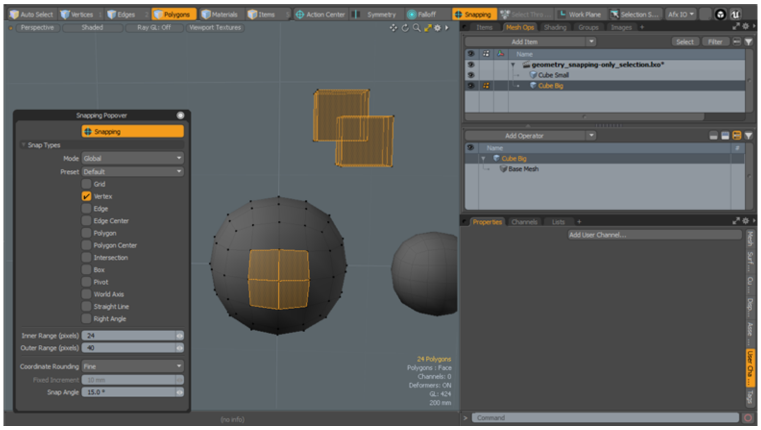

Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Snapping.lxo
Cube Small
Cube Big
Camera
Directional Light

3. On the application menu, click **Polygons** to enable this selection mode.
4. From the **Select** > **Use Selection Set** menu, set **Name** to **Polygons** and **Mode** to **Select**, and then click **OK**.


Name polygons
Mode Select
OK Ret
Cancel Esc



5. From the application menu, click **Snapping** and press **F11** to display the **Snapping Popover** dialog.
6. Enable the **Vertex** option.



7. On the Toolbox, located on the left panel, open the **Curve** sub-tab and click **Curve** to activate the tool.

8. Using the **Curve** tool, draw points over non-selected and selected polygons on both meshes.
9. With Curve tool still active, enable **Only Selected** in left side panel under the **Geometry Snap** options.
10. Continue using Curve tool to draw points over non-selected and selected polygons on both meshes.

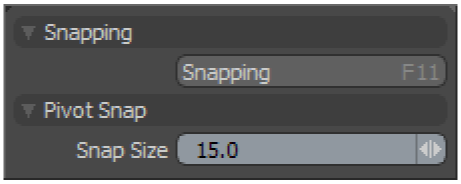
    As you move the Curve handle, notice the Geometry Snapping **Only Selected** option locates the vector points and highlight them as you move over the polygon selection set of the Foreground.

Depth Limitation: When enabled, it limits the snapping range to the specified **Depth**. Snapping ignores elements further away from the source position. This is useful when working with complex scenes with lots of background objects, and you want to snap to specific things.

Depth: The maximum snap distance along screen direction used by the Depth Limitation option.

Pivot Snap Mode

Snaps selected element to items' pivot elements. This option can be helpful when several items need to pivot rotate from the same location.



Snap Size 15.0

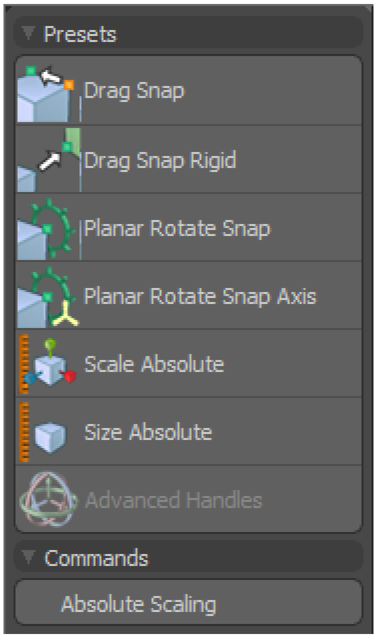
Snap Size: This values defines how close an element must be to snap to the pivot position. Larger values snap elements more easily to the pivot position of an item layer.

Snaps and Precision Palette

The **Snaps and Precision** Palette provides a handy floating palette that holds all the settings and tools related to snapping in a single viewport. You can open it from the menu bar by clicking **Edit** > **Snapping** > **Snaps and Precision**.

The palette includes several tool options you can use for precision transforms. Created with the power of the **Tool Pipe**, they are automated combinations of transform actions, snap settings and Specifying Action Centers and Falloffs .

Tip: Once opened, the visibility of all floating palettes can be toggled on and off by pressing the ` (backtick) key.



Snaps and Precision Palette - Presets

Drag Snap	This tool combines the Element Move transform with Geometry Snap. This allows components of a mesh to be quickly dragged and snapped to other components in the project. By default, the Element Move transform is set to **Auto**, which allows you to click and drag a vertex, edge or polygon regardless of the current selection mode. See Using Falloffs and Applying Snapping for more information.
Drag Snap Rigid	This tool is very useful for moving connected geometry by dragging it from a single component such as a vertex, edge or polygon. The tool combines with snapping, so that you can quickly snap the current component to another component in the project while dragging the rest of the connected polygons along in a rigid manner.
Planar Rotate Snap	This tool is excellent for precisely aligning elements. To use the tool, click on any component in the scene. This determines the element upon which the tool handles are centered. Next, use the outermost handle to establish the resting orientation for the rotation control by dragging that handle to any component in the project. The handle snaps to vertices, edges or polygons. Once the rest orientation has been determined, use the inner round handle to apply some rotation. This rotation action can be freehand or you can once again leverage geometry snapping to very accurately set the rotation.

Planar Rotate Snap Axis	This tool is similar to the **Planar Rotate Snap** tool with the extension of a second axis handle for controlling the plane of rotation. This is an advanced tool that is invaluable in certain complex rotation scenarios that require absolute precision. Like the **Planar Rotate Snap** tool, the first click sets the center and axis for the tool based on the component under the mouse.
Scale Absolute	This combines the Scale tool (**Absolute Handles** option active) with the Geometry Snap modifier. The tool utilizes a two step process. In the first step you use the Absolute handles to set the starting scale point for the tool. Next you use the Scale handles to actually modify the geometry. During both steps, the handles can snap to the geometry, thus allowing you to use one mesh component to set the initial scale and a second component to create the Absolute scale.
Size Absolute	This option opens an absolute sizing option.
Advanced Handles	This option toggles the visibility of Advanced handles for the selected tools. This often places additional controls into the screen viewport making it easier to produce more accurate results.
Absolute Scaling	This option opens the **Absolute Scaling** window.

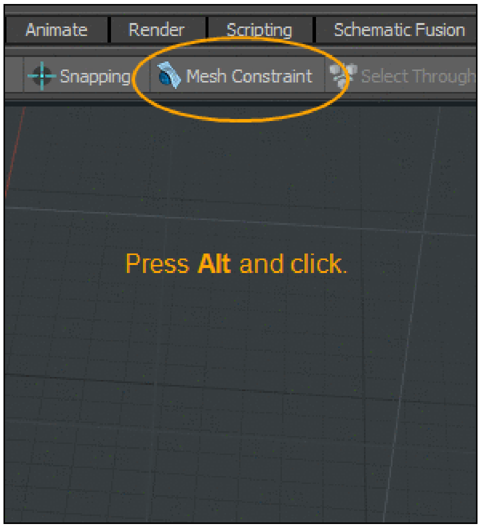
Note: For more information on the Absolute Scaling tool, see Absolute Scaling

Mesh Constraints

The Mesh Constraints tools allow you to quickly apply different types of constraints on a mesh item to position geometry in a scene; **Guide**, **Background**, and **Primitive**. These Mesh Constraint types allow you to use visual guides to assist you in aligning mesh items in your scene, enhance curve tool handles for better transform operations, and to create a simple low polygon count model from a complex mesh.

Mesh Constraint Types

Constraints are found in the menu bar below the interface layout tabs. The Mesh Constraints button is displayed in orange when a constraint is active. Selecting any mode enables the selected constraint and its associated settings appear in the upper portion of the panel, allowing you to customize how the constraint performs. The constraint affects the modeling procedure differently, depending on the type activated.



- **Guide** - Creates visual guide lines in the 3D viewport to use as reference or to enhance snapping.

  For more information, see Constrain to Guide.
- **Background** - Constrains the movement or creation of objects in the foreground layer from passing through any geometry in a background layer to create an impression or the background mesh. This is useful when you have imported a complex model created in another application, such as ZBrush, and want to create a copy using less polygons.

  For more information, see Constrain to Background.

  For a procedural version of this tool, see Procedural Surface Constraint Tool.
- **Primitive** - Constrains the handles of a curve tool, such as a Tube tool, Curve, Bezier, or Pen tool, using a virtual primitive shape to position the geometry. Similar in concept to Background, however, no physical geometry needs to be present for the Primitive option.

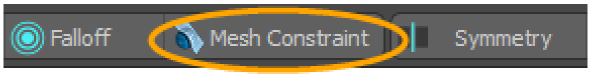
  For more information, see Constrain to Primitive.

Benefits of Using Mesh Constraints

- Define custom guide lines in the scene for visual reference or to enhance snapping.
- Add virtual primitives to a tool handle to enhance your transform actions.
  - Rather than having to select each vertices and align them to the target model , the Mesh Constraint Background option automatically aligns all vertices to the surface of the mesh item onto the target model. For example, if you wanted to add a button to a jacket and have it align to the surface, use the Mesh Constraint Background option.
- Quickly create a low polygon count mesh item from a complex mesh item. Use the Background Mesh Constraint rather than applying the Automatic Retopology tool to generate a new mesh item from a complex input surface.

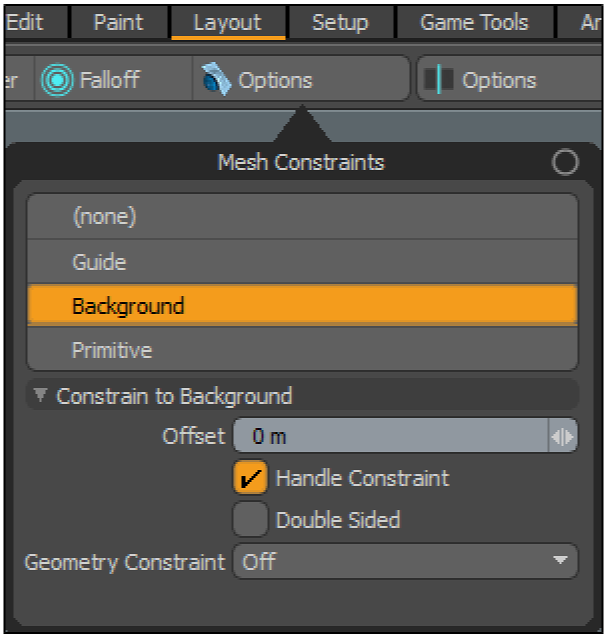
Activating Mesh Constraint Modes

1. Below the interface layout tabs, click the **Mesh Constraint** button to enable the default **Background** constraint mode.


Falloff
Mesh Constraint
Symmetry

2. Press **Alt** and click **Options** to view the Constrain to Background options.

   The Mesh Constraints dialog opens.


Edit
Paint
Layout
Setup
Game Tools
Ar
Falloff
Options
Options
Mesh Constraints
(none)
Guide
Background
Primitive
Constrain to Background

Offset 0 m

☑ Handle Constraint

☐ Double Sided

Geometry Constraint Off

3. Select one of the constraint modes: **Guide**, **Background**, or **Primitive**..

   The icon button displays the name of your selected constraint mode in orange.

To disable a mesh constraint mode:

- Click the constraint button highlighted in orange to dismiss it. Alternatively, press **Alt** + click the Mesh Constraint button and select **None** from the dropdown menu.

Constrain to Background

Setting **Mesh Constraints** to **Background** constrains the movement or creation of objects in the foreground layer from passing through any geometry in a background layer. This is the most common type of constraint.

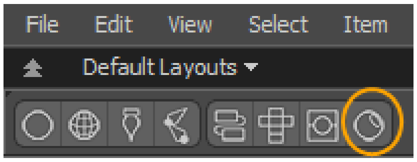
Any item layer that is visible and selected is considered a foreground layer and any layer that is visible but not selected is considered a background layer.

Tip: To see your results better, change the shading style of the mesh items in your scene by clicking the ⚙ gear icon on the top right corner of the 3D viewport. Open both the **Active Meshes** and **Inactive Meshes** tabs and set the **Shading Style** to **Shaded** or **Advanced**. For more information, see 3D (OpenGL) Viewport.

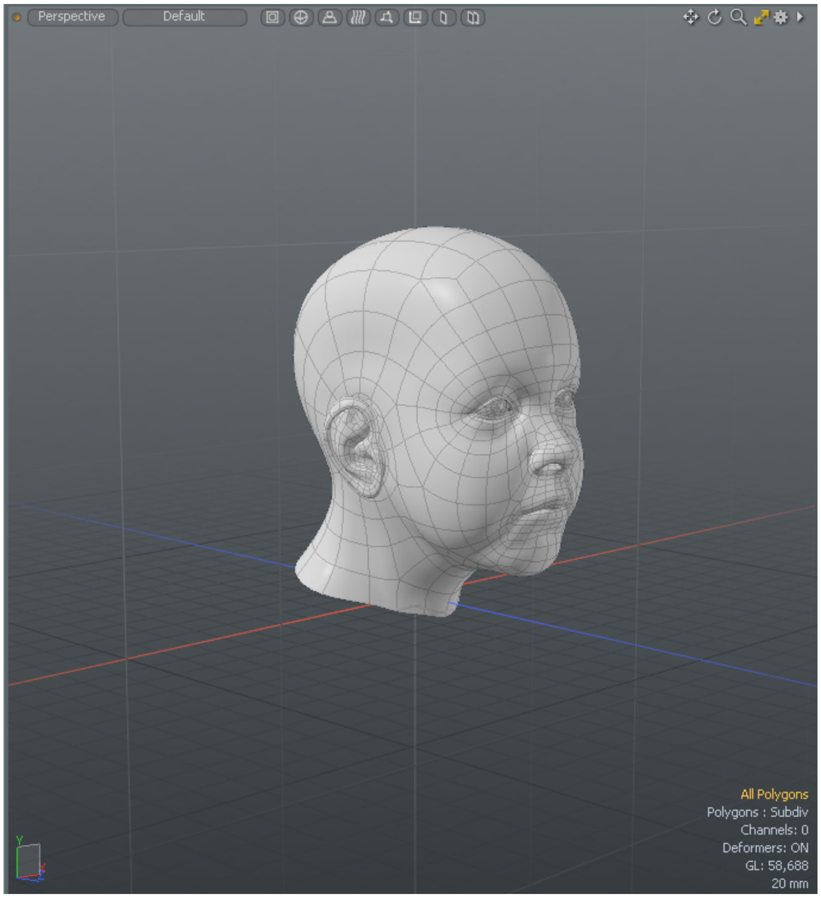
Background Constraint Example

The following procedure uses a Modo preset and a primitive to demonstrate how a background constraint is applied.

1. Click the **Preset Browser** palette icon on the top left corner and double-click on a preset to add it to your scene.



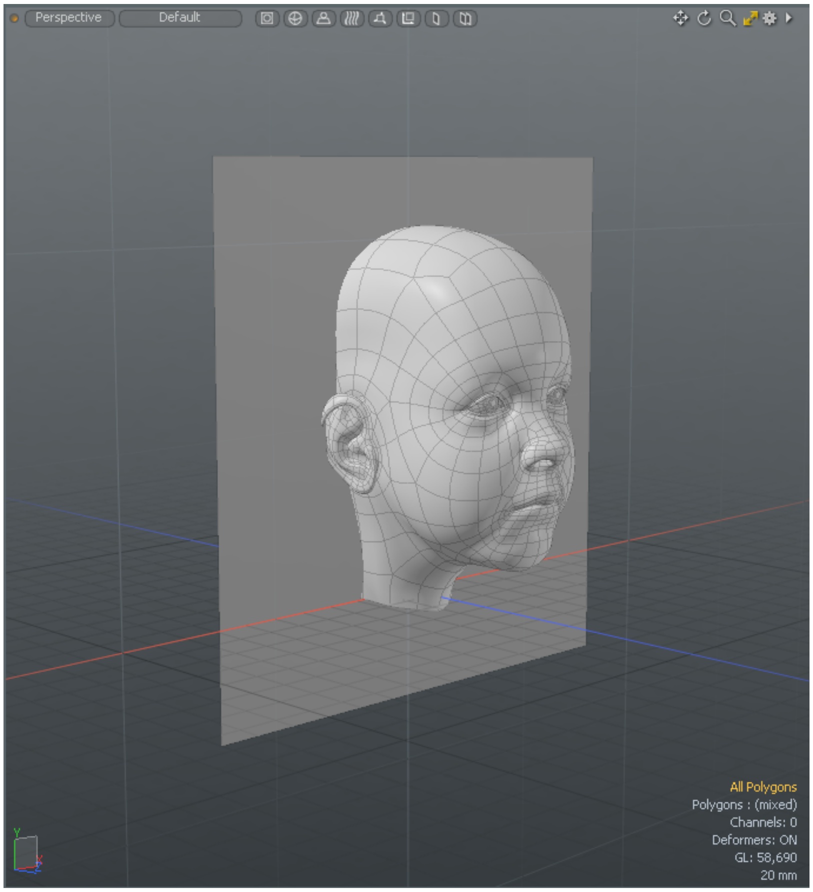
In this example we are using the **Cloud Assets** > **Meshes** > **Human** > **Child Head 01** preset. If you have previously downloaded this cloud asset, you will find it in the **Assets** > **Meshes** > **Human** directory.


Perspective
Default
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 58,688
20 mm

2. Close the Preset Browser window.
3. On the right panel, open the **Items** tab, click **Add Item**, and select **Mesh**.

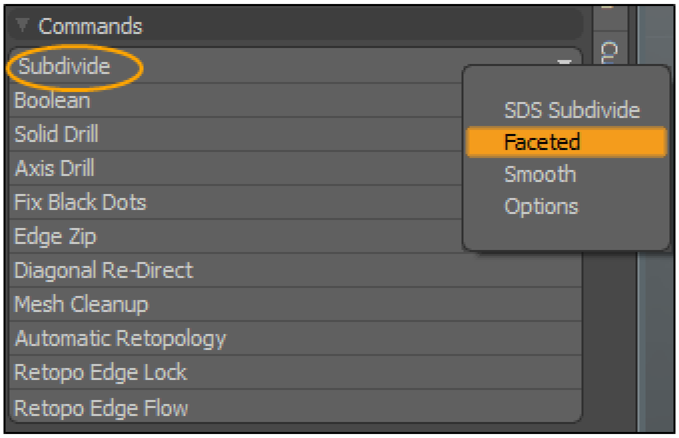
> **Note:** A foreground mesh item must be in a separate layer in the Items list. It can be placed anywhere in the Items List.

4. On the left panel, open the **Basic** tab, click **Cube**, and click-and-drag in the 3D viewport to draw a plane around the preset.


Perspective
Default
All Polygons
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 58,690
20 mm

Tip: Use the move tool handles to position the plane where you want it.

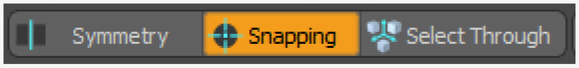
5. Press the **spacebar** to drop the move tool.
6. Under the layouts menu bar, click **Polygons** and double-click on the cube to select all of the polygons.
7. On the left panel, open the **Mesh Edit** tab.
8. Under **Commands**, click the **Subdivide** dropdown menu, and click **Faceted** several times.

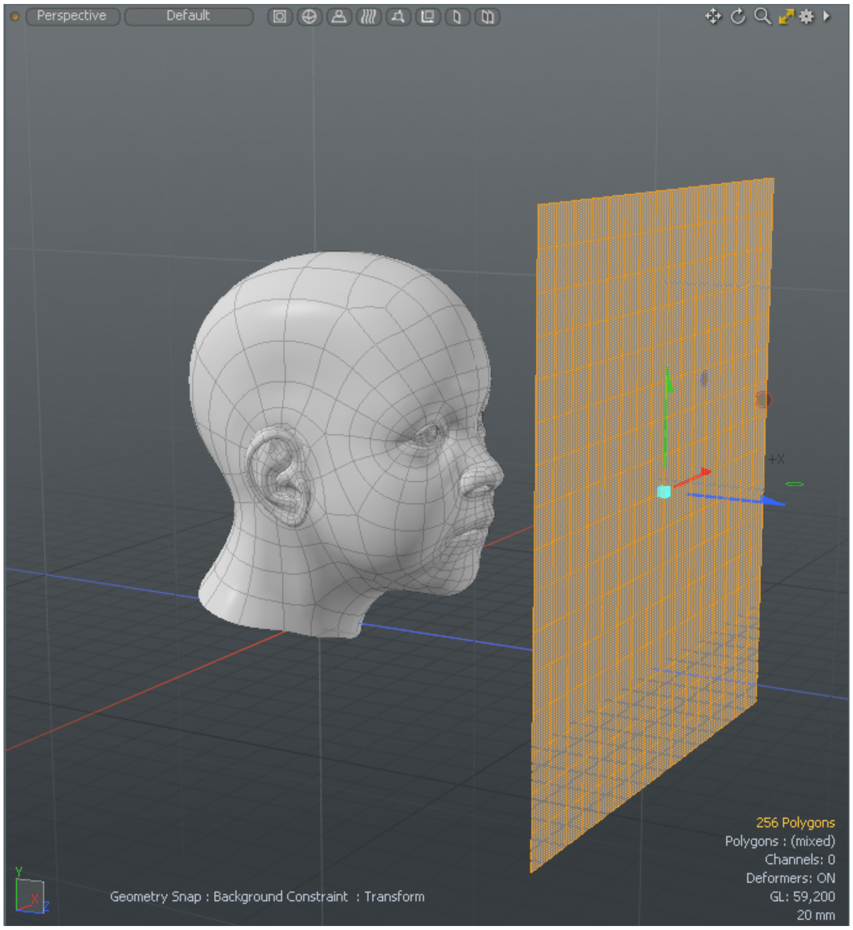

Commands
Subdivide
Boolean
Solid Drill
Axis Drill
Fix Black Dots
Edge Zip
Diagonal Re-Direct
Mesh Cleanup
Automatic Retopology
Retopo Edge Lock
Retopo Edge Flow
SDS Subdivide
Faceted
Smooth
Options

Tip: Subdividing the polygons a number of times provides more details for our new mesh item.

9. **Optional** - On the right panel, open the **Mesh Ops** tab, click **Add Operator**, select **Polygons** > **Set Polygon Type**
10. **Optional** - Open the **Set Polygon Type** tab and set the **Polygon Type** to **Catmull-Clark** to get a smooth finish.
11. In the 3D viewport, press **W** and click-and-drag the plane to move it in front of the preset.

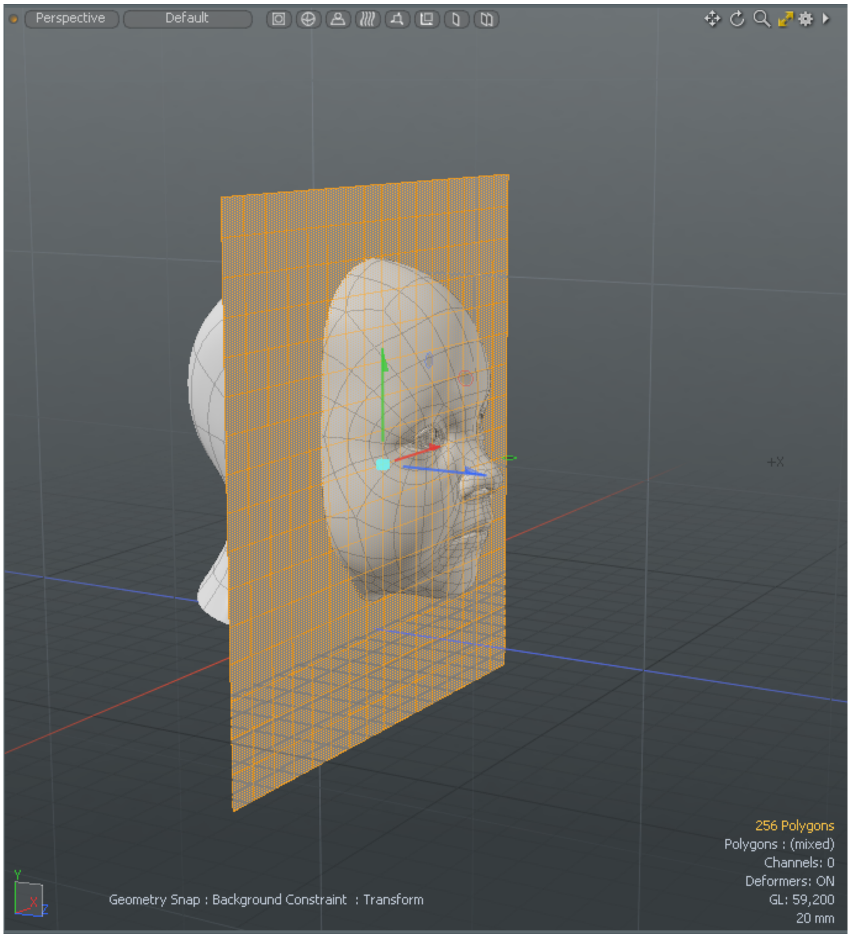
Note: If snapping is on, you may have difficulties aligning the plane in front of the preset. Click the button to turn Snapping off.


Symmetry
Snapping
Select Through


Perspective
Default
256 Polygons
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 59,200
20 mm
Geometry Snap : Background Constraint : Transform

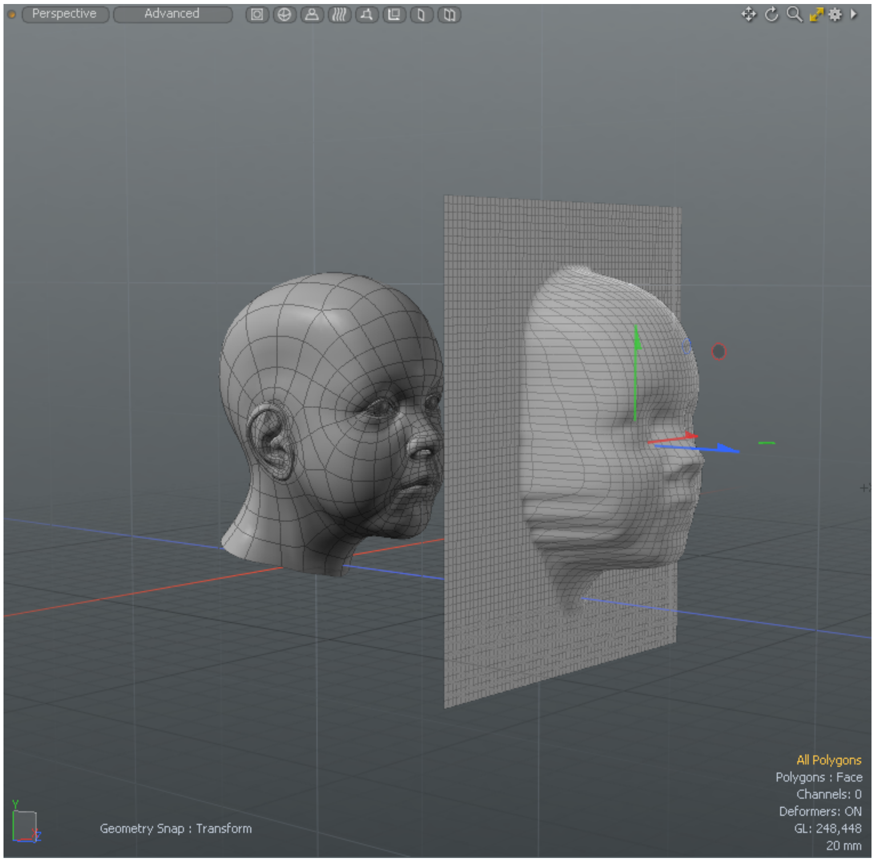
12. Under the layouts menu bar, click **Mesh Constraint**.

    Background mode is activated and the Move tool remains active in your scene.
13. On the left panel, under **Constrain to Background**, set **Geometry Constraint** to **Vector**.
14. Click-and-drag the plane to move the plane half way into the preset.

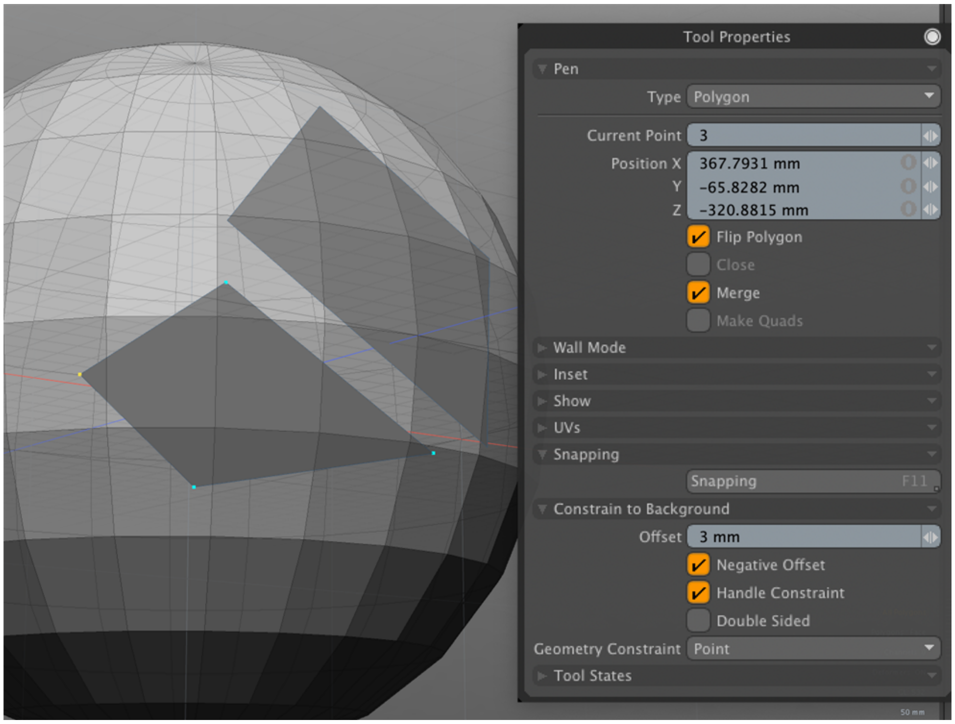

Perspective
Default
+X
Y
X
Z
Geometry Snap : Background Constraint : Transform
256 Polygons
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 59,200
20 mm

15. Under the layouts menu bar, click **Background** to disable the Mesh Constraint button.
16. Click-and-drag the plane to move the cube in front of the face.

    An imprint of the preset is created into the mesh item.


Perspective
Advanced
All Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 248,448
20 mm
Geometry Snap : Transform

Constrain to Background Properties



Tool Properties

Pen

Type: Polygon

Current Point: 3

Position X: 367.7931 mm

Y: -65.8282 mm

Z: -320.8815 mm

☑ Flip Polygon

☐ Close

☑ Merge

☐ Make Quads

Wall Mode

Inset

Show

UVs

Snapping

Snapping F11

Constrain to Background

Offset: 3 mm

☑ Negative Offset

☑ Handle Constraint

☐ Double Sided

Geometry Constraint: Point

Tool States

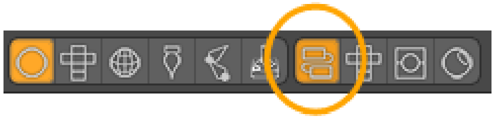
Offset	Allows you to define a distance away from a surface for constraint. This allows something like a tube being drawn over a background object to rest on the surface. If the tube has a diameter of 10 cm, applying a 5 cm offset rests the tube right on the surface as the tube is generated outward from its center.
Negative Offset	When enabled, this option allows you to enter a negative offset value for the background constraint. This has the effect of making the constrained geometry appear under the surface.
Handle Constraint	For tools where a curve of some type is drawn, such as the Tube tool, curves, Bezier, or the Pen tool, the handles can be constrained also by enabling the **Handle Constraints** setting, otherwise, only the polygonal geometry is constrained.
Double Sided	When enabled, this setting treats background polygons as double sided, so geometry can be constrained regardless of the constraining surface's normal facing direction.
Geometry Constraint	Several options are available to define how Modo searches for the background geometry to constrain against. • **Off** - Temporarily disables the constraint. • **Screen Axis** - The constraining surface is evaluated by searching in a perpendicular direction to the current view position, stopping once it hits any geometry. This is the most basic constraint type. • **Vector** - Uses the translation/transform direction set by the active tool. If you have the Move tool active and start dragging the handle on the X axis, the constraint checks that direction to see what background elements it hits and then stops. If performing a bend, vertices move in many directions, as each vertex has its own vector, which is effectively the direction the vertex is currently moving toward. • **Point** - This constraint acts like a magnet, translated/transformed vertices are kept against the background geometry ensuring they are always resting directly on its surface.

Procedural Surface Constraint Tool

The alternative to the direct modeling version of the Constrain to Background option is the procedural mesh operation called **Surface Constraint**. Use this tool to procedurally constrain geometry to another piece of geometry in the scene. Surface constraints can be applied to normal mesh items, as well as more complex mesh items, such as a Static Mesh Item and a Fusion Item.

Activating the Surface Constraint Tool

- On the right panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Deform** > **Surface Constraint**.
- Alternatively, on the top left corner of the interface, click the **Schematic** palette icon , click **Add**, and double-click **Mesh Operations** > **Deform** > **Surface Constraint**.



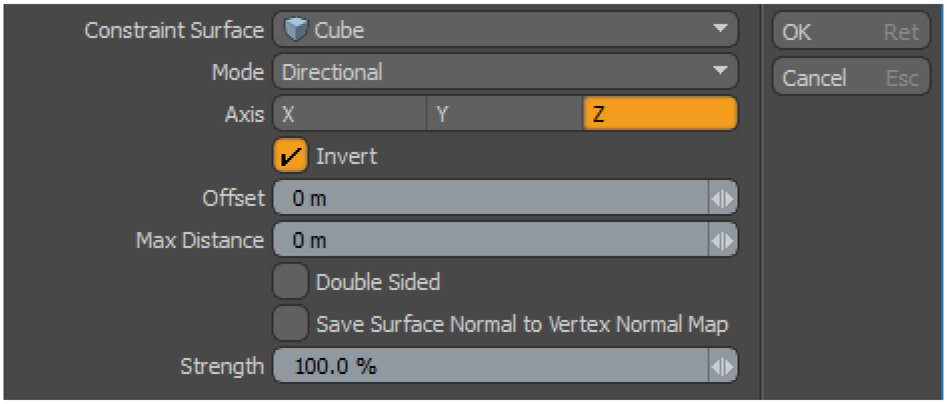
Example

The following is a simple example, which demonstrates how the geometry reacts when changing the **Mode** options of the Surface Constraint tool.

1. On the left panel, on the **Basic** tab, press **Ctrl/Cmd** + click on the cube icon.
2. On the right panel, under the **Mesh Ops** tab, click the **Add Item** dropdown menu, and double-click **Mesh**.
3. On the menu bar, click **Geometry** > **Primitive Tools** > **Teapot**.
4. On the left panel, set the teapot property **Size** to **1m** and click **Apply**.
5. Use the move and rotate tools to position the teapot beside the cube.

Tip: Press **W** to move and press **E** to rotate. Press the **Spacebar** to drop the tool

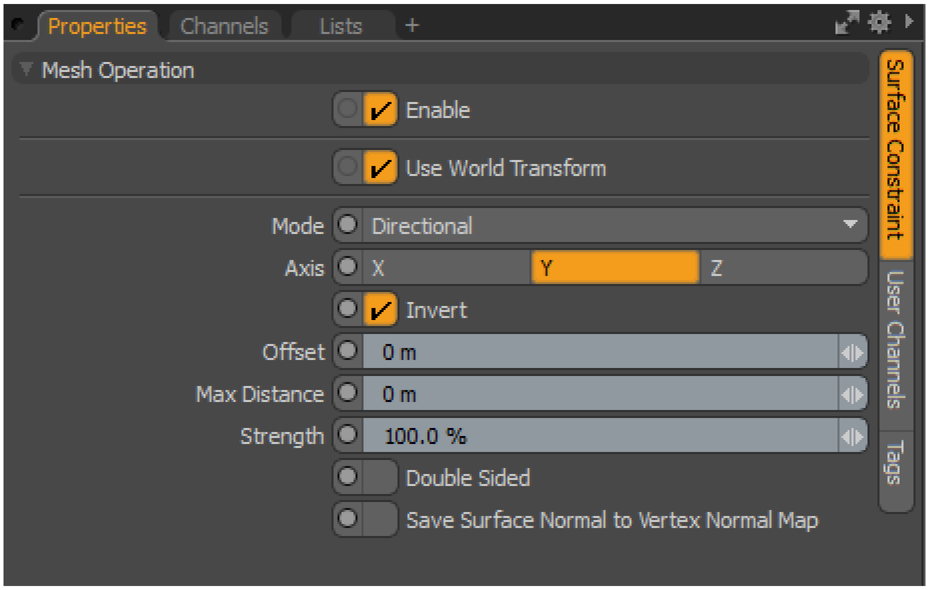
6. With teapot selected, on the right panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Deform** > **Surface Constraint**.
7. From the **Create Surface Constraint Operator** dialog, set the **Constrain Surface** to **Cube**, **Axis** to **Z**, enable **Invert**, and click **OK**.



Constraint Surface: Cube
Mode: Directional
Axis: X | Y | **Z**
☑ Invert
Offset: 0 m
Max Distance: 0 m
☐ Double Sided
☐ Save Surface Normal to Vertex Normal Map
Strength: 100.0 %

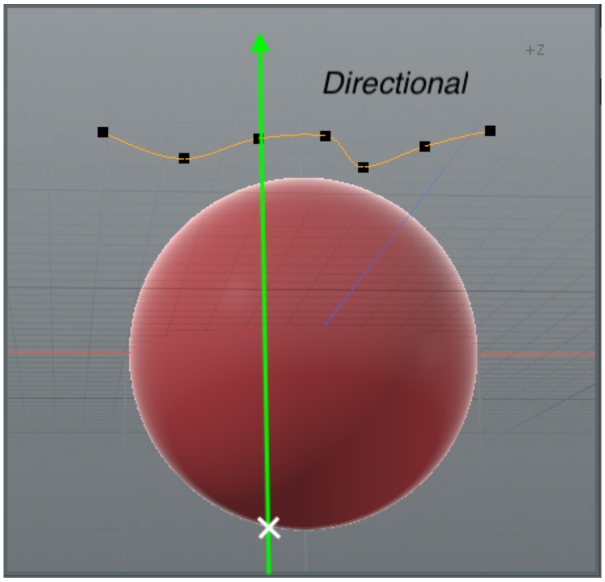
8. In the 3D viewport, swelect the teapot mesh item, press **W**, and drag in the 3D viewport to move it into the cube.

Surface Constraint Options

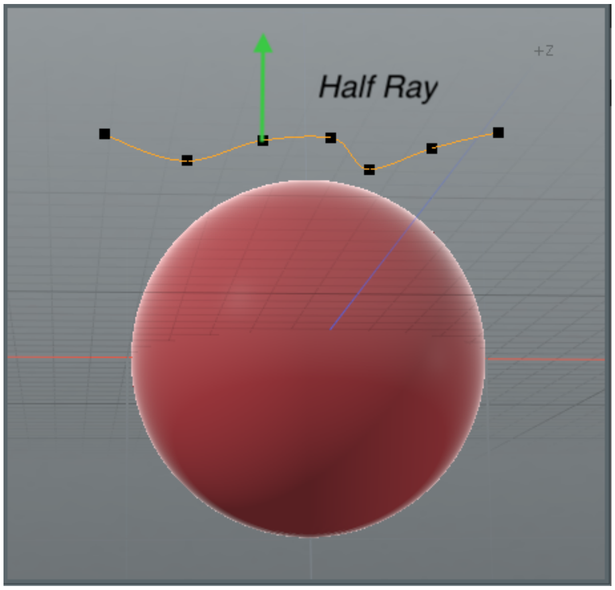


Mesh Operation
☑ Enable
☑ Use World Transform
Mode: Directional
Axis: X | **Y** | Z
☑ Invert
Offset: 0 m
Max Distance: 0 m
Strength: 100.0 %
☐ Double Sided
☐ Save Surface Normal to Vertex Normal Map

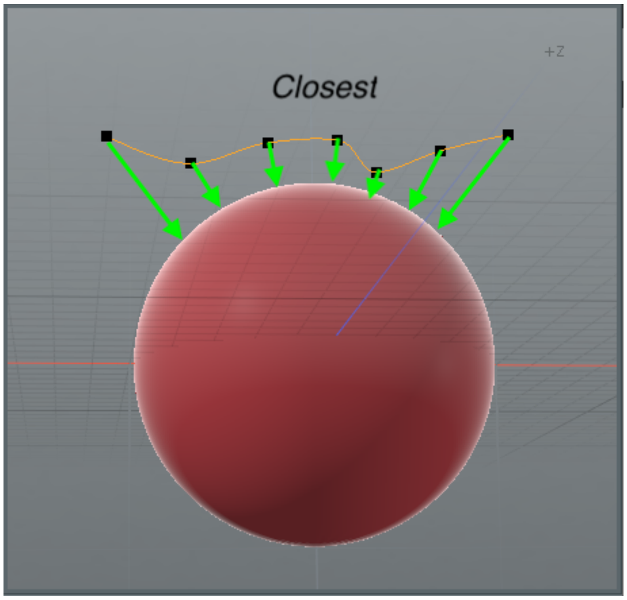
- **Enable** - Enables or disabled the Push tool.
- **Use World Transform** - Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
- **Mode** - Controls the way the vertices of a mesh item are constrained. The following options are available.
  - **Directional**- Forces the entire object onto the surface of another object based on a defined **Axis**. The vector is an infinite directional.



- **Half Ray** - Allows you to move an item around and have it’s geometry constrained to the surface of another object as it connects with that surface.



- **Closest** - Constrains the points of an object to another surface based on the shortest distance to that surface.



- **Axis** - Specifies the axis for **Directional** or **Half Ray** modes. The axis vector is defined for item transform space.
- **Invert** - Inverts the axis vector.
- **Offset** - Allows you to define a distance away from a surface for the constraint.
- **Max Distance** - Allows you to define a maximum distance to be constraint. If distance from point postion to surface object is greater than this value, The point is not moved. 0 m of Max Distance is for infinite distance. Point position is constrainted to object surface with any distance.
- **Strength** - Modulates the strength of the constraint. At 100%, Modo deforms the mesh fully at the constraint position.
- **Double Sided** - When enabled, this setting treats the background polygons as double sided, so geometry can be constrained regardless of the constraining surface's normal facing direction.
- **Save Surface Normal to Vertex Normal Map** - When enabled, the surface normal vector on constraint surface item at the constraint position is saved into vertex normal map of operating geometry mesh.

Constrain to Guide

Modo allows the creation of visual guide lines in the scene to be used as reference or to enhance snapping. You can create guide lines in 3D space and then use them as a snapping tool.

Constrain to Guide Properties

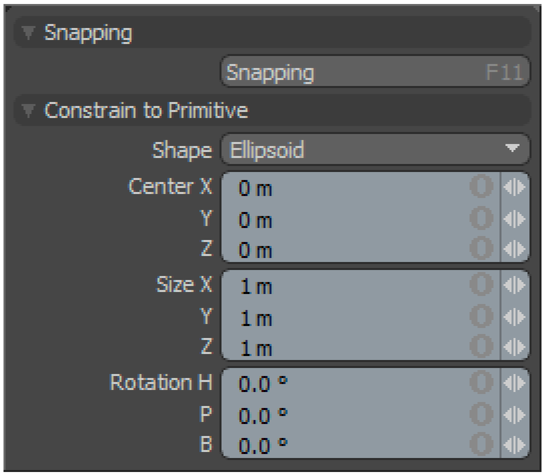
Current Guide	Allows you to choose which guide to create or edit by choosing from one of

	the ten options, each with independent settings.
Enable Guide	Enabled or disabled for snapping.
Show Guide	Visibility of the current guide.
Snap to Guide	Snapping to the guide.
Lock Guide	Once you have set the appropriate guide options, it can be helpful to lock the guide with this toggle to avoid accidentally changing the guide's path. When set, the main guide area changes from yellow to cyan.
Distance	Determines the length of the path.
Position	When using the **Two Points** mode of the guide function, the position option determines the start point of the guide. A straight line is drawn between this value and the end value that represents the guide line itself. When using the **Planar Ray** option, the **Position** value determines the home position for the guide.
Mode	You can define guides in two ways: **Planar Ray** for only planar facing guides, or **Two Points** for guides of any angle. • **Planar Ray** -You can define a single point, a plane on which the line sits, and an angle to determine the guide line. You can use the handle in the viewport to rotate the guide line. • **Two Points** - You can define two single points in 3D space and the guide line is created through the intersection of those two points. You can precisely position the line by dragging the two handles in the 3D viewport.
Plane	When choosing the **Planar Ray** guide mode, this option sets which plane (of the 3 axial planes) that the ray rests upon.
Angle	When choosing the **Planar Ray** guide mode, the **Angle** option determines the rotation of the guide around the **Position** point.
End	When choosing the **Two Point** guide mode, the **End** option determines the position of the second point.

Constrain to Primitive

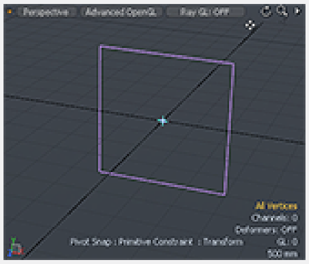
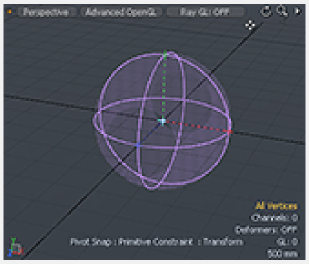
Setting the **Mesh Constraint** to **Primitive** creates a virtual primitive shape onto which tool handles can be constrained to. Similar in concept to **Background**, however, no physical geometry needs to be present for the **Primitive** option.

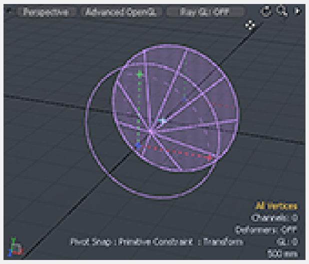
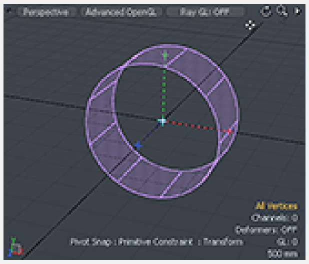
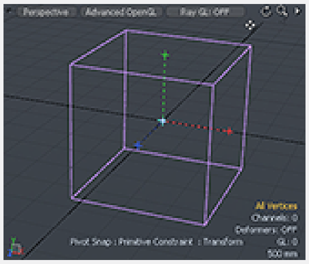
When active, the following attributes appear in the **Properties** panel under the active tool.



Snapping
Snapping F11
Constrain to Primitive
Shape: Ellipsoid
Center X: 0 m
Y: 0 m
Z: 0 m
Size X: 1 m
Y: 1 m
Z: 1 m
Rotation H: 0.0 °
P: 0.0 °
B: 0.0 °

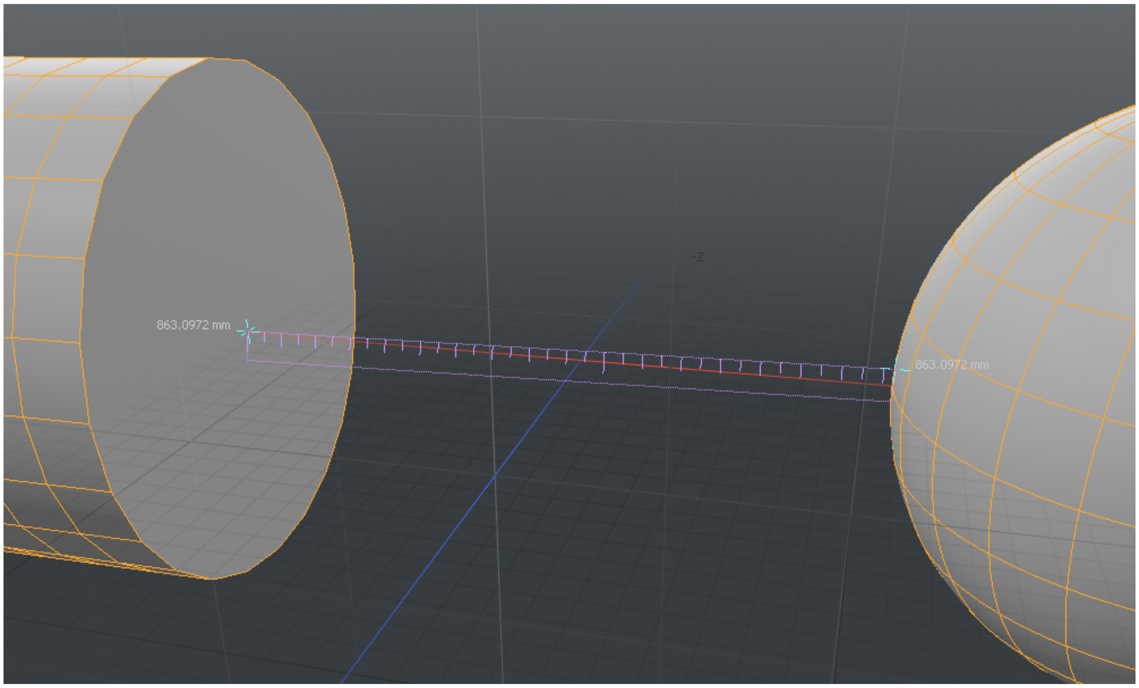
Constrain to Primitive Properties

Shape	You can choose from the various primitive shapes available for constraint:  **Plane**  **Ellipsoid**

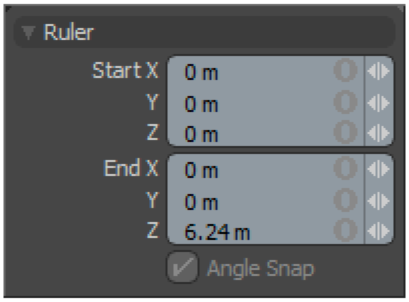
	 Perspective Advanced OpenGL Ray GL: OFF All Vertices Channels: 0 Deformers: OFF GL: 0 500 mm Pivot Snap : Primitive Constraint : Transform **Cone**  Perspective Advanced OpenGL Ray GL: OFF All Vertices Channels: 0 Deformers: OFF GL: 0 500 mm Pivot Snap : Primitive Constraint : Transform **Cylinder**  Perspective Advanced OpenGL Ray GL: OFF All Vertices Channels: 0 Deformers: OFF GL: 0 500 mm Pivot Snap : Primitive Constraint : Transform **Cube**
Center	Defines the center position of the virtual constraint element, based on the center of the bounding box.
Size	Defines the bounding box, which sets the size of the virtual constraint element.
Rotation	Defines the rotation amount for the virtual constraint element.

Using Utilities

Ruler Tool



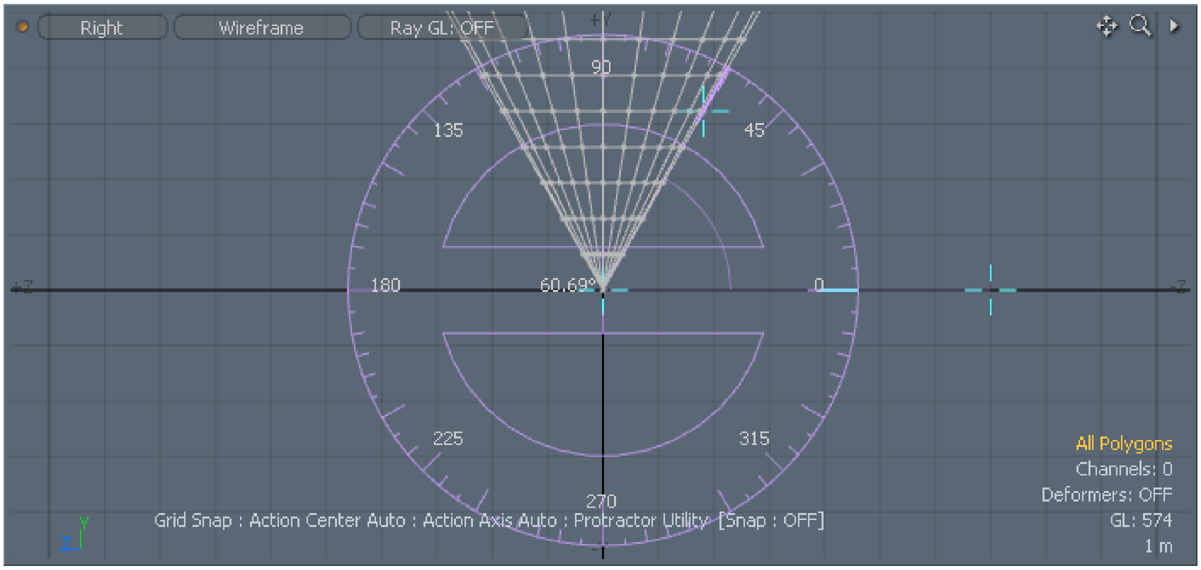
The Ruler command creates a virtual measuring device in the viewport that lets you quickly measure the distance between two locations. To activate the tool, in the menu bar, click **View** > **Ruler Tool**. Once activated, the Ruler becomes visible in the 3D viewport, adding an additional tool **Properties** panel. You can hover the mouse cursor directly over the handles at either end and click and drag to position the handles in the scene. The values on the ruler interactively update as the ruler is positioned in the scene. The value displays are view-dependent, so moving the view position of the viewport adjusts the values for ease of reading. At either end, there is a readout of the ruler's total length. To reposition the ruler itself, you can click and drag the ruler element, repositioning it within the scene. You can also use the Ruler **Start** and **End** fields of the property form to define a fixed size ruler within the scene. When combined with the Applying Snapping function, the ruler is a powerful option for measuring elements in a scene.



Ruler

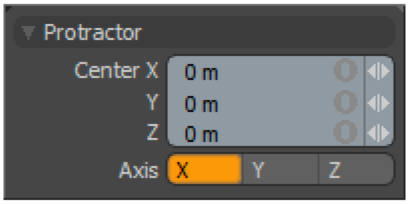
Start X 0 m
Y 0 m
Z 0 m
End X 0 m
Y 0 m
Z 6.24 m
☑ Angle Snap

Protractor Tool



Grid Snap : Action Center Auto : Action Axis Auto : Protractor Utility [Snap : OFF]

All Polygons
Channels: 0
Deformers: OFF
GL: 574
1 m

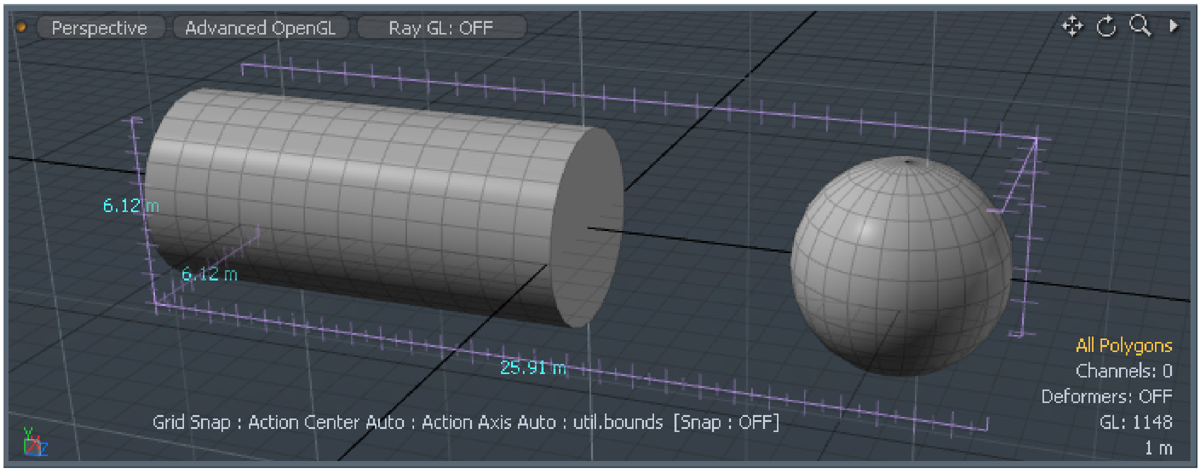


Protractor

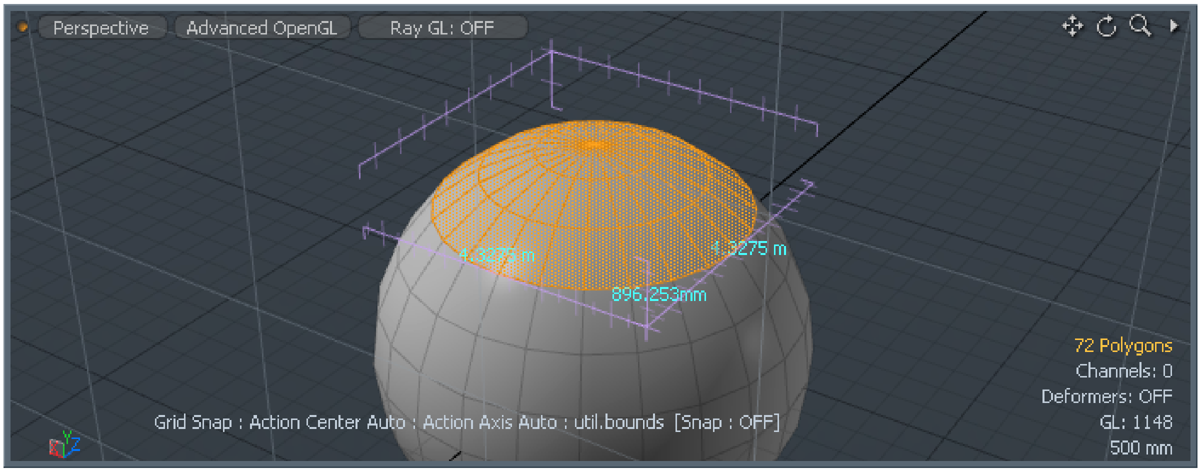
Center X: 0 m
Y: 0 m
Z: 0 m
Axis: [✓ X] Y Z

The Protractor tool provides you with a simple visual method for accurately measuring angles in the 3D viewport. You can activate the tool in the menu bar, in **View** > **Protractor Tool**. Once activated, the Protractor tool becomes visible in the 3D viewports. There are three handles you can use to position and measure elements in the scene. The handle at the center of the Protractor itself positions the tool within the scene. The second tool handle controls the rotation of the tool representation and the third handle measures the angle, producing the readout display. To adjust a handle, you can simply hover the mouse pointer over the handle representation and click and drag to adjust. You can also use the tool's **Properties** panel to position the center of the Protractor. When combined with the Snapping function, the Protractor is a powerful option for measuring angles in a scene.

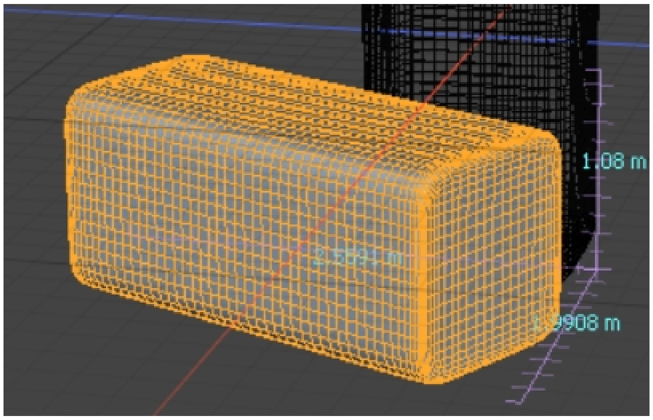
Dimensions Tool


Perspective
Advanced OpenGL
Ray GL: OFF
6.12 m
6.12 m
25.91 m
All Polygons
Channels: 0
Deformers: OFF
GL: 1148
1 m
Grid Snap : Action Center Auto : Action Axis Auto : util.bounds [Snap : OFF]

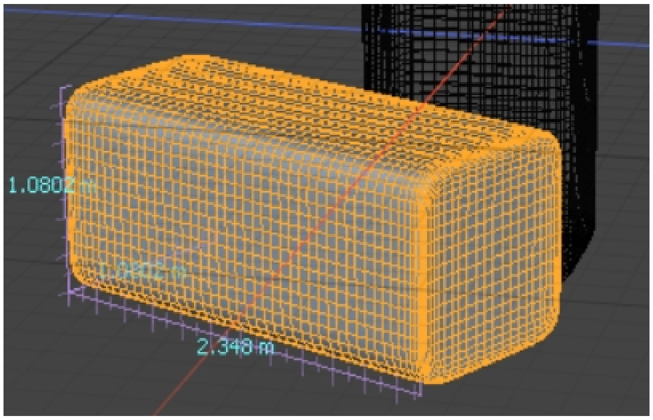
The Dimensions tool displays the bounding box dimensions of any object or component selection. When activated through the menu command **View** > **Dimensions Tool**, the dimensions become visible within the 3D viewport, displaying the size of the element in all three axes. The Dimensions tool requires no user input or interaction outside of making a selection.


Perspective
Advanced OpenGL
Ray GL: OFF
4.3275 m
4.3275 m
896.253mm
72 Polygons
Channels: 0
Deformers: OFF
GL: 1148
500 mm
Grid Snap : Action Center Auto : Action Axis Auto : util.bounds [Snap : OFF]

Enabling the **Orient to Selection** control displays the dimensions relative to the selected element, rather than aligned to the major axes. This makes the measurements easier to read in the context of your selection.



Orient to Selection disabled.



Orient to Selection enabled.

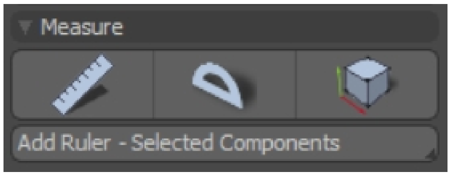
The Absolute Scaling and Uniform Scaling tools also include the **Orient to Selection** control, which causes the **Grab Size** button to account for the orientation of your selection and performs any uniform scaling relative to that orientation, rather than along the major axes.

Measure Scene Distances with the Ruler Array

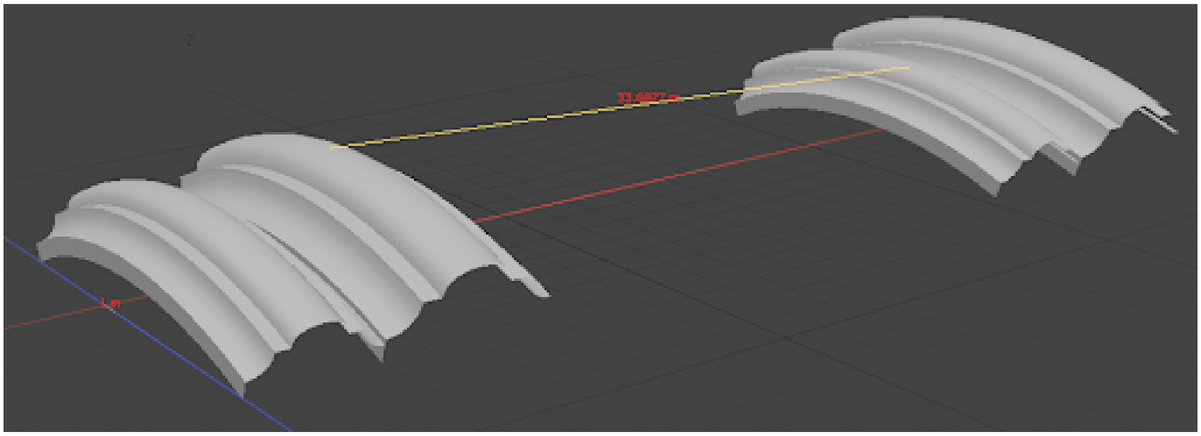
The ruler allows you to select vertices on two objects and then draw a polyline to calculate their distance. This is useful when working to a specific scale or scene layout where you may be working to a reference or blueprints. Guides created by this operation can also be used as guides for snapping. The ruler is available as a MeshOperation or as a direct modeling tool.

To use the ruler as a direct modeling tool:

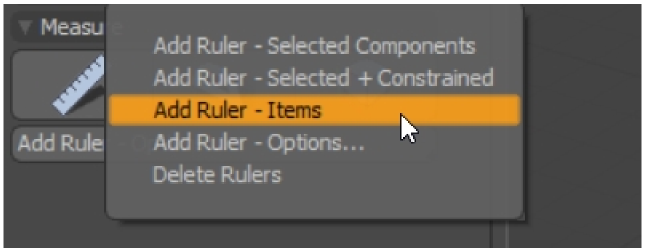
1. Select two or more meshes in the item tree to make them active in the scene.
2. Switch to **Vertices** mode using the button above the viewport.
3. Select two or more vertices on separate, active meshes.
4. In the **Create** tab in the toolbar, click the **Add Ruler** button.



A ruler is created between the selected vertices.



5. Right-click the **Add Ruler** button to apply different types of ruler from a dropdown list:

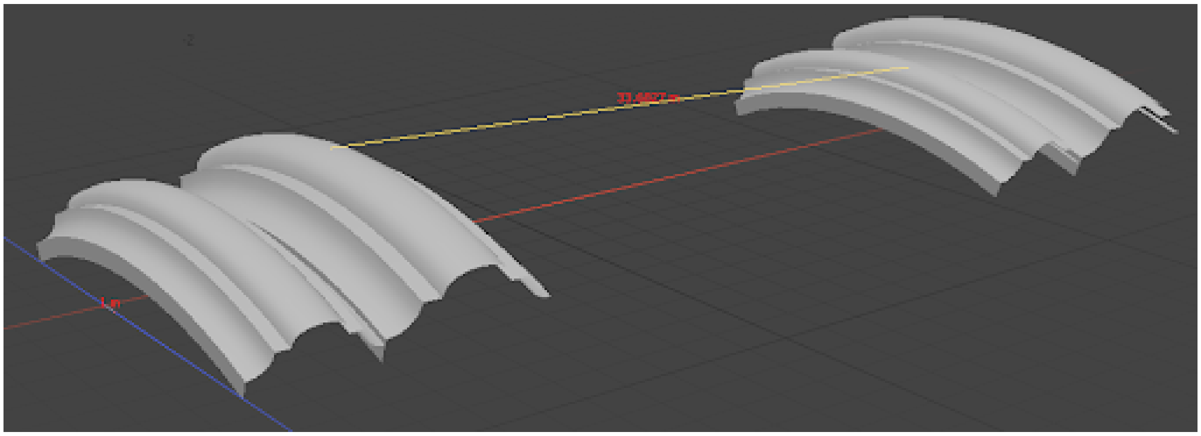

Measu
Add Rule
Add Ruler - Selected Components
Add Ruler - Selected + Constrained
Add Ruler - Items
Add Ruler - Options...
Delete Rulers

- **Add Ruler - Selected Components**: Applies a ruler between two or more selections, but the ruler does not update if any selected components are moved
- **Add Ruler - Selected + Constrained**: Adds a ruler between two or more selections and constrains the ruler to the selection origin. This means if you move any of your selected objects, the ruler updates to match any changes in distance.
- **Add Ruler - Items**: Allows you to create rulers between multiple items instead of selected points on an item. Like **Selected + Constrained**, a ruler created by the items option automatically updates the distance displayed if any items are moved.
- **Add Ruler - Options...**: Allows you to specify that you want a ruler created only between selected items and enables you to constrain the ruler. You can also specify the axis the ruler is locked to in the options window.
- **Delete Ruler**: deletes any ruler arrays present in the scene.

To use the ruler as a MeshOp:

1. Select two or more meshes in the item tree to make them active in the scene.
2. Switch to **Vertices** mode using the button above the viewport.
3. Select two or more vertices on separate, active meshes.
4. In the **Mesh Operation** viewport, add a **Ruler** MeshOp.
5. In the Ruler **Properties** tab, change the **Input Method** to **Selected Vertices**.

A ruler is created between the selected vertices.



Ruler MeshOp Properties

Using the Ruler MeshOp gives you greater control over your ruler and allows you to adjust how it looks with parameters such as **Tick Offset**, **Tick Length**, and **Text Scale**.

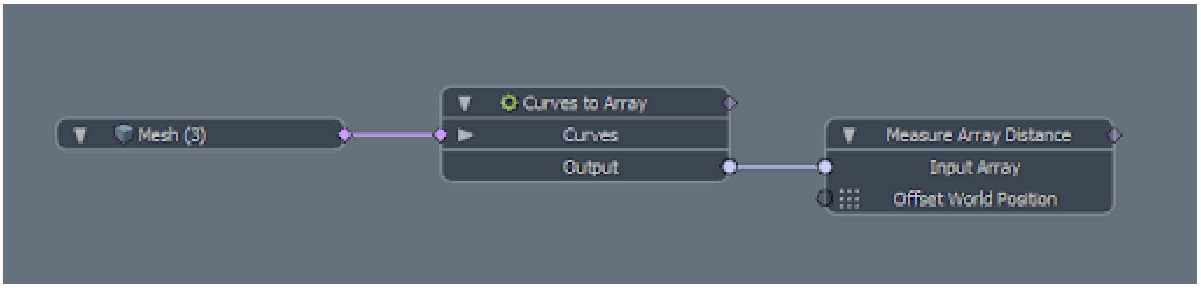
Input Method	• Local Values - Uses the A and B values at the bottom of the form. • Array - Uses the input array channel to get positions. • Selected Vertices - Creates rulers between selected vertices on the current mesh. If no vertices are selected, Modo attempts to connect all vertices.
Facing Axis	Sets the axis to which the ticks and distance face.
Draw Ends	When enabled, draw polylines to represent the end of the ruler.
Draw Ticks	When enabled, draws ticks at even intervals between the start and end of the ruler.
Tick Offset	When **Draw Ticks** is enabled, this control determines how far apart the ticks are placed.
Tick Length	When **Draw Ticks** is enabled, this control determines the length of each tick.
Tick Vertices	When Draw Ticks is disabled, this control determines if ticks are still drawn or not.
Draw Distance	When enabled, draws a text polygon showing the distance between the

	selected vertices.
Font	Sets the font used to display distances.
Text Offset	Sets how far from the ruler the text is drawn.
Text Along Ruler	When enabled, the text is drawn along the ruler. When disabled, the text is drawn horizontally.
Text Scale	Sets the size of the display text.
Use Offset	Allows a World Position Matrix to be plugged in, controlling where the ruler starts.
Link To Offset	When enabled, draws polylines offset from the original selected positions.
Offset Draw Style	When Link to Offset is enabled, controls how the offset lines are drawn.
Extend Distance	Extends the ruler in all directions by this value.

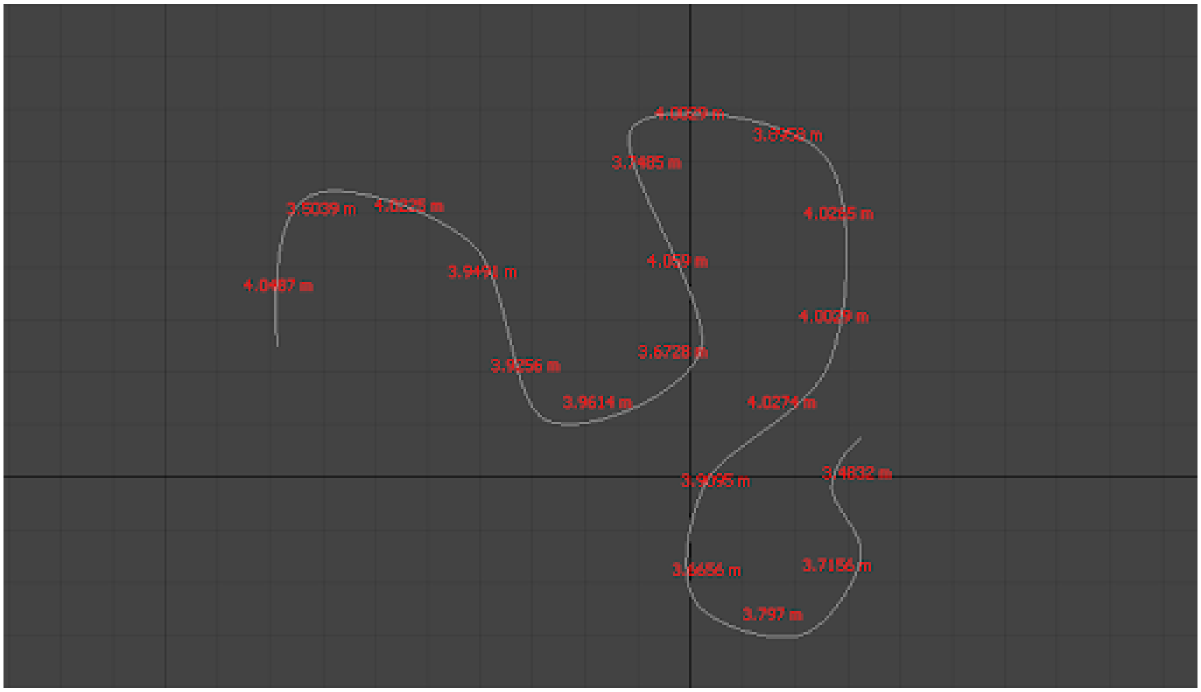
Drawing the Distance Between Arrays

The **Measure Array Distance** node, available in the Schematic viewport, also allows you to measure the distance between arrays.

To use the node, plug your chosen array into the **Input Array** channel in the **Measure Array Distance** node.



Once plugged in, you will be able to see the distances between arrays marked in the 3D viewport.


4.0029 m
3.8958 m
3.7485 m
3.5039 m
4.0225 m
4.0265 m
4.059 m
4.0487 m
3.9491 m
4.0029 m
3.6728 m
3.9256 m
3.9614 m
4.0274 m
3.4832 m
3.9095 m
3.6656 m
3.7156 m
3.797 m

In the **Properties** tab of the **Measure Array** node, you have the option to offset the distances from their respective points and to draw a GL line between them.

Transforming Items

With the transform tools in Modo you can quickly, easily, and most importantly, accurately apply transformations to items — whether hauling or applying values numerically. All the transform tools are universal in how they work. They respect the settings for falloffs, action centers, and the work plane. You can apply them in any mode: component modes (Vertices, Edges, and Polygons), Items, Center, Pivot, and even Materials modes. For information on using a specific tool, click any link below to see its topic.

Tip: You can middle mouse button click without dragging to repeat the last modeling or transform tool action.

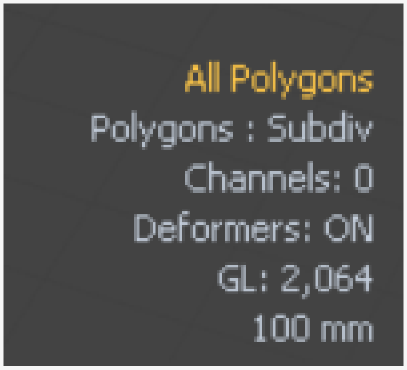
Cage Mode

Cage Mode is a tool used in SubDiv Modeling that allows you to edit a smoothed or subdivided mesh whilst working with a low poly wireframe (or cage). Working with the cage is a good way to make large edits to a high poly mesh quickly. When cage mode is enabled, models will still render at a higher render layer, also known as a Render Level.

Enabling Cage Mode

In Modo 15 or later, **Cage Mode** is equivalent to zero sub divisions. This means that **Cage Mode** no longer needs to be toggled on and off in the **Subdivision** menu found in the **Surface** tab of the **Properties** window and can now be enabled by setting the **Subdivision Level** to zero.

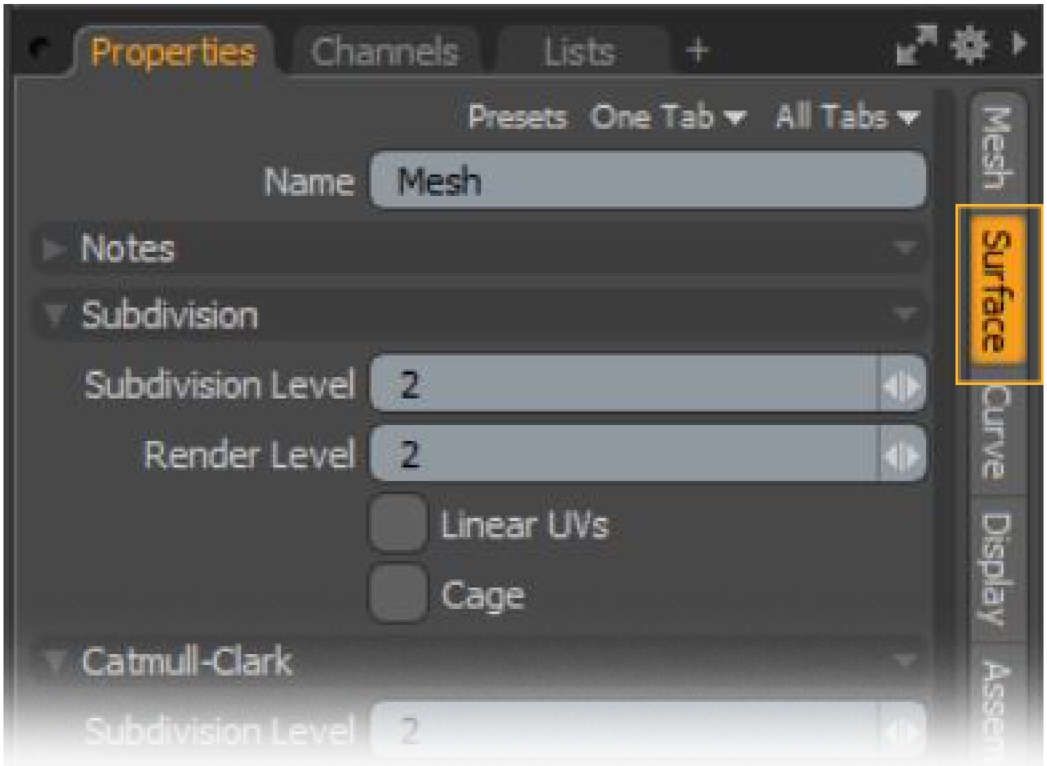
Tip: To enter the SubDiv modeling space, press the **Tab** key. You will be able to tell if you are in the SubDiv space by whether your model is Smooth or not. A smooth model means you are in the SubDiv space, while a faceted model means you are not.



All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 2,064
100 mm

You can also check the current polygon type of your mesh in the bottom right corner of the 3D Viewport, as shown in the image above.

To activate the tool, navigate to the **Surface** tab within the **Properties** editor.


Properties
Channels
Lists
+
Presets
One Tab
All Tabs
Mesh
Surface
Curve
Display
Assem

Name: Mesh

Notes

Subdivision

Subdivision Level: 2

Render Level: 2

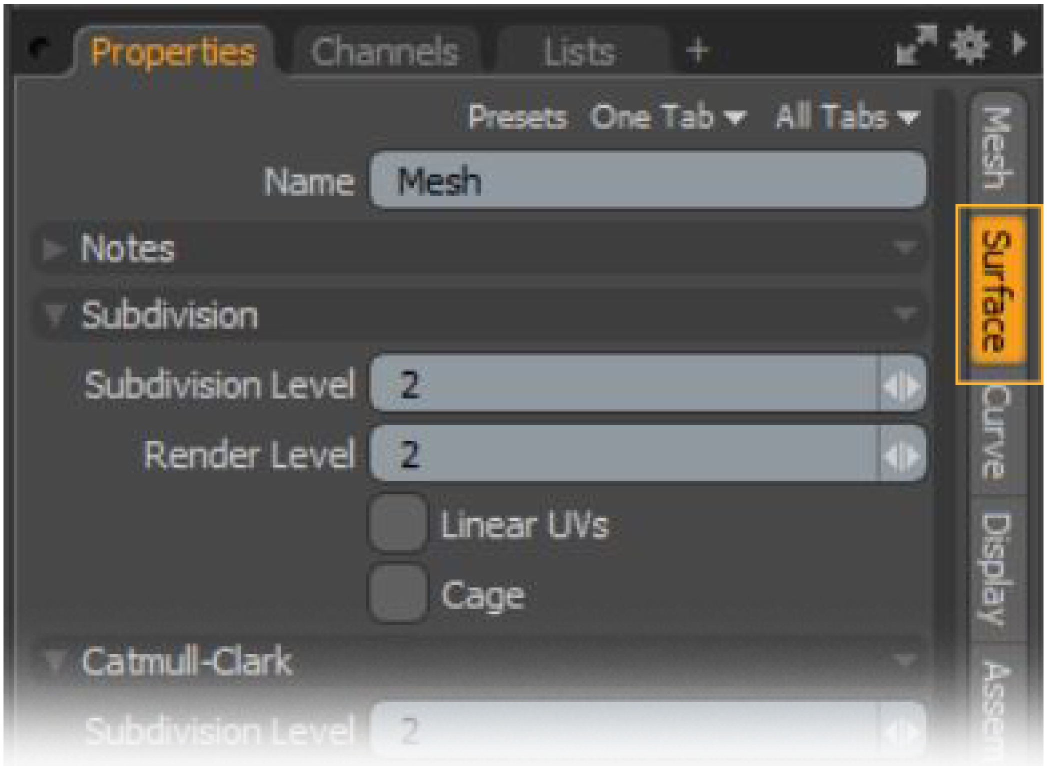
☐ Linear UVs

☐ Cage

Catmull-Clark

Subdivision Level: 2

Under the **Subdivsion** menu, set **Subdivision Level** to zero. This will enable Cage Mode.


Properties
Channels
Lists
+
Presets
One Tab
All Tabs
Mesh
Surface
Curve
Display
Assem

Name: Mesh

Notes

Subdivision

Subdivision Level: 2

Render Level: 2

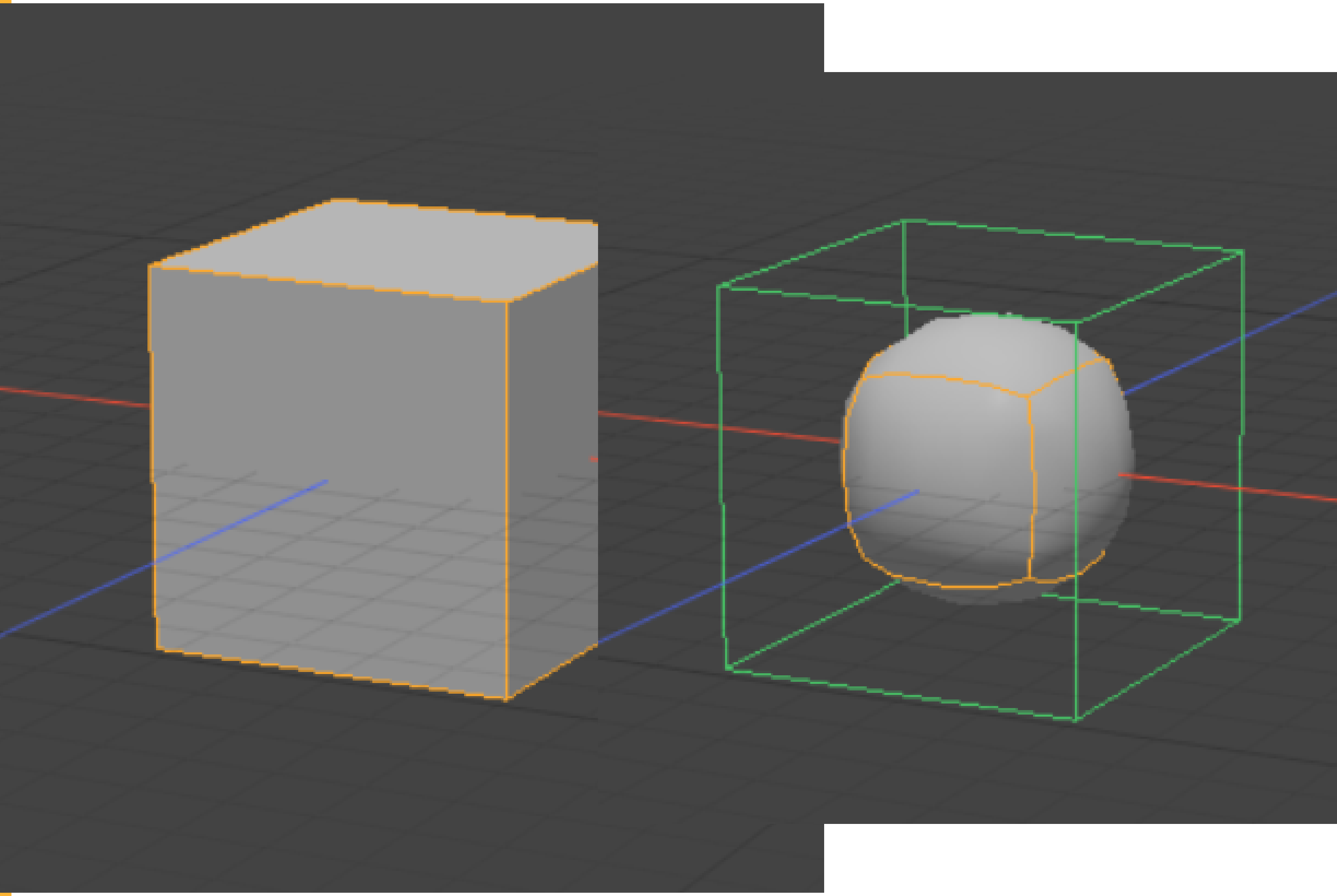
☐ Linear UVs

☐ Cage

Catmull-Clark

Subdivision Level: 2

Tip: To view your cage even while subdivisions are enabled, press **O** on your keyboard, then navigate to the **Active Meshes** tab and select **Show Cages**.



Cage Enabled

Cage Disabled

Duplicating Geometry with the Transform Tools

You can transform and duplicate geometry at the same time by using the transform tools (Transform, Move, Scale, Rotate). To do this, activate the tool, then hold **Ctrl**/**Cmd**+**Shift**, and drag on the handles in the 3d viewport.

Depending on the selection type, you can do the followings:

- In **Items** mode, create an instance of the item.

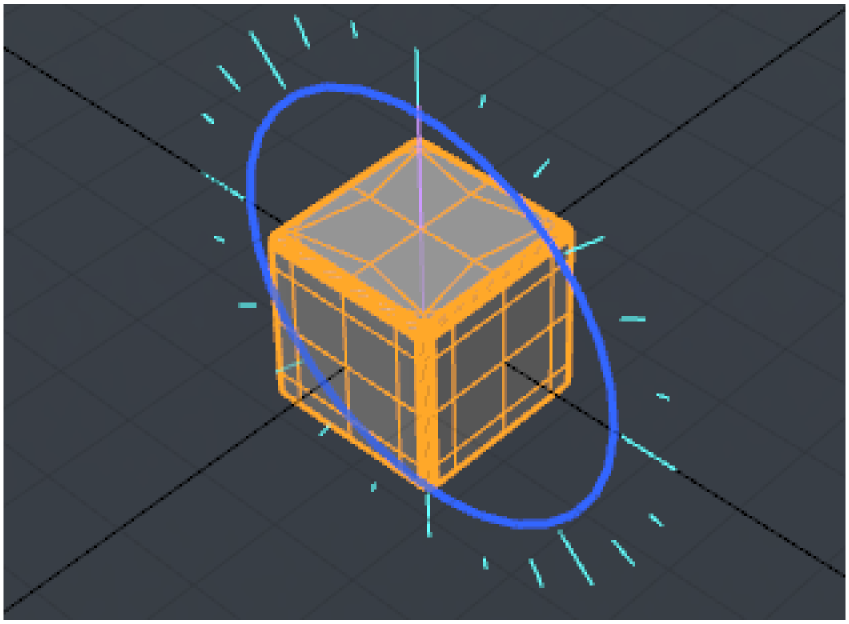
Note: You can convert an instance into actual geometry by right-clicking the instance, then selecting **Change Type** > **Mesh**.

- In **Polygons** mode, copy the selected polygons.
- In **Edges** mode, extend the edge. Clicking on the move handles edits the current segment, clicking outside of the handles creates a new segment.
- In **Vertices** mode, slide the vertex along one of the connecting edges, the one that is most aligned to the transformation vector.

Transform		
	Transform	Combines Move, Rotate, and Scale transformations into a single tool.
	Move	Moves or translates a selection across a specific distance.
	Rotate	Rotates a selection by a specific number of degrees.
	Axis Rotate	A Rotate tool that it is limited to editing a single axis at a time.
	Scale	Makes a selection larger or smaller.
	Uniform Scale	Constrains all three axes, forcing uniform scaling operation.
	Tack	Moves and rotates a polygon and/or all connected vertices to any other polygon in the model.
	Center Commands	Commands that center a geometry selection at the World Origin position.

Axis Rotate

The **Axis Rotate** tool allows you to rotate a selection by a specific number of degrees. The **Axis Rotate** tool is different from the standard **Rotate** tool in that it is limited to editing a single axis at a time, determined by the **Work Plane** position, when activated. It also has a different on-screen control display than the other transform tools, providing a greater degree of control over the amount of rotation depending on how long or short you drag out the handle length (only when the **Advanced Handle** option is enabled). Holding the **Ctrl** key constrains the rotation to fixed intervals defined by the **Snap Angle** amount. After applying a transform amount, you may press the middle mouse button to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the **Basic** sub-tab of the **Modeling** viewports. You may need to press the **Shift** key to show alternate tools in the toolbox. You can also activate the tool from the menu bar under **Edit** >**Transform** > **Axis Rotate** or by pressing the **Alt**+**Shift**+**E** keyboard shortcut.



In the component modes (**Vertices**, **Edges**, and **Polygons**), the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (highlighted) in the Items (Scenes) List. In **Items** mode the tool affects all the currently-selected items in the **Item List**, if nothing is selected, then the tools appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the **Center** and **Pivot** modes, it modifies the Center or Pivots respectively.

Note: You need to physically select either the Center or the Pivot itself (depending on the mode) prior to activating the tool. As in **Items** mode, with no selection, the tool appears to do nothing.

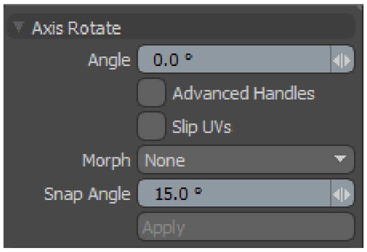
When you define any of the various falloffs Modo provides, application of the tool can be attenuated across a user-defined area. For more on using falloffs, please refer to the Using Falloffs topic.

Once the tool becomes active, Modo draws the relevant tool handle within the viewport. At this point, you can simply click and haul anywhere inside the desired viewport to interactively rotate the geometry.

Depending on the action center, this may also set the rotation center, based on the mouse click position intersecting the **Work Plane**.

Additionally, you may prefer to utilize the on-screen display, either clicking on the wheel to rotate to a specific amount, or by dragging the handles on the value bar across the top of the viewport (when **Advanced Handles** is enabled). When clicking on the wheel, a handle is drawn, displaying the number of degrees. When using this handle the rotation sensitivity is adjusted based on the distance the mouse is from the handles as you drag. This can be thought of as extending a lever. The longer the lever, the more mouse movement is required to rotate the element and, thus, the granularity of control is increased. To interactively scale the size of the handles themselves, you may use the **-** (minus) and **=** (equals) keys on the keyboard to adjust them: minus makes them smaller, while equals makes them larger.

The **Axis Rotate** tool also respects the settings for the Action Center, and originates the action from the defined position. For more information on understanding and using Action Centers, please refer to the Specifying Action Centers and Falloffs topic.



Axis Rotate

Angle 0.0 °

☐ Advanced Handles

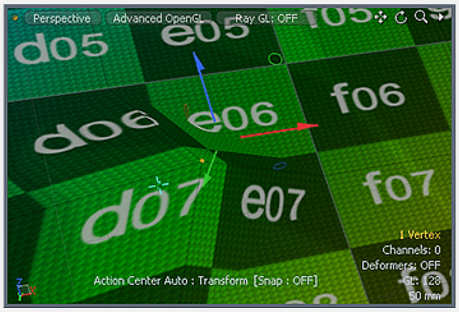
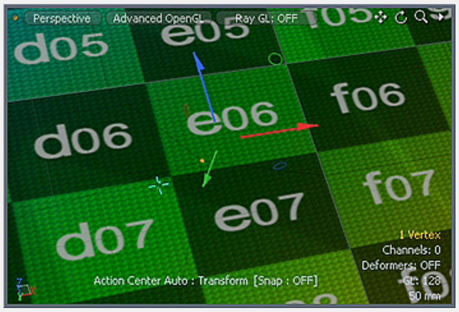
☐ Slip UVs

Morph None

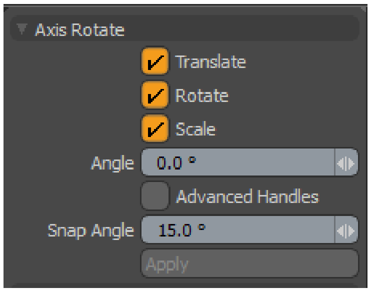
Snap Angle 15.0 °

The following **Axis Rotate** options are available when in component mode:

Angle	Use the **Angle** input field to apply specific rotational value, calculated in degrees. When adjusting an object interactively in the viewport, the **Angle** input field displays the current rotation amount, providing useful feedback.
Advanced Handles	Determines whether or not the rotation display widget displays the additional controls and information in the viewport. When **Advanced Handles** are enabled, the handle itself can be snapped to the angle indicators in the display. Press and hold the **X** key to temporarily toggle snapping.
Slip UVs	Only in component selection modes. UV values are generally fixed to specific vertices; subsequently, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways, requiring you to adjust the map or redo it altogether. To avoid this undesirable result, you can enable the **Slip UVs** function, so as to not disturb any existing UV mapping applied to the geometry.

	 **Slip UVs** function disabled: texture warping occurs.  **Slip UVs** function enabled: texture remains even.
Morph	Only available in component modes. Determines how Modo treats stored **Morph** information, when applying transforms to geometry (Move, Rotate, Scale). In previous versions of Modo, in order to transform a morph, along with its base, it needed to be selected in the **Vertex Map** list. If it wasn't, relative morph map data, when recalled, would produce distorted, undesirable results. If you are unaware of the requirements, it's easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the morph map vertex data is dealt with when applying any transforms: • **None** - Unselected **Morph** data is not affected, though selected (visible) morphs can still be transformed, independent of their source. • **Transform** - With this option selected, **Morph** data is transformed, along with the base mesh. • **Keep Positions** - With this option selected, **Morph** data is converted into an absolute morph map and all vertices retain their pre-transformed positions.
Snap Angle	Defines the angle quantization amount when holding down the **Ctrl** key. Make sure to press **Ctrl** prior to clicking in the 3D viewport.

The following additional **Axis Rotate** options are available when in **Items** mode:



Axis Rotate

☑ Translate

☑ Rotate

☑ Scale

Angle 0.0 °

☐ Advanced Handles

Snap Angle 15.0 °

Apply

Translate	Move all of the currently-selected items in a single axis dimension.
Rotate	Rotate all of the currently-selected items in a single axis dimension.
Scale	Scale all of the currently-selected items larger or smaller in a single axis dimension.

For information about the other options available in **Items** selection mode, see the table above.

The **Snapping** and **Geometry Snap** options are also displayed when the Axis Rotate tool is activated. For more information, see Applying Snapping.

Center Commands

Modo offers a number of centering commands that allow you to quickly center a geometry selection at the World Origin (0,0,0) position. The commands can be invoked from the menu bar **Edit** > **Center Selected**. There is no dialog box or user-specified setting for these commands.

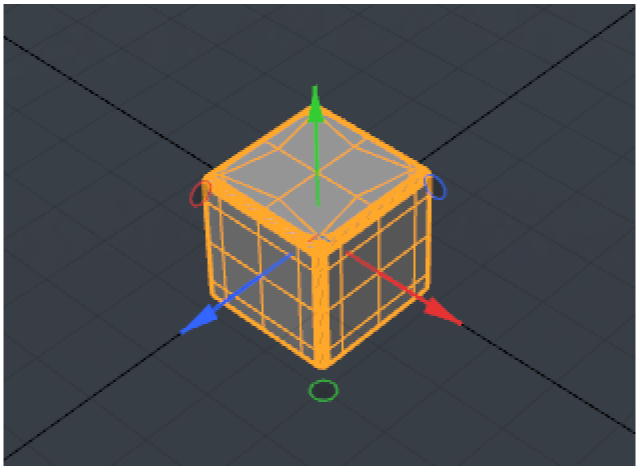
The behavior of **Center** depends on the selected data and the command arguments used. If a single vertex is selected, center moves the vertex to position 0 for the defined axis (or axes). When an edge is selected the center point of the edge is placed at the centered axis (or axes). With a polygon selected, the polygon is moved so that the center of the polygon rests at the center of the defined axis (or axes). When there are more than one vertex, edge, or polygon selected, the center of the bounding volume of that selection is placed at the center of the defined axis (or axes).

The **Center** commands are meant to work on component selections. Should you wish to center an item while in **Item** mode, they should use the **Reset** function available in the items properties, returning that item layers center to the World Origin. You should note that this only centers the item itself, and does not modify the geometry within the item layer.

Warning: If you apply the **Center** commands to a Mesh Item that already has a transformation applied, it is recommended that you also apply the freeze option in the item's properties panel, returning its center to the World Origin. If you don't, strange behavior may occur.

Move

The **Move** tool allows you to move, or translate, a selection across a specific distance. By using the various handles, you can easily constrain the movement to one or two axes. Holding the **Ctrl** key constrains to a single axis when hauling. After applying a transform amount, you may press the middle mouse button to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the **Basic** sub-tab of the **Modeling** viewports, in the menu bar under **Edit** > **Transform** > **Move**, or by pressing the **W** keyboard shortcut.



In the component modes, **Vertices**, **Edges**, and **Polygons**, the tool affects the selected geometry directly; when nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (highlighted) in the Items (Scenes) List. In **Items** mode, the tool affects all the currently-selected items in the **Item List**. If nothing is selected, then the tools appear to do nothing. In **Material** mode, it works the same way as in any of the component modes. Lastly, in the **Center** and **Pivot** modes, it modifies the center or pivots respectively.

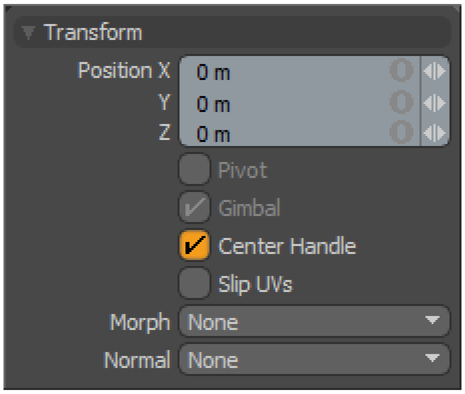
Note: You need to select either the center or the pivot itself (depending on the mode), prior to activating the tool, as in **Items** mode. With no selection the tool appears to do nothing.

When you define any of the various falloffs Modo provides, application of the tool can be attenuated across a user-defined area. For more on using falloffs, please refer to Using Falloffs.

Once the tool becomes active, Modo draws the relevant tool handles within the viewport. At this point, you can simply click and haul anywhere inside the desired viewport to interactively position the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the function to one or two axes.

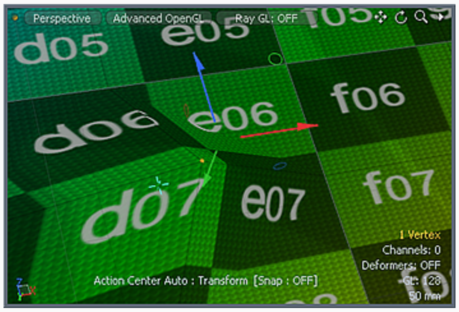
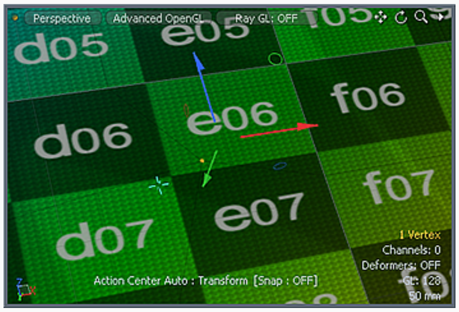
The various tool handles are color-coded to match the coloring of the different axes: red for X, green for Y, and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then click and drag to modify. To interactively scale the size of the handles themselves, you may use the **-** (subtraction) and **=** (equals) keys on the keyboard to adjust them: subtraction makes them smaller, while equals makes them larger.

The **Transform** tool also respects the settings for the **Action Center**, and originates the appropriate action from the defined position. For more information on understanding and using Actions Centers, please refer toSpecifying Action Centers and Falloffs .


Transform
Position X 0 m
Y 0 m
Z 0 m
☐ Pivot
☑ Gimbal
☑ Center Handle
☐ Slip UVs
Morph None
Normal None

The following options are available for adjusting the **Move** tool.

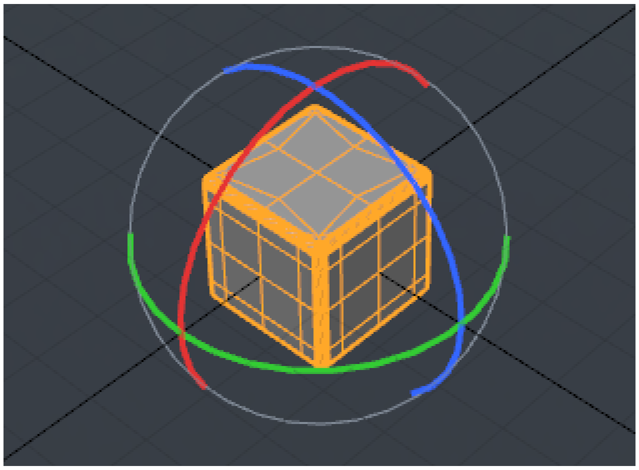
Position	Use the X, Y, and Z input fields to apply specific offset distance values (calculated from the position the object was in when the tool was activated). When adjusting an object interactively in the viewport, the **Position** input fields display the current offset distance applied, providing useful user feedback.
Pivot	Only available in **Items** mode. Allows you to easily modify an item's pivot position without the need to go into **Pivots** mode. **Note:** The **Gimbal** option must be disabled in order to adjust the **Pivot** option.
Gimbal	Disabled in Move mode, as it only applies to rotation transforms.
Slip UVs	Only in component selection modes. UV values are generally fixed to specific vertices; subsequently, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways, requiring you to adjust the map or redo it altogether. To avoid this undesirable result, you can enable

	the **Slip UVs** function, so as to not disturb any existing UV mapping applied to the geometry.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled: texture warping occurs.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled: texture remains even.
Morph	• Only available in component modes. Determines how Modo treats stored **Morph** information, when applying transforms to geometry (Move, Rotate, Scale). In previous versions of Modo, in order to transform a morph, along with its base, it needed to be selected in the **Vertex Map** list. If it wasn't, relative morph map data, when recalled, would produce distorted, undesirable results. If you are unaware of the requirements, it's easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the morph map vertex data is dealt with when applying any transforms: • **None** - Unselected **Morph** data is not affected, though selected (visible) morphs can still be transformed, independent of their source. • **Transform** - With this option selected, **Morph** data is transformed, along with the base mesh. • **Keep Positions** - With this option selected, **Morph** data is converted into an absolute morph map and all vertices retain their pre-transformed positions.
Normal	Only in component modes. When working with vertex normal maps, the **Normal** options control how the map values are affected. Refer to Transform Vertex Normals for more information. • **None** - Vertex normal map values won't be affected at all by Transform operations. • **Edit** - Only vertex normal map values are modified by the tool, and only in the selected Vertex Map. • **Update** - All vertex normal map values, selected or otherwise, and any selected geometry, is modified by the Transform operation.

Child Compensate	Only available in **Items** mode. When disabled, passes along the transform value to any child items of the current selection. When enabled, the child items ignore any transform values. This is useful to transform a single item in a hierarchy, such as adjusting a joint, allowing it to be re-positioned relative to other elements without moving child items.
Center Handle	Toggles the visibility of the small cube icon at the center of the transform handles in the viewport. When enabled, you can click and drag on the handle to move the selection relative to Using the Work Plane. When disabled, the handle is not visible.

Rotate

The **Rotate** tool allows you to rotate a selection by a specific number of degrees. By using the various handles, you can easily constrain the movement to a single axis. Holding the **Ctrl** key constrains the rotation to fixed 15° intervals. After applying a transform amount, you may press the middle mouse button to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the **Basic** sub-tab of the **Modeling** viewports, in the menu bar under **Edit** > **Transform** > **Rotate**, or by pressing the **E** keyboard shortcut.



In the component modes, **Vertices**, **Edges**, and **Polygons**, the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (highlighted) in the Items (Scenes) List. In **Items** mode, the tool affects all the currently-selected items in the **Item List**. If nothing is selected, then the tools appear to do nothing. In **Material** mode, it works the same way as in any of the component modes. Lastly, in the **Center** and **Pivot** modes, it modifies the Center or Pivots respectively.

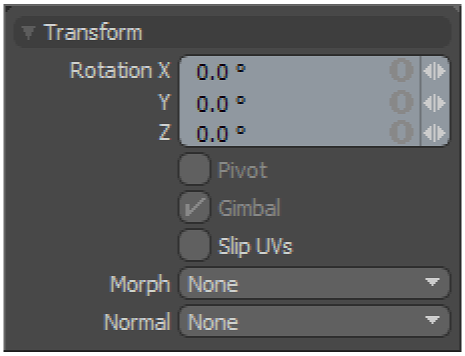
Note: You need to select either the Center or the Pivot itself (depending on the mode), prior to activating the tool, as in the **Items** mode. With no selection, the tool appears to do nothing.

When you define any of the various falloffs Modo provides, application of the tool can be attenuated across a user-defined area. For more on using falloffs, please refer to Using Falloffs.

Once the tool becomes active, Modo draws the relevant tool handles within the viewport. At this point, you can simply click and haul anywhere inside the desired viewport to interactively transform the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the function to a single axis.

The tool handles are color-coded to match the coloring of the different axes: red for X, green for Y, and blue for Z. In some action center modes, there is an additional gray screen space handle around the outside of the tool handles, which rotates the selection based on the viewport cameras postion. Hover the mouse over a specific handle until it changes to a yellow color, and then click and drag to modify. To interactively scale the size of the handles themselves, you can use the **-** (subtraction) and **=** (equals) keys on the keyboard to adjust them: subtraction makes them smaller, while equals makes them larger.

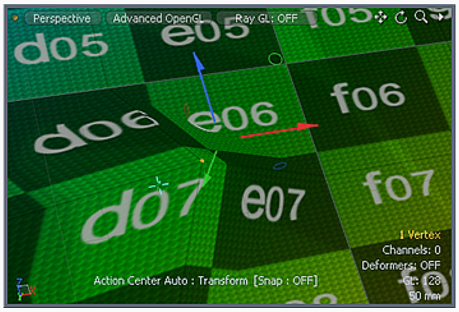
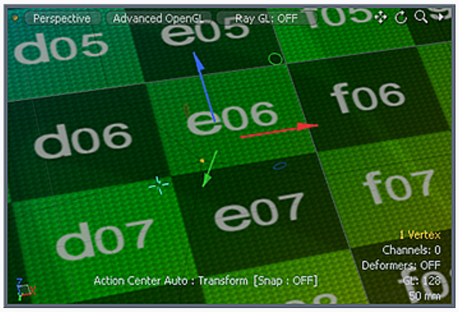
The **Rotate** tool also respects the settings for the **Action Center**, and originates the appropriate action from the defined position. For more information on understanding and using Actions Centers, please refer to Specifying Action Centers and Falloffs .



Transform
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
☐ Pivot
☑ Gimbal
☐ Slip UVs
Morph None
Normal None

The following **Transform** options are available for adjusting the **Rotate** tool:

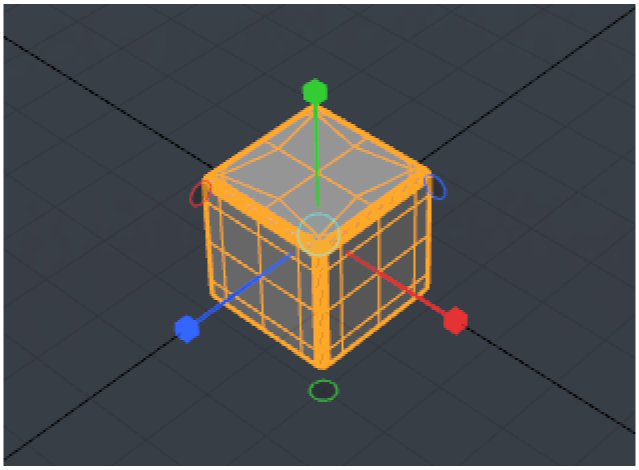
Rotation	Use the X,Y, and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, the **Rotation** input fields display the current offset distance applied, providing useful user feedback.
Pivot	Only available in **Items** mode. Allows you to easily modify an item's pivot position without the need to go into **Pivots** mode.

	Note: The **Gimbal** option must be disabled in order to adjust the **Pivot** option.
Gimbal	When enabled, you can assign rotation values numerically, but may arrive at a situation that produces "gimbal lock" - the loss of one rotation axis when another axis' value is too close. You can avoid this by disabling this setting, but you are then unable to apply specific values within the input fields.
Slip UVs	Only in component selection modes. UV values are generally fixed to specific vertices; subsequently, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways, requiring you to adjust the map or redo it altogether. To avoid this undesirable result, you can enable the **Slip UVs** function, so as to not disturb any existing UV mapping applied to the geometry.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled: texture warping occurs.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled: texture remains even.
Morph	Only available in component modes. Determines how Modo treats stored **Morph** information, when applying transforms to geometry (Move, Rotate, Scale). In previous versions of Modo, in order to transform a morph, along with its base, it needed to be selected in the **Vertex Map** list. If it wasn't, relative morph map data, when recalled, would produce distorted, undesirable results. If you are unaware of the requirements, it's easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the morph map vertex data is dealt with when applying any transforms: • **None** - Unselected **Morph** data is not affected, though selected (visible) morphs can still be transformed, independent of their source. • **Transform** - With this option selected, **Morph** data is transformed, along with the base mesh. • **Keep Positions** - With this option selected, **Morph** data is converted into an

	absolute morph map and all vertices retain their pre-transformed positions.
Normal	Only in component modes. When working with vertex normal maps, the **Normal** options control how the map values are affected. Refer to Transform Vertex Normals for more information. • **None** - Vertex normal map values won't be affected at all by Transform operations. • **Edit** - Only vertex normal map values are modified by the tool, and only in the selected Vertex Map. • **Update** - All vertex normal map values, selected or otherwise, and any selected geometry, is modified by the Transform operation.
Child Compensate	Only available in **Items** mode. When disabled, passes along the transform value to any child items of the current selection. When enabled, the child items ignore any transform values. This is useful to transform a single item in a hierarchy, such as adjusting a joint, allowing it to be re-positioned relative to other elements without moving child items.

Scale

The **Scale** tool allows you to make any selection larger or smaller by changing the scale percentage factor. By using the various handles, you can quickly scale in a single axis dimension. After applying a transform amount, you may press the middle mouse button to re-apply the previous offset amount, making it easy to apply an edit multiple times. The command can be activated in the **Basic** sub-tab of the **Modeling** viewports, in the menu bar under **Edit** > **Transform** > **Scale**, or by pressing the **R** keyboard shortcut.



In the component modes, **Vertices**, **Edges**, and **Polygons**, the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (highlighted) in the Items (Scenes) List. In **Items** mode, the tool affects all the currently-selected items in the **Item List**. If nothing is selected, then the tools appear to do nothing. In **Material** mode, it works the same way as in any of the component modes. Lastly, in the **Center** and **Pivot** modes, it modifies the Center or Pivots respectively.

Note: You need to select either the Center or the Pivot itself (depending on the mode), prior to activating the tool, as in the **Items** mode. With no selection, the tool appears to do nothing.

When you define any of the various falloffs Modo provides, application of the tool can be attenuated across a user-defined area. For more on using falloffs, please refer to Using Falloffs.

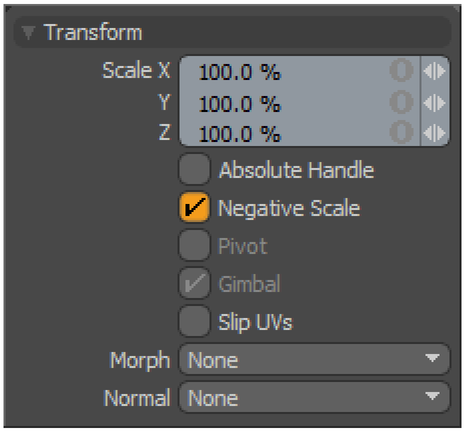
Once the tool becomes active, Modo draws the relevant tool handles within the viewport: arrows for moving, boxes for scaling, and circles for rotating. At this point, you can simply click and haul anywhere inside the desired viewport to interactively transform the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the function to one or all three axes.

The various tool handles are color-coded to match the coloring of the different axes: red for X, green for Y, and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color, and then click and drag to modify. To interactively scale the size of the handles themselves, you can use the **-** (subtraction) and **=** (equals) keys on the keyboard to adjust them: subtraction makes them smaller, while equals makes them larger. Additionally, for the **Scale** tool, there is a small circle at the intersection of all three handles that you can use to uniformly scale the selection interactively. To move your object in the 3rd dimension (forward and backward) in the **3D Viewport** press **Shift** + **Ctrl/Cmd** + **right-mouse-button** and drag.

The **Transform** tool also respects the settings for the **Action Center**, and originates the appropriate action from the defined position. For more information on understanding and using Actions Centers, please refer to Specifying Action Centers and Falloffs .

Transform Scale Options

The following **Transform** options are available for adjusting the **Scale** tool:



Transform

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

☐ Absolute Handle

☑ Negative Scale

☐ Pivot

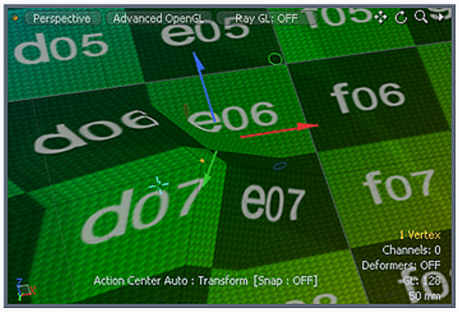
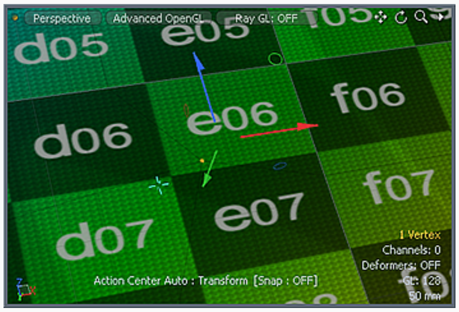
☑ Gimbal

☐ Slip UVs

Morph None

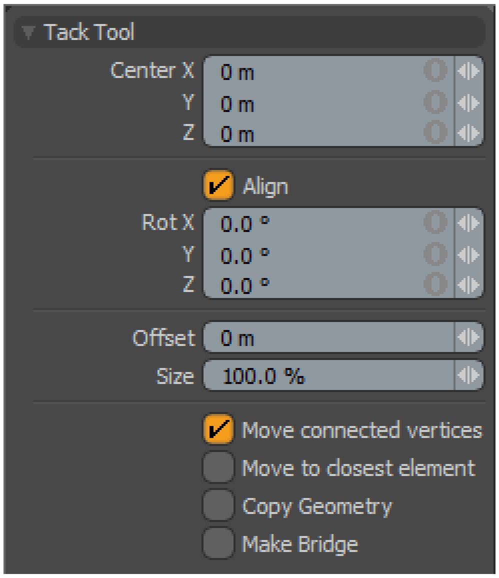
Normal None

Scale	Use the X, Y, and Z input fields to apply specific scaling values. When adjusting an object interactively in the viewport, the **Scale** input fields display the current scale factor applied, providing useful user feedback.
Absolute Handle	Applies an additional function to the tool handles, allowing you to scale the handle itself without affecting the transform amount. In some cases, this provides finer control over very large or very small adjustments.
Negative Scale	Enables/disables Modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, you can always input negative values in any of the input fields directly.
Pivot	Only available in **Items** mode. Allows you to easily modify an item's pivot position without the need to go into **Pivots** mode. **Note:** The **Gimbal** option must be disabled in order to adjust the **Pivot** option.
Gimbal	When enabled, you can assign rotation values numerically, but may arrive at a situation that produces "gimbal lock" - the loss of one rotation axis when another axis' value is too close. You can avoid this by disabling this setting, but you are then unable to apply specific values within the input fields.
Slip UVs	Only in component selection modes. UV values are generally fixed to specific vertices; subsequently, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways, requiring you to adjust the map or redo it altogether. To avoid this undesirable result, you can enable the **Slip UVs** function, so as to not disturb any existing UV mapping applied to the geometry.

	 Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled: texture warping occurs.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled: texture remains even.
Morph	Only available in component modes. Determines how Modo treats stored **Morph** information, when applying transforms to geometry (Move, Rotate, Scale). In previous versions of Modo, in order to transform a morph, along with its base, it needed to be selected in the **Vertex Map** list. If it wasn't, relative morph map data, when recalled, would produce distorted, undesirable results. If you are unaware of the requirements, it's easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the morph map vertex data is dealt with when applying any transforms: • **None** - Unselected **Morph** data is not affected, though selected (visible) morphs can still be transformed, independent of their source. • **Transform** - With this option selected, **Morph** data is transformed, along with the base mesh. • **Keep Positions** - With this option selected, **Morph** data is converted into an absolute morph map and all vertices retain their pre-transformed positions.
Normal	Only in component modes. When working with vertex normal maps, the **Normal** options control how the map values are affected. Refer to Transform Vertex Normals for more information. • **None** - Vertex normal map values won't be affected at all by Transform operations. • **Edit** - Only vertex normal map values are modified by the tool, and only in the selected Vertex Map. • **Update** - All vertex normal map values, selected or otherwise, and any selected geometry, is modified by the Transform operation.
Child Compensate	Only available in **Items** mode. When disabled, passes along the transform value to any child items of the current selection. When enabled, the child items ignore any transform values. This is useful to transform a single item in a hierarchy, such as adjusting a joint, allowing it to be re-positioned relative to other elements without moving child items.

Tack

The **Tack** tool allows you to quickly move and rotate a polygon and/or all connected vertices to any other polygon in the model. To use the tool, simply choose a single polygon in the mesh you wish to move and re-align. With the tool active, click on another polygon in your project. The initial mesh moves and rotates so that the selected polygon is aligned to the polygon that you clicked on.



Tack Tool

Center X 0 m
Y 0 m
Z 0 m

☑ Align

Rot X 0.0 °
Y 0.0 °
Z 0.0 °

Offset 0 m
Size 100.0 %

☑ Move connected vertices
☐ Move to closest element
☐ Copy Geometry
☐ Make Bridge

The following **Transform** options are available for adjusting the **Tack** tool:

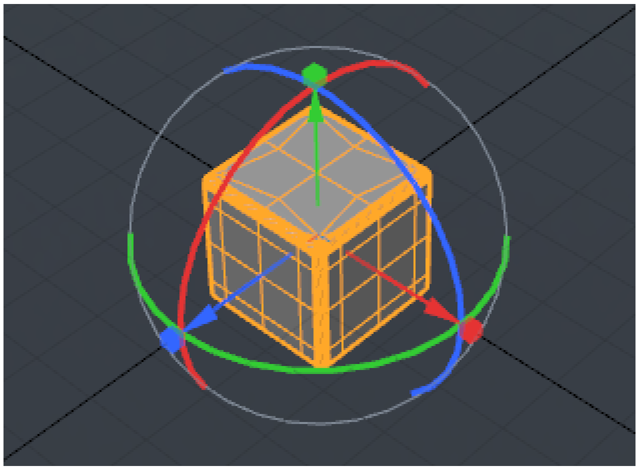
- **Center** - this XYZ triplet is automatically set with each click of the tool. The value represents the 3D coordinates of the element clicked on. After the value has been set automatically, it's possible to edit the values directly, to move the mesh on any of the three axes.
- **Align** - with this option active, the mesh is moved and rotated to align to the surface normal of the target element. Disabling this option results in the mesh only moving to the target element.
- **Rotate** - this XYZ triplet is automatically set with each click of the tool, but only when the **Align** option is active. After the value has been set automatically, it can be manually edited to offset the rotation from the element normal.
- **Offset** - use this value to determine an offset distance perpendicular to the surface geometry it's tacked on to.
- **Size** - use this value to scale the tacked geometry.
- **Move connected vertices** - by default, this option is active so that the selected polygon and all connected vertices are moved and aligned to the target element. When this option is disabled, only the selected polygon is transformed.

- **Move to Closest Element** - snaps the geometry centered onto the closest component element - polygon, edge, or vertex.
- **Copy Geometry** - duplicates the geometry when tacking, instead of simply moving the original. Simultaneously holding down **Shift** when clicking makes a copy per mouse click.
- **Make Bridge** - when this setting is enabled, Modo attempts a **Bridge** operation between the selected polygon and the polygon you've clicked on which to tack the object, making a single connected mesh from the two surfaces.

Transform

The Transform tool combines Move, Rotate, and Scale transformations into a single tool. By using the various handles, you can quickly position, rotate, and scale from one convenient tool. Alternatively, you can click-and-drag in the 3D viewport to position your selection. Holding the **Ctrl** key constrains the position transform to a single axis when hauling, and constrains the rotation transform to 15° increments. After applying a transform amount, you may press the middle mouse button to re-apply the previous offset amount, making it easy to apply an edit multiple times. You can also hold the **Shift** + **Ctrl/Cmd** and drag to duplicate your selected object. In addition, you can move your selected object in the 3rd dimension (forward and backward) in the **3D Viewport** by pressing **Shift** + **Ctrl/Cmd** + **right-mouse-button**.

The command can be activated in the **Basic** sub-tab of the **Model** layout, or by pressing the **Y** keyboard shortcut.



In the component modes, **Vertices**, **Edges**, and **Polygons**, the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected (highlighted) in the Items (Scenes) List. In **Items** mode, the tool affects all the currently-selected items in the **Item List**. If nothing is selected, then the tools appear to do nothing. In

Material mode, it works the same way as in any of the component modes. Lastly, in the **Center** and **Pivot** modes, it modifies the Center or Pivots respectively.

Note: You need to select either the Center or the Pivot itself (depending on the mode), before activating the tool, as in **Items** mode. With no selection, the tool appears to do nothing.

When you define any of the various falloffs Modo provides, application of the tool can be attenuated across a user-defined area. For more on using falloffs, please refer to Using Falloffs.

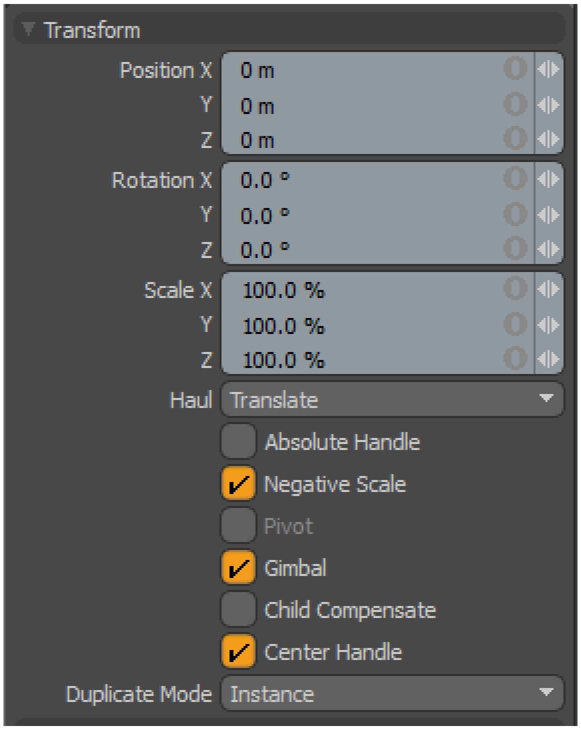
Once the tool becomes active, Modo draws the relevant tool handles within the viewport: arrows for moving, boxes for scaling, and circles for rotating. At this point, you can simply click and haul anywhere inside the desired viewport to interactively transform the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the function to one or two axes.

The various tool handles are color-coded to match the coloring of the different axes: red for X, green for Y, and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color, and then click and drag to modify. To interactively scale the size of the handles themselves, you can use the **-** (subtraction) and **=** (equals) keys on the keyboard to adjust them: subtraction makes them smaller, while equals makes them larger.

The **Transform** tool also respects the settings for the **Action Center**, and originates the appropriate action from the defined position. For more information on understanding and using Actions Centers, please refer to Specifying Action Centers and Falloffs .

Transform Options

The following options are available for adjusting the **Transform** tool.



Transform

Position X 0 m
Y 0 m
Z 0 m

Rotation X 0.0 °
Y 0.0 °
Z 0.0 °

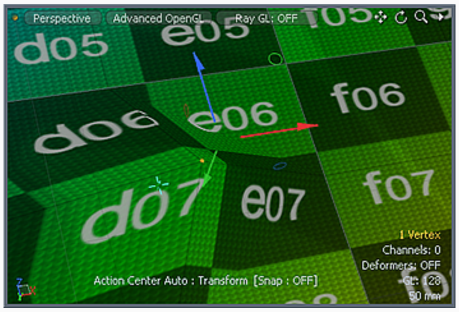
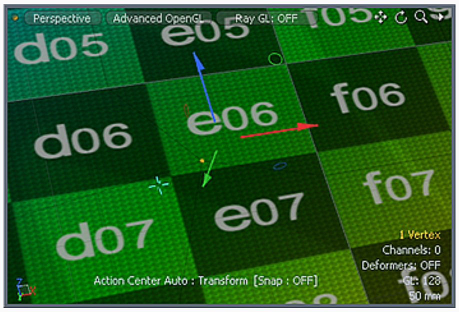
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Haul Translate

☐ Absolute Handle
☑ Negative Scale
☐ Pivot
☑ Gimbal
☐ Child Compensate
☑ Center Handle

Duplicate Mode Instance

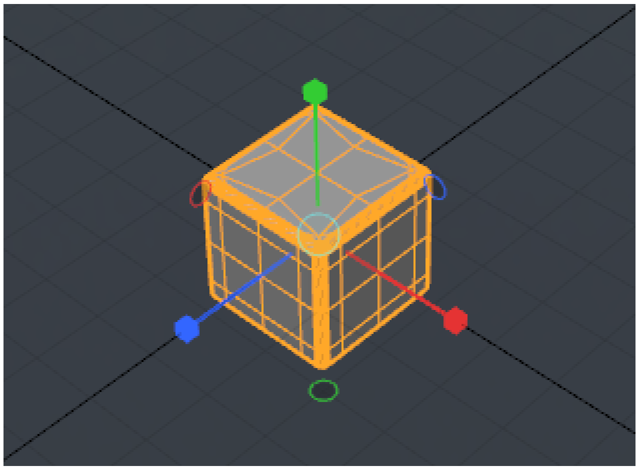
Position	Use the X, Y, and Z input fields to apply specific offset distance values (calculated from the position the object was in when the tool was activated). When adjusting an object interactively in the viewport, the **Position** input fields display the current offset distance applied, providing useful user feedback.
Rotation	Use the X, Y, and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, the **Rotation** input fields display the current offset distance applied, providing useful user feedback.
Scale	Use the X, Y, and Z input fields to apply specific scaling values. When adjusting an object interactively in the viewport, the **Scale** input fields display the current scale factor applied, providing useful user feedback.
Haul	You have three options available when hauling in the viewport (dragging the mouse when not using the tool handles): **Translate** allows you to move the selection, **Rotate** allows you to rotate, and **Scale** allows you to scale the selection.
Absolute Handle	Applies an additional function to the tool handles, allowing you to scale the handle itself without affecting the transform amount. In some cases, this provides finer control over very large or very small adjustments.

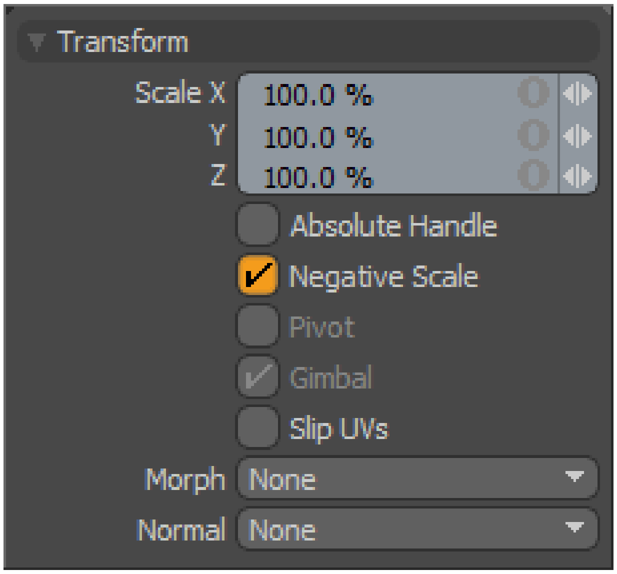
Negative Scale	Enables/disables Modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, you can always input negative values in any of the input fields directly.
Pivot	Only available in **Items** mode. Allows you to easily modify an item's pivot position without the need to go into **Pivots** mode. **Note:** The **Gimbal** option must be disabled in order to adjust the **Pivot** option.
Gimbal	When enabled, you can assign rotation values numerically, but may arrive at a situation that produces "gimbal lock" - the loss of one rotation axis when another axis' value is too close. You can avoid this by disabling this setting, but you are then unable to apply specific values within the input fields.
Center Handle	Toggles the visibility of the small cube icon at the center of the transform handles in the viewport. When enabled, you can click and drag on the handle to move the selection relative to Using the Work Plane. When disabled, the handle is not visible.
Slip UVs	Only in component selection modes. UV values are generally fixed to specific vertices; subsequently, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways, requiring you to adjust the map or redo it altogether. To avoid this undesirable result, you can enable the **Slip UVs** function, so as to not disturb any existing UV mapping applied to the geometry.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled: texture warping occurs.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled: texture remains even.
Morph	Only available in component modes. Determines how Modo treats stored **Morph** information, when applying transforms to geometry (Move, Rotate,

	Scale). In previous versions of Modo, in order to transform a morph, along with its base, it needed to be selected in the **Vertex Map** list. If it wasn't, relative morph map data, when recalled, would produce distorted, undesirable results. If you are unaware of the requirements, it's easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the morph map vertex data is dealt with when applying any transforms: • **None** - Unselected **Morph** data is not affected, though selected (visible) morphs can still be transformed, independent of their source. • **Transform** - With this option selected, **Morph** data is transformed, along with the base mesh. • **Keep Positions** - With this option selected, **Morph** data is converted into an absolute morph map and all vertices retain their pre-transformed positions.
Normal	Only in component modes. When working with vertex normal maps, the **Normal** options control how the map values are affected. Refer to Transform Vertex Normals for more information. • **None** - Vertex normal map values won't be affected at all by Transform operations. • **Edit** - Only vertex normal map values are modified by the tool, and only in the selected Vertex Map. • **Update** - All vertex normal map values, selected or otherwise, and any selected geometry, is modified by the Transform operation.
Child Compensate	Only available in **Items** mode. When disabled, passes along the transform value to any child items of the current selection. When enabled, the child items ignore any transform values. This is useful to transform a single item in a hierarchy, such as adjusting a joint, allowing it to be re-positioned relative to other elements without moving child items.

Uniform Scale

The **Uniform Scale** tool is identical to the **Scale** tool, except that you may haul anywhere in the viewport and all three axes are constrained, forcing all scaling operation to be uniform. This ensures the mesh does not become distorted when scaling.





Transform

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

☐ Absolute Handle

☑ Negative Scale

☐ Pivot

☑ Gimbal

☐ Slip UVs

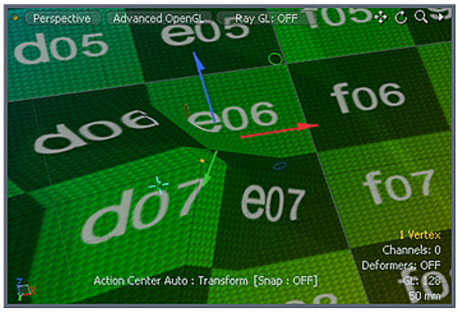
Morph None

Normal None

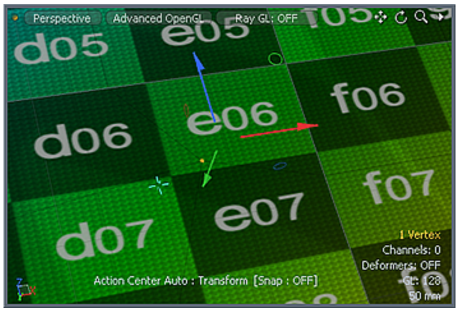
The following **Transform** options are available for adjusting the **Uniform Scale** tool:

- **Scale** - you may use the X,Y, and Z input fields to apply specific scaling values. When adjusting an object interactively in the viewport, the **Scale** input fields display the current scale factor applied, providing useful user feedback.
- **Absolute Handle** - applies an additional function to the tool handles, allowing you to scale the handle itself without affecting the transform amount. In some cases, this provides finer control over very large or very small adjustments.
- **Negative Scale** - enables/disables Modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, you can always input negative values in any of the input fields directly.
- **Pivot** - allows you to easily modify an item's pivot position without the need to go into **Pivots** mode.
- **Gimbal** - setting is grayed out in **Scale** mode.
- **Slip UVs** - (only in component modes) UV values are generally fixed to specific vertices; subsequently, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways,

requiring you to adjust the map or redo it altogether. To avoid this undesirable result, you can enable the **Slip UVs** function, so as to not disturb any existing UV mapping applied to the geometry.


Perspective
Advanced OpenGL
Ray GL: OFF
d05
e05
f05
d06
e06
f06
d07
e07
f07
1 Vertex
Channels: 0
Deformers: OFF
GL: 128
50 mm
Action Center Auto : Transform [Snap : OFF]

Slip UVs function disabled: texture warping occurs.


Perspective
Advanced OpenGL
Ray GL: OFF
d05
e05
f05
d06
e06
f06
d07
e07
f07
1 Vertex
Channels: 0
Deformers: OFF
GL: 128
50 mm
Action Center Auto : Transform [Snap : OFF]

Slip UVs function enabled: texture remains even.

- **Morph** - (only in component modes) determines how Modo treats stored **Morph** information, when applying transforms to geometry (Move, Rotate, Scale). In previous versions of Modo, in order to transform a morph, along with its base, it needed to be selected in the **Vertex Map** list. If it wasn't, relative morph map data, when recalled, would produce distorted, undesirable results. If you are unaware of the requirements, it's easy to accidentally mess up a model. To remedy this problem, there are now three options controlling how the morph map vertex data is dealt with when applying any transforms:
  - **None** - unselected **Morph** data is not affected, though selected (visible) morphs can still be transformed, independent of their source.
  - **Transform** - with this option selected, **Morph** data is transformed, along with the base mesh.
  - **Keep Positions** - with this option selected, **Morph** data is converted into an absolute morph map and all vertices retain their pre-transformed positions.
- **Normal** - (only in component modes) when working with vertex normal maps, the **Normal** options control how the map values are affected. Refer to Transform Vertex Normals for more information.
  - **None** - vertex normal map values won't be affected at all by **Transform** operations.
  - **Edit** - only vertex normal map values are modified by the tool, and only in the selected Vertex Map.
  - **Update** - all vertex normal map values, selected or otherwise, and any selected geometry, is modified by the **Transform** operation.
- **Child Compensate** - (only in **Items** mode) when disabled, passes along the transform value to any child items of the current selection. When enabled, the child items ignore any transform values. This is useful to transform a single item in a hierarchy, such as adjusting a joint, allowing it to be re-positioned relative to other elements without moving child items.

Absolute Scaling

Absolute Scaling allows you to determine the dimensions of your mesh and then rescale it uniformly, explicitly or in reference to the dimensions of your mesh using precise numeral values. While it is possible to rescale a mesh item once it has been created, hauling the scale tool or adjusting the **Scale X/Y/Z** channels in the **Mesh Properties** window can often be imprecise.

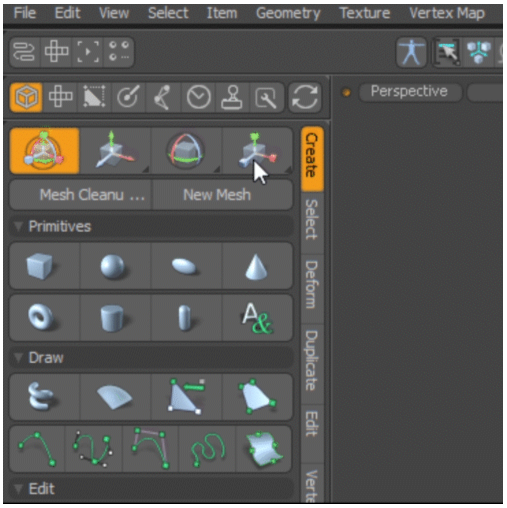
> **Note:** If transforms are applied to your mesh, your mesh may not scale as you would expect. To work around this, you must freeze all transformations on your mesh. For more information on freezing your mesh, see Freezing Geometry.

There are two ways **Absolute Scaling** can be enabled:

- Selecting it from the **Snaps and Precision Palette**

  The **Snaps and Precision Palette** can be opened via the menu bar by clicking **Edit** > **Snapping** > **Snaps and Precision**

- Holding down the **Scale** tool button, and then selecting **Absolute Scaling** from the pop-up menu.

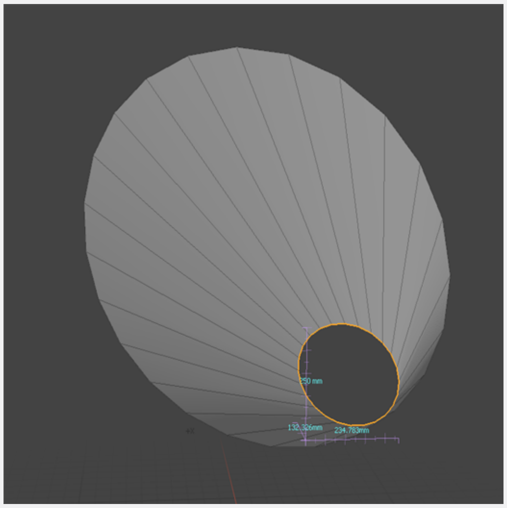
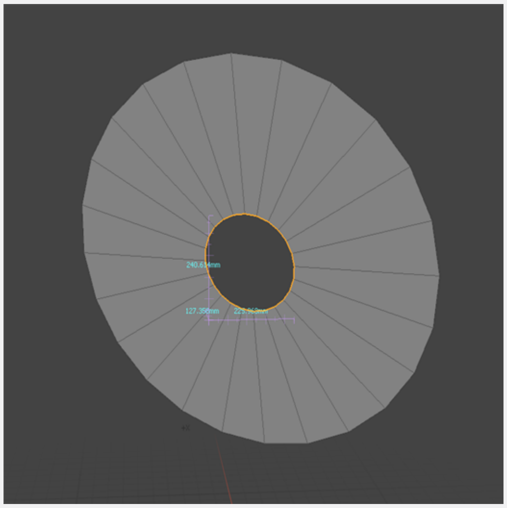

File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Perspective
Mesh Cleanu ...
New Mesh
Primitives
Draw
Edit
Create
Select
Deform
Duplicate
Edit
Verte

The following commands are available within the **Absolute Scaling** palette:

Grab Size	This command grabs the actual bounding box size and inputs it into the **Uniform** and **Explicit** data fields as a point of reference for sizing the selection.

Uniform

The **Uniform Scale** options provide a means to uniformly scale all axes of a selection in a single step. To use, first choose the **Grab Size** command. This captures the longest axis length and inserts it into the **Uniform** value input field as a point of reference. Next, select a scaling center if you want it to be other than the default bounding box center. Enter a new explicit value into the **Uniform** input field. In **Uniform Axis**, define the axis you wish to scale absolutely, and then finally click the **Uniform Scale** button to perform the scale operation.

Uniform Scale	Uniformly scales the bounding box for the selected item by the value in the **Uniform** input field, based on the **Uniform Axis** field.
Uniform Axis	When scaling the bounding box, **Uniform Axis** determines on which axis the scaling command operates, with the other two axes scaled in proportional amount in relation to the defined axis.
Orient to Selection	**Orient to Selection** prevents warping from occurring when scaling selected edges or faces that are not aligned with a major plane or axis.  Unaligned rescaled edge without **Orient to Selection**  Unaligned rescaled edge with **Orient to Selection**

Explicit

The **Explicit Scale** option provides a means to define a specific size to the bounding box of the selected elements. To use it, first choose the **Grab Size** command. This captures the bounding box volume size of the entire selection. Next, select a scaling center if you want it to be other than the default bounding box center. Enter new size values into the **X**,**Y** and **Z** input fields, and click the **Explicit Scale** option to scale the object to the specified size.

Explicit Scale	Scales the bounding box for the selected item by the value in the **X**,**Y** and **Z** input fields.
Reference Scale	The selected section of geometry (assuming it's a smaller section of a whole) defines the scaling amount as a reference, but all connected geometry is scaled proportionately to the selected area. For instance, if you know the size of the cockpit of a race car, you can scale the entire car by the proportional amount based on just the selected cockpit area.

Scaling Center

Scaling Center X/Y/Z	When scaling a selected area, these values define the origin of the scaling operation as defined from the drop-down selection. The **Low** option defines the side toward the negative end of the axis, **Center** refers to the averaged center position of all vertices. The **High** is the side of the bounding box nearest the positive values of the axis.

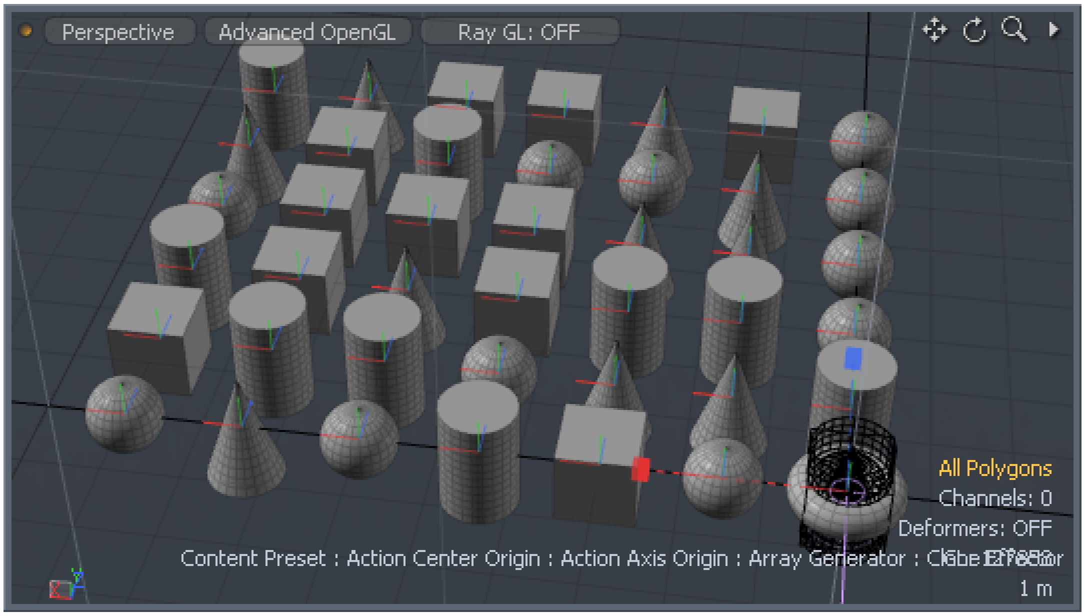
Duplicating Items

Often, it's the details that make objects compelling, and Modo's toolbox helps you focus on details when needed. By providing a variety of cloning and array tools, Modo makes it easy for you to produce multitudes of seemingly complex forms from basic geometry. With the Radial Sweep and Bridge tools you can extend and shape a surface very quickly. Extrusions provide a diverse set of options for generating and extending existing geometry, and therefore, you can take something simple and make it complex. For information on using a specific tool, click a link below to see that tool's topic.

Duplicate		
	Radial Sweep	Creates geometry by revolving the selection around a central point.
	Curve Extrude	Extrudes a geometry selection along the length of a curve.
	Sketch Extrude	Extrudes selected polygons by drawing in the viewport.
	Bezier Extrude	Extrudes a geometry selection along the length of a Bezier curve.
	Pen Extrude	Extrudes a geometry selection along the length of a pen path where each knot creates a new segment.
	B-Spline Extrude	Extrudes a geometry selection along the B-Spline curve.
	Curve Clone	Clones geometry by clicking down control points for a curve.
	Open Curve Offset	Copies an existing curve(s) and positions it parallel to the existing curve at a specified offset value.
	Closed Curve Offset	Copies an existing curve(s), positions it at a specified offset value, and joins them together.
	Bezier Clone	Duplicates an object along a Bezier path a user-defined number of times.
	Pen Clone	Creates linear paths that have perfectly straight line segments between each user-defined point.
	Scatter Clone	Generates a random point cloud onto which Modo places clones of the original geometry.
	Clone	Creates duplicates of the currently selected geometry based on a specified linear offset.
	Array	Produces ordered duplicates of a mesh by cloning a source element into a grid-like pattern.

Duplicate		
	Radial Array	Clones any selected geometry recursively into a circular or helical pattern.
	Mirror	Create a duplicate of the selected geometry in the active layer.
	Mesh Paint	Allows you to directly apply one mesh layer onto the surface of another.
	Bridge	Creates polygons to span the area between selected edges or polygons.
	Paste Tool	Add copies of any geometry currently in the Clipboard.
	Duplicate Commands	Duplicate, Duplicate Hierarchy, Duplicate Options, and Instance commands.
Procedural Duplicate		
	Curve Fill	Fills closed, non-intersecting curves with quadrangles.
	Curve Rebuild	Uses a curve to build a new, similar curve but with a different amount of points.
	Pen Generator	Uses an existing polyline to provide a path for other tools to walk and evaluate.
	Content Preset	Offers a series of ready-made profiles, which are different 2D curve styles that can be used to generate new geometry.
	Sweep Effector	Draws geometry along a curve or line that is generated by tools such as the Pen Generator or Path Generator.
	Lace Geometry	Draws a tube along the path of a curve.
	Symmetrize	Procedural mesh operation that deletes half of the mesh and mirrors the opposite half in its place.

Array


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
Content Preset : Action Center Origin : Action Axis Origin : Array Generator : Clone Effector
1 m

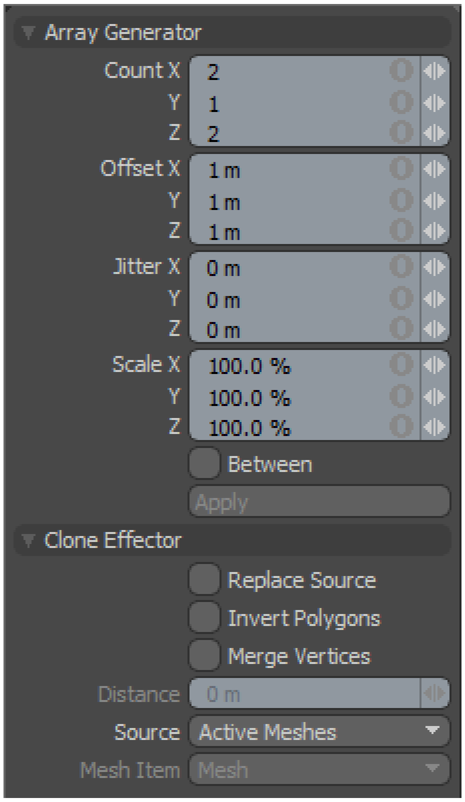
The Array tool produces ordered duplicates of a mesh (or multiple meshes) by cloning a source element into a grid-like pattern. By specifying the **Source**, you can define whether to clone individual items, random items, or a specific, defined item layer.

Usage

You first determine what to clone — either by selecting the target element(s) when cloning into the current foreground layer or by setting the visibility of item layers in the Items (Scenes) List when you want to clone background items. Modo generates cloned items in the topmost foreground layer. Next, activate the Array tool by clicking the **Array** icon in the **Duplicate** sub-tab of the modeling toolbox or by choosing **Geometry** > **Duplicate** > **Array**. Alternatively, to add a procedural Array, in the **Mesh Operations** list, click the **Add Operator** button, and navigate to **Mesh Operations** > **Duplicate** > **Array**, or enter its name in the search bar. For more information, see Array.

Once you have activated the tool, click in any 3D viewport to enable the interactive tool mode and adjust its attributes by dragging in the viewport. You can also click and adjust offset values for the tool handles. Each handle's color corresponds to its axis. Dragging interactively adjusts the spacing of the array. (Horizontal dragging adjusts one axis and vertical dragging adjusts another. The two adjusted axes are entirely dependent on the current position of the Work Plane.) If working in the Perspective view, rotate the viewport's orientation to change the axes being targeted. You can also right-click and drag in the 3D viewport to add or subtract additional clones (interactively adjusting the **Count** value). As before, the adjusted axes are dependent on the direction of dragging and the current Work Plane position. You can constrain any hauling action to the initial dragging direction by pressing **Ctrl** before clicking in the viewport. Alternatively, you can type values in the tool's **Properties** panel to make adjustments while in interactive mode or prior to entering interactive mode and then click **Apply** to invoke the cloning. By right-clicking the **Array** icon in the toolbox, you can choose other related Array tools.

Array Properties



Array Generator

Count X 2
Y 1
Z 2
Offset X 1 m
Y 1 m
Z 1 m
Jitter X 0 m
Y 0 m
Z 0 m
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
☐ Between
Apply

Clone Effector

☐ Replace Source
☐ Invert Polygons
☐ Merge Vertices
Distance 0 m
Source Active Meshes
Mesh Item Mesh

Array Generator	
Count X/Y/Z	Determines the number of clones in each axial direction. The array produces a volume of cloned elements that can increase polygon counts very quickly — even with small numbers. (For example, a 4x4x4 array produces 64 cloned elements.) The use of Instances (Instance Array tool) or Replicators (Replica Array tool) can limit the amount of geometry generated by the Array tool.
Offset X/Y/Z	Determines the distance away from the source element to produce the cloned elements, which Modo calculates as the distance between the source and the first order of clones. Subsequent clones are offset the same amount again. If you enable

Array Generator	
	Between, this value determines the distance between the first and last clones with all others evenly spaced between those two.
Jitter X/Y/Z	Provides random variation to be added to any of the three axes (X, Y, or Z). Modo positions each copied mesh with the **Offset** and a random variation using the **Jitter** distance as the maximum possible offset variation (offset from the new position within the array).
Scale X/Y/Z	Scales the cloned elements to resize them based on the specified values. All cloned elements in the array receive the same amount of scaling.
Between	When enabled, uses the **Offset** value to generate the clones between the calculated first position and the last position.
Clone Effector	
Replace Source	When enabled, removes the source geometry (the initial geometry being cloned) and replaces it with a cloned element. By enabling this Modo can apply **Jitter** and **Scale** values to the source, which would otherwise remain unmodified.
Invert Polygons	When enabled, reverses the normal direction on the cloned geometry. This can be useful in instances where negatively cloned elements flip their normals inward, producing an inside-out geometry.
Merge Vertices	When enabled, automatically joins vertices into a single vertex based on the **Distance** setting.
Distance	Determines the maximum distance between neighboring vertices for Modo to join them into a single vertex. This only applies when you enable **Merge Vertices**. A setting of 0 m only joins vertices that are collocated.
Source	Determines the source mesh that is cloned into the array. • **Active Meshes** - Uses the currently selected geometry as the clone source. • **Specific Mesh** - Specifies a mesh item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position.

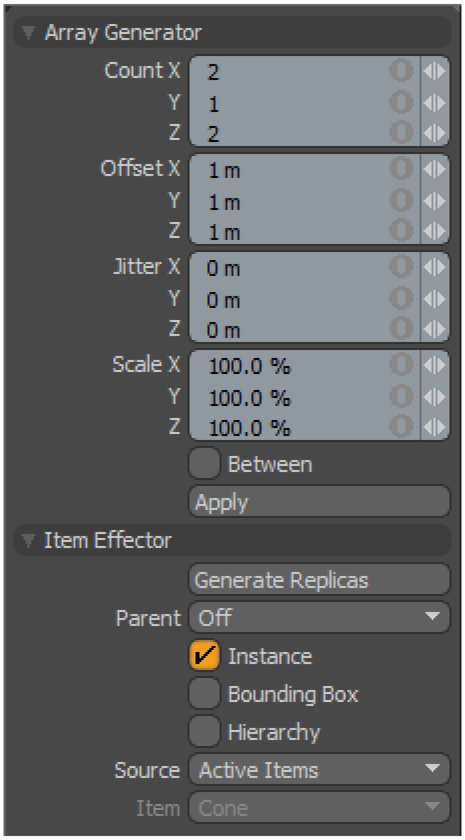
Array Generator	
	• **Preset Shape** - Clones a profile shape selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a profile before activating the tool.
Mesh Item	Specifies the mesh to be cloned. This is only relevant if you set **Source** to **Specific mesh**.

Tip: When the you click in the 3D viewport to activate the interactive mode for the tool, Modo defines an initial offset value that is automatically set to the bounding box size of the selected element. This makes it possible to clone an array of elements that are side by side with no space between them.

Instance Array

The Instance Array tool is similar to the standard Array tool, but produces instances instead of Mesh Items or geometry. Instances are useful in that they don't contain any physical geometry, but reference a source item to keep scene files small. Additionally, Modo propagates any edits made to the source geometry to all the instanced clones to make it easy to update the geometry if necessary. There is no visual difference between an instance or the actual geometry. To activate the Instance Array tool, right-click the **Array** icon in the **Model** layout, Toolbox **Duplicate** sub-tab and choose **Instance Array**.

Instance Array Properties



Count X 2
Y 1
Z 2
Offset X 1 m
Y 1 m
Z 1 m
Jitter X 0 m
Y 0 m
Z 0 m
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
☐ Between
Parent Off
☑ Instance
☐ Bounding Box
☐ Hierarchy
Source Active Items
Item Cone

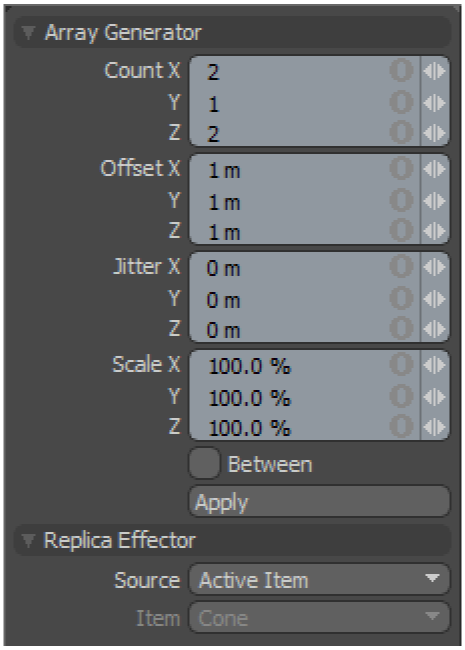
Array Generator	
Count X/Y/Z	Determines the number of clones in each axial direction. Each clone is an instance of the original mesh and contains no geometry, itself, but references the source mesh. If you make any edits or modifications to the source, Modo propagates those changes to each instance.
Offset X/Y/Z	Determines the distance away from the source element to produce the cloned elements, which Modo calculates as the distance between the source and the first order of clones. Subsequent clones are offset the same amount again. If you enable **Between**, this value determines the distance between the first and last clones with all others evenly spaced between those two.
Jitter X/Y/Z	Provides some random variation to be added to any of the three axes (X, Y, or Z). Modo positions each copied mesh with the **Offset** and a random variation using the **Jitter** distance as the maximum possible offset variation.

Array Generator	
Between	When enabled, uses the **Offset** value to generate the clones between the calculated first position and the last position.
Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**. • **Off** - Clones source item as additional instanced layers in the **Item List**. • **Hierarchy** - Creates a cascading array where each subsequent clone is a child of the previous clone. • **Root** - Creates instances as child items of the source mesh. • **Primary** - Creates Instances as children of a group locator that is parented to the source mesh.
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewports by pink wireframes. (This is enabled by default.)
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo clones into the array. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the mesh to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Replica Array

The Replica Array tool produces an array of cloned items (similar to the Array tool), but this tool generates the cloned items by using a Replicator. Much like instances, replicas don't add additional geometry to the scene. Modo creates a point source Mesh Item layer with vertices representing the item positions and a Replicator Item properly defined with a source and prototype. To activate the Replica Array tool, right-click the **Array** icon and choose **Replica Array**.

Replica Array Properties


Array Generator
Count X 2
Y 1
Z 2
Offset X 1 m
Y 1 m
Z 1 m
Jitter X 0 m
Y 0 m
Z 0 m
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
☐ Between

Replica Effector
Source Active Item
Item Cone

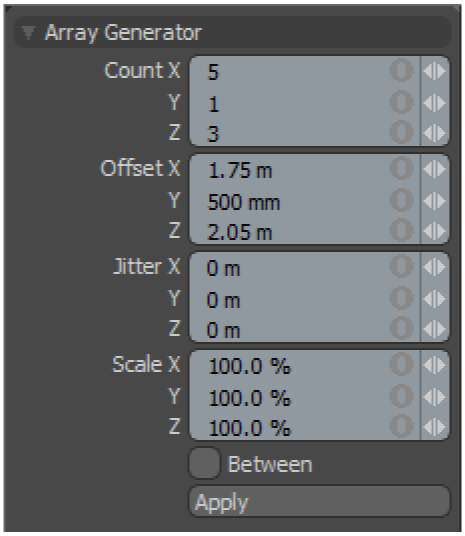
Array Generator	
Count X/Y/Z	Determines the number of clones in each axial direction. Each clone is an instance of the original mesh and contains no geometry, itself, but references the source mesh. If you make any edits or modifications to the source, Modo propagates those changes to each instance.
Offset X/Y/Z	Determines the distance away from the source element to produce the cloned elements, which Modo calculates as the distance between the source and the first order of clones. Subsequent clones are offset the same amount again. If you enable **Between**, this value determines the distance between the first and last clones with all others evenly spaced

Array Generator	
	between those two.
Jitter X/Y/Z	Provides some random variation to be added to any of the three axes (X, Y, or Z). Modo positions each copied mesh with the **Offset** and a random variation using the **Jitter** distance as the maximum possible offset variation.
Between	When enabled, uses the **Offset** value to generate the clones between the calculated first position and the last position.
Replica Effector	
Source	Determines the source mesh that Modo clones into the array. • **Active Item** - Use the currently selected geometry as the clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Preset Shape** - Clones a profile shape selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a profile before activating the tool.
Item	Specifies the mesh to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Transform Array

The Transform Array tool creates ordered arrays of existing items. To use the Transform Array tool, you need to select the target items prior to activating the tool. Regardless of the target items' current positions, Modo moves the items to the new array positions based on their center points. To activate the Transform Array tool, right-click the **Array** icon and choose **Transform Array**.

Transform Array Properties


Array Generator

Count X 5
Y 1
Z 3

Offset X 1.75 m
Y 500 mm
Z 2.05 m

Jitter X 0 m
Y 0 m
Z 0 m

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

☐ Between

Array Generator	
Count X/Y/Z	Determines the number of clones in each axial direction. Each clone is an instance of the original mesh and contains no geometry, itself, but references the source mesh. If you make any edits or modifications to the source, Modo propagates those changes to each instance.
Offset X/Y/Z	Determines the distance away from the source element to produce the cloned elements, which Modo calculates as the distance between the source and the first order of clones. Subsequent clones are offset the same amount again. If you enable **Between**, this value determines the distance between the first and last clones with all others evenly spaced between those two.
Jitter X/Y/Z	Provides some random variation to be added to any of the three axes (X, Y, or Z). Modo positions each copied mesh with the **Offset** and a random variation using the **Jitter** distance as the maximum possible offset variation.
Between	When enabled, uses the **Offset** value to generate the clones between the calculated first position and the last position.

Array in the Procedural

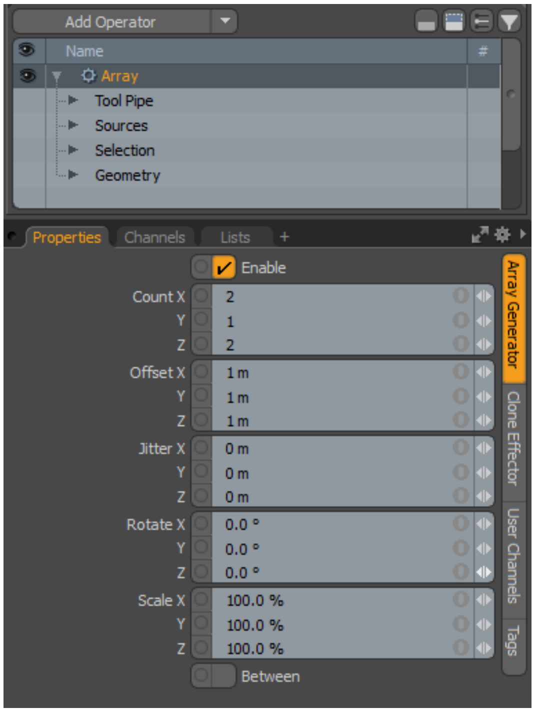
Array is also available as a procedural mesh operation. It is a combination of the Clone Effector and Array Generator sub-tools. You can find the tool in the **Mesh Ops** tab, on the right-side of the **Model** layout.

To add the Array mesh operation:

1. Click the **Add Operator** button, and navigate to **Mesh Operations** > **Duplicate** > **Array**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

   The Array mesh operation duplicates the geometry below it in the list. You can adjust the operation using its **Properties** panel in the lower-right.

You can expand the Array item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses:



☑ Enable
Count X 2
Y 1
Z 2
Offset X 1 m
Y 1 m
Z 1 m
Jitter X 0 m
Y 0 m
Z 0 m
Rotate X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
☐ Between

- **Tool Pipe** - Allows you to add sub-tools to the operation. For the Array tool, Array Generator is added by default and the options are displayed in the Properties tab.
- **Sources** - Allows you to specify the geometry you want to duplicate.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

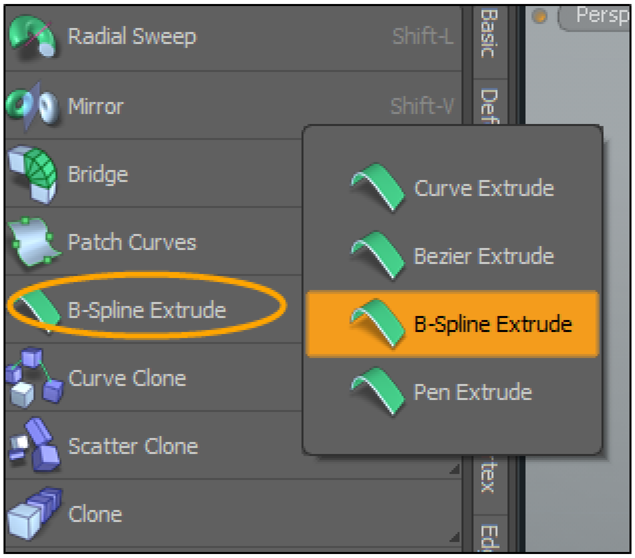
- **Selection** - Allows you to select the elements to which you want to apply the tool.

Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

The Array mesh operation has the same properties as the traditional Array tool. See Array Properties.

B-Spline Extrude

The **B-Spline Extrude** tool allows you to quickly extrude a geometry along the B-Spline curve.


Radial Sweep
Shift-L
Mirror
Shift-V
Bridge
Patch Curves
B-Spline Extrude
Curve Clone
Scatter Clone
Clone
Curve Extrude
Bezier Extrude
B-Spline Extrude
Pen Extrude
Basic
Def
tex
Ed
Persp

The Toolbox tools displays the last tool used. To active the **B-Spline Extrude** tool, open the **Duplicate** sub-tab in the Toolbox, right-click on the dropdown arrow on the current tool listed under the **Extrude** tools. From the menu, select **B-Spline Extrude** to view the tool options.

Using the Tool

The following examples demonstrate how this tool can be used.

Applying the Tool to a Mesh Item

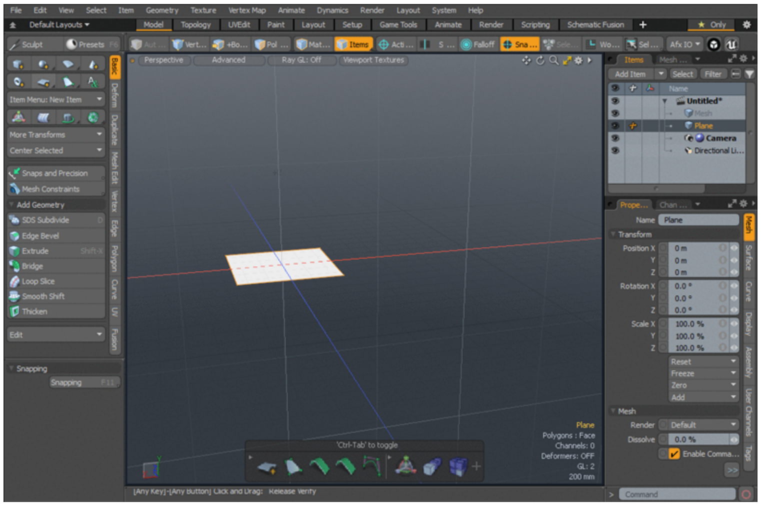
In this example, there is no pre-existing curve to start with. We add a **Plane** primitive and then drawn a curve using the **B-Spline Extrude** tool.

To apply to an existing mesh:

1. Open the **Model** layout.
2. In the left panel, open the **Basic** sub-tab, and **Shift** + click **Plane**.
3. Activate the **Polygons** selection mode or **Items** selection mode and select the **Plane**.
4. On the left panel, open the **Duplicate** sub-tab, and click **B-Spline Extrude**.
5. In **3D Viewport**, click a number of times to draw a curve.

> **Tip:** Alternatively, change **B-Spline Path** options while tool is active to position the points of the curve.

The polygon selected is extruded along the B-Spline curve. Click the image below to view an animation.



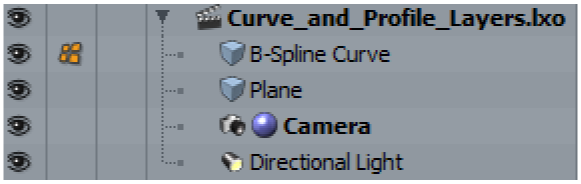
Applying the Tool Using Separate Mesh Ops Item Layers

In this example, we add a **Curve** and **Mesh** item to our scene. Each have their own layer in the **Mesh Ops** List. The **Mesh** item is used to create a profile along the curve created by the **B-Spline Extrude** tool.

1. Open the **Model** layout.
2. On the left panel, open **Curve** sub-tab and click **B-Spline**.
3. Click a number of time in the 3D Viewport to draw a B-Spline curve.

4. On the right panel, under **Items**, rename this layer **B-Spline Curve**.
5. On the right panel, under the **Items** tab, click **Add Item**.
6. On the left column, click **Items** and then double-click **Mesh**.

   A separate mesh layer is added to your scene.
7. On the right panel, under Items, rename this layer **Plane**.



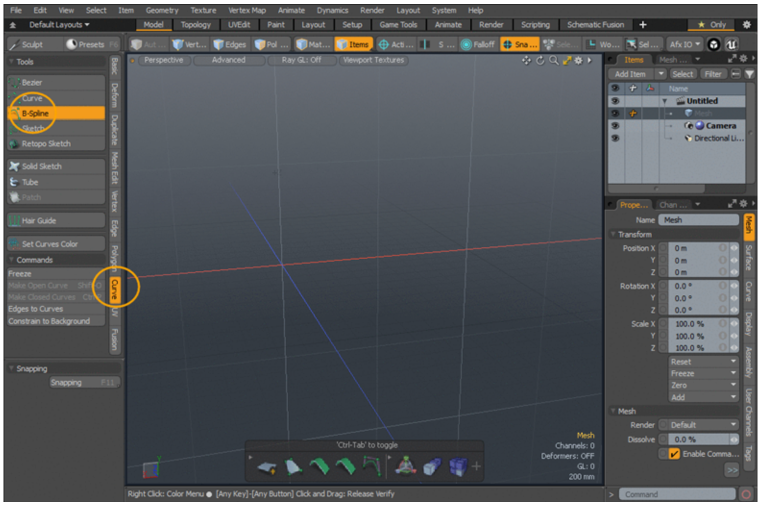
Curve_and_Profile_Layers.lxo
B-Spline Curve
Plane
Camera
Directional Light

8. On the left panel, open the **Basic** sub-tab, **Shift** + click **Plane**.

   Two separate layers display in the **Items** tab.
9. On the left open the **Duplicate** sub-tab and click **B-Spline Extrude**.
10. Click in the **3D Viewport**.

    The plane is extruded along the B-Spline curve.
11. Click and drag the last vertex on the B-Spline curve.

    The extruded profile updates. Click the image below to view an animation.


File Edit View Select Item Geometry Texture Vertex Map Animate Dynamics Render Layout System Help
Model Topology UVEdit Paint Layout Setup Game Tools Animate Render Scripting Schematic Fusion
Bezier
Curve
B-Spline
Sketch
Retopo Sketch
Solid Sketch
Tube
Hair Guide
Set Curves Color
Commands
Freeze
Edges to Curves
Constrain to Background
Snapping
Perspective
Advanced
Items
Untitled
Camera
Directional Li...
Mesh
Transform
Position
Rotation
Scale
Reset
Freeze
Zero
Add
Render
Dissolve
Enable Comma...

Applying the Tool Using One Mesh Ops Item Layer

In this example, we add a **Curve** and **Mesh** item to our scene and they are in same layer in the **Mesh Ops** List. The **Mesh** item is used to create a profile along the curve created by the **B-Spline Extrude** tool.

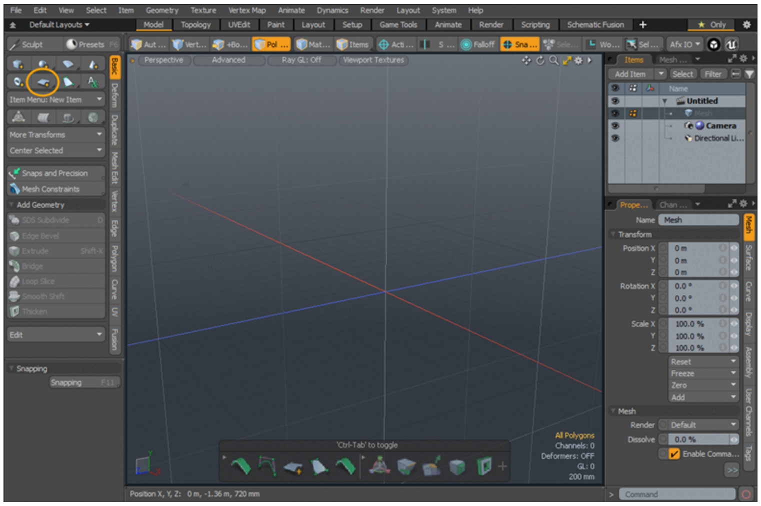
1. Open the **Model** layout.
2. On the left panel, open the **Basic** sub-tab, and **Shift**+click **Plane**.
3. Activate **Polygon** selection mode and select the **Plane**.

4. On the left panel, open the **Curve** sub -tab and click **B-Spline**.
5. Draw the Curve in the 3D Viewport.

   Notice on the right panel, under the **Items** tab, both the Plane and the Curve are in the same **Mesh Ops** layer.
6. Press the spacebar twice to drop the tool and selection.
7. Select the curve in the 3D Viewport.
8. On the left panel, open the **Duplicate** sub-tab, and click **B-Spline Extrude**.
9. Click in the **3D Viewport**.

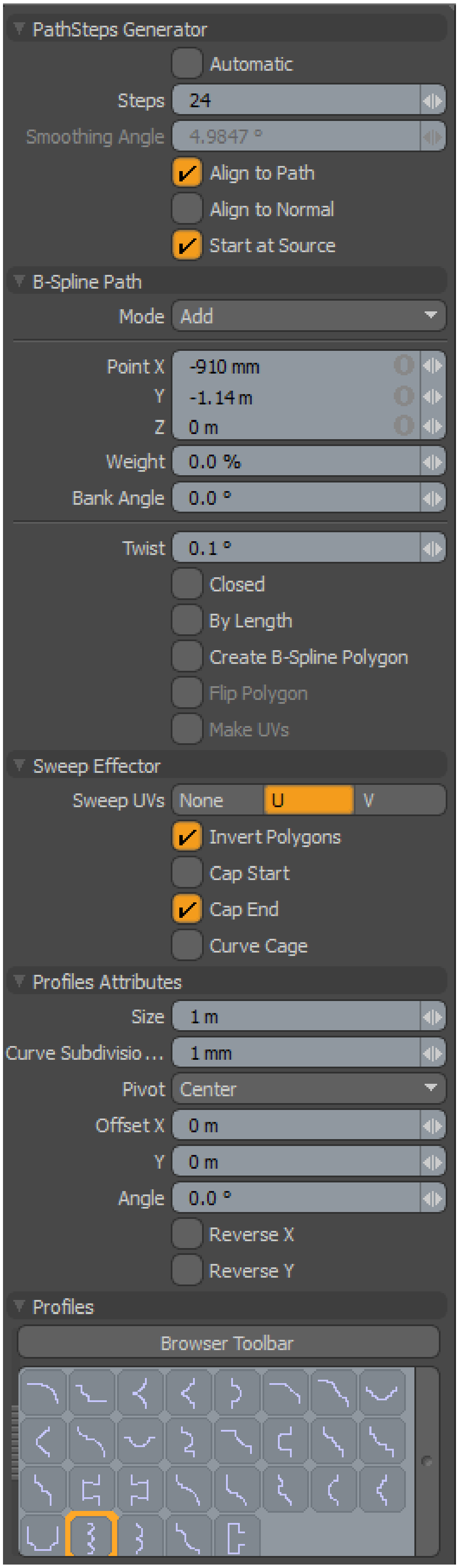
   The Polygon is extruded along the curve.
10. Click and drag the last vertex on the B-Spline curve.

    The extruded profile updates. Click the image below to view an animation.


File Edit View Select Item Geometry Texture Vertex Map Animate Dynamics Render Layout System Help
Default Layouts
Model Topology UVEdit Paint Layout Setup Game Tools Animate Render Scripting Schematic Fusion
Perspective Advanced Ray GL: Off Viewport Textures
Item Menu: New Item
More Transforms
Center Selected
Snaps and Precision
Mesh Constraints
Add Geometry
Edit
Snapping
Items
Add Item Select Filter
Untitled
Camera
Directional Li...
Name Mesh
Transform
Position X Y Z
Rotation X Y Z
Scale X Y Z
Reset Freeze Zero Add
Mesh
Render Default
Dissolve 0.0 %
Enable Comma...
'Ctrl-Tab' to toggle
All Polygons
Channels: 0
Deformers: OFF
GL: 0
200 mm
Position X, Y, Z: 0 m, -1.36 m, 720 mm
Command

B-Spline Options

The following options are available:



PathSteps Generator

☐ Automatic

Steps 24

Smoothing Angle 4.9847 °

☑ Align to Path

☐ Align to Normal

☑ Start at Source

B-Spline Path

Mode Add

Point X -910 mm

Y -1.14 m

Z 0 m

Weight 0.0 %

Bank Angle 0.0 °

Twist 0.1 °

☐ Closed

☐ By Length

☐ Create B-Spline Polygon

☐ Flip Polygon

☐ Make UVs

Sweep Effector

Sweep UVs None [✓ U] V

☑ Invert Polygons

☐ Cap Start

☑ Cap End

☐ Curve Cage

Profiles Attributes

Size 1 m

Curve Subdivisio… 1 mm

Pivot Center

Offset X 0 m

Y 0 m

Angle 0.0 °

☐ Reverse X

☐ Reverse Y

Profiles

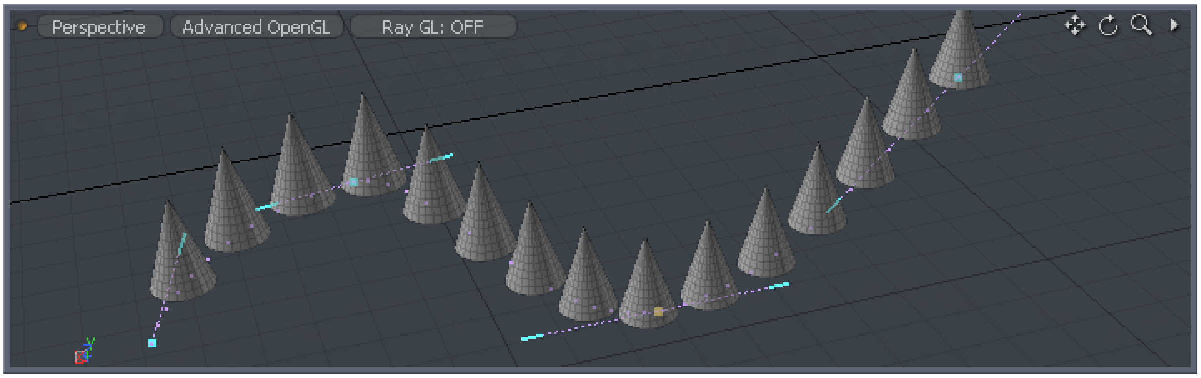
PathSteps Generator	
Automatic	Forces a computed number of clones, based on the **Smoothing Angle** value, such that a clone is placed along the path to maintain a level of smoothness at or above the user-set level (determined by the **Smoothing Angle**).
Steps	Active only when the **Automatic** option is disabled. The defined **Steps** value determines the number of particles generated along the curve path that feed into the Effector, essentially controlling the number of segments of the resulting extrusion.
Smoothing Angle	Determines the angle along the curve where a clone is created when the **Automatic** option is enabled.
Align to Path	When enabled, the generated particles include rotation information so the newly-created clone aligns to the path vector direction at the location where it is created.
Align to Normal	When enabled, the generated particles include rotation information so the newly-created clone aligns with any stored normal direction for a vertex, such as for a curve extracted from geometry.
Start at Source	Often a curve used for cloning has its initial control knot imperfectly aligned with the geometry to be cloned. With **Start at Source** active, the curve is used as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly-created geometry to move away from the intended original location.
Curve Path	
Mode	By default, this is set to **Add**. In **Add** mode, clicking on a curve point and dragging edits the curve's position. Clicking anywhere in the 3D viewport other than the handles creates a new curve control point. To avoid creating new control points when dragging existing points around, you can set this mode to **Edit**. The **Delete** mode allows quick removal of control points by simply clicking them.
Point X/Y/Z	The XYZ position of the currently-selected knot. You can interactively edit the position by clicking over the knot indicated by the cyan dot. It turns yellow and you can now drag its position around in the viewport, or manually enter a position by editing any of the data fields.

PathSteps Generator	
Bank Angle	For each path position, a bank angle can be defined that controls the rotation angle of elements cloned through that position.
Twist	Works just like the **Bank Angle** option, but instead of being calculated per knot, the rotation amount is defined across the entire length of the curve, where the **Twist** value represents the amount of twisting rotation from the first knot in the curve to the last knot.
Closed	Causes the curve to become a closed loop, connecting the end point to the start.
Start Control	Enabling this toggle causes the first knot in the curve to become a control point, but not an actual segment of the curve. This provides a simple method for changing the curvature at the start of the curve.
End Control	Enabling this toggle causes the last knot in the curve to become a control point, but not an actual segment of the curve. This provides a simple method for changing the curvature at the end of the curve.
By Length	Forces the newly-created geometry to be evenly distributed across the entire length of the curve rather than spacing based on the control knots.
Create B-SplinePolygon	Generates a b-spline curve polygon with the path.
Flip Polygon	Inverts the faces of the generated polygons.
Make UVs	When this toggle is active, the tool populates the current UV map with data for the generated polygons.
Sweep Effector	
Sweep UVs	Allows you to choose between the U or V axis when placing UV data for newly-created geometry.
Invert Polygons	Reverses the normal facing direction of the polygons generated by the extrusion. This is useful when extruded geometry appears inverted or inside-out.
Cap Start/End	Forces the back end of the extruded geometry to be capped or closed.

PathSteps Generator	
Profile Attributes	
Size	The size of 1D or 2D profile to sweep. If the size is 0.0, it uses the real size.
Curve Subdivision Tolerance	This is the minimum angle when it subdivides the Bezier curve to a string of two-point polygons for extrusion. This setting basically controls the detail tolerance of the conversion from the original curve to the resulting polygon extrusion.
Pivot	Specifies the pivot position of the profile.
Offset X/Y	The offset amount on the profile space.
Reverse X/Y	Reverses the selected profile to horizontal or vertical.
Profiles	
Profiles	A mini **Preset Browser** for viewing various profiles. Works the same way as the standard Preset Browser. Selecting any of the various profiles modulates the extrusion width with the selected shape over its length. To return to the original polygon selection for the extrusion, **Ctrl**+click the selected profile to deselect it.

Bezier Clone

The **Bezier Clone** tool duplicates an object along a Bezier path a user-defined number of times. The path can be pre-generated, or with the tool active, can be drawn and edited interactively while cloning the target element(s). A Bezier Curve is a type of curve defined by additional off-curve control handles, providing you with an intuitive means to control the amount of incoming and outgoing curvature for each control point.



The **Bezier Clone** tool is found in the **Duplicate** sub-tab of the default modeling toolbox. You may have to right-click to open the button pop-up menu to find it (by default, it is located under the **Curve Clone** tool button). Alternately, the tool is also found in the menu bar under **Geometry** > **Duplicate** > **Bezier Clone**. Click the button to activate the tool. Next, in the 3D viewport, click to position the first control point and then continue to drag the mouse cursor to create the Bezier handle. Releasing the mouse button sets the handle length. Subsequent clicking and dragging draws additional control points and handles with a curve segment drawn between each vertex. Cloned items (as defined in the tool's properties panel) interactively display on each new segment as it is drawn.

When drawing out the Bezier for cloning, control points are always created at the intersection of the **Work Plane** and the mouse click. The curvature of the segments between control points is defined by the Bezier handles (called **endpoint tangent vectors**). While the tool is active, you can hover over any of the control points (changing them from blue to yellow) then click and drag to reposition it. Points always move on the two axes relative to the current **Work Plane**, so rotating the viewport to change an axis may be necessary. Handles are edited in the same manner. Pressing and holding the **Ctrl** key when editing handles break tangency, allowing opposing handles to be different lengths and/or angles. Editing any handle without pressing **Ctrl** restores tangency. When editing (moving) control points, pressing and holding **Shift** moves all downstream control points as a single unit.

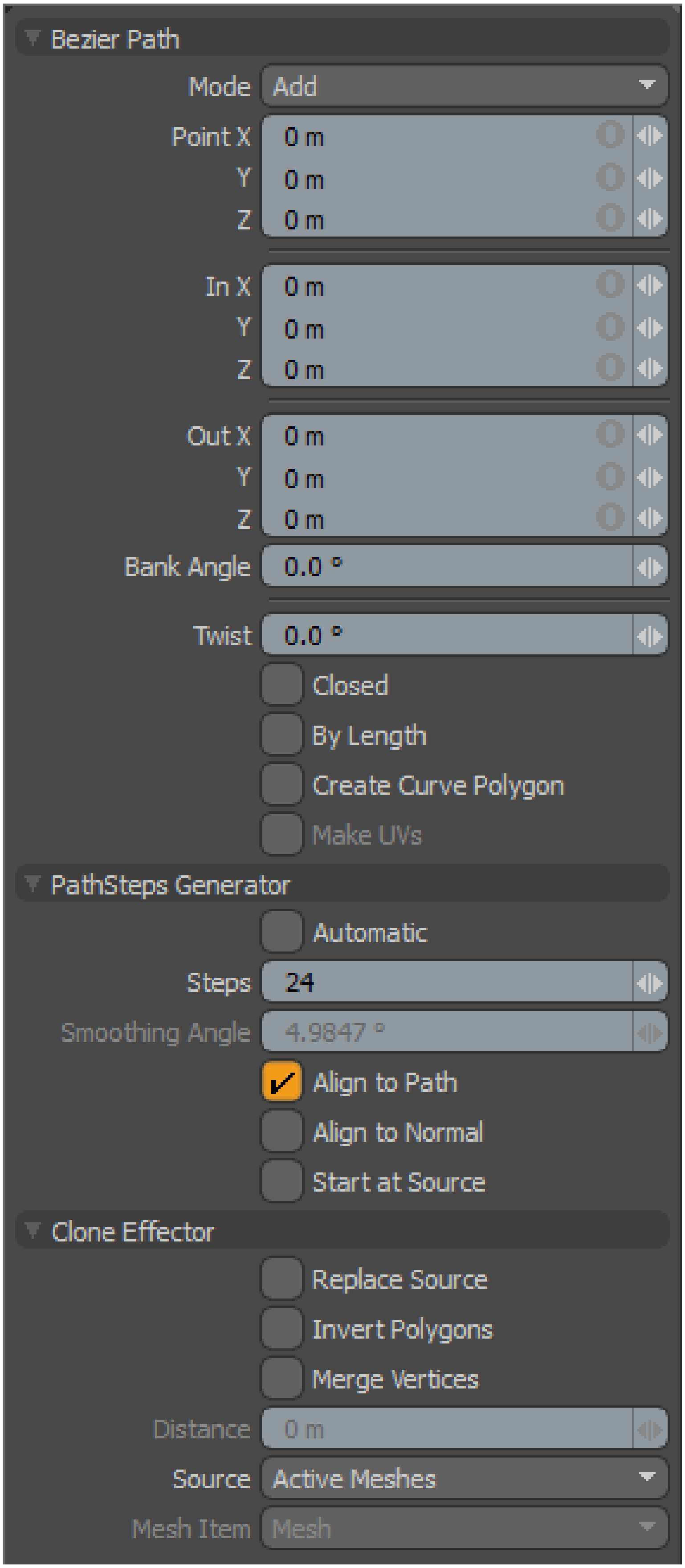
There are three Bezier editing modes: **Add** any additional control point after the most recently-selected control point, **Edit** to change a control point's location, and **Delete** removes any control point that is clicked. Control points along the Bezier Curve can also be positioned precisely during editing by selecting the target vertex and using the **Point X/Y/Z** fields of the tool's properties panel. Once the tool is dropped, by pressing the **spacebar**, the cloned items are, in effect, set and no further editing of the path is possible. However, using a pre-drawn path allows you to delete the resulting clones and re-create them with modified settings.

Pre-drawn Path

To use a pre-drawn Bezier path for cloning, you need to make sure the path itself is selected prior to activating the tool. This is done while in the **Polygons selection mode**. With the target curve selected, activate the tool (by clicking the tools button) and then again clicking in the 3D viewport. This triggers the cloning of the selected element along the Bezier. If desired, the Bezier can then be edited in the same manner as if it were drawn with the tool.

Clones

The number of clones is determined by the **PathSteps Generator**. Adjusting the **Steps** value increases or decreases the number of clones along the path. The position of clones can be further adjusted with the **By Length** option, which evenly distributes the clones along the path, regardless of the control points. By default, what is cloned is determined by what is selected. Generally, you would define a selection within a single layer, activate the clone tool, and then draw the curve that results in the cloned elements. If what is to be cloned is on a different layer, the you can adjust the settings of the **Clone Effector** to specify alternate clone sources. This allows you to specify a specific layer, or use multiple layers randomly for each clone location, based on their foreground/background visibility.



Bezier Path

Mode: Add

Point X: 0 m
Y: 0 m
Z: 0 m

In X: 0 m
Y: 0 m
Z: 0 m

Out X: 0 m
Y: 0 m
Z: 0 m

Bank Angle: 0.0 °

Twist: 0.0 °

☐ Closed
☐ By Length
☐ Create Curve Polygon
☐ Make UVs

PathSteps Generator

☐ Automatic

Steps: 24

Smoothing Angle: 4.9847 °

☑ Align to Path
☐ Align to Normal
☐ Start at Source

Clone Effector

☐ Replace Source
☐ Invert Polygons
☐ Merge Vertices

Distance: 0 m

Source: Active Meshes

Mesh Item: Mesh

The following properties are available for the Bezier Clone tool.

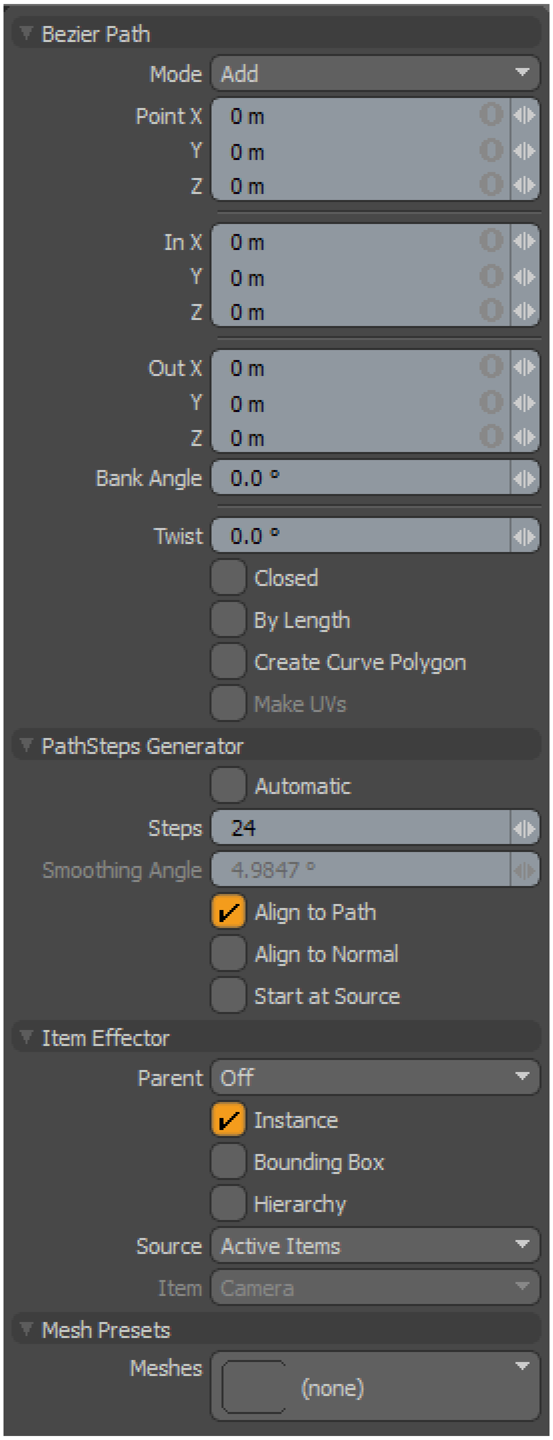
Bezier Path	
Mode	The various Bezier curve mode options available are: • **Add** - the default mode; when you click in the viewport, additional points are added to the curve. At each control point position, two handles extend out that allow you to adjust the curvature of the curve segment between vertices. While drawing you can hover over any point or handle, where it turns yellow, you can then click and drag the control point or handle to further edit the curve. Control points may be added mid-curve by selecting the preceding control, highlighting it yellow, and clicking the position where the new control point is desired (point order is defined by the initial order in which the curve was created, press the **F** key prior to activating the tool to invert the order). • **Edit** - in **Edit** mode, you can freely click and drag any of the control points or handles along the curve to change the look to the desired shape. While pressing the **Ctrl** key, you may break the tangency of the control handles, allowing for sharp corners. • **Delete** - in **Delete** mode, you can click on any control point along the curve to remove it from the Bezier.
Point X/Y/Z	If you want finer point control, you can assign specific XYZ values in these input fields for the currently-selected (highlighted) control point.
Bank Angle	The orientation of the cloned elements is determined by the orientation of the geometry being cloned, and the **Align** options, if enabled. The **Bank Angle** option provides you an additional control, per path vertex, for banking or angling the clones from side to side. Bank value then smoothly blends with the values of the previous and later control point positions.
Twist	Works just like the **Bank Angle** option, but instead of being calculated per knot, the rotation amount is defined across the entire length of the curve, where the **Twist** value represents the amount of twisting rotation from the first knot in the curve to the last knot.
In X/Y/Z/Out X/Y/Z	If you want finer handle control can assign specific XYZ values in these **In** and **Out** input fields for the currently-selected (highlighted) control point.

Bezier Path	
Closed	Adds an automatic curve segment between the first and last control point positions producing a closed curve.
By Length	When the **By Length** option is enabled, cloned items are evenly distributed along the path by equal lengths rather than by the span (space between control points), which may produce uneven placement.
Create Curve Polygon	When the **Create Curve Polygon** option is enabled, dropping the tool leaves the Bezier Curve path instead of removing it from the scene. This is useful should you wish to use the resulting curve for additional purposes.
Make UVs	If the **Create Curve Polygon** option is enabled, this toggle activates auto-generation of UV texture coordinates along the curve. The generated UV values be of a single vertical line ('V' axis in UV) positioning all the control point vertices evenly between 0 and 1. For example, this can be useful for applying a Gradient to a rendered curve (the **Render Curves** option is available in the Mesh Item's properties.
Path Steps Generator	
Automatic	When enabled, Modo assign the number of clones to the path based on the number of control points combined with the **Smoothing Angle** option. This method bunches-up the most clones around the curvy areas, and produces fewer clones on the straight areas of the curve, which may be desirable. For even spacing of clones, enable the **By Length** option, above, and disable **Automatic**.
Steps	Available only when **Automatic** is disabled. This option allows you to specify an exact number of clones along the path.
Smoothing Angle	When enabled, the **Smoothing Angle** determines the distribution of the clones along the curve. Smaller values produce more clones, and larger values reduce the number of overall clones. The **Smoothing Angle** basically defines what the difference between two positions along the curve is, if the next position falls within the **Smoothing Angle** threshold, a new clone is generated.
Align to Path	When enabled, changes the orientation of the cloned elements along the curve. The first clone retain its original orientation and subsequent

Bezier Path	
	clones be angled away from the first based on the changing (vector) angle of the Bezier curve.
Align to Normal	When enabled, Modo tries to align the direction of the source objects to a straight up position before cloning the source item(s). If the item is already upright when cloned, this option likely has no effect.
Start at Source	By default, Modo clones items at the actual World Position along the Bezier curve. When the **Start at Source** option is enabled, Modo offsets each cloned item the distance between the source items center point and the first curve point, producing an effect as if the curve were initiated precisely at the center of the first source item.
Clone Effector	
Replace Source	When this option is enabled, Modo removes the source geometry and positions it at the first point along the path.
Invert Polygons	When enabled, inverts the normal direction of the cloned geometry.
Merge Vertices	When enabled, merges coinciding vertices into a single vertex.
Distance	Used in conjunction with the **Merge Vertices** option, the **Distance** value determines the maximum distance between vertices where merging occur. Vertices outside this value remain unchanged.
Source	Determines the mesh that be cloned along the Pen tool path. • **Active Meshes** - uses the currently-selected geometry as a clone source. • **Specific Mesh** - allows you to specify a specific Mesh Item as the clone source using the **Mesh Item** option below. • **All BG** - combines all the currently-visible background items cloning them as a single mesh. • **Random BG** - chooses a random background item from all the currently-visible background items at each clone position. • **Preset Shape** - allows you to clone **Profile** shapes from the **Preset Browser**. Open the **Preset Browser** (press **F6**) and select a profile before activating the tool.
Mesh Item	Works when the **Source** is set to **Specific Mesh** and allows you to choose the mesh layer to be cloned.

Bezier Instance

The **Bezier Instance** tool duplicates an object along a Bezier path a user-defined number of times, creating instances of the source item, rather than actual geometry. You can activate the tool by right-clicking the **Bezier Clone** tool and selecting **Bezier Instance** from the list.



Bezier Path
Mode: Add
Point X: 0 m
Y: 0 m
Z: 0 m
In X: 0 m
Y: 0 m
Z: 0 m
Out X: 0 m
Y: 0 m
Z: 0 m
Bank Angle: 0.0 °
Twist: 0.0 °
☐ Closed
☐ By Length
☐ Create Curve Polygon
☐ Make UVs

PathSteps Generator
☐ Automatic
Steps: 24
Smoothing Angle: 4.9847 °
☑ Align to Path
☐ Align to Normal
☐ Start at Source

Item Effector
Parent: Off
☑ Instance
☐ Bounding Box
☐ Hierarchy
Source: Active Items
Item: Camera

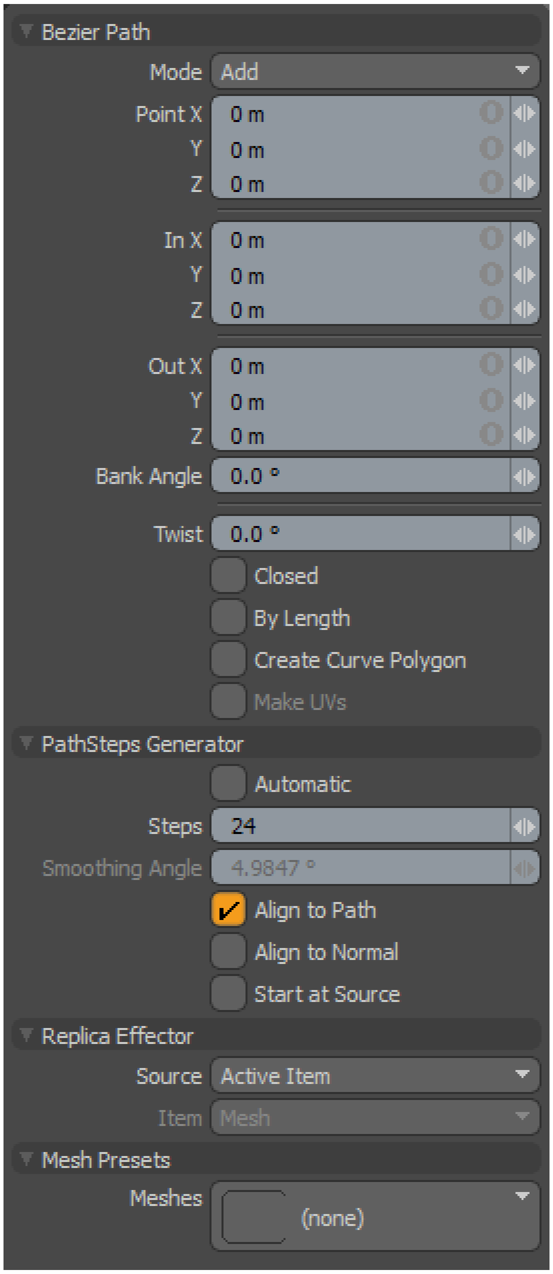
Mesh Presets
Meshes: (none)

In addition to the **Bezier Path** and **PathSteps Generator** properties described above, the following parameters are available for the Bezier Instance tool.

Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**.
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewports by pink wireframes. (This is enabled by default.)
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo uses.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Bezier Replica

The **Bezier Replica** tool duplicates an object along a Bezier path a user-defined number of times, creating replicas of the source item. You can activate the tool by right-clicking the **Bezier Clone** tool and selecting **Bezier Replica** from the list.



Bezier Path
Mode: Add
Point X: 0 m
Y: 0 m
Z: 0 m
In X: 0 m
Y: 0 m
Z: 0 m
Out X: 0 m
Y: 0 m
Z: 0 m
Bank Angle: 0.0 °
Twist: 0.0 °
☐ Closed
☐ By Length
☐ Create Curve Polygon
☐ Make UVs

PathSteps Generator
☐ Automatic
Steps: 24
Smoothing Angle: 4.9847 °
☑ Align to Path
☐ Align to Normal
☐ Start at Source

Replica Effector
Source: Active Item
Item: Mesh

Mesh Presets
Meshes: (none)

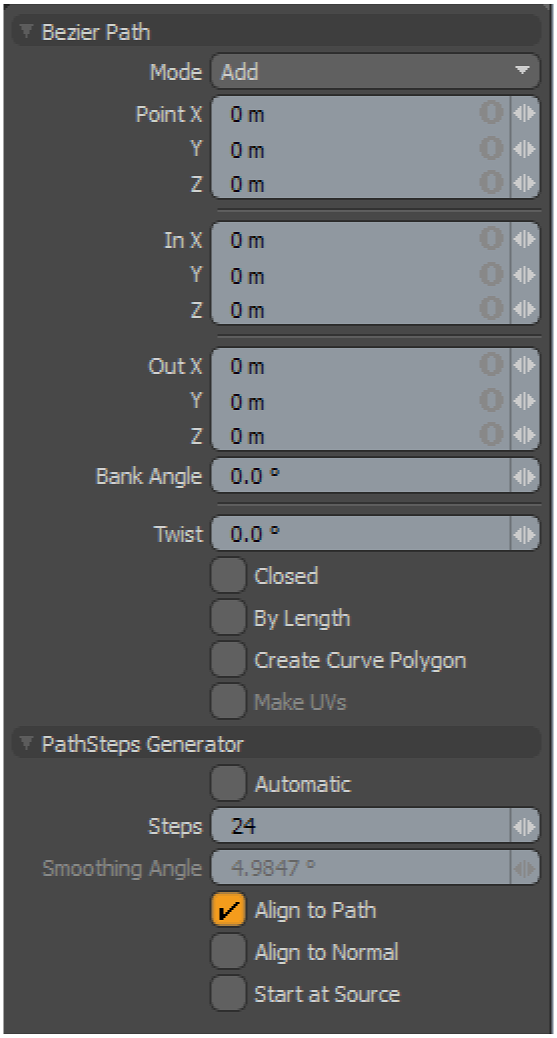
In addition to the **Bezier Path** and **PathSteps Generator** properties described in the Bezier Clone topic, the following properties are available for the Bezier Replica tool.

Replica Effector	
Source	Determines the source mesh that Modo uses. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on

Replica Effector	
	the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Bezier Transform

The Bezier Transform tool helps you arrange your existing items along a Bezier path. To activate the Bezier Transform tool, right-click the **Bezier Clone** icon and select **Bezier Transform** in the list.



Bezier Path

Mode: Add

Point X: 0 m
Y: 0 m
Z: 0 m

In X: 0 m
Y: 0 m
Z: 0 m

Out X: 0 m
Y: 0 m
Z: 0 m

Bank Angle: 0.0 °

Twist: 0.0 °

☐ Closed
☐ By Length
☐ Create Curve Polygon
☐ Make UVs

PathSteps Generator

☐ Automatic

Steps: 24

Smoothing Angle: 4.9847 °

☑ Align to Path
☐ Align to Normal
☐ Start at Source

For the Bezier Transform tool, you can set the **Bezier Path** and **PathSteps Generator** properties. For more information on these parameters, see Bezier Clone.

Bezier Extrude

The **Bezier Extrude** tool allows you to quickly extrude a geometry selection along the length of a Bezier curve. The path can be pre-generated, or with the tool active, can be drawn and edited interactively while cloning the target element(s). A Bezier curve is a type of curve defined by additional off-curve control handles, providing you with an intuitive means to control the amount of incoming and outgoing curvature for each control point.

The **Bezier Extrude** tool is found in the **Duplicate** sub-tab of the default modeling toolbox. Right-click to open the button pop-up menu to find it (by default, it is located under the **Curve Extrude** tool button).

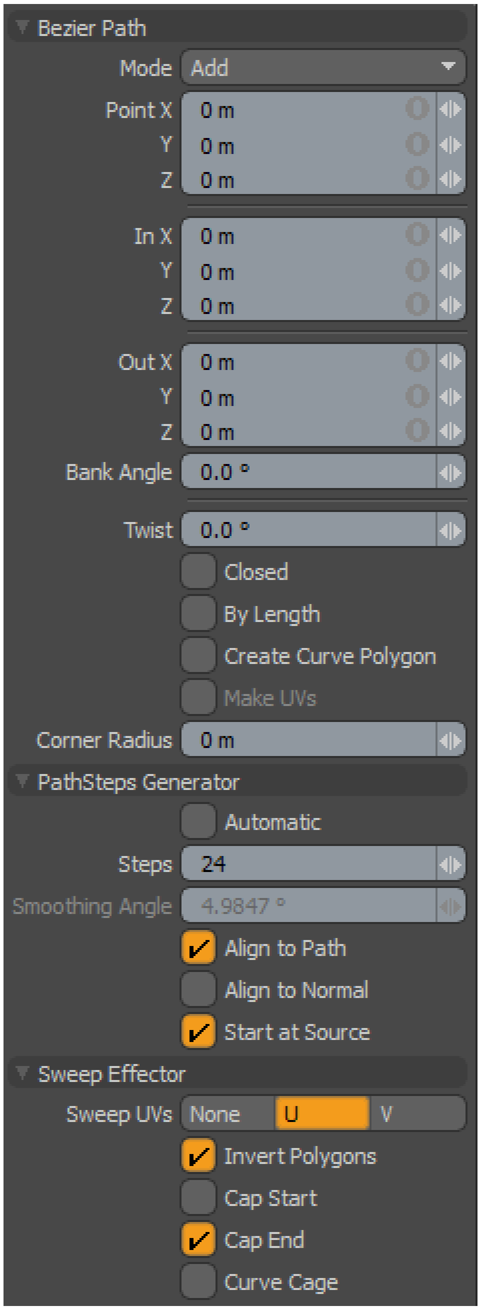
Alternately, the tool is also found in the menu bar under **Geometry** > **Duplicate** > **Bezier Extrude**. Click the button to activate the tool. Next, in the 3D viewport, click to position the first control point and then continue to drag the mouse cursor to create the Bezier handle. Releasing the mouse button sets the handle length. Subsequent clicking and dragging draws additional control points and handles with a curve segment drawn between each vertex. The extruded geometry interactively displays across each new segment as it is drawn.

When drawing out the Bezier for extruding, control points are always created at the intersection of the **Work Plane** and the mouse click. The curvature of the segments between control points is defined by the Bezier handles (called **endpoint tangent vectors**). While the tool is active, you can hover over any of the control points (changing them from blue to yellow), then click and drag to reposition it. Points always move on the two axes relative to the current **Work Plane**, so rotating the viewport to change an axis may be necessary. Handles are edited in the same manner. Pressing and holding the **Ctrl** key when editing handles breaks tangency, allowing opposing handles to be different lengths and/or angles. Editing any handle without pressing **Ctrl** restores tangency. When editing (moving) control points, pressing and holding **Shift** moves all downstream control points as a single unit.

There are three Bezier editing modes: **Add** any additional control point after the most recently-selected control point, **Edit** to change a control points location, and **Delete** removes any control point that is clicked. Control points along the Bezier Curve can also be positioned precisely during editing by selecting the target vertex and using the Point X/Y/Z fields of the tools properties panel. Once the tool is dropped, by pressing the **spacebar** key, the cloned items are, in effect, set and no further editing of the path is possible. However, using a pre-drawn path allows you to delete the resulting extrusion and re-create it with modified settings.

Pre-drawn Path

To use a pre-drawn Bezier path for extrusion, you need to make sure the path itself is selected prior to activating the tool. This is done while in the **Polygons selection mode**. With the target curve selected, activate the tool (by clicking the tool's button) and then again clicking in the 3D viewport. This triggers the cloning of the selected element along the Bezier. If desired, the Bezier can then be edited in the same manner, as if it were drawn with the tool.



Bezier Path

Mode Add

Point X 0 m
Y 0 m
Z 0 m

In X 0 m
Y 0 m
Z 0 m

Out X 0 m
Y 0 m
Z 0 m

Bank Angle 0.0 °

Twist 0.0 °

☐ Closed
☐ By Length
☐ Create Curve Polygon
☐ Make UVs

Corner Radius 0 m

PathSteps Generator

☐ Automatic

Steps 24

Smoothing Angle 4.9847 °

☑ Align to Path
☐ Align to Normal
☑ Start at Source

Sweep Effector

Sweep UVs None U V

☑ Invert Polygons
☐ Cap Start
☑ Cap End
☐ Curve Cage

Bezier Path	
Mode	The various Bezier curve mode options available are: • **Add** - the default mode; when you click in the viewport, additional points are added to the curve. At each control point, position two handles that extend out and allow you to adjust the curvature of the curve segment between vertices. While drawing, you can hover over any point or handle. Where it turns yellow, you

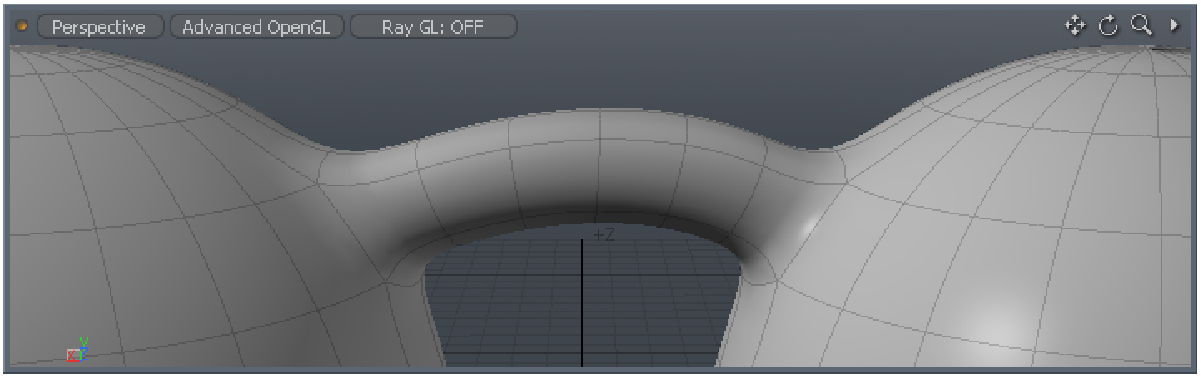
Bezier Path	
	can then click and drag the control point or handle to further edit the curve. Control points may be added mid-curve by selecting the preceding control, highlighting it yellow, and clicking the position where the new control point is desired (point order is defined by the initial order in which the curve was created, press the **F** key prior to activating the tool to invert the order). • **Edit** - in **Edit** mode you can freely click and drag any of the control points or handles along the curve to change the look to the desired shape. While pressing the **Ctrl** key, you may break the tangency of the control handles, allowing for sharp corners. • **Delete** - in **Delete** mode, you can click on any control point along the curve to remove it from the bezier.
Point X/Y/Z	If you want finer point control, you can assign specific XYZ values in these input fields for the currently-selected (highlighted) control point.
In X/Y/Z/Out X/Y/Z	If you want finer handle control, you can assign specific XYZ values in these **In** and **Out** input fields for the currently-selected (highlighted) control point.
Bank Angle	The orientation of the extruded geometry is determined by the orientation of the initial polygon and the **Align** options, when enabled. The **Bank Angle** option provides you an additional control, per path vertex, for banking or angling the extrusion from side to side. The bank value then smoothly blends with the values of the previous and later control point positions.
Twist	The **Twist** option works just like the **Bank Angle** option, but instead of being calculated per knot, the rotation amount is defined across the entire length of the curve, where the **Twist** value represents the amount of twisting rotation from the first knot in the curve to the last knot.
Closed	The **Closed** toggle option adds an automatic curve segment between the first and last control point positions, producing a closed extrusion.
By Length	When the **By Length** option is enabled, segments of the extrusion are evenly distributed along the path by equal lengths rather than by the span (space between control points), which may produce uneven segments.
Create Curve Polygon	When the **Create Curve Polygon** option is enabled, dropping the tool leaves the Bezier Curve path instead of removing it from the scene. This is useful should you wish to use the resulting curve for additional purposes.

Bezier Path	
Make UVs	If the **Create Curve Polygon** option is enabled, this toggle activates auto-generation of UV texture coordinates along the curve. The generated UV values are of a single vertical line ('V' axis in UV), positioning all the control point vertices evenly between 0 and 1. For example, this can be useful for applying a Gradient to a rendered curve (the **Render Curves** option is available in the Mesh Item properties.
Corner Radius	Sets the minimum step on either side of a sharp corner.
PathSteps Generator	
Automatic	When the **Automatic** option is enabled, Modo assign the number of segments to the path based on the number of control points combined with the **Smoothing Angle** option. This method bunches-up the most segments around the curvy areas, and produce fewer segments on the straight areas of the curve, which may be desirable. For even spacing of segments, enable the **By Length** option, above, and disable **Automatic**.
Steps	Available only when **Automatic** is disabled, this option allows you to specify an exact number of segments along the path.
Smoothing Angle	When the **Automatic** toggle is enabled, the **Smoothing Angle** determines the distribution of the segments along the curve. Smaller values produce more segments, and larger values reduce the number of segments. The **Smoothing Angle** basically defines what the difference between two positions along the curve is, if the next positions falls within the **Smoothing Angle** threshold, a new segment is generated.
Align to Path	This option, when enabled, changes the orientation of the segments along the curve. The first segment retain its original orientation and subsequent segments be angled away from the first based on the changing (vector) angle of the Bezier Curve.
Align to Normal	When this option is enabled, Modo tries to align the direction of the source objects to a straight up position before cloning the source item(s). If the item is already upright when cloned, this option likely has no effect.
Start at Source	By default, Modo clones items at the actual World Position along the Bezier Curve. When the **Start at Source** option is enabled, Modo offsets each cloned item the distance between the source items center point and the first curve point, producing an effect as if the curve were initiated precisely at the center of the first source item.

Bezier Path	
Sweep Effector	
Sweep UVs	This pop-up control allows you to choose between the U or V axis when placing auto-generated UV values for the newly created geometry.
Invert Polygons	Reverses the normal facing direction of the polygons generated by the extrusion. This is useful for when the extrusion travels along a negative axis causing inside-out geometry.
Cap Start/End	This toggle forces the back end of the extruded geometry to be capped or closed.

Bridge

The Bridge tool creates polygons to span the area between selected edges or polygons. The Bridge tool has several options to control how Modo generates the spanning polygons to connect the two locations.



The **Bridge** tool is found in the **Model** layout **Toolbox**, under the **Duplicate**, **Edge**, and **Polygon** sub-tabs. It is also found in the **Topology** layout **Tools** sub-tab. In addition, you can also activate it by choosing **Geometry** > **Duplicate** > **Bridge**.

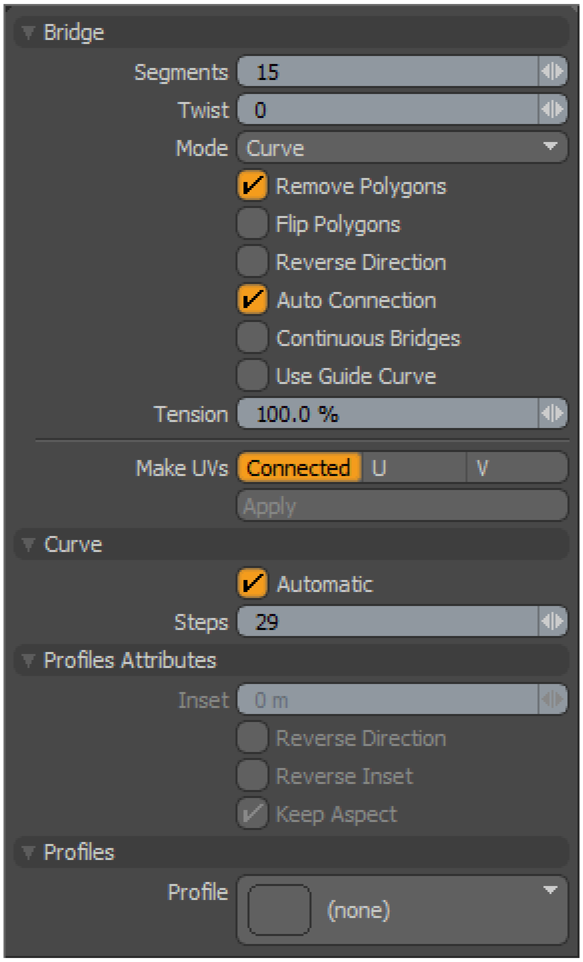
Prior to activating the tool, you must select at least two polygons that do not share any edges. The first polygon (or cluster of polygons) selected connects with the second polygon (or cluster of polygons). Make sure each cluster (or individual polygon) does not share any edges with the first selection for Modo to determine how to bridge from one location to the next.

Note: You can use Edge selections as well — as long as they don't share any vertices between the two or more selections.

Once you specify the selection, clicking the **Bridge tool** icon activates the tool. Clicking again in the 3D viewport enables the interactive mode for you to drag and interactively adjust the number of segments between the polygon clusters.

Bridge Tool Properties

You can adjust additional options in the tool's **Properties** panel.



Bridge
Segments 15
Twist 0
Mode Curve
☑ Remove Polygons
☐ Flip Polygons
☐ Reverse Direction
☑ Auto Connection
☐ Continuous Bridges
☐ Use Guide Curve
Tension 100.0 %
Make UVs Connected | U | V
Apply
Curve
☑ Automatic
Steps 29
Profiles Attributes
Inset 0 m
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
Profiles
Profile (none)

Bridge Properties

Segments: Determines the number of polygons that Modo creates across the length of the bridge. You can set this value interactively by dragging in the 3D viewport while the tool is active or numerically in the

Properties panel. For curved bridge spans or those modulated by a profile, having more segments produces smoother results.

Twist: Rotates the bridge polygons between the original connection points when set to a non-zero number. This is useful to clean up bridges between disparate polygon selections where the bridge can become overlapped. By specifying the **Twist** value you may be able to untangle the bridge manually.

Mode: Sets the interpolation style of the generated bridge polygons.
Linear- Creates a bridge in a straight line between the original selection.
Curve- Creates an arc between the original selection points. You can further refine the curve by adjusting the **Tension** setting.
Smooth- Creates an "ease in/ease out" blend between the original selection sections.

Remove Polygons: When enabled, removes the originally selected polygons used to create the bridge if you selected polygons prior to activating the tool. This is enabled by default and is the preferred behavior.

Flip Polygons: When enabled, flips the normals of the newly created polygons in the bridge. Use when a negative bridge action causes the resulting polygons to appear inside-out.

Reverse Direction: When enabled, reverses the normal direction of the target polygon cluster when bridging. This can help resolve a skewed bridge span that can't be resolved by adjusting the **Twist** value alone.

Auto Connection: When enabled, adds segments automatically, connects them to adjoining boundary edges when bridging opposing edges within the same surface, and attempts to retain the surrounding curvature.

Continuous Bridge: When enabled, bridges successive clusters of polygons or edges into a continuous length or tube with the spans evenly divided between each individual span. Selection order determines the order in which Modo bridges the spans.

Use Guide Curve: When enabled, uses a guide curve to determine a single continuous edge along the final bridged span that may eliminate a twisting issue. When using the Bridge tool to connect consecutive curves created with the Contour tool, twisting may occur in the resulting geometry. You can select a single vertex from each curve and press **Shift**+**O** on the keyboard to create an open curve to serve as a guide curve. Then switch to Polygon selection mode and select all the curves in order, with the guide curve last. Do this before activating the Bridge Tool and enabling **Use Guide Curve**.

Tension: Provides additional control over the bridge path used when **Curve** mode is active. Increasing this percentage value accentuates the curved effect and decreasing it mutes the effect toward the linear path.

Make UVs: Determines how Modo generates a UV map automatically for the resulting geometry.
Connected- Connects the edges between existing UV values as a straight line.
U/V- Fills the entire UV map 1-0 area with the resulting bridged geometry in the specified orientation.

Curve Properties

It is possible to bridge between selections of two curves or, by enabling **Continuous Bridge**, a series of successive curves. You can control the smoothness of the bridging in these cases by adjusting the **Segments** between spans and the **Steps** across the curve. The **Tension** setting controls how smoothly one span blends into the next.

Automatic: When enabled, determines automatically the **Steps** value by subdividing the curve based on the Mesh Item layer's **Curve Refinement Angle** setting.

Steps: Determines the the number of curve subdivisions when **Automatic** is disabled. Larger values produce smoother surfaces; smaller values produce more faceted surfaces.

Profile Attributes Properties

Inset: Specifies the X direction offset amount of 1D profile.

Reverse Direction: When enabled, evaluates the 1D profile from top to bottom.

Reverse Inset: When enabled, reverses the **Inset** value to mirror the profile.

Keep Aspect: Sets the **Inset** value automatically based on the aspect ratio of the profile.

Profiles: Provides a list (essentially a mini Preset Browser) for viewing various profiles. This works the same as the standard Preset Browser.

Join Meshes Procedurally Using the Bridge MeshOp

The bridge MeshOp allows you to quickly bridge between two or more faces or edges on a single mesh and join them. Unlike the direct modeling bridge tool, the Bridge MeshOp also allows you to adjust parameters such as segments, twist, and tension even after the tool is applied and can continuously be adjusted at any stage of your modeling process without needing to rebuild geometry or reapply the operation.

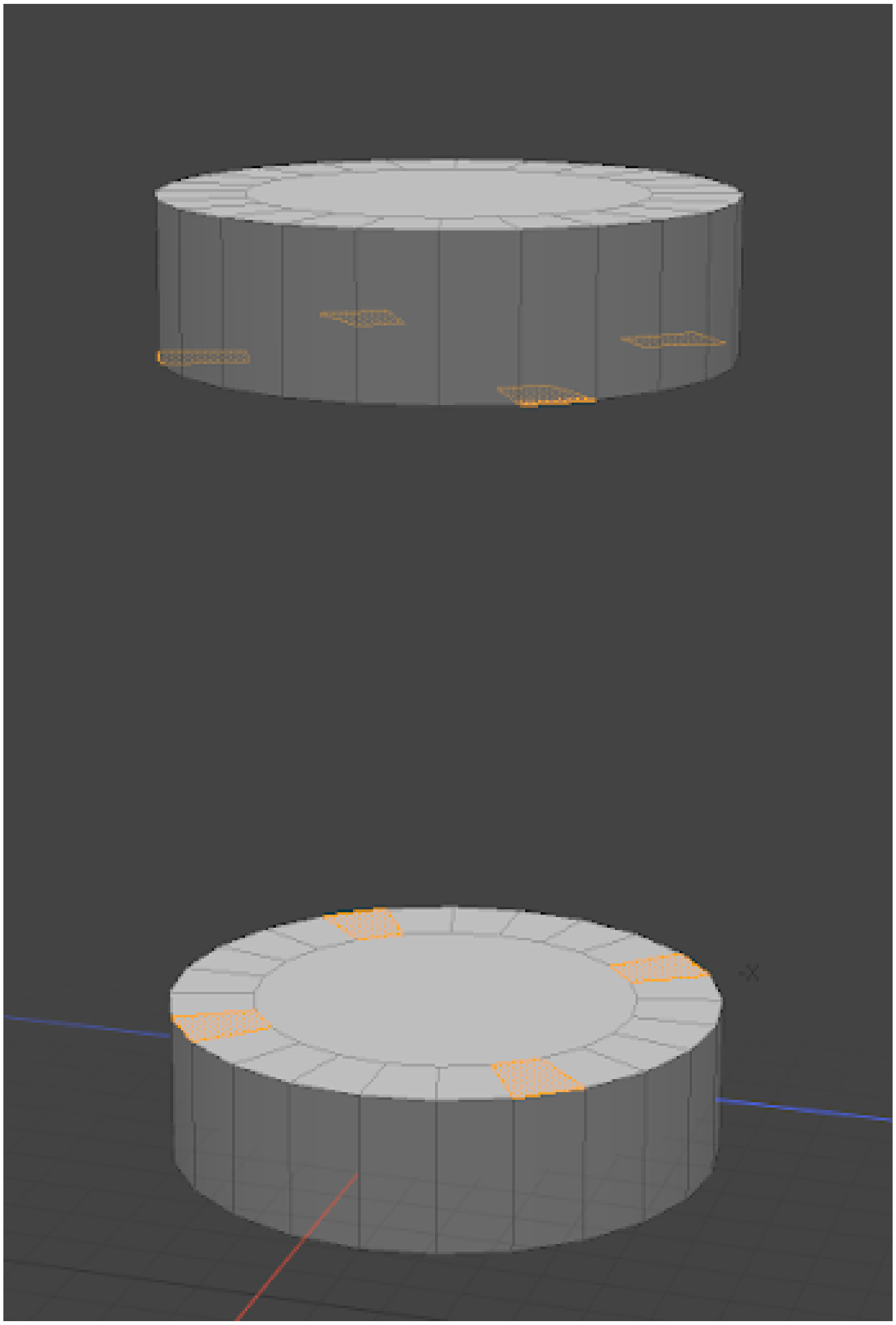
Activate the Bridge MeshOp by choosing **Mesh Operations** > **Edit**> **Bridge** in the Mesh Operations viewport, or by going to **Add** > **Mesh Operations** > **Edit** > **Bridge** in the Schematic viewport.

You need to select at least two polygons that do not share any edges or at least two edges that do not share any vertices to join meshes with the Bridge tool. The Bridge creates a polygon that spans from the first edge or face selected to the second face or edge selected.

To apply a Bridge MeshOp:

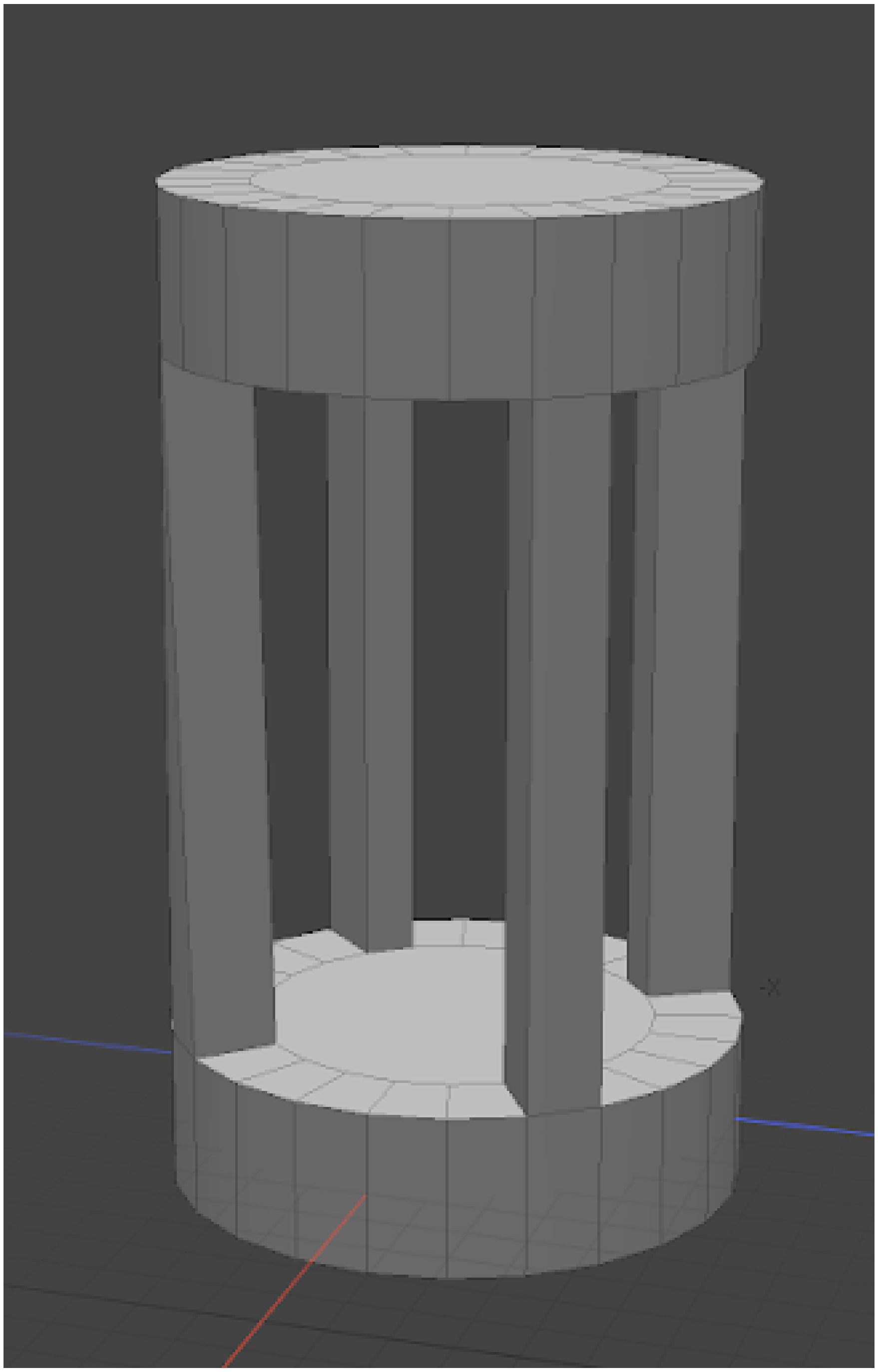
1. Select two or more faces or edges on your geometry. In this example, we will be selecting faces, but the workflow is the same for bridging edges.
2. In the Mesh Operations viewport, click the **Add Operation** button.
3. Add a Bridge operation.

> **Tip:** The Bridge MeshOp can also be applied via the Schematic viewport by dragging your mesh from the Items List into the Schematic viewport, selecting the faces you want to be bridged and then clicking **Add**.



A bridge is applied between the mesh selections.

4.



Bridge MeshOp Properties

Selection Type	Specifies whether **Polygons** or **Edges** are being bridged. Modo sets this automatically depending on what is selected when the operation is applied, but you can change the type at any time.

Segments	Allows you to set the number of polygons that Modo creates across the length of the bridge. Increase or decrease the number of segments by dragging in the 3D viewport, or by entering a number in the **Properties** panel. For curved bridge spans, a higher number of segments produces smoother results.
Twist	Enter a number higher than zero to rotate the bridged polygons between the original connection points.
Mode	Sets the interpolation style of the bridge polygons generated: • **Linear** - Creates a bridge in a straight light between the original selections. • **Curve** - creates an arc between the original selections. The curve can be further refined by adjusting the **Tension** setting. • **Smooth** Creates an "ease in/ease out" blend between the original selections.
Remove Polygons	When enabled, removes the polygons originally selected to create the bridge. This is enabled by default and preferred behavior.
Flip polygons	When enabled, flips the normals of newly created polygons in the bridge. You can enable **Flip Polygons** when a negative bridge action causes the resulting polygons to appear inside-out.
Reverse Direction	When enabled, reverses the normal direction of the target polygon cluster. This can help resolve a skewed bridge span that can't be corrected by adjusting the **Twist** value alone.
Auto Connection	When enabled, segments are created automatically and connected to adjoining boundary edges when bridging opposite edges within the same surface. Auto connection also helps to retain the curvature of surrounding geometry.
Continuous Bridge	Bridges all selected polygons or edges into one continuous length or tube with spans divided between each span. The selection order determines the order in which Modo bridges the spans. **Continuous Bridge** is enabled by default, but should be disabled if multiple independent polygons or edges are being bridged at once.
Close Bridges	When this option and **Continuous Bridge** are enabled, the last selection is connected to the first selection. This is useful for making radial or sweeping bridged shapes.
Use Guide Curve	When enabled, Modo uses a guide curve to determine a single continuous edge along the final bridged span that may help to eliminate twisting.

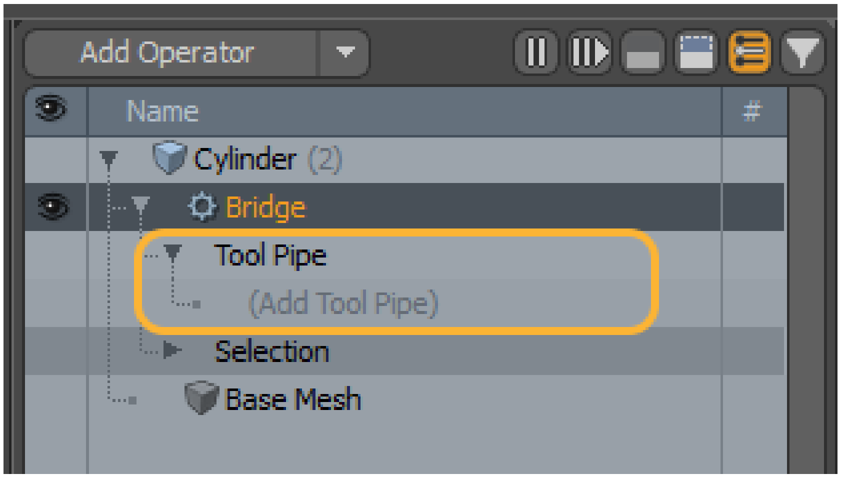
	Tip: When using the Bridge tool to connect consecutive curves created with the Contour tool, twisting may occur in the resulting geometry. You can select a single vertex from each curve and press **Shift**+**O** on the keyboard to create an open curve to serve as a guide curve. Then switch to polygon selection mode and select all the curves in order, with the guide curve last. Do this before activating the Bridge Tool and enabling **Use Guide Curve**.
Tension	Provides additional control over the bridge path used when **Mode** is set to **Curve**. Increasing this percentage value accentuates the curved effect and decreasing it mutes the effect toward the linear path.
Make UVs	Determines how Modo generates a UV map automatically for the resulting geometry.
Connected	Connects the edges between existing UV values as a straight line.
U/V	Fills the entire UV map 1-0 area with the resulting bridged geometry in the specified orientation.

Using Content Presets With the Bridge MeshOp

The Bridge MeshOp can also be used with content presets to change the shape and silhouette of your bridge. Because of the procedural nature of the Bridge MeshOp, some extra setup is required to use content presets than there would be with the direct modeling version of the Bridge tool. Content presets are added to the Bridge MeshOp from the Mesh Operations viewport or via the Schematic viewport.

To add content presets in the Mesh Operations viewport:

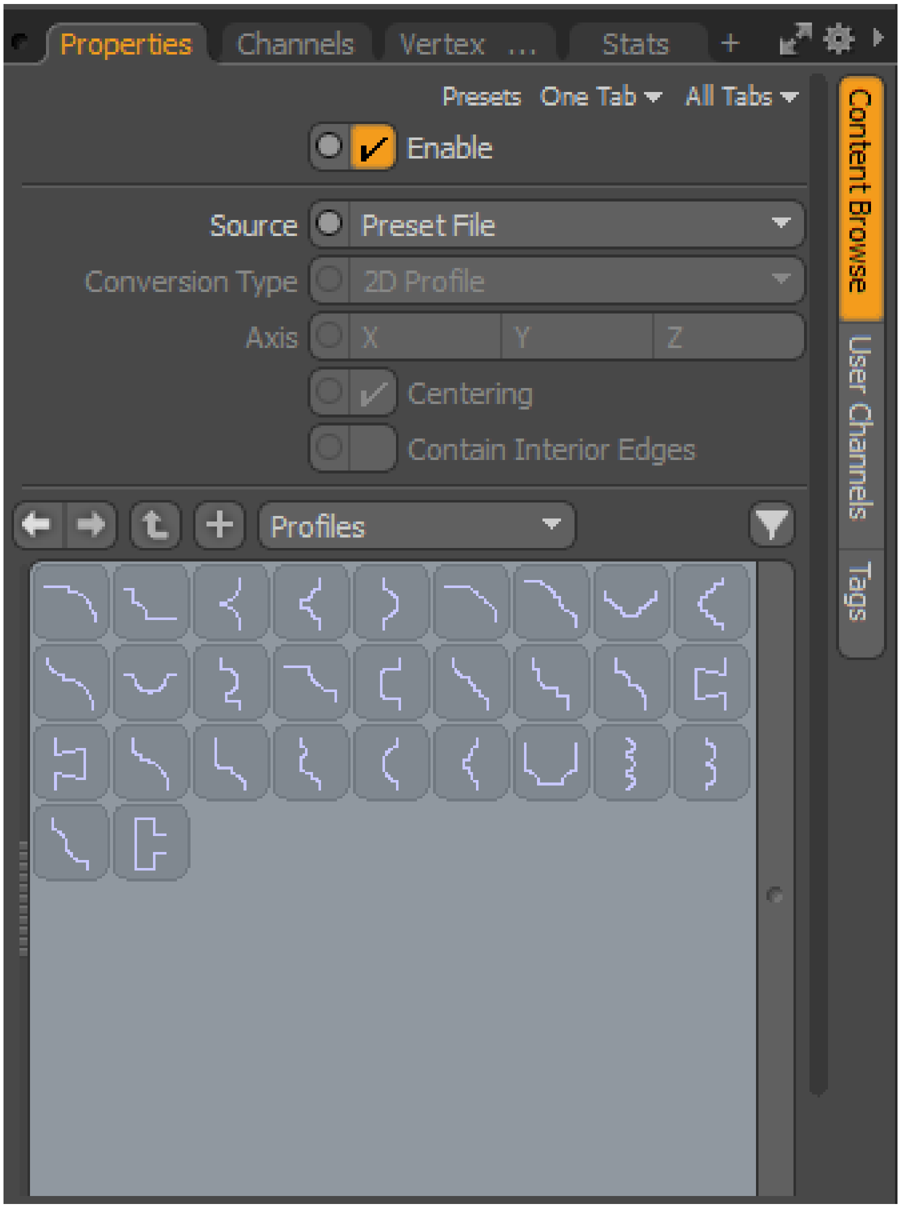
1. In the Mesh Ops list, expand the Bridge MeshOp so that the **Tool Pipe** parameter is exposed.


Add Operator
Name
#
Cylinder (2)
Bridge
Tool Pipe
(Add Tool Pipe)
Selection
Base Mesh

2. Click **(Add Tool Pipe)**.
3. Add a Content Preset MeshOp by navigating to **Tool Pipe** > **Sub Tools** > **Content Preset**,

   OR

   By tying **Content Preset** in the search field.
4. In the **Properties** panel, select a preset to use.



Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Content Browse
User Channels
Tags
Profiles

☑ Enable

Source: Preset File

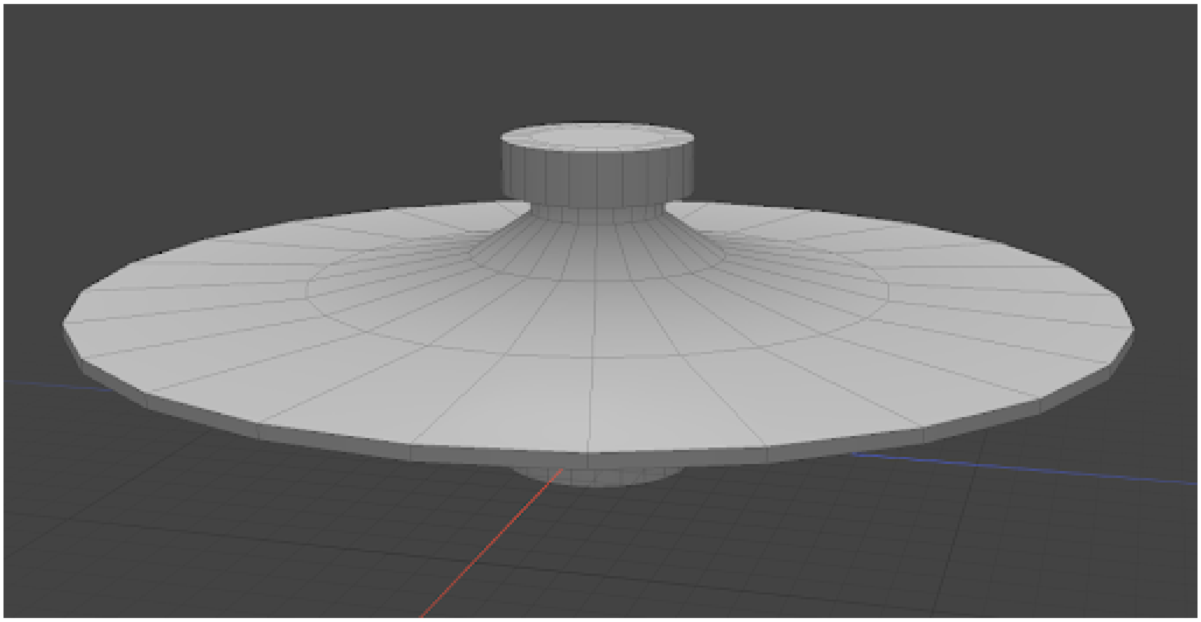
Conversion Type: 2D Profile

Axis: X Y Z

☑ Centering

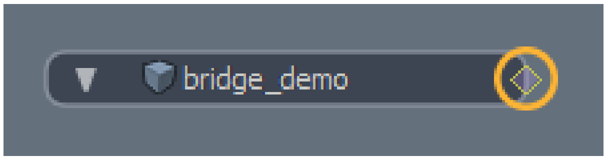
☐ Contain Interior Edges

The selected preset is applied to the bridge.



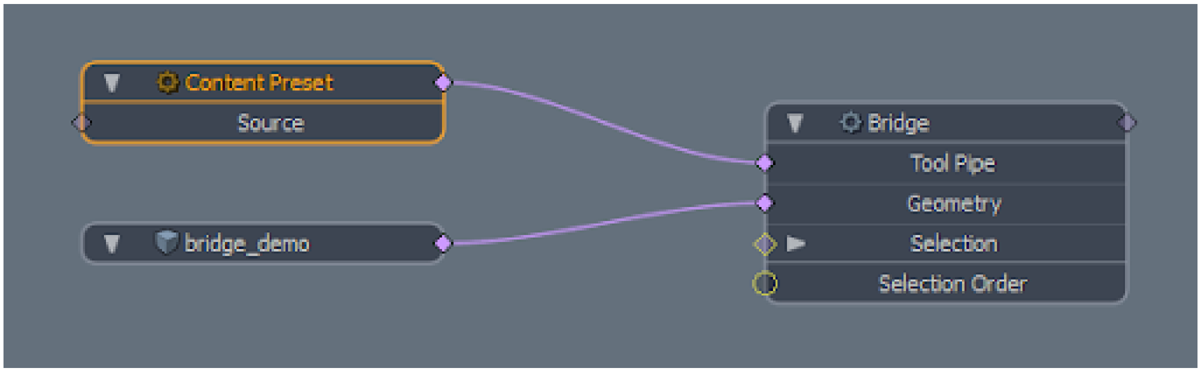
To add presets in the Schematic Viewport:

1. Drag the mesh item containing the Bridge MeshOp from the Item List into the Schematic viewport and double-click the diamond port on the node to expose connected nodes.

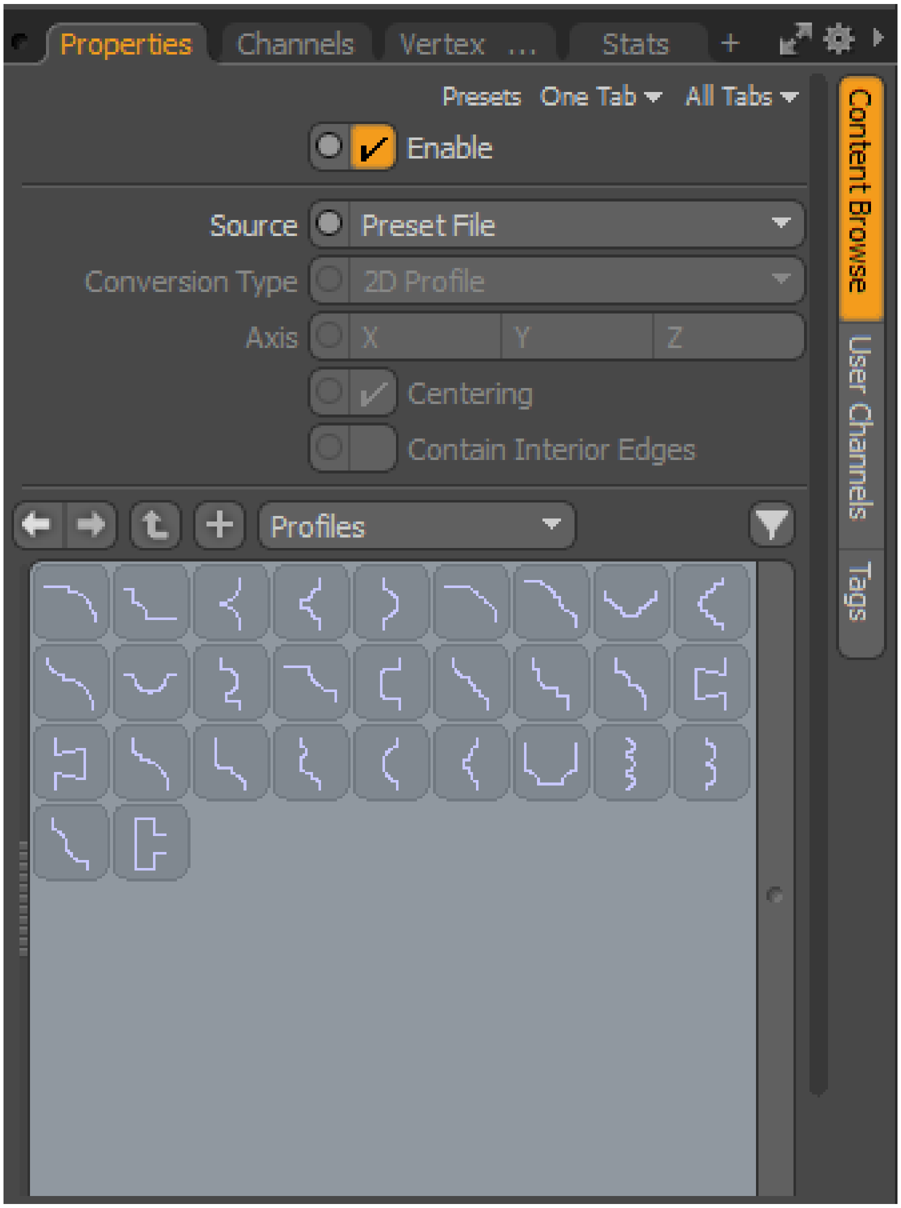

bridge_demo

2. In the Schematic viewport, click **Add** and navigate to **Mesh Operations** >**Tool Pipe** > **Sub-Tools** > **Content Preset** or by type **Content Preset** in the search field.

3. Connect the Content Preset to the **Tool Pipe** port on the Bridge node.


Content Preset
Source
bridge_demo
Bridge
Tool Pipe
Geometry
Selection
Selection Order

4. In the **Properties** panel, select a preset to use.


Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Content Browse
User Channels
Tags
Profiles

☑ Enable

Source: Preset File

Conversion Type: 2D Profile

Axis: X Y Z

☑ Centering

☐ Contain Interior Edges

The preset is applied. In both cases, once the preset is applied, you can continue to adjust the properties of the Bridge MeshOp such as **Segment**, **Twist**, and **Tension** within the **Properties** panel.

Clone

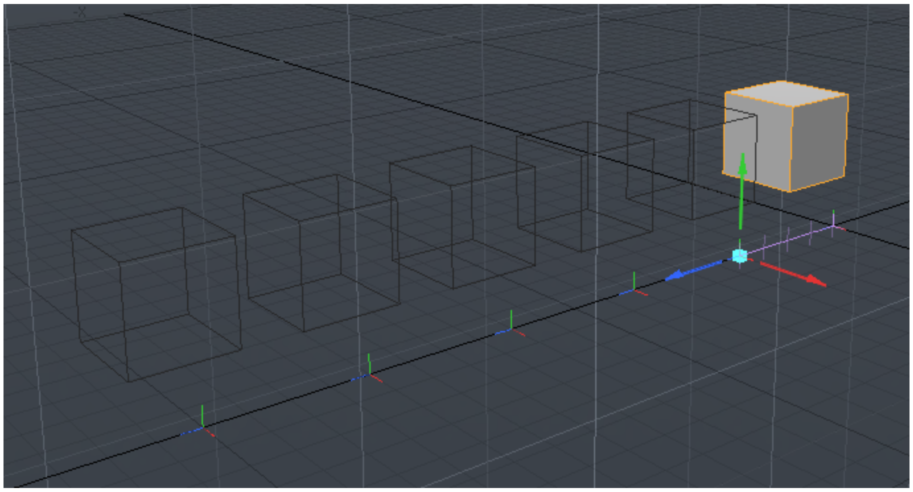
The Clone tool creates duplicates of the currently-selected geometry based on a specified linear offset. You can activate the Clone tool by clicking its icon on the **Duplicate** tab of the Modeling toolbox:

You can also find the tool in the menu bar under **Geometry** > **Duplicate** > **Clone**.

To Clone an item:

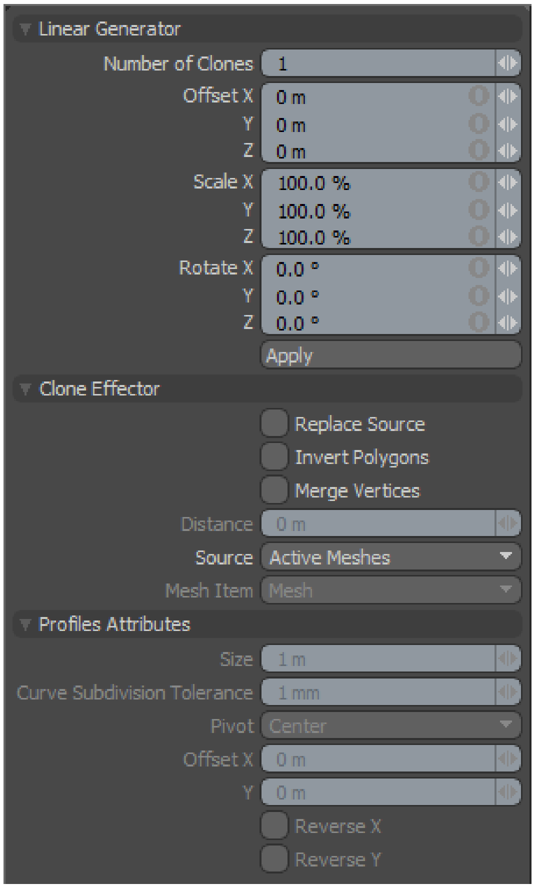
1. Activate the Clone tool.
2. Click and drag in the viewport to specify the linear offset along which you want to place the clones.

   Modo draws the clones in the viewport.

3. When the tool is active, its properties are displayed in the lower-left area of the screen. You can set the number of clones and other properties here.

Clone Properties

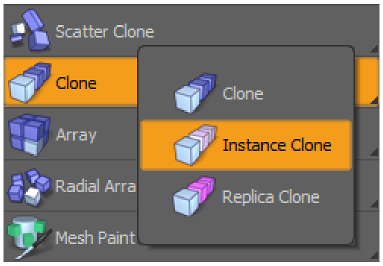


Number of Clones 1
Offset X 0 m
Y 0 m
Z 0 m
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Rotate X 0.0 °
Y 0.0 °
Z 0.0 °
☐ Replace Source
☐ Invert Polygons
☐ Merge Vertices
Distance 0 m
Source Active Meshes
Mesh Item Mesh
Size 1 m
Curve Subdivision Tolerance 1 mm
Pivot Center
Offset X 0 m
Y 0 m
☐ Reverse X
☐ Reverse Y

Clone	
Number of Clones	Determines the number of copies of the original geometry Modo should create.
Offset X/Y/Z	Determine how far each clone is from the previous one. For example, setting this value to **1 m** places each successive cloned object 1 meter away from the previous clone or from the original geometry (in the case of the first clone).
Scale X/Y/Z	Specifies the maximum amount of randomized fluctuation for any given clone.
Rotate X/Y/Z	Indicates the maximum amount of randomized rotation for any given clone.
Replace Source	When enabled, removes the original geometry and creates a replacement clone for it at the center of the scatter clone effect.

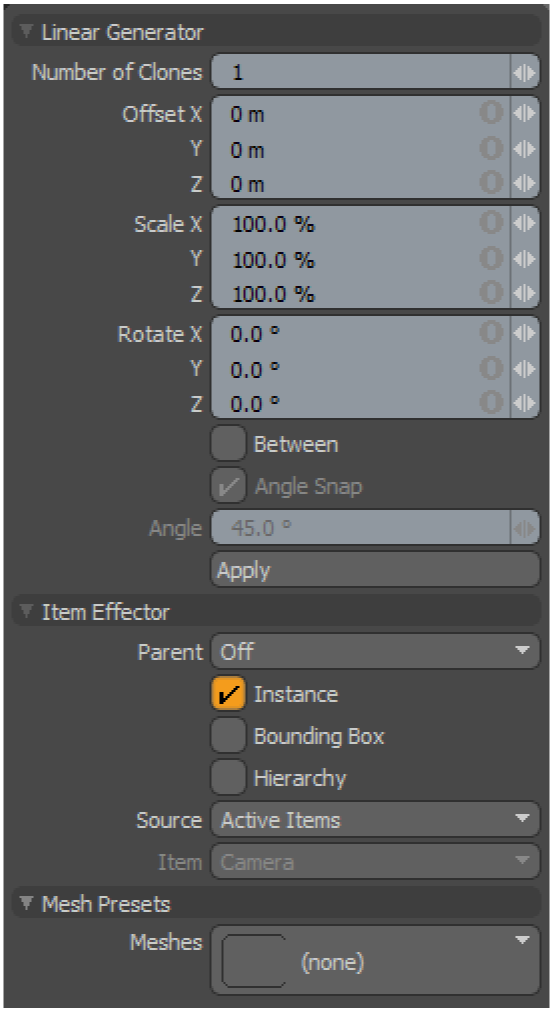
Instance Clone

The Instance Clone tool creates instances of the selected geometry based on a specified linear offset. To activate the tool, right-click on the **Clone** button and select **Instance Clone** from the list:


Scatter Clone
Clone
Array
Radial Arra
Mesh Paint
Clone
Instance Clone
Replica Clone

When the tool is active, its properties are displayed in the lower-left area of the screen.

Instance Clone Properties



Linear Generator

Number of Clones: 1

Offset X: 0 m
Y: 0 m
Z: 0 m

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Rotate X: 0.0 °
Y: 0.0 °
Z: 0.0 °

☐ Between

☑ Angle Snap

Angle: 45.0 °

Item Effector

Parent: Off

☑ Instance

☐ Bounding Box

☐ Hierarchy

Source: Active Items

Item: Camera

Mesh Presets

Meshes: (none)

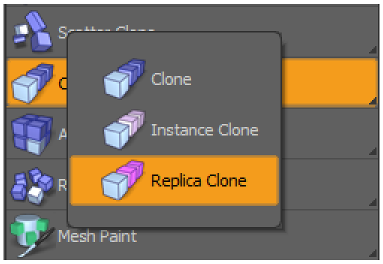
In addition to the **Linear Generator** properties described above for the Clone tool, you can set the following properties for Instance Clone.

Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**. • **Off** - Clones source item as additional instanced layers in the **Item List**. • **Hierarchy** - Creates a cascading array where each subsequent clone is a child of the previous clone. • **Root** - Creates instances as child items of the source mesh.

Item Effector	
	• **Primary** - Creates Instances as children of a group locator that is parented to the source mesh.
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewports by pink wireframes. This is enabled by default.
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, Modo clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo clones. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

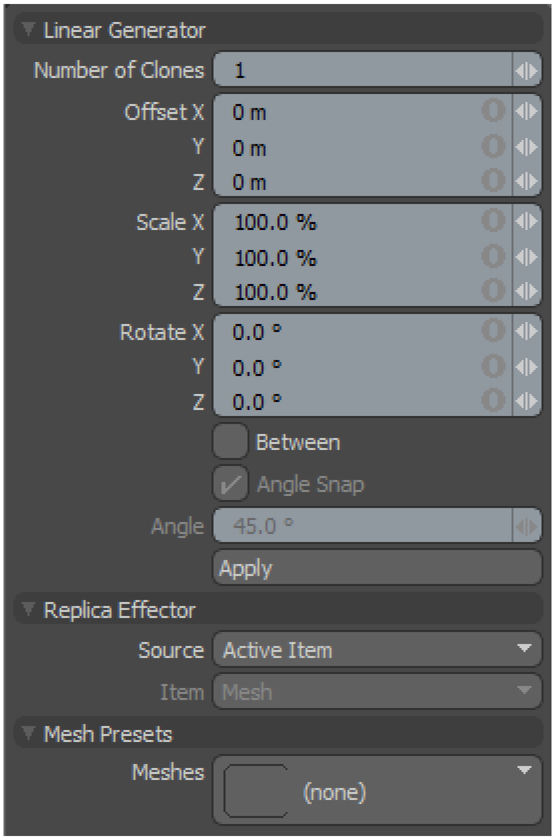
Replica Clone

The Replica Clone tool creates clones of the selected geometry using a Replicator, based on a specified linear offset. To activate the tool, right-click on the **Clone** icon and select **Replica Clone** from the list.



When the tool is active, its properties are displayed in the lower-left area of the screen.

Replica Clone Properties



Linear Generator

Number of Clones 1

Offset X 0 m
Y 0 m
Z 0 m

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Rotate X 0.0 °
Y 0.0 °
Z 0.0 °

☐ Between

☑ Angle Snap

Angle 45.0 °

Apply

Replica Effector

Source Active Item

Item Mesh

Mesh Presets

Meshes (none)

In addition to the **Linear Generator** properties described above for the Clone tool, you can set the following properties for Replica Clone.

Replica Effector	
Source	Determines the source mesh that Modo clones along the path. • **Active Items** - Uses the currently selected item(s) as a clone source.

Replica Effector	
	• **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Procedural Clone

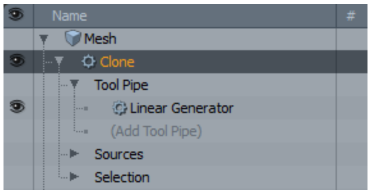
Clone is also available as a procedural mesh operation. It is a combination of the Clone Effector and Linear Generator sub-tools.

You can add the operation from the **Mesh Ops** tab, on the right-side of the screen.

1. Click the **Add Operator** button, and navigate to **Mesh Operations** > **Duplicate** > **Clone**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

   The Clone mesh operation is added to the list and the geometry below it in the list is duplicated. You can adjust its settings in the operation's **Properties** panel.

You can expand the Clone item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses.



Clone requires the following inputs:

- **Tool Pipe** - Allows you to add sub-tools to the operation. For the Clone tool, Linear Generator is added by default.
- **Sources** - Allows you to specify the geometry you want to duplicate.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.
- **Selection** - Allows you to select the elements to which you want to apply the tool.

Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

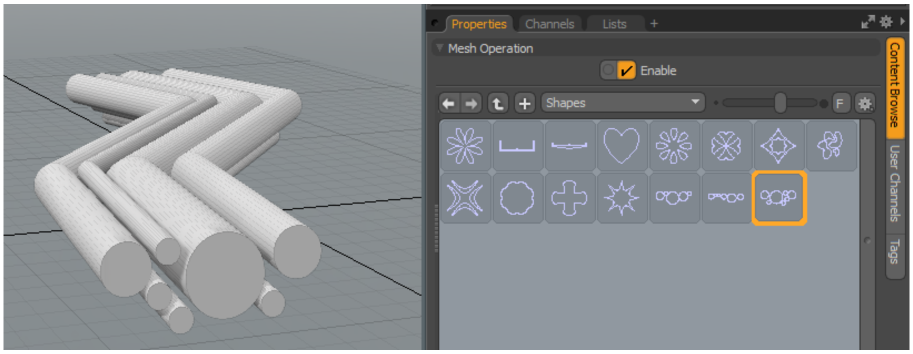
The Clone mesh operation has the same properties as the traditional Clone tool. See Clone.

Content Preset

The procedural **Content Preset** sub-tool offers a series of ready made Profiles, which are different 2D curve styles that can be used to generate new geometry. The **Content Preset** tool is used as an input to the **Tool Pipe** of tools such as the Bridge tool or **Sweep Effector**. These can use the Profile like a brush, and draw the Profile along a curve or line.

The image below shows the **Content Preset** tool on the right. In this example, the **Content Preset** tool has been added to a **Sweep Effector** that is drawing the output of a Pen Generator. A Profile consisting of a series of circles has been used to draw the tubes on the left.

Note: You may need to widen the **Properties** panel to see the toolbar above the Profiles. When the panel is narrow, the toolbar can be opened by clicking **Browser Toolbar**.



The Profiles are presented by category. For example, the screenshot above shows the Profiles in the **Shapes** category. To select a category, click the category dropdown above the Profiles.

Note: For more information on Profiles, see Preset Browser.

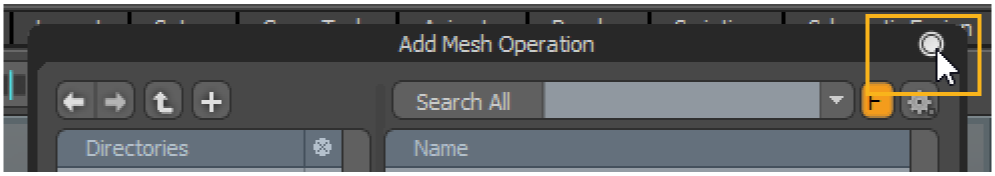
Using the Content Preset

The following steps show how to use the **Content Preset** sub-tool and a **Pen Generator** to draw the tube mesh shown in the screenshot above.

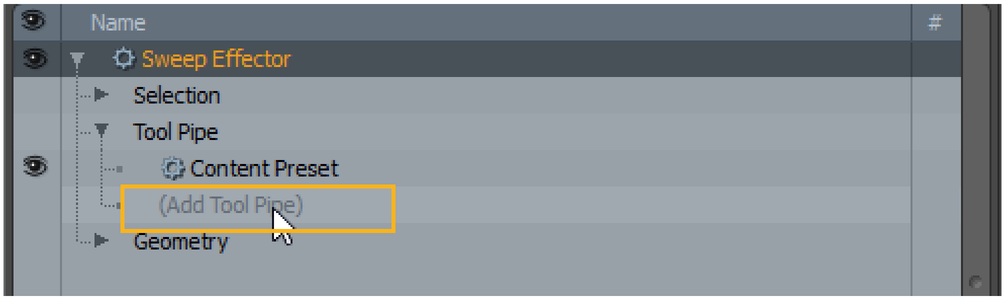
1. In the **Model** layout, click the **Procedural** tab (upper right), and click **Add Item** above the lower half of the panel (the **Mesh Operations** list). This opens the Procedural Preset Browser.

Note: The **Procedural** tab has two **Add Item** buttons, make sure you click the lower button.

2. Click the small circle in the upper right corner of the **Preset Browser** to keep it open. This avoids having to repeatedly click the **Add Item** button to add procedural tools.



3. Add a **Sweep Effector**: Select **Procedural** > **Duplicate**, and double-click **Sweep Effector**.
4. Add a **Content Preset**: Select **Procedural** > **Tool Pipe** > **Sub-Tools**, and double-click **Content Preset**. A **Content Preset** is added to the **Tool Pipe** of the **Sweep Effector**.
5. Click the small circle in the upper-right corner of the **Preset Browser** to close it.
6. Add a **Pen Generator**: In the **Mesh Operations** list, expand the **Tool Pipe** for the **Sweep Effector**, and click **(Add Tool Pipe)**.



Select **New** and double-click **Pen Generator**. This is added to the **Tool Pipe** of the **Sweep Effector**.

7. You can now add a polyline as a guide along which to sweep the profile that is currently selected in the **Content Preset**. Expand **Polyline** under the **Pen Generator** and click **(Add Polyline)**.
8. Add a new mesh for the polyline: In the **Preset Browser**, select **New**, and double-click **Mesh**.

Note: The new geometry is drawn in the original mesh (not the new, selected line mesh), and so may be seen in wireframe if you're using the default display settings.

9. To draw a polyline in the new mesh, select the Pen tool in the **Basic** tab of the modeling toolbox, set the **Type** to **Polyline** (in the **Tool Properties**), and draw a line in the 3D viewport.

Note: You won't see the swept shape unless one is selected in the **Content Preset**.

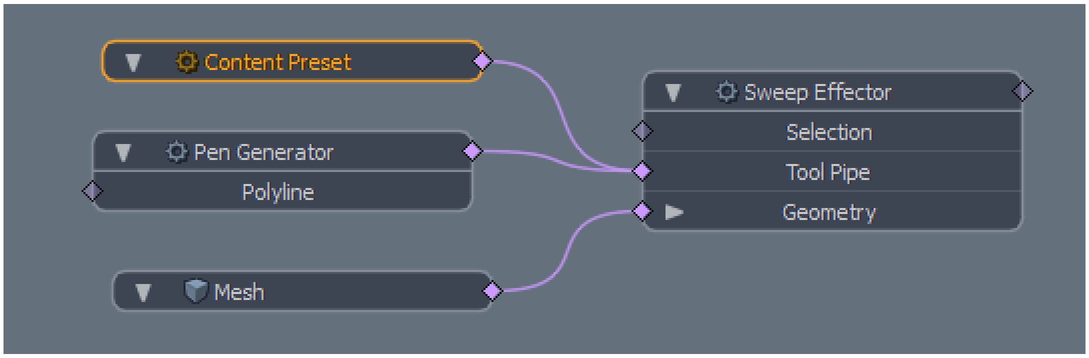
10. You can change the profile that is swept along the line. Select the **Content Preset** in the **Mesh Operations** list to see the profiles in the item's **Properties** panel (lower-right). Click on a profile to select it. In the example above, the last Profile in the **Shapes** category was used.

Tip: Try changing the **Invert Polygons** setting for the **Sweep Effector**. In some cases, this fills the resulting mesh, creating a solid appearance.

Content Preset Schematic

The **Content Preset** has a single output that connects to the **Tool Pipe** of a parent tool.

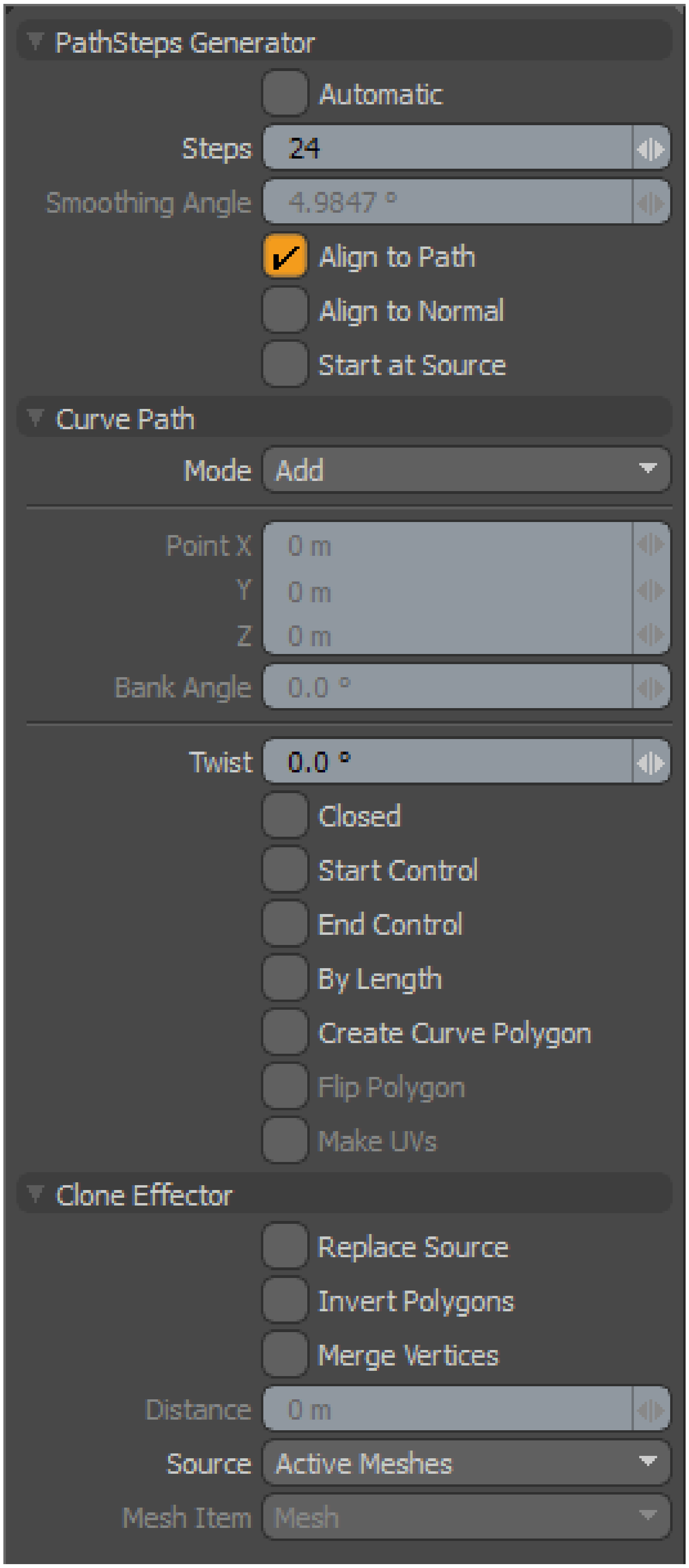
The example below shows a **Content Preset** used by a Sweep Effector to draw a **Profile** along a polyline that is generated by a **Pen Generator**.


Content Preset
Pen Generator
Polyline
Mesh
Sweep Effector
Selection
Tool Pipe
Geometry

Curve Clone

The **Curve Clone** tool allows you to quickly clone geometry by clicking down control points for a curve. The cloned element is determined by the settings of the **Source** option of the **Clone Effector**. As the curve is created, the specified geometry duplicates itself along the curve in real-time (depending on system performance). You may want to begin with a pre-created curve; simply put the curve in a visible, but deselected Mesh Item, prior to activating the tool (also known as a **Background** item). Once the second vertex of the curve has been created with the tool, the clone effector creates geometry along it, using the various controls.

Note: In addition to editing the control points in the 3D viewport, you can also edit them in the **Render Display**.



PathSteps Generator

☐ Automatic

Steps: 24

Smoothing Angle: 4.9847 °

☑ Align to Path

☐ Align to Normal

☐ Start at Source

Curve Path

Mode: Add

Point X: 0 m

Y: 0 m

Z: 0 m

Bank Angle: 0.0 °

Twist: 0.0 °

☐ Closed

☐ Start Control

☐ End Control

☐ By Length

☐ Create Curve Polygon

☐ Flip Polygon

☐ Make UVs

Clone Effector

☐ Replace Source

☐ Invert Polygons

☐ Merge Vertices

Distance: 0 m

Source: Active Meshes

Mesh Item: Mesh

The following properties are available for the Curve Clone tool.

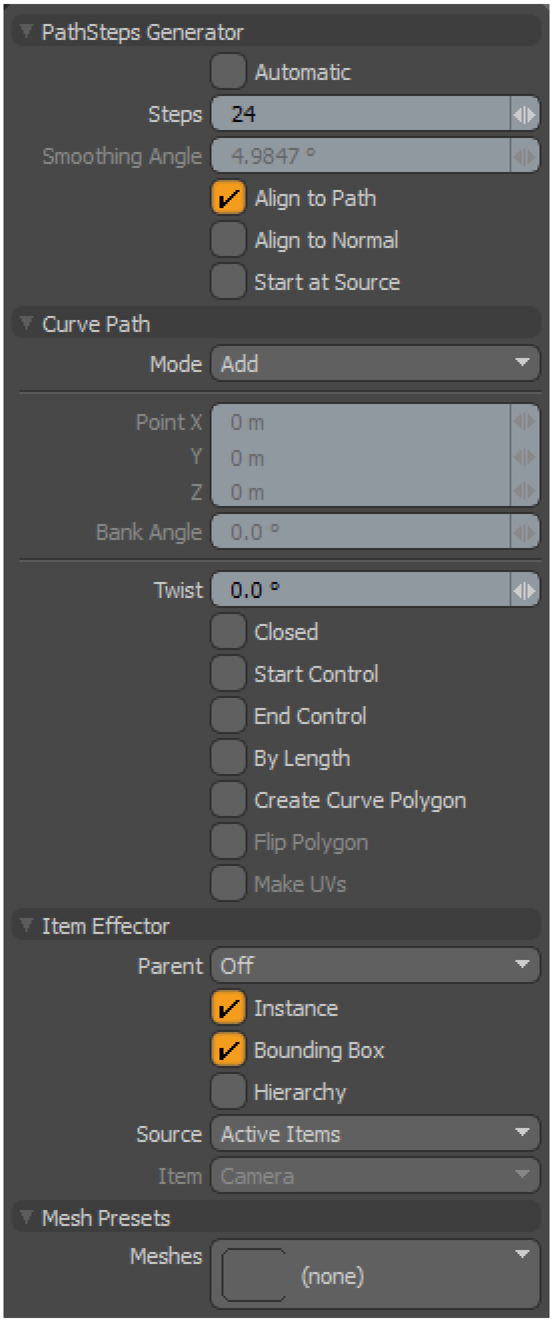
Path Steps Generator	
Automatic	Forces a computed number of clones based on the **Smoothing Angle** value, such that a clone is placed along the path to maintain a level of smoothness at or above the user-set level (determined by the **Smoothing Angle**).
Steps	Active only when the **Automatic** option is disabled. The defined **Steps** value determines the number of particles generated along the curve path that feed into the **Effector**, essentially controlling the number of clones.
Smoothing Angle	Determines the angle along the curve where a clone is created when the **Automatic** option is enabled.
Align to Path	When enabled, the generated particles include rotation information so the newly-created clone aligns to the path vector direction at the location where it is created.
Align to Normal	When enabled, the generated particles include rotation information so the newly-created clone aligns with any stored normal direction for a vertex, such as for a curve extracted from geometry.
Start at Source	Often a curve used for cloning has its initial control knot imperfectly aligned with the geometry to be cloned. With **Start at Source** active, the curve is used as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly-created geometry to move away from the intended original location.
Curve Path	
Mode	By default, the mode is set to **Add**. In the **Add** mode, clicking on a curve point and dragging edits the curve's position. Clicking anywhere in the 3D viewport other than the handles create a new curve control point. To avoid creating new control points when dragging existing points around you can set this mode to **Edit**. The **Delete** mode allows quick removal of control points by simply clicking on them.
Point X/Y/Z	The XYZ position of the currently selected knot. You can interactively edit the position by clicking over the knot indicated by the cyan dot. It turns yellow and you can now drag its position around in the viewport, or manually enter a position by editing any of the data fields.

Path Steps Generator	
Bank Angle	For each path position, a bank angle can be defined that controls the rotation angle of elements cloned through that position.
Twist	Works just like the **Bank Angle** option, but instead of being calculated per knot, the rotation amount is defined across the entire length of the curve, where the **Twist** value represents the amount of twisting rotation from the first knot in the curve to the last knot.
Closed	Causes the curve to become a closed loop, connecting the end point to the start.
Start Control	Enabling this toggle causes the first knot in the curve to become a control point, but not an actual segment of the curve. This provides a simple method for changing the curvature at the start of the curve.
End Control	Enabling this toggle causes the last knot in the curve to become a control point, but not an actual segment of the curve. This provides a simple method for changing the curvature at the end of the curve.
By Length	Forces the newly-created geometry to be evenly distributed across the entire length of the curve rather than spacing based on the control knots.
Create Curve Polygon	Generates a curve type polygon that remains after the tool has been dropped. This is useful if you want to re-use the curve you create with the current tool, with additional curve-based tools later in your project.
Flip Polygon	Invert the faces of the generated polygons.
Make UVs	When this toggle is active, the tool populates the current UV map with data for the generated polygons.
Clone Effector	
Replace Source	When this option is enabled, Modo removes the source geometry and position it at the first **Point** in the path.
Invert Polygons	When enabled, this option inverts the normal direction of the cloned geometry.
Merge Vertices	When enabled, this option merges coinciding vertices into a single

Path Steps Generator	
	vertex (based on a specified **Distance** value).
Distance	Used in conjunction with the **Merge Vertices** option, the **Distance** value determines the maximum distance between vertices where merging occurs. Vertices outside this value remain unchanged. A value of zero only removes collocated vertices.
Source	Determines the mesh that is cloned along the **Pen** tool path. • **Active Meshes** - uses the currently-selected geometry as a clone source. • **Specific Mesh** - allows you to specify a specific Mesh Item as the clone source using the **Mesh Item** option below. • **All BG** - combines all the currently-visible background items, cloning them as a single mesh. • **Random BG** - chooses a random background item from all the currently-visible background items at each clone position. • **Preset Shape** - allows you to clone **Profile** shapes from the **Preset Browser**. Open the **Preset Browser** (**F6**) and select a profile before activating the tool.
Mesh Item	Works when the **Source** is set to **Specific Mesh** and allows you to set the mesh to be cloned.

Curve Instance

With the Curve Instance clone you can create instances of a geometry by clicking down control points for a curve. To activate the tool, right-click the **Curve Clone** tool icon in the **Duplicate** sub-tab of the modeling toolbox, and select **Curve Instance**.



PathSteps Generator

☐ Automatic

Steps 24

Smoothing Angle 4.9847 °

☑ Align to Path

☐ Align to Normal

☐ Start at Source

Curve Path

Mode Add

Point X 0 m

Y 0 m

Z 0 m

Bank Angle 0.0 °

Twist 0.0 °

☐ Closed

☐ Start Control

☐ End Control

☐ By Length

☐ Create Curve Polygon

☐ Flip Polygon

☐ Make UVs

Item Effector

Parent Off

☑ Instance

☑ Bounding Box

☐ Hierarchy

Source Active Items

Item Camera

Mesh Presets

Meshes (none)

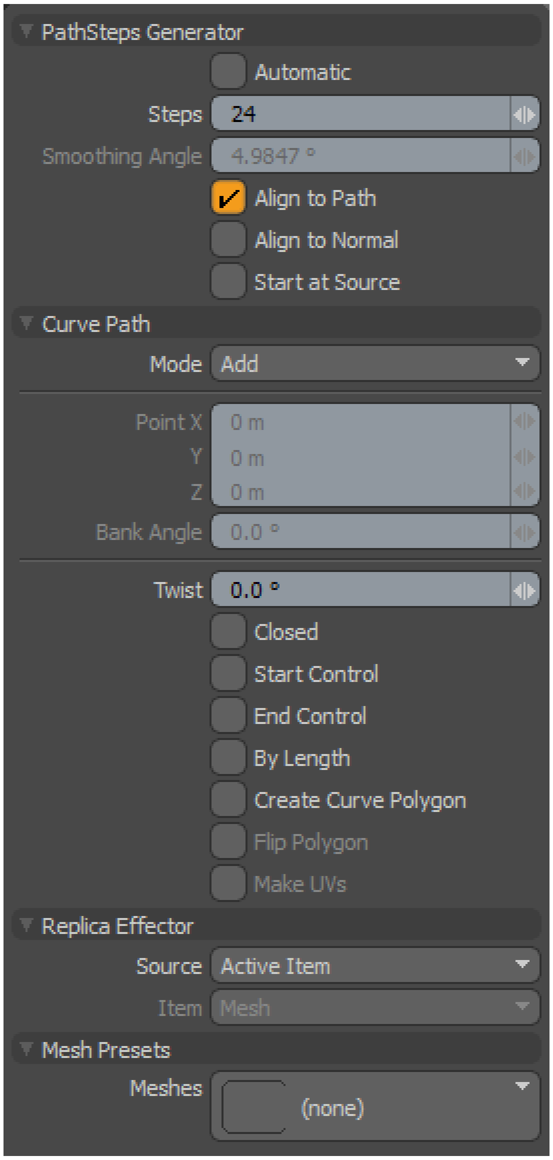
In addition to the **PathSteps Generator** and **Curve Path** properties described in the Curve Clone topic, the following properties are available for the Curve Instance tool:

Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**.
Instance	When enabled, each cloned item is an instance of the source mesh.

Item Effector	
	Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewports by pink wireframes. (This is enabled by default.)
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	
Source	Determines the source mesh that Modo clones onto the curve.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the Source.

Curve Replica

The Curve Replica tool works like the Curve Clone tool, but creates replicas instead of actual meshes. To activate the tool, right-click the **Curve Clone** tool icon in the **Duplicate** sub-tab of the modeling toolbox, and select **Curve Replica**.



PathSteps Generator

☐ Automatic

Steps 24

Smoothing Angle 4.9847 °

☑ Align to Path

☐ Align to Normal

☐ Start at Source

Curve Path

Mode Add

Point X 0 m

Y 0 m

Z 0 m

Bank Angle 0.0 °

Twist 0.0 °

☐ Closed

☐ Start Control

☐ End Control

☐ By Length

☐ Create Curve Polygon

☐ Flip Polygon

☐ Make UVs

Replica Effector

Source Active Item

Item Mesh

Mesh Presets

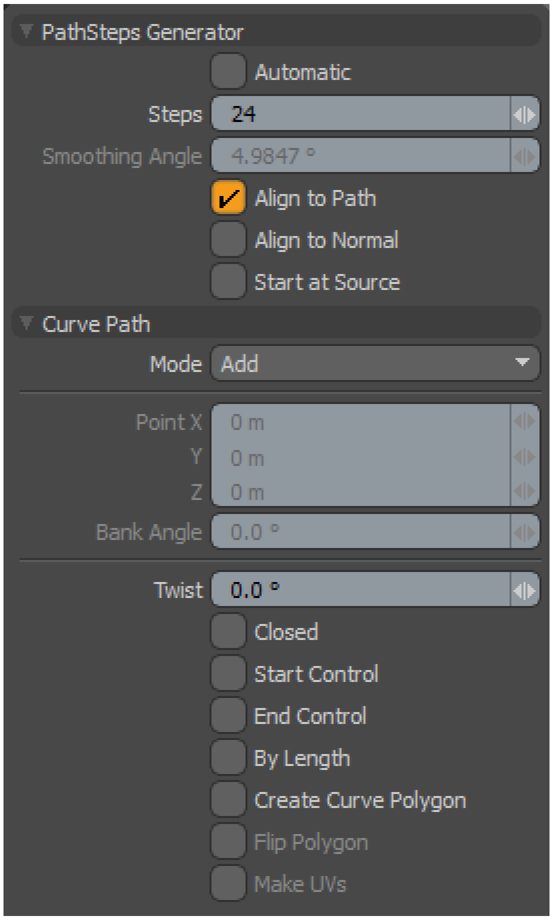
Meshes (none)

In addition to the **PathSteps Generator** and **Curve Path** properties described in the Curve Clone section, the following properties are available for the Curve Replica tool:

Replica Effector	
Source	Determines the source mesh that Modo clones onto the curve.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the Source.

Curve Transform

With the Curve Transform tool, you can organize your existing geometry along the path of a curve. To activate the tool, right-click the **Curve Clone** tool icon in the **Duplicate** sub-tab of the modeling toolbox, and select **Curve Transform**.



PathSteps Generator

☐ Automatic

Steps: 24

Smoothing Angle: 4.9847 °

☑ Align to Path

☐ Align to Normal

☐ Start at Source

Curve Path

Mode: Add

Point X: 0 m

Y: 0 m

Z: 0 m

Bank Angle: 0.0 °

Twist: 0.0 °

☐ Closed

☐ Start Control

☐ End Control

☐ By Length

☐ Create Curve Polygon

☐ Flip Polygon

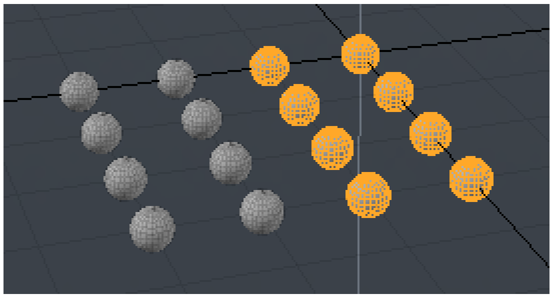
☐ Make UVs

Using the Curve Transform

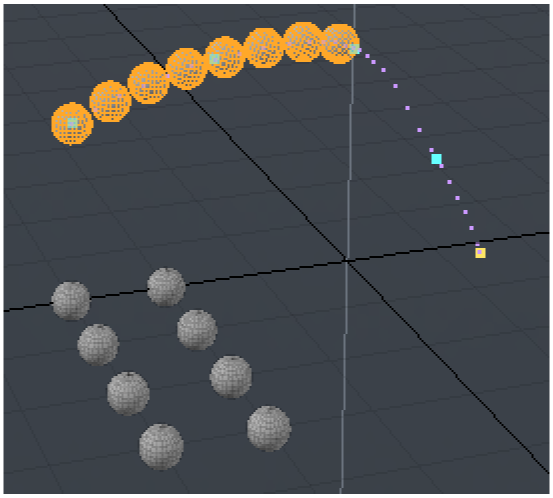
The **Curve Transform** tool moves selected geometry along the path of a curve that you draw in the 3D viewport. The tool repositions all selected geometry, providing the number of items does not exceed the **Steps** value. If you try and move more items than the **Steps** value, the extra items will not be repositioned.

The following steps are illustrated by transforming a series of spheres.

1. Select the geometry that you want to transform.

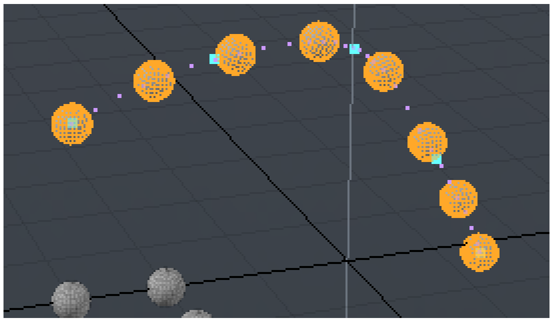


2. Select the **Curve Transform** tool.
3. Click in the 3D viewport and draw the curve using a series of clicks to lay control points.



4. Edit the tool parameters. See Curve Clone for details.

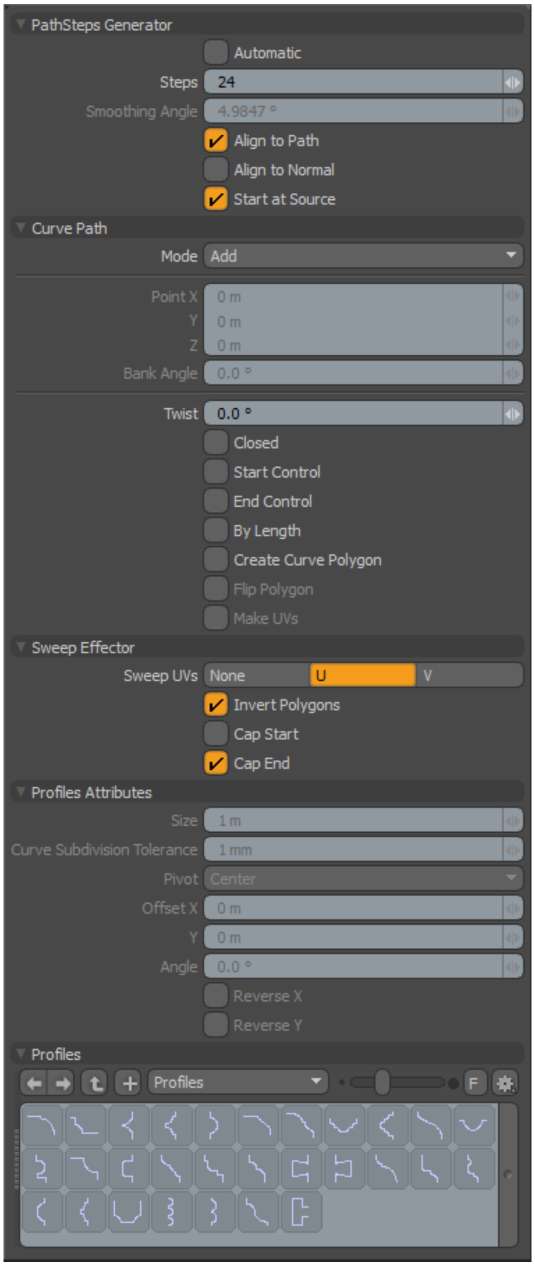
   For example, to evenly space out the items along the curve, set the **Steps** value to equal the number of items.



5. You can alter the curve by clicking and dragging the blue control points.

Curve Extrude

The **Curve Extrude** tool allows you to quickly extrude a geometry selection along the length of a curve. If the curve is pre-existing, select it first, then the polygonal geometry for extrusion. **Curve Path** also allows you to create a path interactively. To use, select the tool and make it interactive by clicking in the viewport. Once activated, a number of properties are available.



PathSteps Generator

☐ Automatic

Steps 24

Smoothing Angle 4.9847 °

☑ Align to Path

☐ Align to Normal

☑ Start at Source

Curve Path

Mode Add

Point X 0 m

Y 0 m

Z 0 m

Bank Angle 0.0 °

Twist 0.0 °

☐ Closed

☐ Start Control

☐ End Control

☐ By Length

☐ Create Curve Polygon

☐ Flip Polygon

☐ Make UVs

Sweep Effector

Sweep UVs None [✓ U] V

☑ Invert Polygons

☐ Cap Start

☑ Cap End

Profiles Attributes

Size 1 m

Curve Subdivision Tolerance 1 mm

Pivot Center

Offset X 0 m

Y 0 m

Angle 0.0 °

☐ Reverse X

☐ Reverse Y

Profiles

PathSteps Generator	
Automatic	Forces a computed number of clones, based on the **Smoothing Angle** value, such that a clone is placed along the path to maintain a level of smoothness at or above the user-set level (determined by the **Smoothing Angle**).
Steps	Active only when the **Automatic** option is disabled. The defined **Steps** value determines the number of particles generated along the curve path that feed into the Effector, essentially controlling the number of segments of the resulting extrusion.
Smoothing Angle	Determines the angle along the curve where a clone is created when the **Automatic** option is enabled.
Align to Path	When enabled, the generated particles include rotation information so the newly-created clone aligns to the path vector direction at the location where it is created.
Align to Normal	When enabled, the generated particles include rotation information so the newly-created clone aligns with any stored normal direction for a vertex, such as for a curve extracted from geometry.
Start at Source	Often a curve used for cloning has its initial control knot imperfectly aligned with the geometry to be cloned. With **Start at Source** active, the curve is used as a relative guide using the original geometry as the true starting location. Disabling this option may cause the newly-created geometry to move away from the intended original location.
Curve Path	
Mode	By default, this is set to **Add**. In **Add** mode, clicking on a curve point and dragging edits the curve's position. Clicking anywhere in the 3D viewport other than the handles creates a new curve control point. To avoid creating new control points when dragging existing points around, you can set this mode to **Edit**. The **Delete** mode allows quick removal of control points by simply clicking them.
Point X/Y/Z	The XYZ position of the currently-selected knot. You can interactively edit the position by clicking over the knot indicated by the cyan dot. It turns yellow and you can now drag its position around in the viewport, or manually enter a position by editing any of the data fields.

PathSteps Generator	
Bank Angle	For each path position, a bank angle can be defined that controls the rotation angle of elements cloned through that position.
Twist	Works just like the **Bank Angle** option, but instead of being calculated per knot, the rotation amount is defined across the entire length of the curve, where the **Twist** value represents the amount of twisting rotation from the first knot in the curve to the last knot.
Closed	Causes the curve to become a closed loop, connecting the end point to the start.
Start Control	Enabling this toggle causes the first knot in the curve to become a control point, but not an actual segment of the curve. This provides a simple method for changing the curvature at the start of the curve.
End Control	Enabling this toggle causes the last knot in the curve to become a control point, but not an actual segment of the curve. This provides a simple method for changing the curvature at the end of the curve.
By Length	Forces the newly-created geometry to be evenly distributed across the entire length of the curve rather than spacing based on the control knots.
Create Curve Polygon	Generates a curve-type polygon that remains after the tool has been dropped. This is useful if you want to re-use the curve you create with the current tool with additional curve-based tools later in your project.
Flip Polygon	Inverts the faces of the generated polygons.
Make UVs	When this toggle is active, the tool populates the current UV map with data for the generated polygons.
Sweep Effector	
Sweep UVs	Allows you to choose between the U or V axis when placing UV data for newly-created geometry.
Invert Polygons	Reverses the normal facing direction of the polygons generated by the extrusion. This is useful when extruded geometry appears inverted or inside-out.

PathSteps Generator	
Cap Start/End	Forces the back end of the extruded geometry to be capped or closed.
Profile Attributes	
Size	The size of 1D or 2D profile to sweep. If the size is 0.0, it uses the real size.
Curve Subdivision Tolerance	This is the minimum angle when it subdivides the Bezier curve to a string of two-point polygons for extrusion. This setting basically controls the detail tolerance of the conversion from the original curve to the resulting polygon extrusion.
Pivot	Specifies the pivot position of the profile.
Offset X/Y	The offset amount on the profile space.
Reverse X/Y	Reverses the selected profile to horizontal or vertical.
Profiles	
Profiles	A mini **Preset Browser** for viewing various profiles. Works the same way as the standard Preset Browser. Selecting any of the various profiles modulates the extrusion width with the selected shape over its length. To return to the original polygon selection for the extrusion, **Ctrl**+click the selected profile to deselect it.

Curve Extrude in the Procedural

Modo has a procedural Curve Extrude mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the stack in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Item** button above the **Mesh Operations** list. Under **Mesh Operations** > **Duplicate**, double-click **Curve Extrude**.

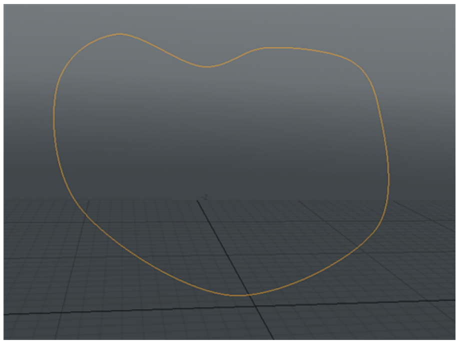
When you add the operation, Modo prompts you to specify the **Path** you want to use. Here, select the mesh that contains your curve.

You can expand the Curve Extrude item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses. Curve Extrude requires the following inputs:

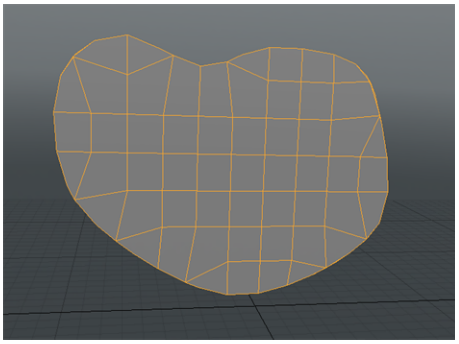
- **Selection** - Allows you to select the elements to which you want to apply the tool.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation. When you add a Curve Extrude, the Path Generator and Path Segment Generator tools are automatically added.

Curve Fill

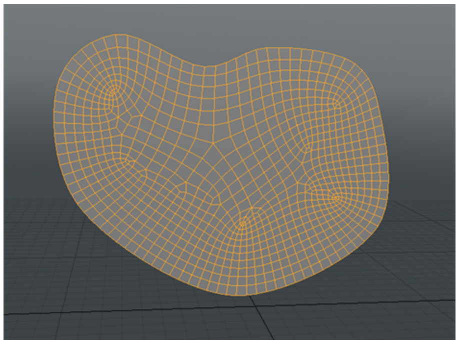
This is a procedural tool for filling closed, non-intersecting curves with quadrangles.



A sample curve.



The curve filled using the **Grid** setting.



The curve filled using the **Paving** setting.

The tool supports two fill methods:

- **Grid** - This forms a grid aligned to Cartesian coordinates. This is a fast method with fewer polygons but the results are not as reliable as **Paving**.
- **Paving** - This method repeatedly creates an inner ring of polygons, starting from the inside of the curve and finishing at the center. As the fill area gets smaller, multiple polygons may be merged or split to add or remove polygons as required. This method is slightly slower than the **Grid** method, but produces a better topology.

Note: Elaborate curve profiles may not result in a successful fill.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection.

Using Curve Fill

The following example stacks a Curve Fill tool on top of a mesh containing a curve. You can also add a curve to the tool using the **Curve** setting in the procedural stack.

1. Draw a curve using the Adding Curves tool, or add a procedural curve to the Mesh Operations List.

   If you have multiple curves and want to target a certain one, select the curve in the **Item List** or the Mesh Operations List.
2. Click **Add Operator** in the Mesh Operations List.

Note: You can also right-click on the mesh in the 3D viewport and select **Item Operations** > **Add Mesh Operation**.

3. Double-click **Curves** > **Curve Fill** to add it to the stack and fill the curve.
4. Adjust the Curve Fill Properties as required.

Using Guide Curves

The Curve Fill mesh operation can use open curves as guide curves for the topology. For example, if you need an edge in a certain place on your filled curve, you can specify a guide curve, and Modo will place an edge along the length of the guide curve. You can also use guide curves to separate your filled curve into individually editable islands.

You can add guide curves within the same mesh layer as the closed curve used for the fill, or in a separate one.

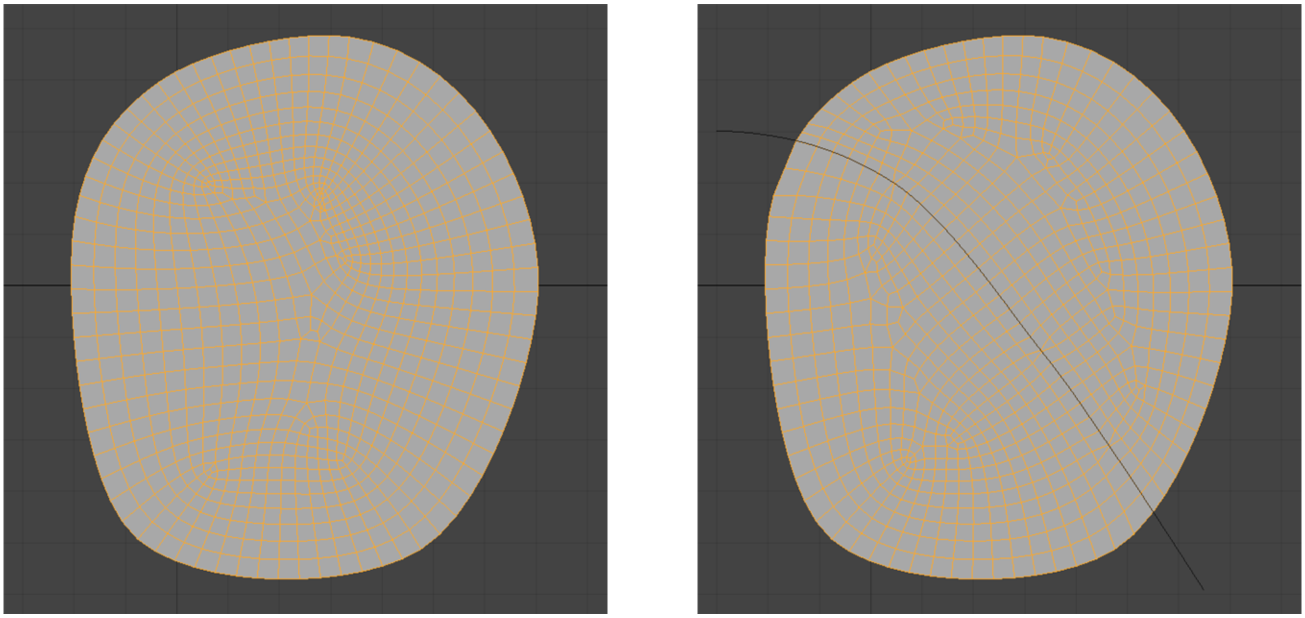
In a Separate Mesh

1. Add a new mesh layer and draw a curve.
2. Select the mesh where your Curve Fill mesh operation is and expand the operation it in the Mesh Operations list by clicking the arrow in front of it.

   This reveals the inputs the operation uses.
3. Expand the **Guide** input and click **(Add Guide)**.

   This opens the **Add Operation** dialog.
4. Under **Existing**, double-click the mesh where your open curve is.

   The curve fill topology now follows the new guide.



Curve Fill without Guide Curves

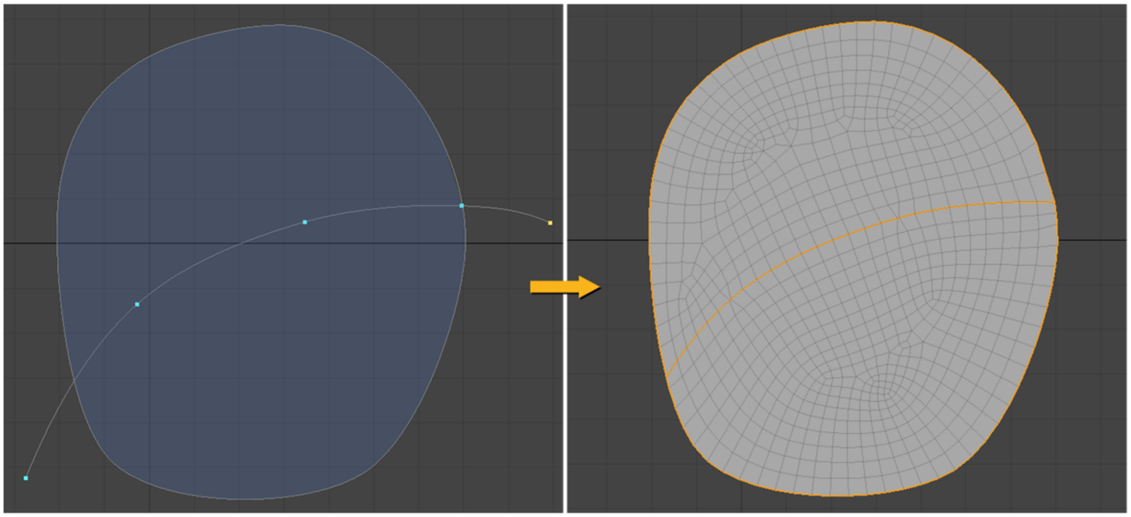
Curve Fill with Guide Curve

In the Same Mesh

To add the closed curve and the guide curves in a single mesh layer:

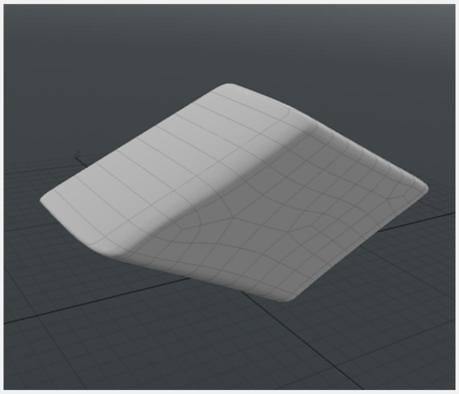
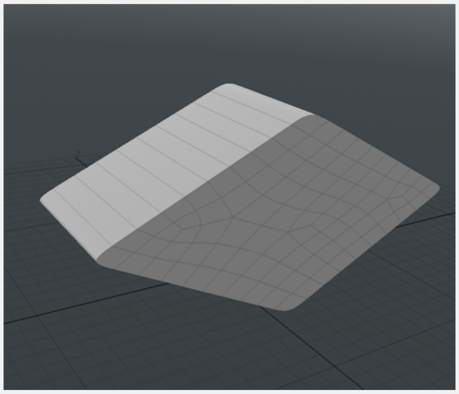
1. Add the Curve Fill mesh operation as described in Using Curve Fill.
2. In the Mesh Operations list, select the **Base Mesh**.
3. Draw your curve.

   The generated topology automatically follows the open curve.



Curve Fill Properties

Curve Fill	
Enable	Toggles the fill on and off.
Curve	Selects the curve to fill.
Guide	Selects the guide curve.
Delete Source Curve	When enabled, the original curve is deleted after it's filled with polygons. When disabled, the curve is preserved after the Curve Fill operation.
Merge Patch	When enabled, Modo treats the Curve Fill topology as one item, regardless of the existence of guide curves. When disabled, the guide curves separate the topology into distinct islands.
Method	Selects the fill method. Choose from: • **Grid** - Create fill quads as a grid, aligned with Cartesian coordinates. • **Paving** - Create fill quads in progressive rings from the outer edge to the center.
Fill Steps	Determines the quad density in the fill surface.
Curve Selection	Select one or more curves from the input mesh. The number counts up from zero, and can be a single value or a range. To specify a range separate values with a hypen, for example: **1-3**.
Polygon Type	Selects the polygon type for the quads: • **Face** - Standard face with no subdivision. • **Subdivision** - Smooth subdivision using Modo's standard algorithm. • **Catmull-Clark** - Smooth subdivision using the Catmull-Clark algorithm.
Edge Weight	When Polygon Type is set to Subdivision or Catmull-Clark, determines the edge weight to modify the amount of sharpness at the edges in a subdivided fill.

	 A diamond shaped curve that has been Catmull-Clark subdvided and thickened using Smooth Shift. The **Edge Weight** is **0%**.  The same mesh but with an increased **Edge Weight**.
Make UVs	Check this to create a UV map for the filled curve.
UV Map Name	If you have checked **Make UVs**, you can edit the name of the UV map.
Pack UVs	Only active when **Make UVs** is enabled. When enabled, packs the UV map to the 0.0 - 1.0 space.

Curve Fill Procedural

The Curve Fill has the following procedural sub-tool:

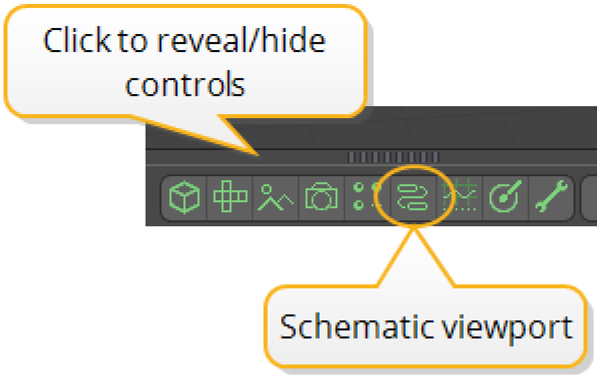
- **Curve** - The curve(s) to fill. This allows you to fill a curve (or curves) on another mesh layer (or from third-party tools that generate curves).
- **Guide** - The guide curve(s) used to generate the fill topology.

Curve Fill Schematic

The Curve Fill node has the same inputs as the procedural version.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.


Click to reveal/hide controls
Schematic viewport

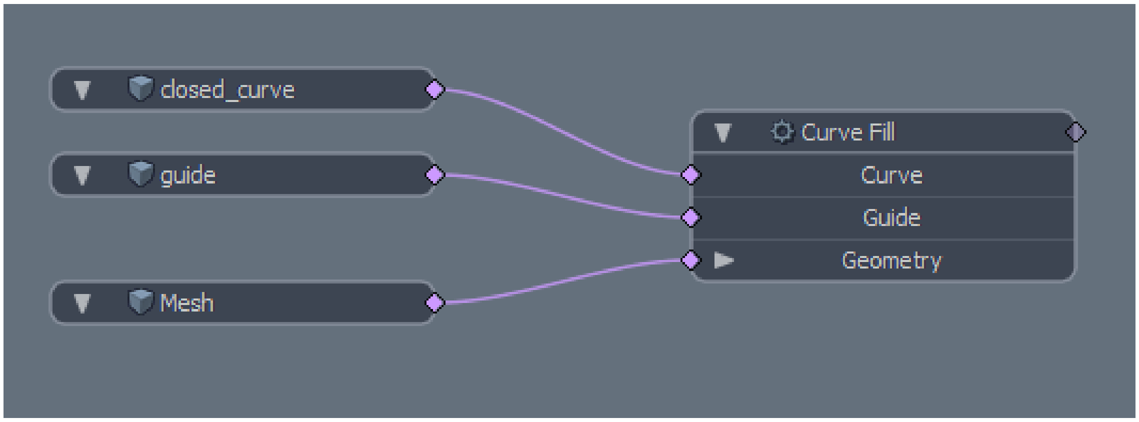
>

MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

To add the Curve Fill node, click **Add...**, then under **Mesh Operations** > **Curves**, double-click **Curve Fill**.

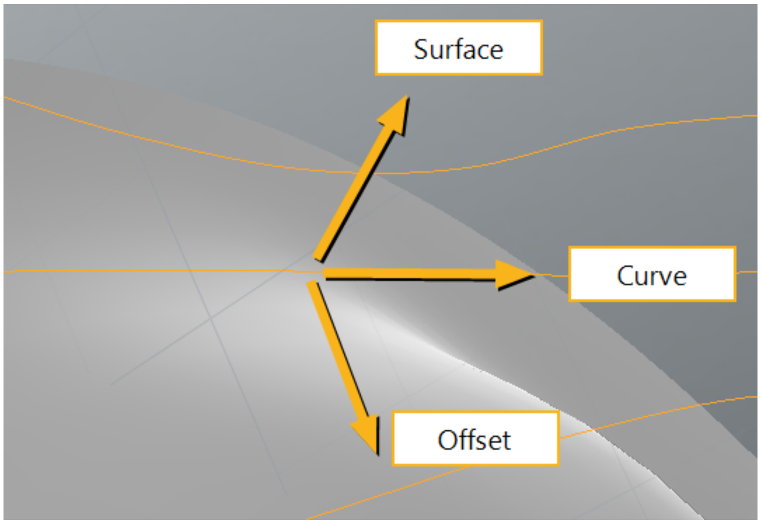
In the following example, the Curve Fill node takes a curve to fill from the mesh layer called **closed_curve**. The guide curve is attached to its **Guide** input, and it generates the geometry in a separate layer called **Mesh**, attached to its **Geometry** input.


closed_curve
guide
Mesh
Curve Fill
Curve
Guide
Geometry

Note: For more information on working in the Schematic, see Schematic Viewport.

Curve Offset by Mesh

This is a procedural tool that creates parallel curves using the curves in the current mesh or curves in a mesh linked to the operation. It uses a separate mesh surface to determine which way to push the curves. Curves are generated in a direction perpendicular to the closest surface normal of the mesh and to the curve tangent or forward direction.

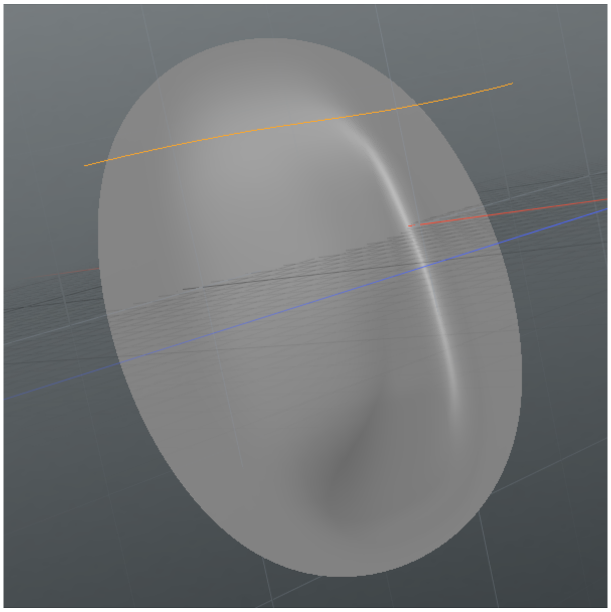


This mesh operation is used by Modo for generating double stitching. For more information, see Stitch Variants.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

To use the Curve Offset by Mesh operation:

Make sure you have your scene prepared with your mesh and your curve. The example below shows a simple scene with the offset mesh in one mesh layer and a curve in another.



1. Select the mesh layer with your curve in it, and open the **Mesh Ops** tab on the right panel.

Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

2. Click **Add Operator**, then under **Curves**, double-click **Curve Offset by Mesh**.

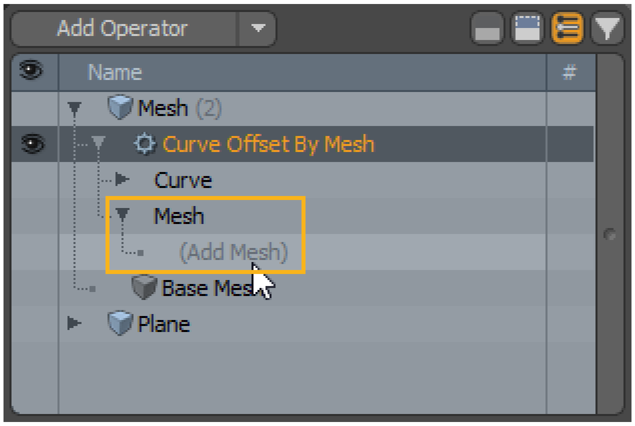
   The **Curve Offset by Mesh** operation is added to the Mesh Operations list.

3. To specify what mesh the operation should use to offset the curve, expand it in the list by clicking the small arrow in front of its name.

   This reveals the mesh operation's inputs: **Curve** and **Mesh**.

   The **Curve** input is optional. If the curve is in a separate mesh layer from the **Curve Offset by Mesh** operation, use this input to specify which one. In this example, we've added the operation to the curve's mesh layer, so you don't need to specify the curve.

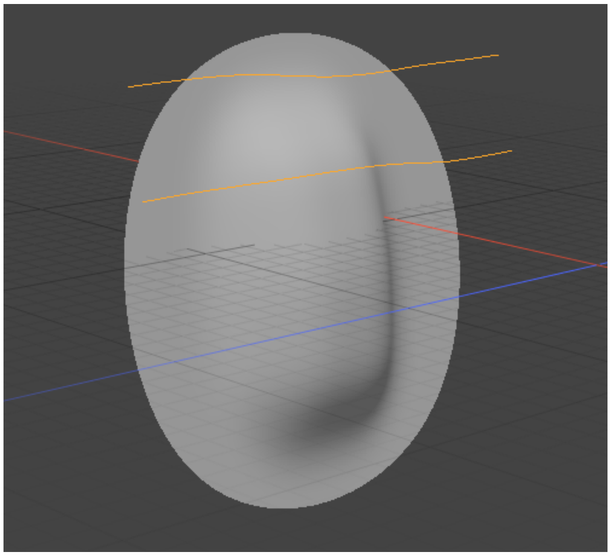
4. Under **Mesh**, click **(Add Mesh)**.



This opens the **Add Operation** dialog.

5. Double-click the mesh you want to use.

   Parallel curves are created for the curve in your mesh. If you select the offset mesh in **Polygons** selection mode and apply transforms to it, the curves shift with it.

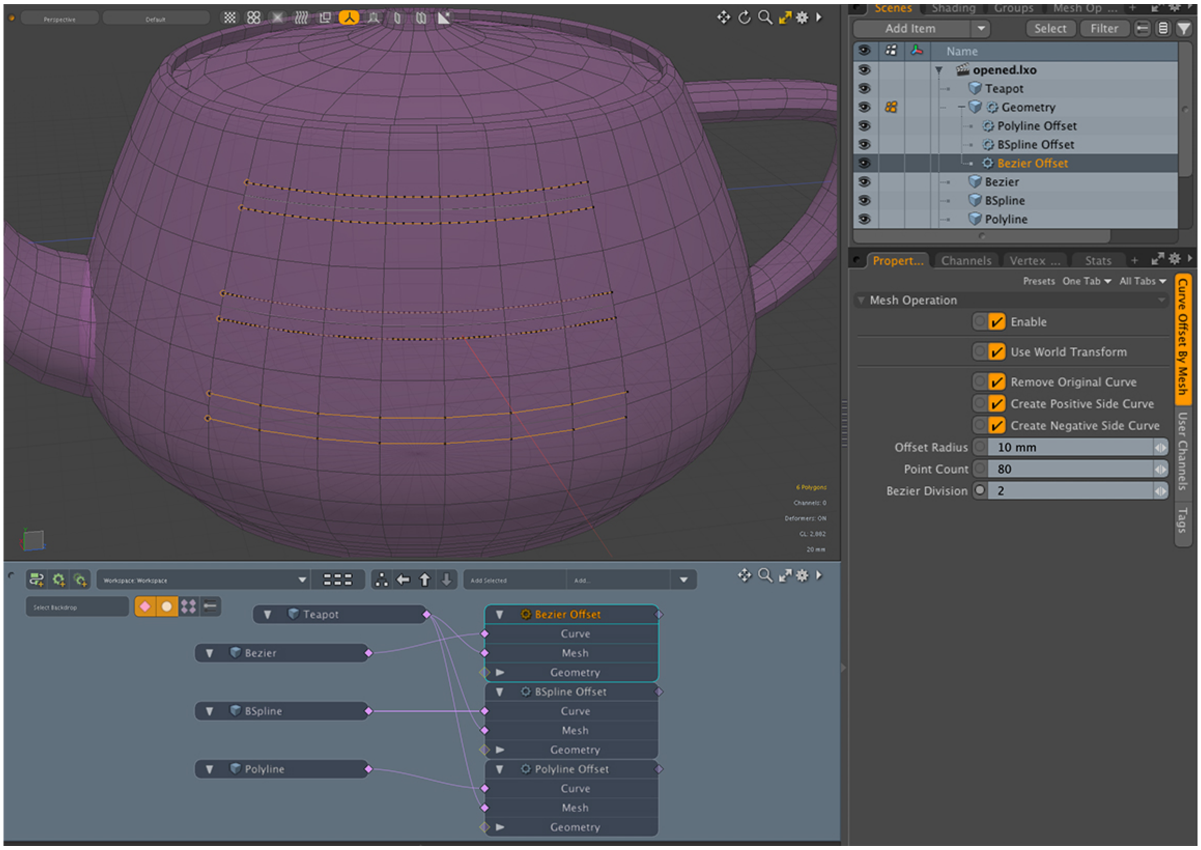


Curve Offset by Mesh Properties

The Curve Offset mesh operation's properties can be found in the **Properties** tab on the lower right panel.

Enable	Toggles curve offset.
Use World Transform	When enabled, the mesh operation maintains the world space on the input layer.
Remove Original Curve	Hide the original central curve. This doesn't affect curves from a different mesh.
Create Positive Side Curve	When enabled, Modo creates the outer curve on the positive side of the central curve.
Create Negative Side Curve	When enabled, Modo creates the outer curve on the negative side of the central curve.
Offset Radius	The distance between the original curve and the new curves.
Point Count	The number of points created on the new curves.

The **Offset Curve by Mesh** operator supports **Bezier**, **B-Spline** and **Polyline** as well as **Curve**. **Curve** and **B-Spline** are resampled by the **Point Count** value. **Bezier Curve** is evenly subdivided by the **Bezier Division** value. **Polyline** is offset to the specified sides without dividing.


Scenes
Shading
Groups
Mesh Op ...
Add Item
Select
Filter
Name
opened.lxo
Teapot
Geometry
Polyline Offset
BSpline Offset
Bezier Offset
Bezier
BSpline
Polyline
Propert...
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Curve Offset By Mesh
User Channels
Tags
Teapot
Bezier
BSpline
Polyline
Bezier Offset
BSpline Offset
Polyline Offset
Curve
Mesh
Geometry

Mesh Operation

☑ Enable

☑ Use World Transform

☑ Remove Original Curve

☑ Create Positive Side Curve

☑ Create Negative Side Curve

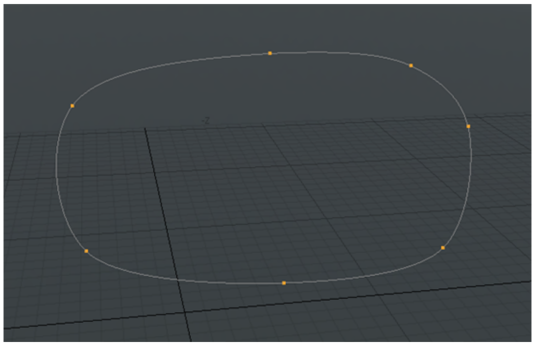
Offset Radius 10 mm

Point Count 80

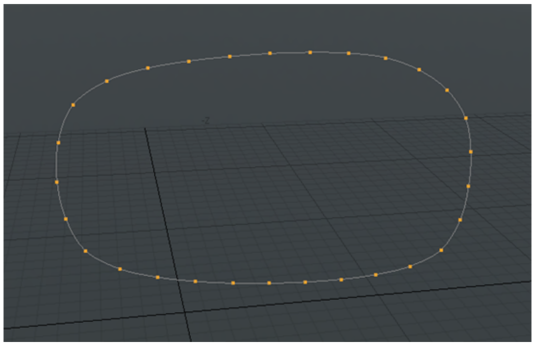
Bezier Division 2

Curve Rebuild

This is a procedural tool that uses a curve to build a new, similar curve but with a different amount of points.



Before



After

Activating the Curve Rebuild Tool

- On the left panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Curves** > **Curve Rebuild**.
- Using the Schematic viewport, click **Add Operator**, and double-click **Mesh Operations** > **Curves** > **Curve Rebuild**.

Applying the Curve Rebuild Tool

The following example stacks a Curve Rebuild tool on top of a mesh containing a curve. You can also add a curve to the tool using the **Curve** setting in the procedural stack.

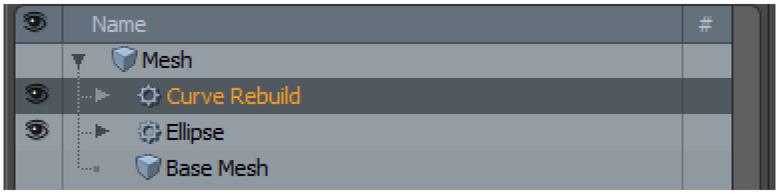
1. Draw a curve using the Adding Curves tool, or add a procedural Curve.

   If you have multiple curve meshes and want to target a certain one, select the curve in the **Item List** or the Mesh Operations List.

2. Click **Add Operator** in the Mesh Operations List.

Note: You can also right-click on the mesh in the 3D viewport and select **Item Operations** > **Add Mesh Operation**.

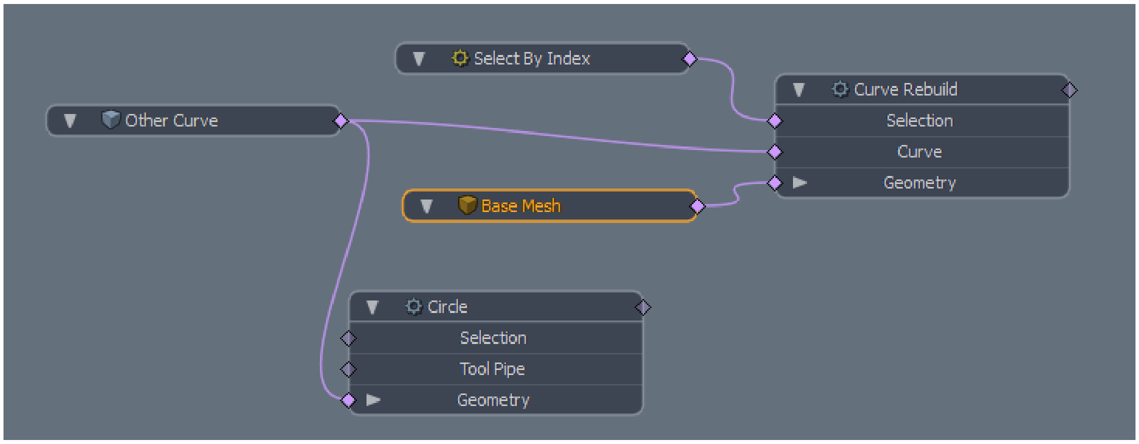
3. Double-click **Curves** > **Curve Rebuild** to add it to the operations stack and rebuild the curve.



4. Adjust the **Mode** and **Point Count** as required.

Applying the Curve Rebuild Tool in the Schematic Viewport

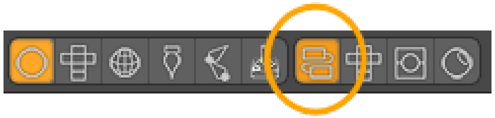
The schematic node has the same three inputs as the procedural version. The image below shows an example.



1. In Model layout, draw a curve using the Adding Curves tool, or add a procedural Curve.

   If you have multiple curve meshes and want to target a certain one, select the curve in the **Item List** or the Mesh Operations List.

2. On the top left corner of the interface, click the Schematic palette icon.



3. In the Schematic viewport, click **Add...** and double-click **Mesh Operations** > **Curves** > **Curve Rebuild**.
4. Drag-and-drop your curve mesh item from the **Mesh Ops** tab onto the Schematic viewport.

   The mesh item is automatically connected to the Curve Rebuild node.

5. On the right panel, select the **Curve Rebuild** layer in the **Mesh Ops** tab, and adjust the properties.

   For more information, see Curve Rebuild Properties.

Schematic sub-tools

The Curve Rebuild tool has the following procedural sub-tools in the Schematic viewport:

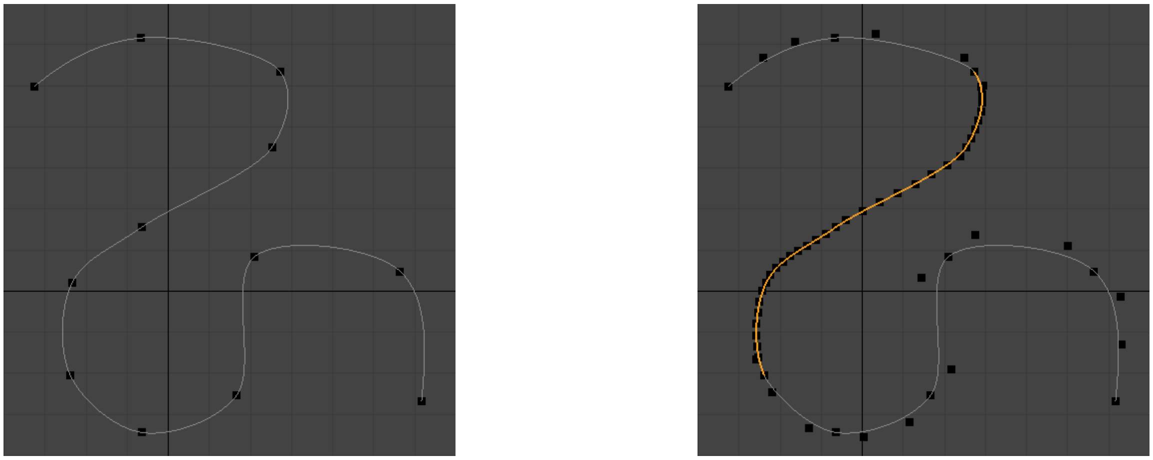
- **Selection** - Allows you to select one or more curves from the input mesh. The selection can be a single figure, or a range such as **0-4**.
- **Curve** - The input mesh to rebuild. This allows you to rebuild a curve on another mesh layer (or from third-party tools that generate curves) to fill.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

Curve Rebuild Properties

Source Curve	The mesh containing the curve(s) to rebuild.
Mode	Sets the point generation method: • **Point Count** - Specifies a discrete number of particles to create along the curve. • **Minimum Distance** - Specifies a minimum distance between particles along the curve. • **Exact Distance** - Specifies an exact distance between particles along the curve. The resulting curve may not be full length.
Point Spacing	Specifies the distance between each particle on the curve. This option is available when **Mode** is set to **Minimum Distance** or **Exact Distance**.

Rebuilding Curve Spans

You can procedurally rebuild sections of a curve using the Curve Rebuild Spans mesh operation. This operation allows you to subdivide and offset the existing spans, the sections between control points on a curve.



Before

After

This way, a curve can be subdivided without altering the existing points. They can remain locked in place while the curve span between them is altered. Curve Rebuild Spans is one of the tools that make up the Lace Item.

Using Curve Rebuild Spans

1. In the Item List, select the mesh containing your curve. If you only want to select certain spans on the curve, use **Edges** selection mode to select the required sections.

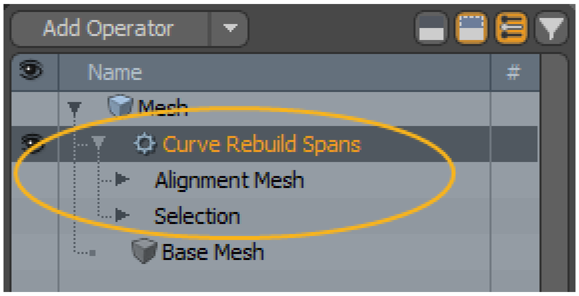
Tip: To help you see the control points on your curve better, do the following: Click the Settings button in the top-right corner of the 3D viewport to open the **3D Viewport Properties**, and in the **Active Meshes** sub-tab, set **Show Vertices** to **Always**. You can also enable **Independent Point Size** and increase the **Point Size** value to make the vertices bigger.

2. Open the **Mesh Ops** tab, and click **Add Operator**.

Note: If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

3. Under **Mesh Operations** > **Curves**, double-click **Curve Rebuild Spans**.

   This adds the Curve Rebuild Spans mesh operation to the Mesh Operations list and opens its properties on the lower right panel.



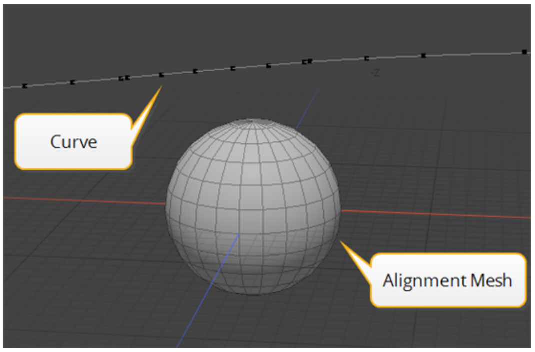
4. Adjust the Curve Rebuild Spans Properties as needed.

Curve Rebuild Spans Inputs

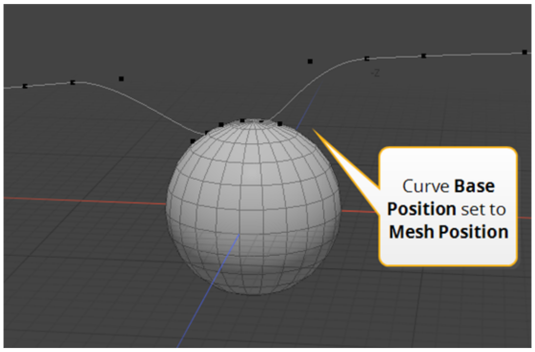
Expanding the mesh operation by clicking the arrow in front of it reveals the operation's inputs.

You can set up the following here:

- **Alignment Mesh** - You can align the newly created curve points to another mesh by specifying one here. The points of the curve then find the nearest positions on the alignment mesh and snap to those. You can adjust the **Base Position**, **Offset**, and **Mesh Control** settings in the Curve Rebuild Spans Properties.



The Sphere is specified as **Aligment Mesh**.



Curve is aligned to the Sphere's surface.

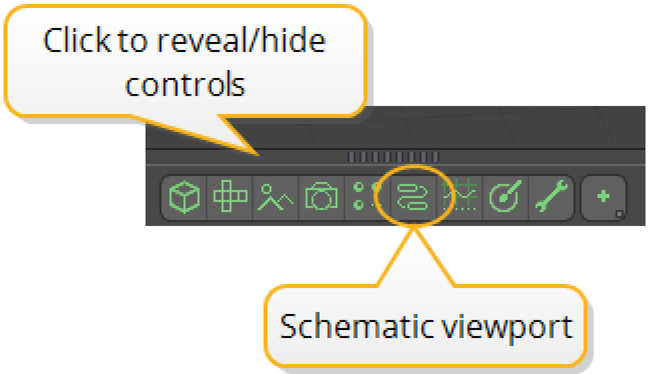
- **Selection** - Select edges or modify your existing selection. For more information on procedural selection, see Procedural Selection.

Curve Rebuild Spans in the Schematic

You can work with the Curve Rebuild Spans mesh operation in the Schematic viewport too.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.



>

MadCap:conditions="Default.NoTranslate">OR

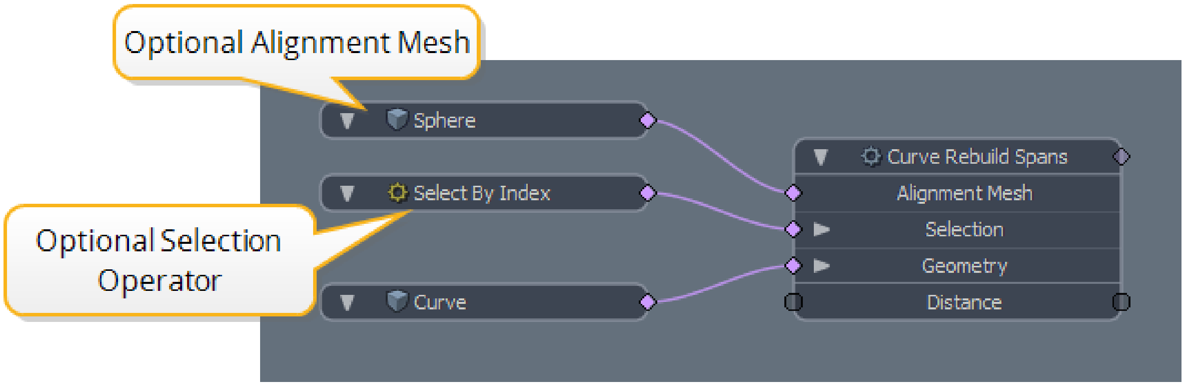
- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

To add the Curve Rebuild Spans node:

1. Click **Add...**, and under **Mesh Operations** > **Curves**, double-click **Curve Rebuild Spans**.

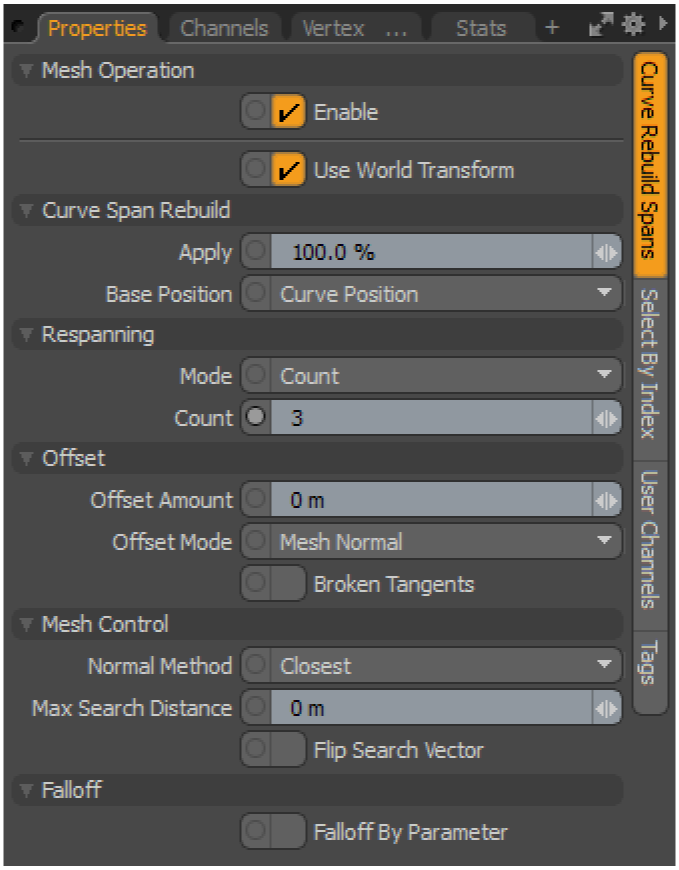
This adds the node to the viewport and opens its properties on the lower right panel.

2. Connect the curve mesh to the Curve Rebuild Spans node's **Geometry** input.
3. If you're using an Alignment Mesh, or a Selection operator, connect them to the Alignment Mesh and Selection inputs, respectively.


Optional Alignment Mesh
Optional Selection Operator
Sphere
Select By Index
Curve
Curve Rebuild Spans
Alignment Mesh
Selection
Geometry
Distance

4. Adjust the Curve Rebuild Spans properties as needed.

Curve Rebuild Spans Properties


Properties
Channels
Vertex ...
Stats
+
Curve Rebuild Spans
Select By Index
User Channels
Tags

Mesh Operation

☑ Enable

☑ Use World Transform

Curve Span Rebuild

Apply: 100.0 %

Base Position: Curve Position

Respanning

Mode: Count

Count: 3

Offset

Offset Amount: 0 m

Offset Mode: Mesh Normal

☐ Broken Tangents

Mesh Control

Normal Method: Closest

Max Search Distance: 0 m

☐ Flip Search Vector

Falloff

☐ Falloff By Parameter

Curve Span Rebuild

Apply	Changing the **Base Position** to use an attached mesh or altering the offset amount moves the newly created points off the existing curve. The **Apply** value controls how much that offset is applied. 0% means that any new points will lie on the existing curve span, while 100% means any potential offsets are fully applied to the new point positions.
Base Position	Determines the non-offset position of the newly created curve points. The options are: • **Curve Position** - All new points are created along the existing curve. • **Mesh Position** - If an input mesh is connected to the mesh operation's **Alignment Mesh** input, the new points can be created on the surface of the connected mesh. The location on the mesh surface is determined by the Mesh Control options.

Respanning

Mode	Determines how new points are created along the existing spans. The options are: • **Count** - New spans are created by subdividing each selected span equally a discrete number of times. • **Distance** - Each span is split until the distance between each new span is less that the specified **Distance** value.
Count	The number of new spans to create for each selected span when **Mode** is set to **Count**.
Distance	The maximum length of a span before it is subdivided when **Mode** is set to **Distance**.

Offset

Offset Amount	The distance by which the points are pushed.

Offset Mode	Specifies the way of determining the direction of the offset. The options are: • **Mesh Normal** - Pushes spans out in the direction of the mesh normal found through the Mesh Control settings. • **Vector** - Pushes spans out in a specific X,Y,Z vector direction. • **Curve Normal** - Finds the orientation of all points on the original curve and pushes the spans in that direction.
Broken Tangents	Tangent handles on either side of a new point may be offset by different amounts, resulting in broken tangent handles. By default, the offsets are averaged to provide a smooth curve point, but enabling **Broken Tangents** turns this behavior off.

Mesh Control

Normal Method	Determines the way the location on the input mesh surface is found. The position/normals from the points on the mesh are then used for the **Base Position** or **Mesh Normal** offset modes. The options are: • **Closest** - Curve points are offset to the closest point on the surface to the new curve point. The new points find their closest matched surface location. • **X, Y, Z Axis** - Curve points find the nearest surface toward the specified axis. • **User Vector** - Identical to the X,Y, and Z axis options, but rather than using a standard direction, you can specify a vector. • **Curve Normal** - Similar to **User Vector**, but the vector is determined by the normal of the original curve.
Max Search Distance	When **Normal Method** is set to **Closest**, this is the maximum distance away from the new points the mesh can be found.
Flip Search Vector	When enabled, Modo looks for geometry in the negative direction. For instance, if the **Normal Method** is set to **Y Axis**, Modo looks for the first geometry from the positive Y position down through the curve.

Falloff

Falloff By	The push amount decreases as the distance of a new point to an original curve

Parameter	point increases.
Falloff Mode	When **Falloff By Parameter** is enabled, here you can specify the Falloff type. The available options are: • **Linear** • **Ease In** • **Ease Out** • **Smooth**

Curve Sweep

The Curve Sweep mesh operation is similar to Curve Extrude, but provides better control and a more streamlined workflow for procedurally extruding geometry along a curve. It is most commonly used when working with the Lace Item to generate lacing.

Note: For more information on lacing, see Lace Item.

Using Curve Sweep

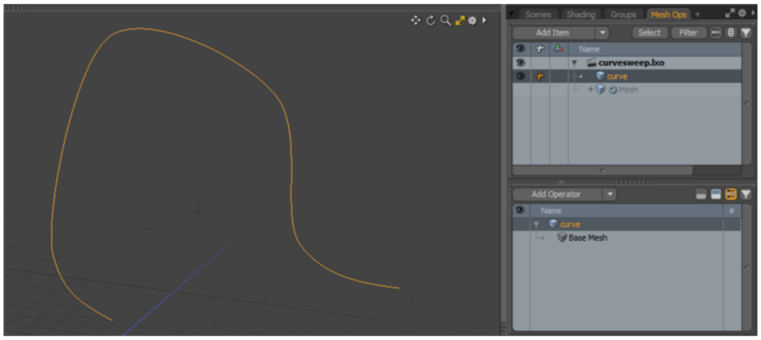
To use the tool, you need at least two mesh layers for the following:

- The curve used as the path
- The Curve Sweep operation and the geometry used as the shape for the sweep operation.

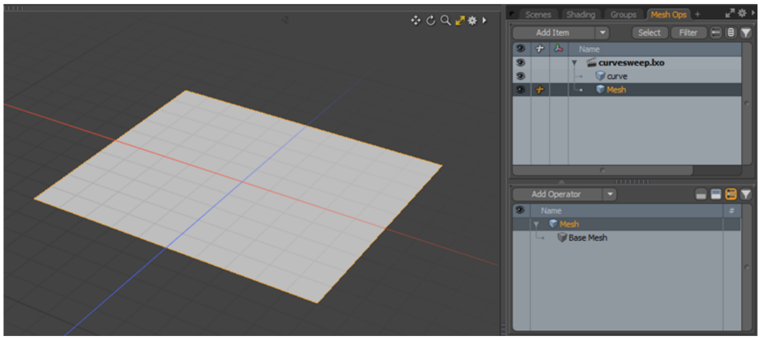
The shape geometry can also be located in a third, separate mesh layer, or can be a preset.

Tip: To add a new mesh layer, press the **N** key, or in the Item List, click **Add Item**, and under **Items**, double-click **Mesh**.

1. Create a curve in an empty mesh.


Scenes
Shading
Groups
Mesh Ops
Add Item
Select
Filter
Name
curvesweep.lxo
curve
Mesh
Add Operator
Name
#
curve
Base Mesh

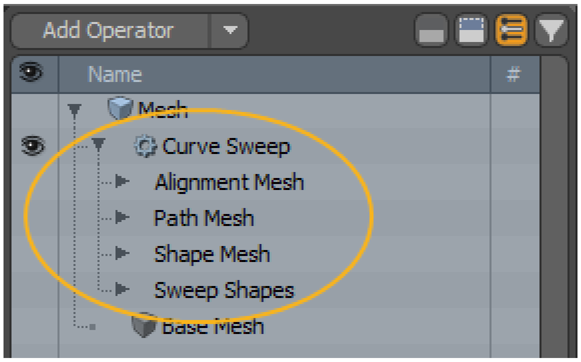
2. Add a new mesh, or select one in the Item List.
3. Add the geometry you want to use to sweep along the curve. For example, a square.


Scenes
Shading
Groups
Mesh Ops
Add Item
Select
Filter
Name
curvesweep.lxo
curve
Mesh
Add Operator
Name
#
Mesh
Base Mesh

4. Staying in the same mesh where you've just added your geometry, in the **Mesh Ops** tab, click **Add Operator**, and under **Mesh Operations** > **Curves**, double-click **Curve Sweep**.

Note: If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

The Curve Sweep mesh operation is added to the Mesh Operations list. And its properties open on the lower right panel.


Add Operator
Name
#
Mesh
Curve Sweep
Alignment Mesh
Path Mesh
Shape Mesh
Sweep Shapes
Base Mesh

Expanding the mesh operation by clicking the arrow in front of it reveals the operation's inputs.

You can set up the following here:

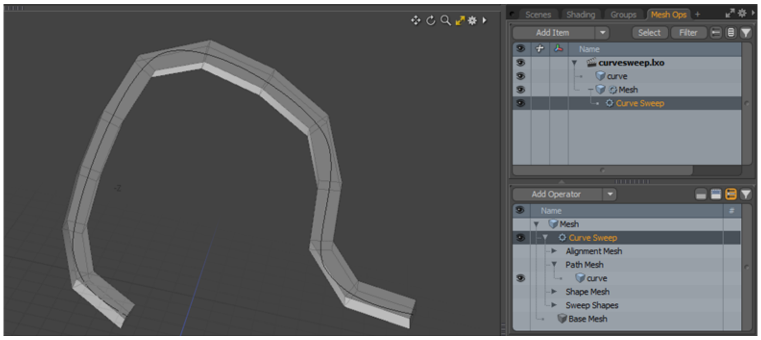
- **Alignment Mesh** - You can use geometry in another mesh to determine the alignment of normals along the curve. The mesh containing the geometry can be added here.
- **Path Mesh** - The path to sweep along. This is the mesh containing your curve.
- **Shape Mesh** - The shape to use for the sweep operation. This is the mesh containing the geometry you want to duplicate along the path. If your geometry is in the same mesh layer as the Curve Sweep operation, you don't need to specify this.
- **Sweep Shapes** - You can use more than one shape for the sweep operation. This input allows you to specify additional shapes.

5. In the Curve Sweep operation's **Properties** panel, set **Path** to the mesh where your curve is.

   If you have multiple curves in your mesh and you only want to use some of them, specify the **Curve Index**. For example, if you have four curves in your mesh, but you only want the first two to be affected by the Curve Sweep, enter '0,1'.

6. Make sure that **Extrude Shape** is set to **Current Mesh**.

   The geometry is duplicated along the length of the curve.

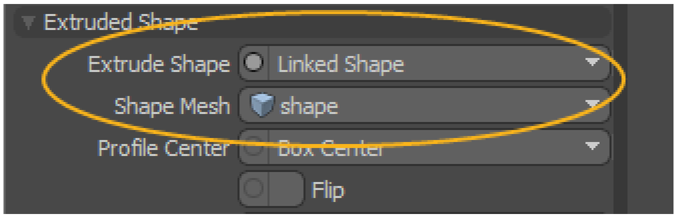

Scenes
Shading
Groups
Mesh Ops
Add Item
Select
Filter
Name
curvesweep.lxo
curve
Mesh
Curve Sweep
Add Operator
Name
Mesh
Curve Sweep
Alignment Mesh
Path Mesh
curve
Shape Mesh
Sweep Shapes
Base Mesh

For more information on other properties you can adjust, see Curve Sweep Properties.

Using a Shape in a Separate Mesh Layer

If the shape to duplicate is not in the same mesh layer as the Curve Sweep operation, you need to specify the mesh in the Curve Sweep **Properties**.

Under **Extruded Shape**, set **Extrude Shape** to **Linked Shape**, and **Shape Mesh** to the mesh layer where your shape is.


Extruded Shape
Extrude Shape: Linked Shape
Shape Mesh: shape
Profile Center: Box Center
○ Flip

The geometry in the specified mesh is duplicated along the length of the curve.

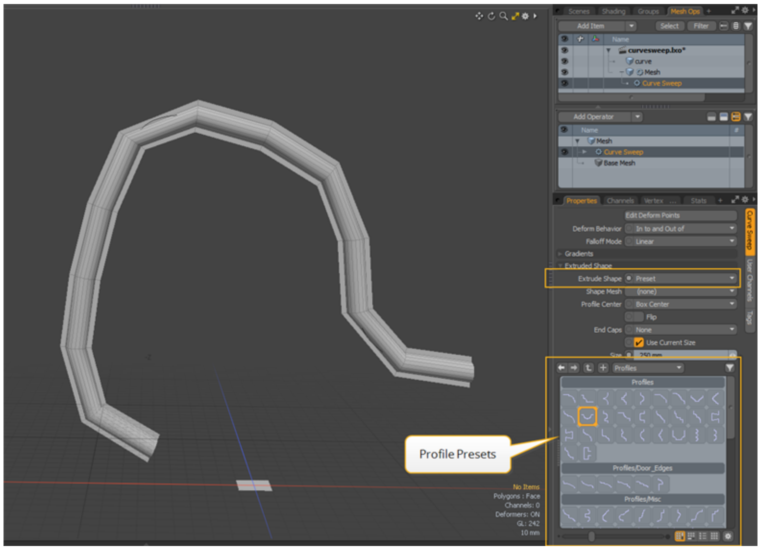
Using a Preset as Shape

In addition to using geometry, you can use one of the Profile presets to sweep along the curve. Once you've added the operation and specified the curve, follow these steps:

1. In the Curve Sweep **Properties**, set **Extrude Shape** to **Preset**,

   A Profiles preset browser opens below the properties.

2. Click one of the presets.



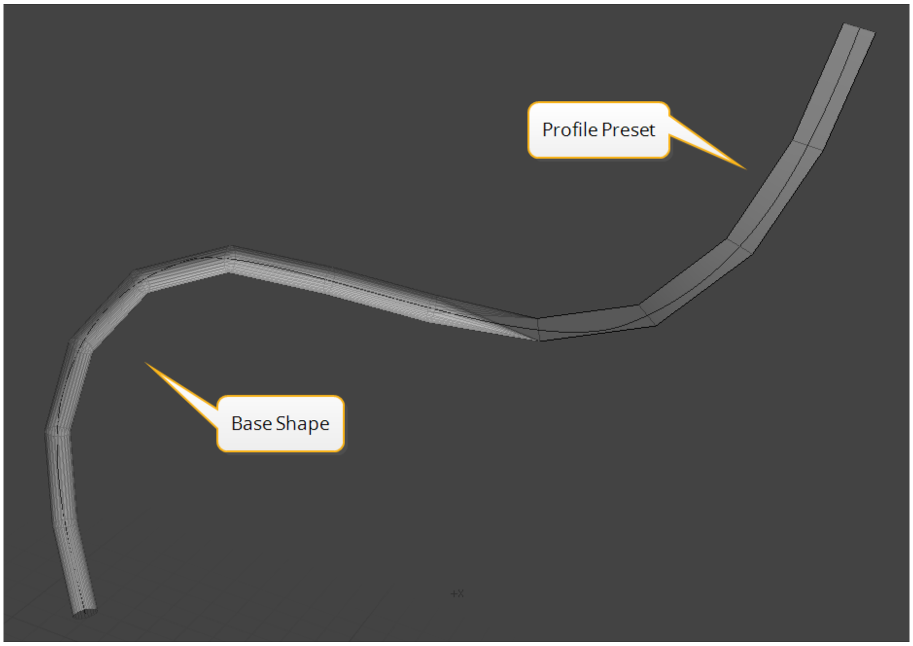
Deform Behavior: In to and Out of
Falloff Mode: Linear
Extrude Shape: Preset
Shape Mesh: (none)
Profile Center: Box Center
End Caps: None
☑ Use Current Size

Profile Presets
Mesh
Curve Sweep
Base Mesh
Properties
Edit Deform Points
Gradients
Extruded Shape
Flip
Profiles
Profiles/Door_Edges
Profiles/Misc

The selected shape is duplicated along the length of the curve.

Using Multiple Shapes

The Curve Sweep mesh operation allows you to use multiple shapes to sweep along different sections of the curve, and blend from one shape into another.


Profile Preset
Base Shape

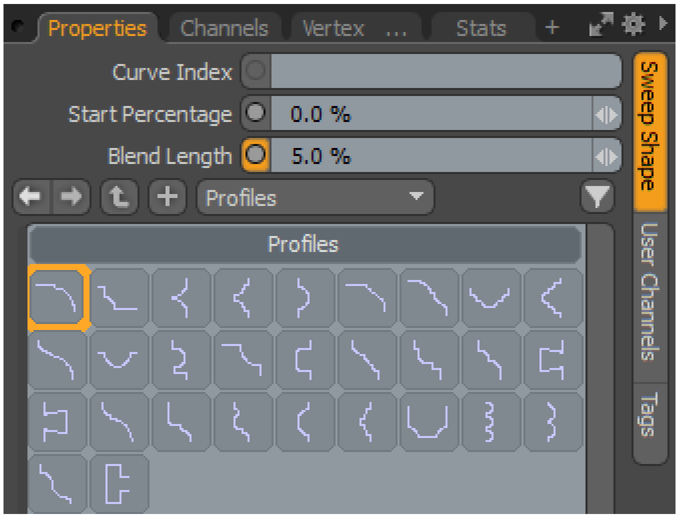
Once you've added the operation and specified the curve and base shape, follow these steps:

1. In the Mesh Operations list, expand the **Curve Sweep** operation, then expand its **Sweep Shapes** input and click **(Add Sweep Shapes)**.

   This opens the **Add Operation** dialog.

2. Under **New**, double-click **Sweep Shape Input**.

   The Sweep Shape Input operation is added to the Mesh Operations list, and its properties open in the **Properties** panel.


Properties
Channels
Vertex ...
Stats
+
Sweep Shape
User Channels
Tags
Profiles
Profiles

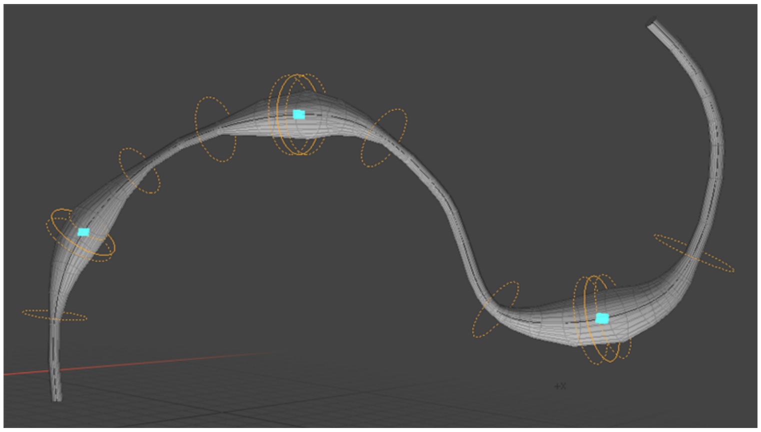
Curve Index
Start Percentage 0.0 %
Blend Length 5.0 %

3. Select a Profile preset, and specify the following:

Curve Index	Specifies the curve to be used. If you have only one curve in your scene, you don't need to use this.
Start Percentage	The point along the curve where Modo starts using the selected shape instead of the base shape.
Blend Length	Specifies how far along the curve the two shapes are blended. The lower the value, the more abrupt the change is from one shape to the other.

You can add as many Sweep Shape Inputs as you like.

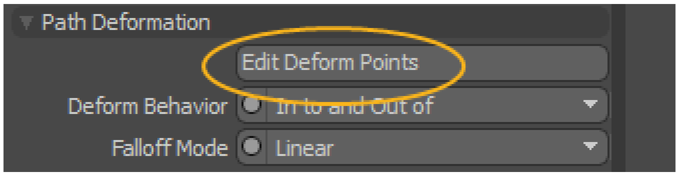
Adding and Editing Deform Points



You can deform points of the swept curve using the Edit Deform Points tool, located in the Curve Sweep operation's **Properties** panel.

To use the tool:

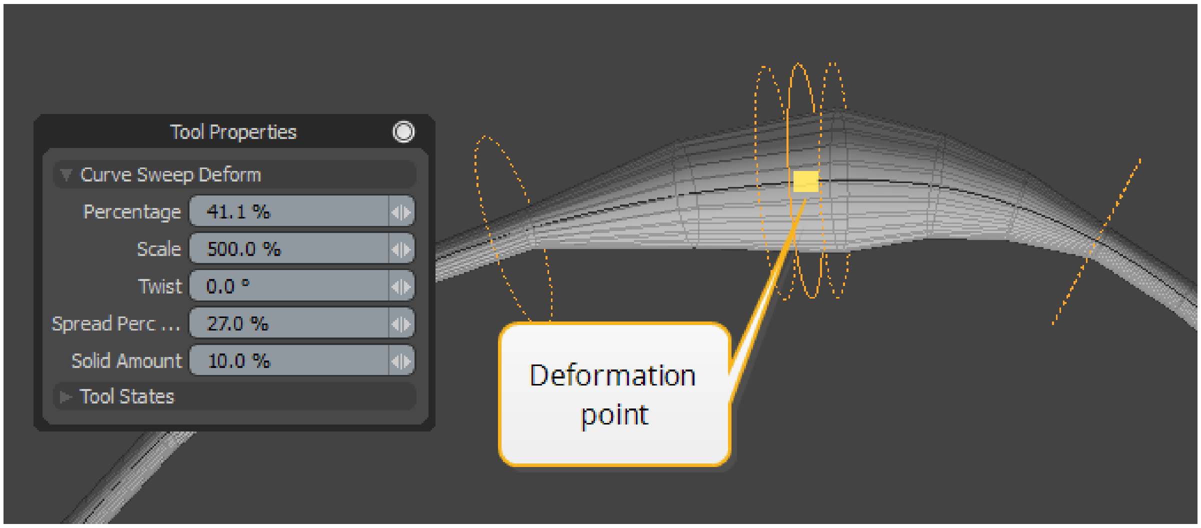
1. Add and set up your Curve Sweep mesh operation, as described in Using Curve Sweep.
2. Click **Curve Sweep** in the Mesh Operations list to open its properties in the **Properties** panel.
3. Under **Path Deformation**, click **Edit Deform Points**.



Deform Behavior: In to and Out of

Falloff Mode: Linear

4. The Tool Properties appear on the left panel. If you're working in the **Modo** layout, click the **Tool Properties** button or press **K** to reveal them.
5. **Shift**+click on the curve to place deformation points.



Tool Properties

Curve Sweep Deform

Percentage: 41.1 %

Scale: 500.0 %

Twist: 0.0 °

Spread Perc ...: 27.0 %

Solid Amount: 10.0 %

Tool States

6. Edit the deformation values in the **Tool Properties** or in the 3D viewport.

Property	Keyboard Shortcut	Description
Percentage	Click-and-drag	The position of the deformation point along the curve.
Scale	Right-click-and-drag	The strength of the deformation.
Twist	**Ctrl**+right-click-and-drag	The rotation around the sweep path's tangent applied to the extruded geometry.
Spread Percent	**Ctrl**+click-and-drag	The range of the deformation's effect.
Solid Amount		The amount of the spread area that the deformation is applied to at 100%. Anything outside of this range is blended into or out of based on the falloff and deformation mode settings.

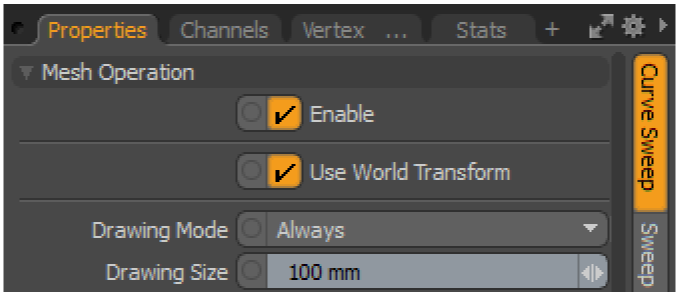
7. Adjust the **Path Deformation** options in the Curve Sweep **Properties**, if needed. For more information on these options, see Path Deformation.

You can remove a deformation point by right-clicking it.

Curve Sweep Properties

Selecting the Curve Sweep mesh operation in the Mesh Operations list opens its properties on the lower right panel.

Mesh Operation



Mesh Operation

☑ Enable

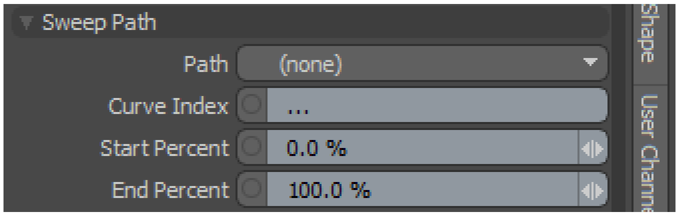
☑ Use World Transform

Drawing Mode: Always

Drawing Size: 100 mm

Enable	Enables or disables the mesh operation.
Use World Transform	Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Drawing Mode	When deformation has been applied along the sweep, these points can be drawn in the viewport. This option specifies when to draw the deformation points. You can set it to **Always**, only when they are **Selected**, or disable it completely by selecting **None**.
Drawing Size	Specifies the size of the drawn deformation controllers.

Sweep Path



Sweep Path

Path: (none)

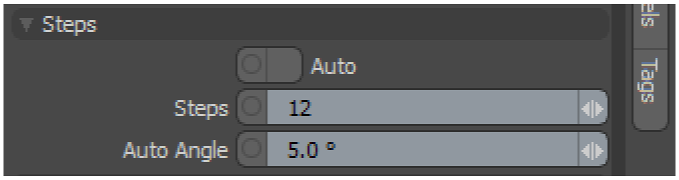
Curve Index: ...

Start Percent: 0.0 %

End Percent: 100.0 %

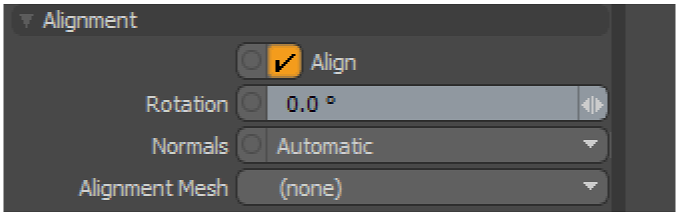
Path	The mesh containing the curve or curves to sweep along.
Curve Index	The index of the curves to sweep along. If you only have one curve in your mesh layer, you don't need to specify this.
Start Percent	The percentage along the curve to start creating geometry.
End Percent	The percentage along the curve to stop creating geometry.

Steps



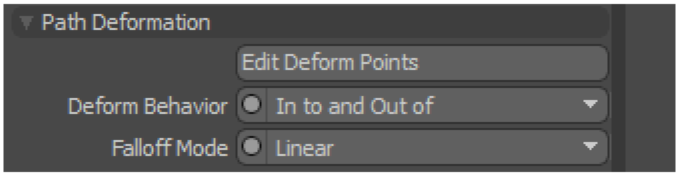
Auto	Automatically determines the number of steps to use based on the **Auto Angle**.
Steps	When **Auto** is disabled, determines the number of steps along the curve path, controlling the number of segments of the resulting sweep.
Auto Angle	The maximum amount of curve along the sweep path allowed without adding a step when **Auto** is enabled.

Alignment

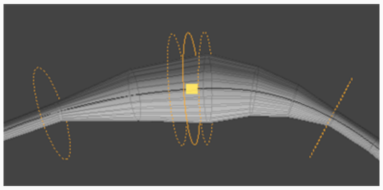
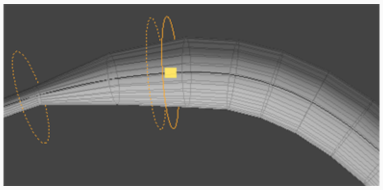


Align	When enabled, Modo aligns the swept geometry to the path.
Rotation	Rotate the geometry around the tangents along the path by the specified amount.
Normals	Specifies the way the path normals are determined. • **Automatic** - Generates smoothly interpolating normals along the path. • **Mesh Normals** - Uses the polygon normal from an alignment mesh. • **Curve Direction** - Aligns the normals towards the curve found in the selected alignment mesh. The normal can be the direction to the closest point on the other curve, or the direction to the point on the other curve with an identical percentage along the curve. • **Path Normal** - Aligns normals to the average normal of the path curve.
Alignment Mesh	Specifies the mesh used when **Normals** is set to **Mesh Normals** or **Curve Direction**.

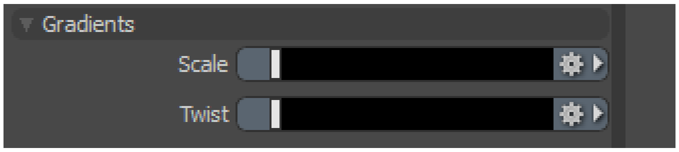
Path Deformation



Deform Behavior: In to and Out of
Falloff Mode: Linear

Edit Deform Points	Activates the Sweep Deformation tool. For more information on how to use the tool, see Adding and Editing Deform Points.
Deform Behavior	Specifies the way the deformation is applied. • **In to and Out of** - Blends from undeformed into deformed, then back to undeformed based on the **Spread Percent**. • **Blend In to** - Blends into the deformed shape and propagates the effect along the rest of the curve.  In to and Out of  Blend In to
Falloff Mode	The falloff shape when blending between deformed and undeformed geometry. The available options are: • **Linear** • **Ease In** • **Ease Out** • **Smooth**

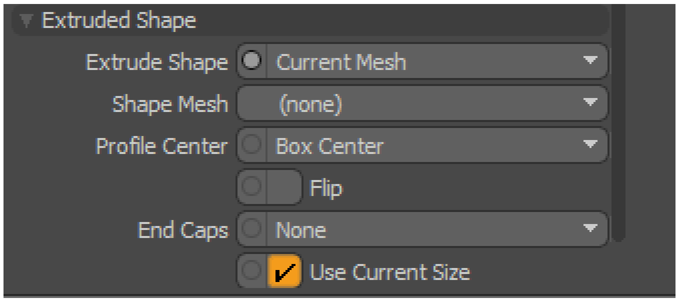
Gradients



Scale	Drives the scale of the extruded geometry over the length of the curve using a gradient.
Twist	Drives the twist of the extruded geometry over the length of the curve using a gradient.

Note: For more information on working with gradients, see Gradient Editor.

Extruded Shape


Extruded Shape

Extrude Shape: Current Mesh

Shape Mesh: (none)

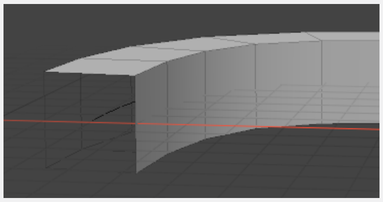
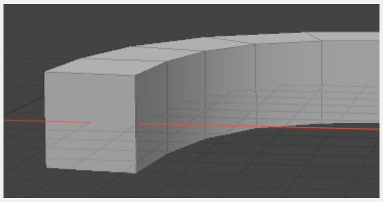
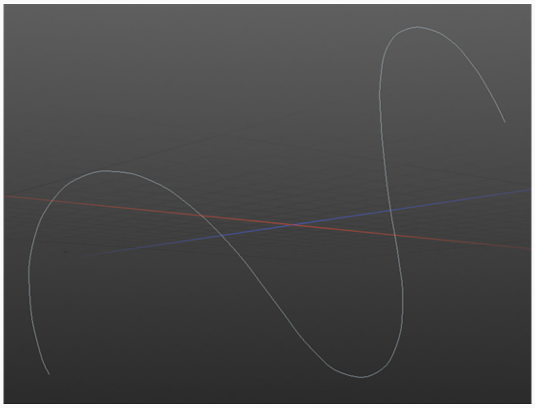
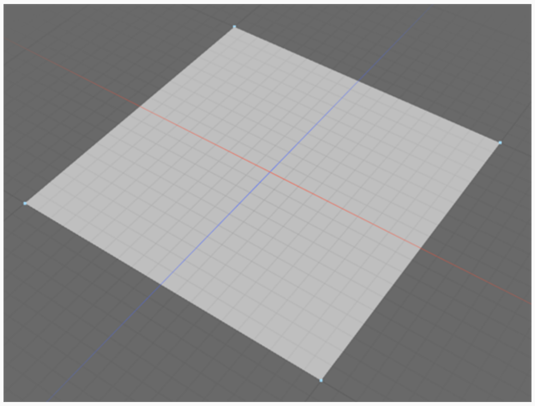
Profile Center: Box Center

☐ Flip

End Caps: None

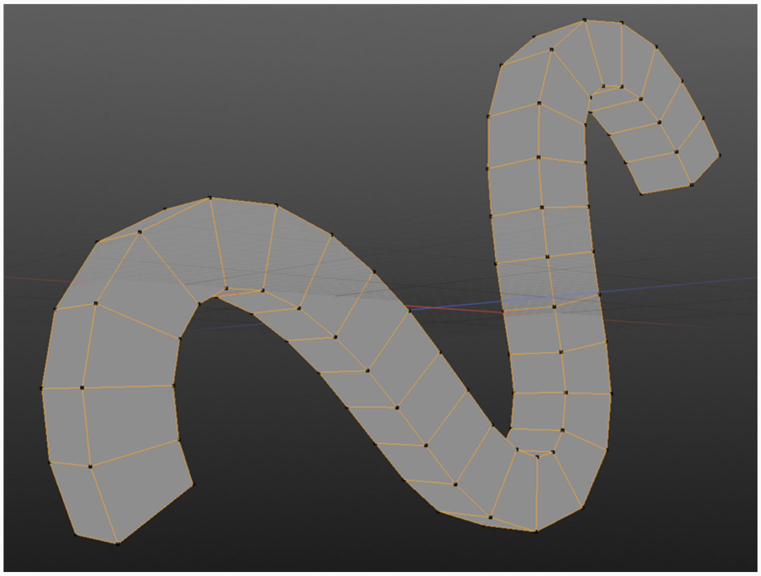
☑ Use Current Size

Extrude Shape	The base shape to extrude. • **Current Mesh** - The mesh where the Curve Sweep operation is located. • **Linked Shape** - A different mesh linked through the **Shape Mesh** input. For more information, see Using a Shape in a Separate Mesh Layer. • **Preset** - A Profile preset. Selecting this option opens a preset browser in the panel. For more information, see Using a Preset as Shape. • **Polygon Strips** - Uses strips of polygons to sweep along the curve.
Shape Mesh	When **Extrude Shape** is set to **Linked Shape**, this controls which mesh is used.
Profile Center	The bounding box pivot of the shape to extrude.
Flip	When enabled, inverts the faces of the generated polygons.
End Caps	Creates closed polygons on the ends of the sweep.

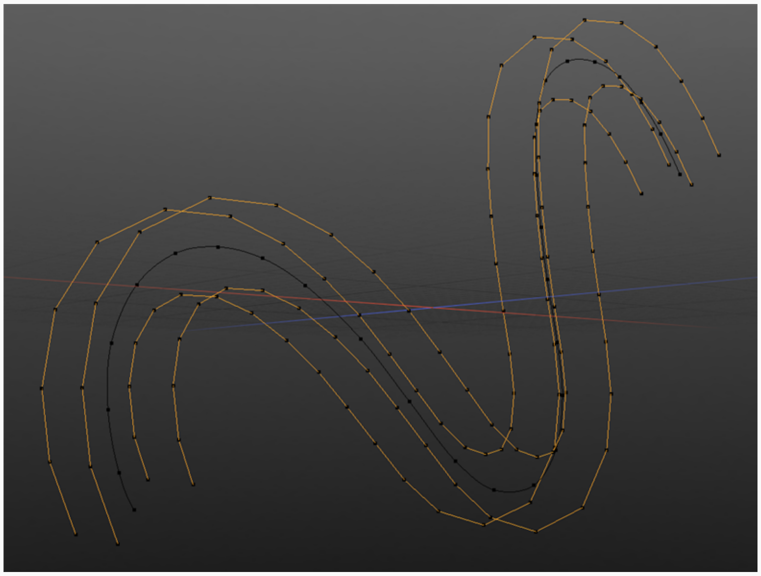
	 No End Caps  End Caps The available options are: • **None** - Both ends are left open. • **Start** - Closing polygons are created at the start. • **End** - Closing polygons are created at the end. • **Both** - Closing polygons are created on both ends. **Note:** The polygon faces might be inverted. In such cases, enable **Flip** to get the expected result.
Use Current Size	Use the current size of the shape to extrude.
Size	Set a different size for the extruded geometry if **Use Current Size** is disabled.
Point Mode	Point Mode can be used to extrude vertices along a curve to produce multiple edge loops, in the form of curves or PolyLines, from a single source mesh. This makes it easy to create a series of edge loops which follow the same curve path. Each point on the source mesh generates an individual edge loop.  Example curve  Example source mesh

The available options are:

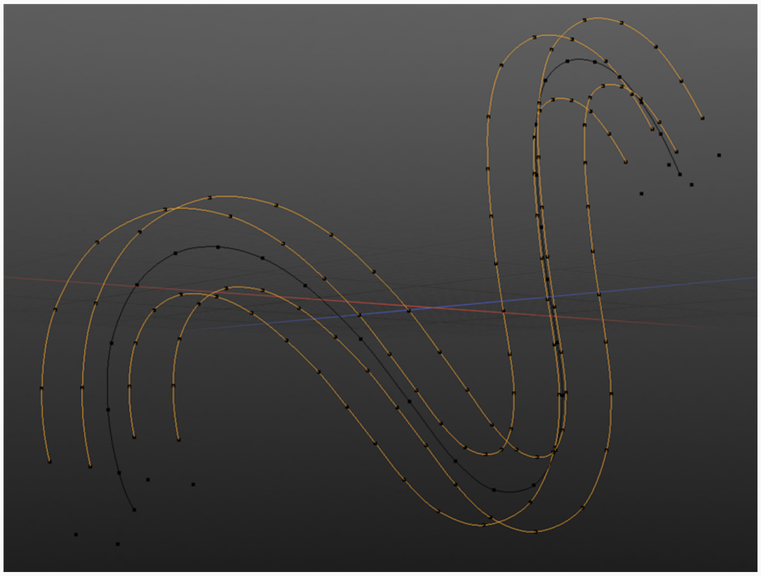
- **None** - The extrusion is applied to the geometry.



- **PolyLines** - Uses PolyLine geometry to create an extrusion from each point, which follows straight lines between vertices along the curve path.

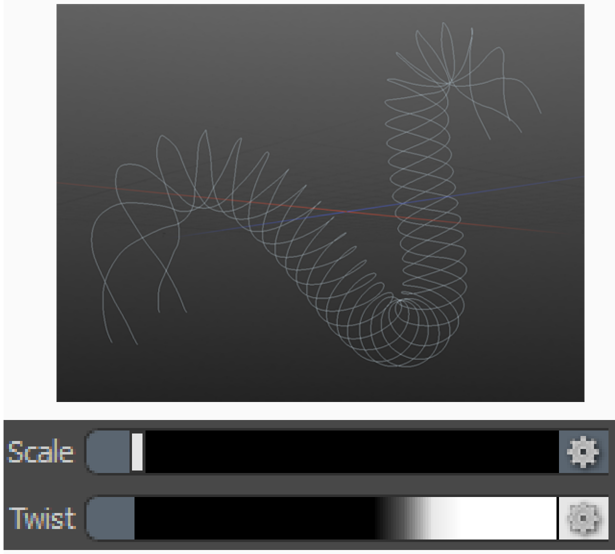


- **Curves** - Uses Curve geometry to create an extrusion from each point, which follows the curve exactly.

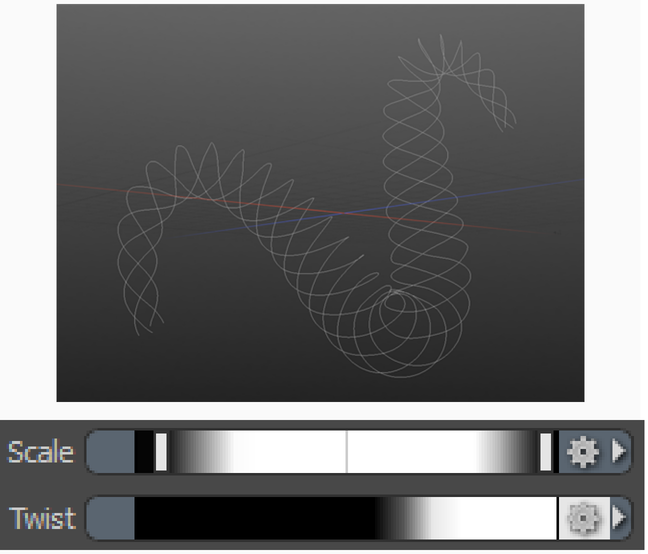


Note: When using the **PolyLines** or **Curves** options, the **Flip** and **End Caps** properties have no result as there are no polygons to affect.

The **Twist** and **Scale** Gradients properties can be adjusted to create a variety of different effects, for example, coil shapes for wires or springs.

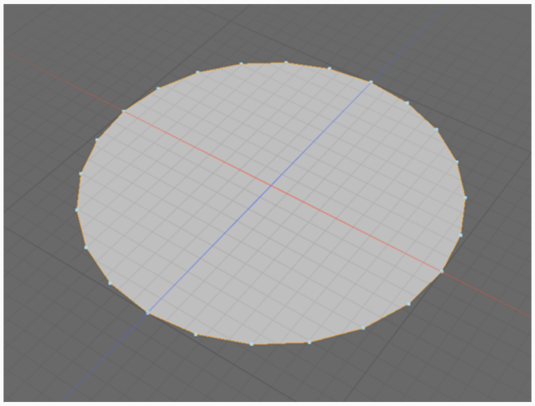


Twist gradient applied

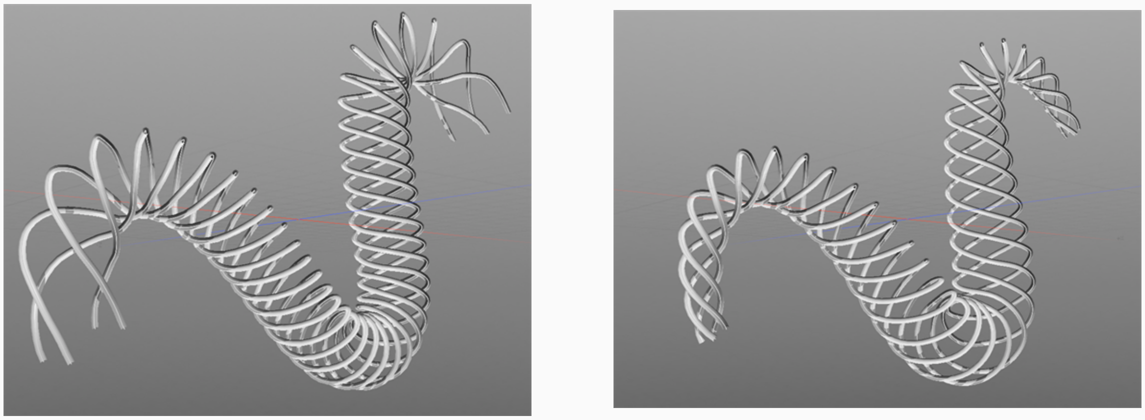


Twist and **Scale** gradient applied

The resulting PolyLines or Curves shown above can then be extruded again using the **Curve Sweep** tool to produce a mesh.

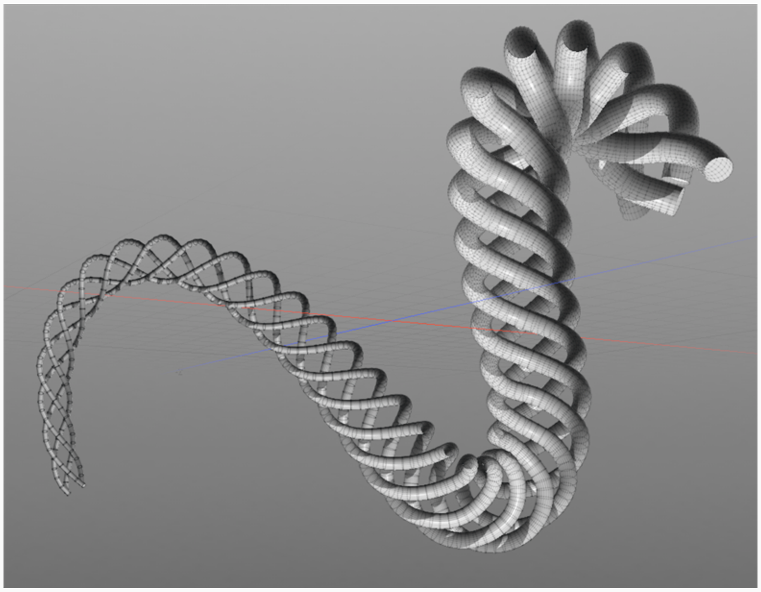


Source mesh



Curve Sweep applied to the source mesh with **Point Mode** set to **None**.

The **Twist** and **Scale** gradients can then be used on the second **Curve Sweep** to further customize the effect.



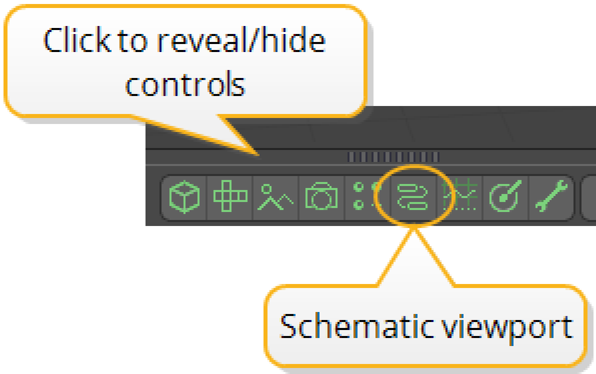
Scale gradient applied to second **Curve Sweep**

Curve Sweep in the Schematic

You can work with the Curve Sweep mesh operation in the Schematic viewport too.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.



>

MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

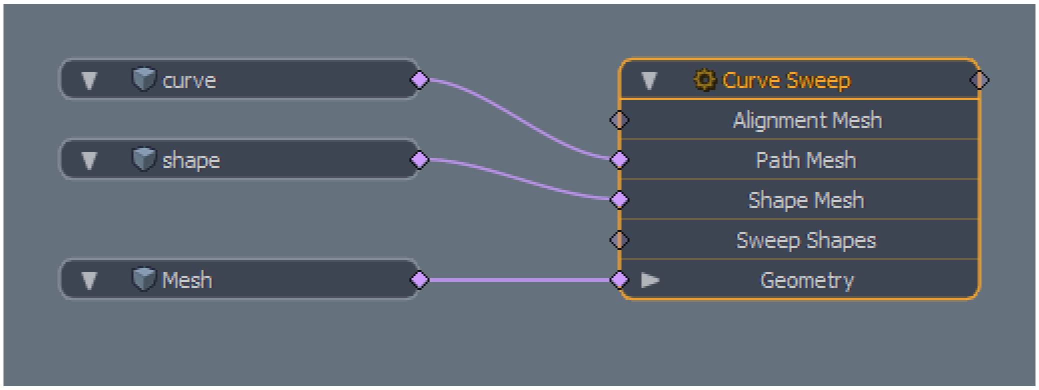
To add the Curve Sweep node:

1. Click **Add...**, and under **Mesh Operations** > **Curves**, double-click **Curve Sweep**.

   This adds the node to the viewport and opens its properties on the lower right panel.

   In the Item List, select the mesh layer containing your curve and the one containing the shape geometry if it's separate.
2. In the Schematic viewport, click **Add Selected**.

   Your meshes are added to the Schematic viewport.
3. Connect the curve mesh to the Curve Sweep node's **Path Mesh** input, and the shape mesh to its **Shape Mesh** input (if it's separate).

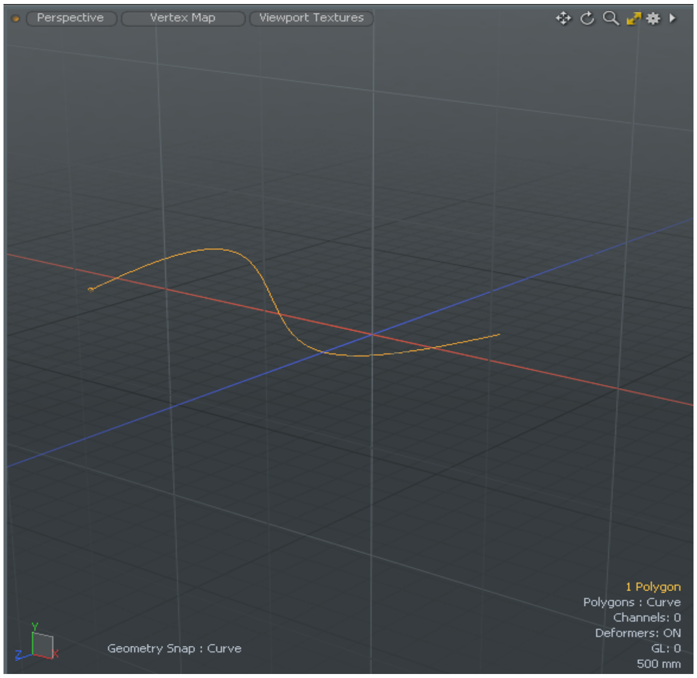


4. Adjust the Curve Sweep properties as needed.

Open Curve Offset

For procedural modeling, use the **Open Curve Offset** and **Open Curve Offset Trio** tools to create additional open curves parallel to an existing curve.

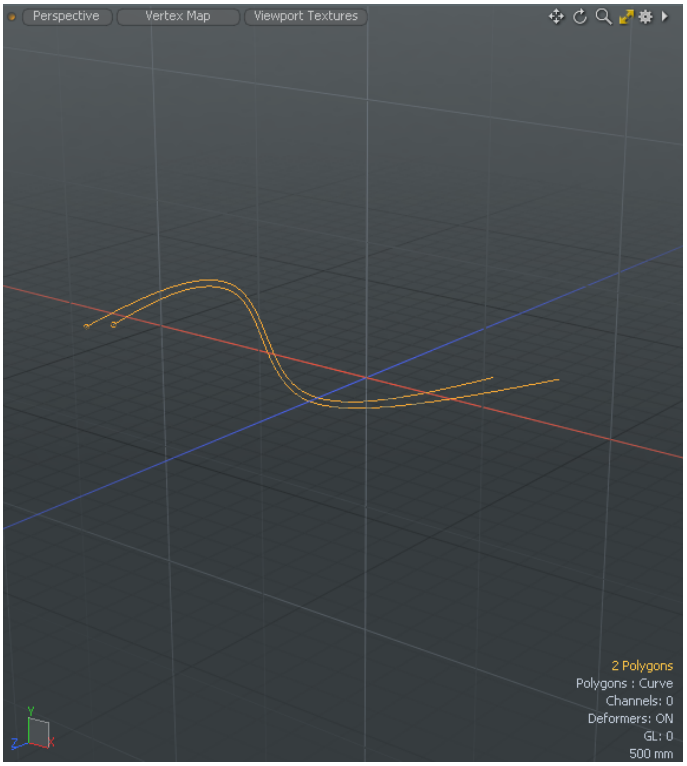
1. In **Model** layout, add a curve to your scene.


Perspective
Vertex Map
Viewport Textures
1 Polygon
Polygons : Curve
Channels: 0
Deformers: ON
GL: 0
500 mm
Geometry Snap : Curve
Y
Z
X

2. Activate the **Polygon** selection mode, and double-click on the curve.
3. With the curve selected, on the right panel open the **Curves** tab, and click **Open Curve Offset**

Tip: If this option is not visible in the left panel. click and hold **Open Curve Offset** and click **Open Curve Offset Trio**.

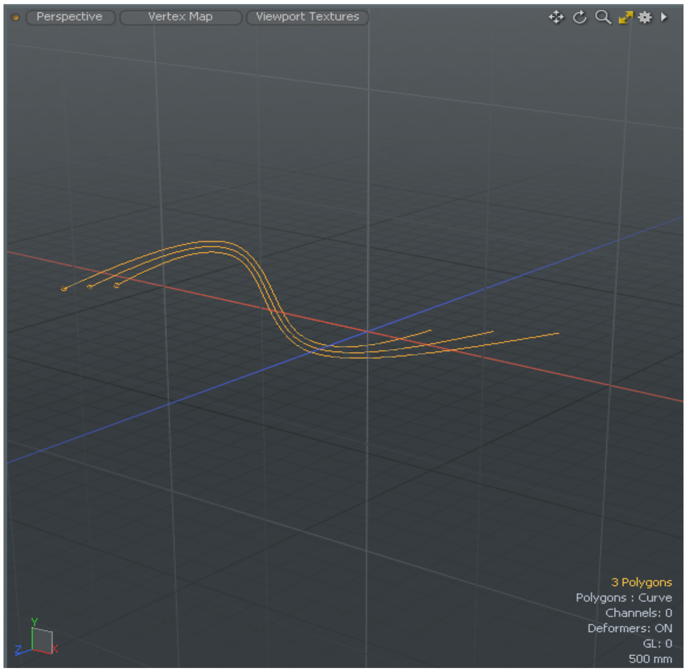
4. Set the Offset to **111 mm** and click **OK**.


Perspective
Vertex Map
Viewport Textures
2 Polygons
Polygons : Curve
Channels: 0
Deformers: ON
GL: 0
500 mm
Y
Z
X

An additional curve is created parallel to the original curve.

5. Press **Ctrl/Cmd** + **Z** to undo.
6. On the left panel, click-and-hold **Open Curve Offset** and select **Open Curve Offset Trio**.
7. Set the Offset to **111 mm** and click **OK**.

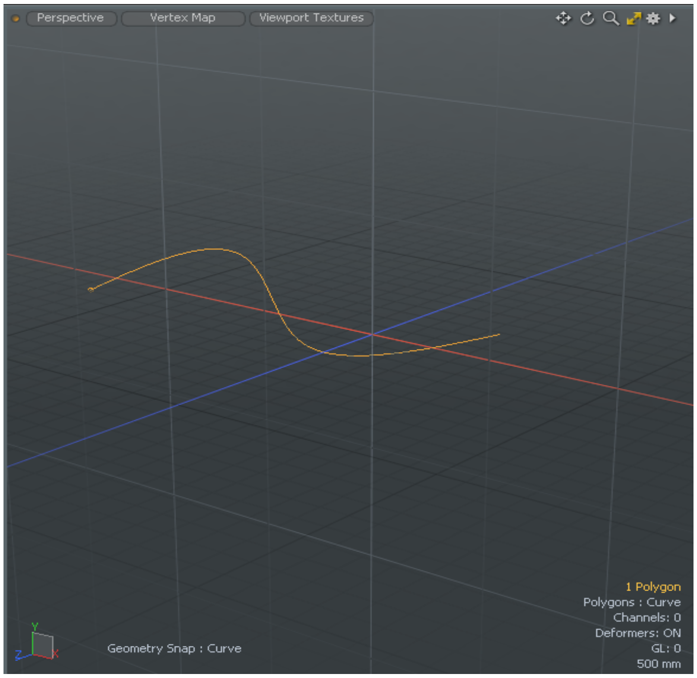
   Two new curves appear on either side of the original curve; one smaller and one larger than the original curve. Both curves are in the same layer in the **Items** list.


Perspective
Vertex Map
Viewport Textures
3 Polygons
Polygons : Curve
Channels: 0
Deformers: ON
GL: 0
500 mm
Y
Z
X

Closed Curve Offset

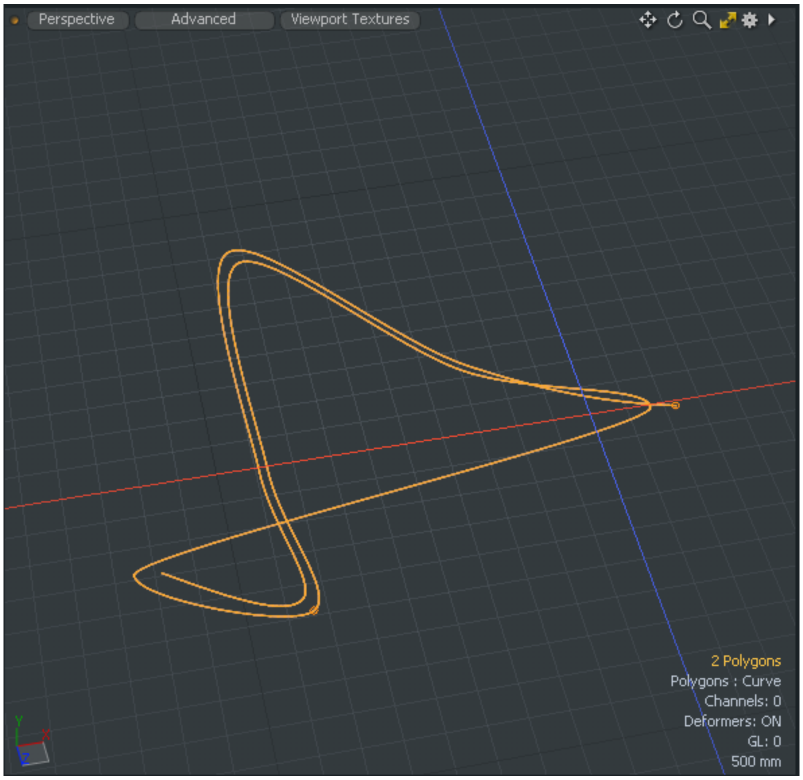
For procedural modeling, use the **Closed Curve Offset** tool to create closed curves offset from an existing curve.

1. In **Model** layout, add a curve to your scene.


Perspective
Vertex Map
Viewport Textures
1 Polygon
Polygons : Curve
Channels: 0
Deformers: ON
GL: 0
500 mm
Geometry Snap : Curve

2. Activate the **Polygon** selection mode, and double-click on the curve.
3. On the **Curves** tab, select **Close Curve Offset**.
4. Set the **Offset** value to **111 mm**.

   A duplicate closed curve displays 111 mm from the original curve. All curves are in the same layer in the **Items** list.


Perspective
Advanced
Viewport Textures
2 Polygons
Polygons : Curve
Channels: 0
Deformers: ON
GL: 0
500 mm

Duplicate Commands

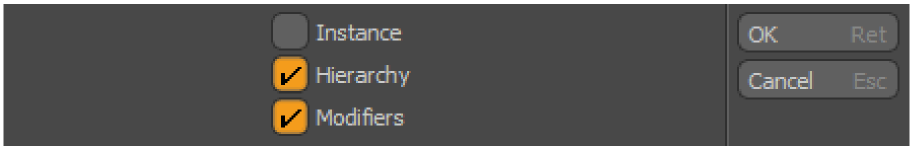
Duplicate

In Modo, using the **Duplicate** command creates a single copy of any item (or duplicates of multiple items if you select more than one item). You can invoke the command in the **Duplicate** sub-tab of the modeling tools panel or by choosing **Item > Duplicate**. (To duplicate geometry components, such as points or polygons, it is better to use the **Copy** and **Paste** commands.) There are no dialogs or options with the **Duplicate** command. Select the item and invoke the command. Then Modo creates an additional item and adds it to the **Item List**. When Modo creates the item, it automatically appends a number to the end of the duplicated item's name to avoid naming conflicts. You can adjust the protocol for the naming in the Modo system Preferences. To rename the item, simply right-click the new item layer and choose Rename.

Duplicate Hierarchy

The **Duplicate Hierarchy** command is similar to the **Duplicate** command; however, it also duplicates all descending hierarchy items as well to retain the parent/child relationships of the originating item. There are no dialogs or options. Select the item and invoke the command, and Modo creates the additional items and adds them to the **Item List**.

Duplicate Options



☐ Instance
☑ Hierarchy
☑ Modifiers

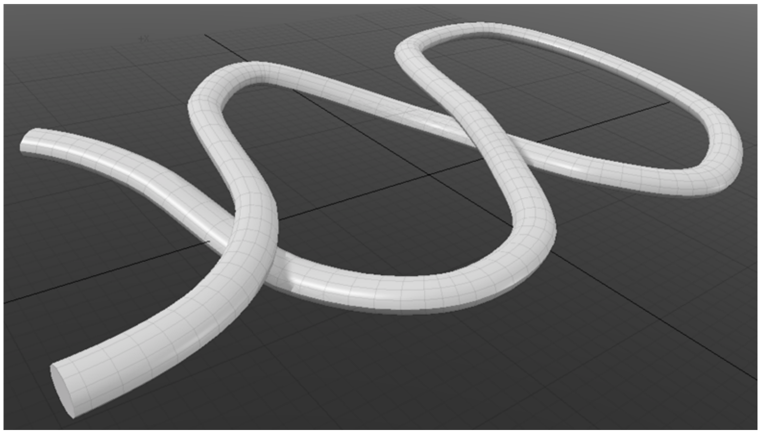
Invoking the **Duplicate Options** command presents a dialog with several options for duplicating items. To invoke this command, choose **Item > Duplicate Options**. If you select the **Instance** checkbox, Modo generates an instanced clone of the original item(s). If you select the **Hierarchy** checkbox, Modo duplicates the descending hierarchical chain of the selected item to retain the parent/child relationships. If you select the **Modifiers** checkbox, Modo properly duplicates any modifiers associated with the selected item(s).

Instance

If you invoke the **Instance** command (**Item > Instance**), Modo creates a single instance (a copy) of the selected item(s). Instances are different from duplicated geometry in that they reference the originating item. Because Modo transmits any component modifications made to a source item automatically to all descendants, they do not require additional scene memory and, therefore, can be duplicated many times over with little to no performance effect on the scene. You can recognize instanced items in the 3D Viewports by their distinct pink wireframe representations and can only modify them by using an Item transform. If you wish to modify an individual instance, convert it to a standard Mesh Item by right-clicking the instanced item's name in the **Item List** and choosing **Change Type > Mesh**.

Lace Geometry

The procedural Lace Geometry tool draws a tube along the path of a curve. By default, the tube cross section is circular, but you can also select from oval, square, rectangle, or a custom profile. The custom profile is drawn in a separate mesh, and is added to the Lace Geometry using the **Profile Shape** input.



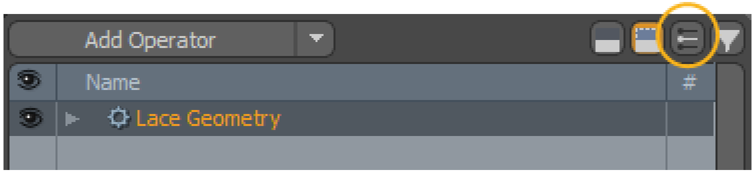
Note: The tool was given its name as it was originally developed to draw the laces for a footwear modeling project.

Using the Lace Geometry

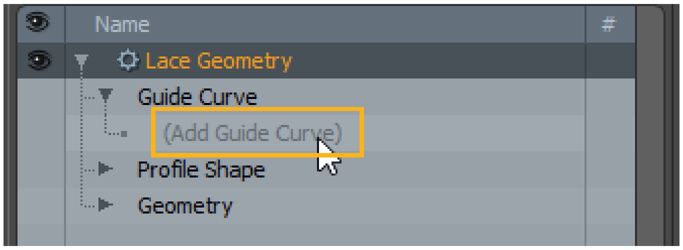
Lace Geometry requires a curve to draw along. This should be drawn in a separate mesh and added to the **Guide Curve** input. You can then select the cross-section shape in the **Properties** panel. To use a custom shape, draw one in a separate mesh and add it to the **Profile Shape** input.

The following steps show you how to draw the mesh pictured above. In this example, the curve is drawn after the Lace Geometry is added, though this is not compulsory. The steps assume you are starting with a new, empty mesh.

1. In the **Model** layout, open the **Mesh Ops** tab and click the **Add Operator** button (above the Mesh Operations List to open the Procedural Preset Browser.
2. Select **Mesh Operations** > **Duplicate** and double-click **Lace Geometry**.
3. Set the tube width to be larger, similar to the one in the image above: In the **Properties** for the **Lace Geometry** (lower-right), set **Radius** to **100 mm**.
4. Ensure the **Mesh Operations** list is not in single item view mode: if the single item view button is highlighted, click to disable it.

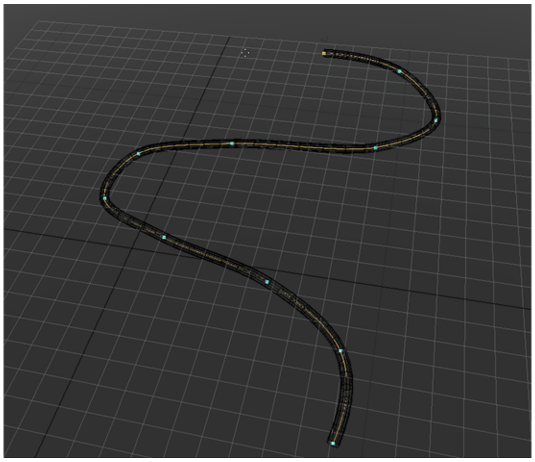


5. Now add a new mesh to hold the curve: Click the **Add Operator** button, select **Mesh Operations** > **Other** and double-click **Mesh**. In the **Item List**, rename this mesh to **Curve**.
6. In the **Mesh Operations** list, expand the **Guide Curve** setting for the **Lace Geometry** and click **(Add Guide Curve)**.



7. Select **Existing** and double-click the mesh named **Curve**.
8. Select the **Curve** mesh in the **Items List** so that you can draw into it.
9. In the **Basic** section of the modeling toolbox (upper-left of the UI), select the **Curve** tool.
10. Draw the curve in the 3D viewport by clicking at each plot point.

    The tube draws along the curve.



Note: With the default viewport settings, the tube is drawn wireframe as it is in the same mesh as the **Lace Geometry**, and not the currently selected **Curve** mesh. You can change how inactive meshes are drawn in the 3D Viewport Properties section of the 3D viewport settings. To open the settings, press **O** when the mouse is over the 3D viewport.

11. The image at the start of this topic shows a tube with an oval cross section. You can replicate this by adjusting the properties: Select the **Lace Geometry** in the **Mesh Operations** list, and select the **Lace Geometry** sub-tab in the **Properties** panel. Change **Shape** to **Oval** and set the **Aspect Ratio** to **50 %**.

You can now adjust the properties to further customize the mesh.

Lace Geometry Properties

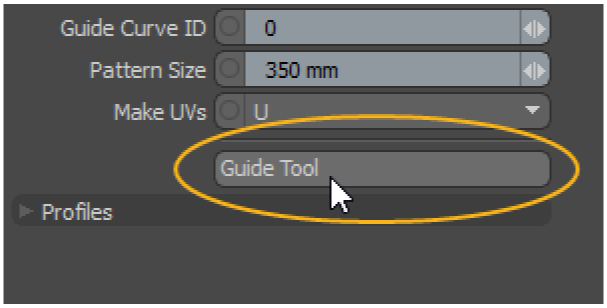
The Lace Geometry can be modified using the following properties:

Lace Geometry	
Shape	The profile shape (cross-section) of the tube. Select from: • **Circle** - You can adjust the number of sides using the **Sides** property. • **Oval** - The height of the oval can be adjusted using the **Aspect Ratio** property. • **Square** - A square with a single edge per side. • **Rectangle** - The height of the rectangle can be adjusted using the **Aspect Ratio** property. • **Custom Profile** - Allows you to use any curve shape for the cross-section. Add a curve to a different mesh and then attach it to the Lace Geometry using the

	Profile Shape input. See Lace Geometry Procedural for more details.
Sides	Sets the number of sides when the **Shape** is **Circle** or **Oval**.
Aspect Ratio	Sets the relative height when the **Shape** is **Oval** or **Rectangle**.
Radius	Sets the radius of the tube shape. This applies to all shapes.
Shape Angle	The angle of rotation of the profile shape in 2D space before it is applied to the lace geometry.
Refinement Angle	This is the threshold when dividing the curve for adaptive subdivision. Smaller values generate more segments.
Start	The proportion of the curve length along which the tube should start.
End	The proportion of the curve length along which the tube should end.
Guide Curve ID	The index of the curve to use as the guide. This is useful if there are multiple curves in the curve mesh, and you want to select a particular one. Curve indices start at 0 and increment for each curve drawn.
Pattern Size	This is the size of UV texture and is mapped to 0-1 in UV space; this corresponds to the size of the mesh in the 3D viewport. For example, if you had a straight rectangular tube that was 2 m in length, you would set the **Pattern Size** to 2 m so that the sides of the tube would fit in the 0-1 UV space. Since any UVs of the Lace Geometry are procedural, it is important to set the appropriate value to make the UV texture with real size. You can see the effect of this property by opening the UV map in the UV editor and adjusting the value.
Make UVs	Sets the direction of the lacing UVs. To disable UV map creation, set this to **None**.

Using the Guide Tool

The **Guide Tool** in the Lace Geometry operation's **Properties** panel gives you further control over the laces. You can reposition the points of the curve and have the laces follow the new positions, twist, or shrink the laces around a selected point.

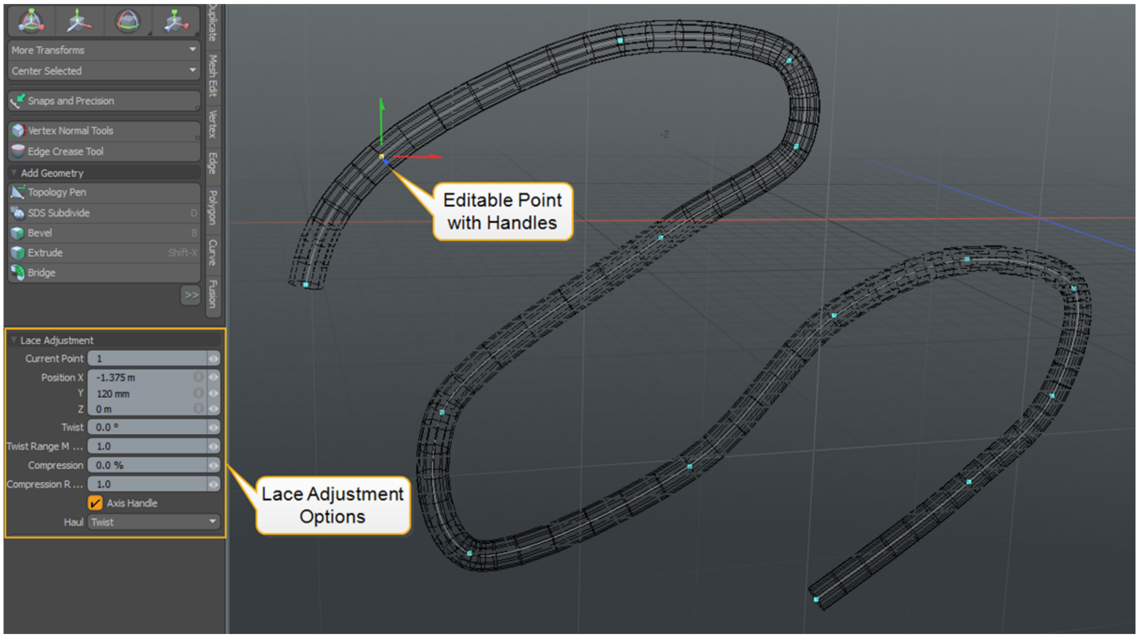


Guide Curve ID: 0
Pattern Size: 350 mm
Make UVs: U

Guide Tool
Profiles

1. Activate the tool by clicking the **Guide Tool** button.

   The tool automatically finds the lace guide curve and its points become editable in the 3D viewport. At the same time, the **Lace Adjustment** options appear on the left panel.



Lace Adjustment
Current Point: 1
Position X: -1.375 m
Y: 120 mm
Z: 0 m
Twist: 0.0 °
Twist Range M ...: 1.0
Compression: 0.0 %
Compression R ...: 1.0
☑ Axis Handle
Haul: Twist

More Transforms
Center Selected
Snaps and Precision
Vertex Normal Tools
Edge Crease Tool
Add Geometry
Topology Pen
SDS Subdivide
Bevel
Extrude
Bridge
Duplicate
Mesh Edit
Vertex
Edge
Polygon
Curve
Fusion
Editable Point with Handles
Lace Adjustment Options

2. Adjust the curve positions by clicking a point in the 3D viewport and dragging the axis handles.

   The lace geometry follows the new curve positions.

3. Edit the **Lace Adjustment** options on the left panel.

   The following options are available:

Current Point	The point selected in the 3D viewport.
Position X/Y/Z	The current point's position.
Twist	Rotates the lace along the tangent direction around the curve point. The influence range is determined by the radius.

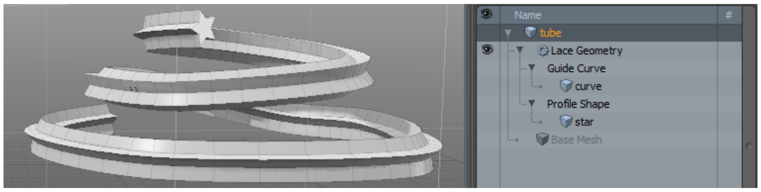
Twist Range Multiplier	Increases/decreases the **Twist**'s influence range.
Compression	Shrinks the lace around the curve point.
Compression Range Multiplier	Increases/decreases the **Compression**'s influence range.
Axis Handle	Enables/disables showing axis handles for the selected point in the 3D viewport.
Haul	Specifies the hauling behavior. Clicking and dragging in the viewport with a point selected twists or compresses the lace around the selected point.

Lace Geometry Procedural

The Lace Geometry operation has the following procedural inputs:

- **Guide Curve** - Set this to the mesh that contains the guide curve. If the mesh contains multiple curves, use the **Guide Curve ID** property (see above) to select the required curve.
- **Profile Shape** - Allows you to add a custom shape to use as the profile shape. This input is used if you set the **Shape** property to **Custom Profile**. To use this input, add a profile curve to a separate mesh, click **(Add Profile Shape)** and select the mesh.

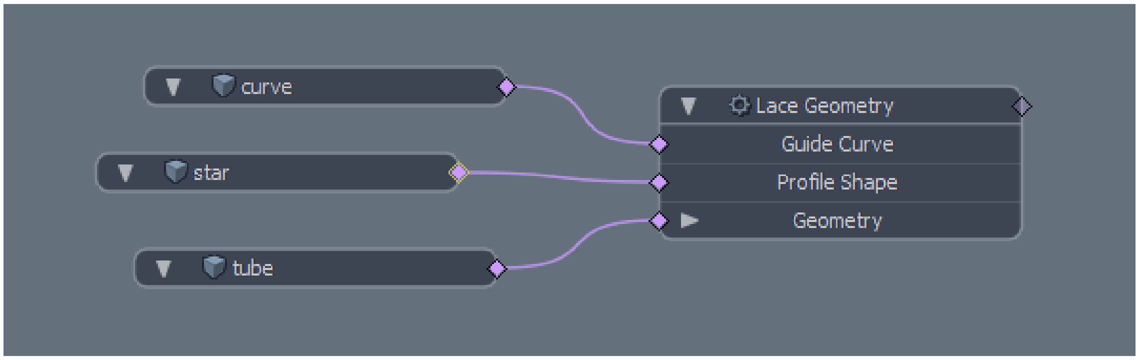
  The image below shows a mesh generated using the Star primitive, together with the corresponding mesh operations.



- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

Lace Geometry Schematic

The Lace Geometry schematic node has the same inputs as the procedural version. The **Profile Shape** input is optional.

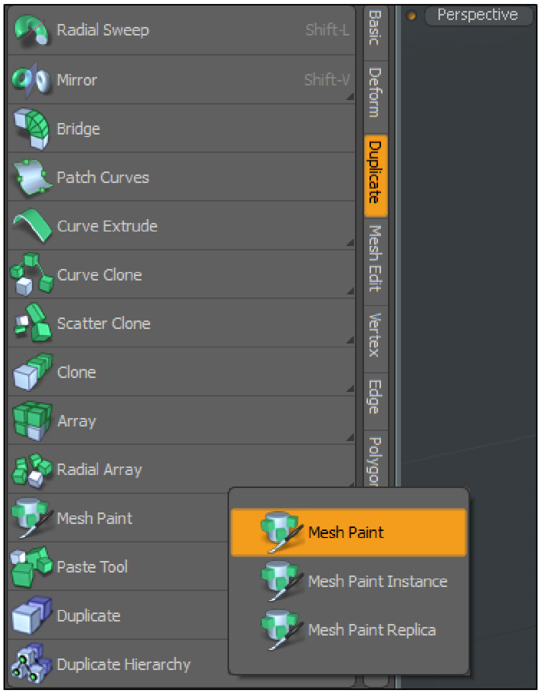

curve
star
tube
Lace Geometry
Guide Curve
Profile Shape
Geometry

Mesh Paint

The Mesh Paint tool provides a method to directly apply one mesh layer onto the surface of another. For example, you can brush many copies of boulders across a landscape.

You can paint geometry, instances, and replicas and accurately place a single copy or apply hundreds of copies with a single stroke.

Mesh Paint Tools


Radial Sweep Shift-L
Mirror Shift-V
Bridge
Patch Curves
Curve Extrude
Curve Clone
Scatter Clone
Clone
Array
Radial Array
Mesh Paint
Paste Tool
Duplicate
Duplicate Hierarchy
Mesh Paint
Mesh Paint Instance
Mesh Paint Replica
Basic
Deform
Duplicate
Mesh Edit
Vertex
Edge
Polygon
Perspective

- **Mesh Paint** - Paints a single copy of the background mesh item onto a surface. Each background item added to the surface of the target is added to the mesh item.
- **Mesh Paint Instance** - Paints instances of an item onto a surface. The first background item added to the surface of the target is added to the mesh item. Adding an additional item adds an instances to the target mesh until you change the source item.
- **Mesh Paint Replica**- Paints replicas of an item onto a surface. The first background item is added to the surface of the target mesh item and a point cloud and replicator are created. Adding additional source items are then added to the point cloud.

To activate the Mesh Paint tool:

- On the left panel, open the **Duplicate** tab, click the down arrow on **Mesh Paint**, and select one of the Mesh Paint Tools.
- Alternatively, click **Geometry** > **Duplicate** > **Mesh Paint**.

Mesh Paint

There must be at least two Mesh Items in the **Item List**. The currently selected item is the target of the tool and the background (visible but unselected) item is the source item applied onto the target.

With this setup, you can drag across the selected Mesh Item and Modo applies the background mesh along the brush stroke.

Tip: You can use **Snapping** with Vertex and Polygon Center options enabled while applying the Mesh Paint Tool. The background item is snapped to the polygons or vertices of the target surface. For more information, see Applying Snapping.
In addition, you can use **Symmetry** while using the Mesh Paint tool to apply mirrored background items across a chosen axis. For more information, see Applying Precision .

Applying Mesh Paint

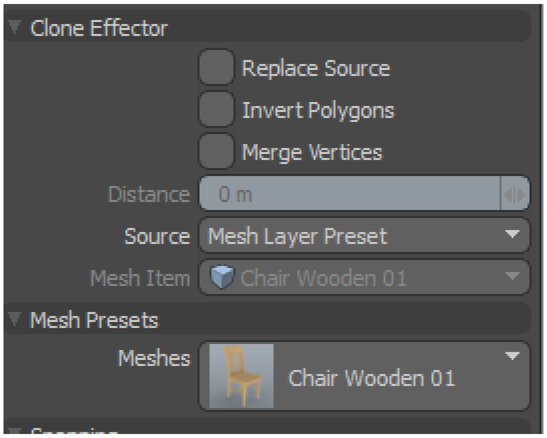
In this example you'll paste a chair model from our Preset Browser over the surface of a sphere.

1. In **Model** layout, on the left panel, open the **Basics** tab, and **Shift** + click **Sphere**.

   The Sphere added to the scene an is selected in the Items tab on the right panel.

2. Under the layout menu bar, click the **Polygon** selection mode icon.
3. On the left panel, open the **Duplicate** tab and click **Mesh Paint**.
4. Set **Clone Effector** > **Source** to **Mesh Layer Preset**.
5. Under the Mesh Presets properties, click and select **Meshes** > **Interior** > **Chairs** > **Chair Wooden 01**.

   For more information about navigating in the Preset Browser, see Browsing the Directory Tree.



Clone Effector

☐ Replace Source

☐ Invert Polygons

☐ Merge Vertices

Distance: 0 m

Source: Mesh Layer Preset

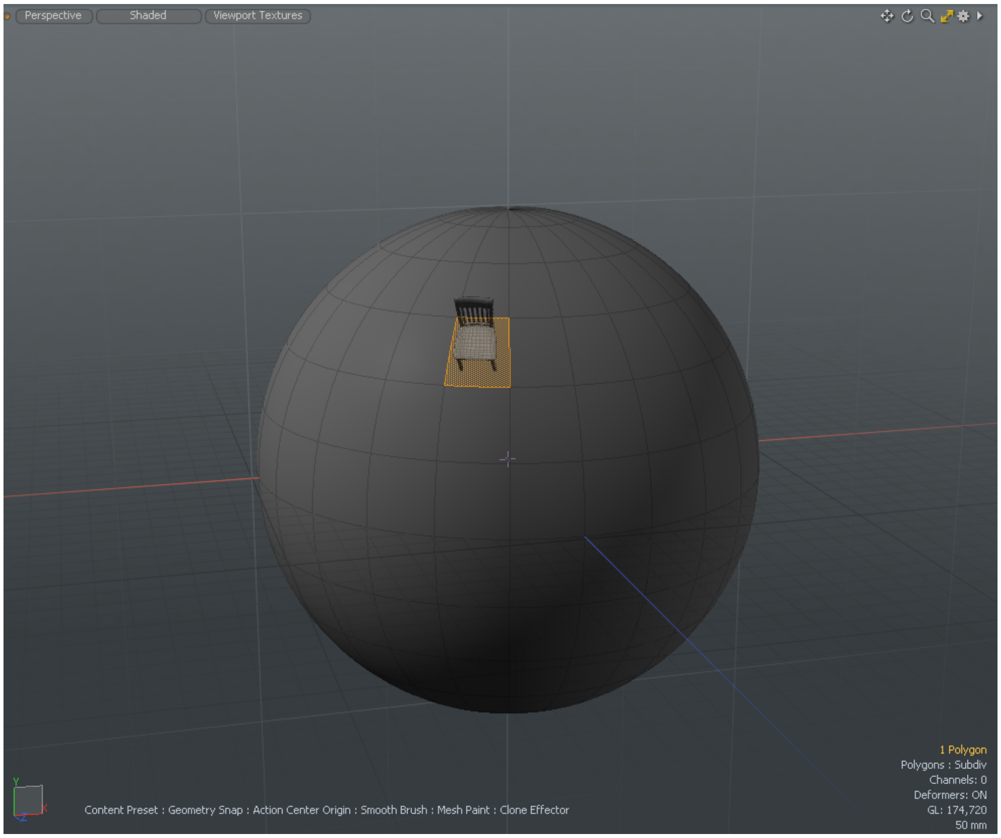
Mesh Item: Chair Wooden 01

Mesh Presets

Meshes: Chair Wooden 01

6. With Polygon selection mode active, click a polygon on the Sphere's surface.

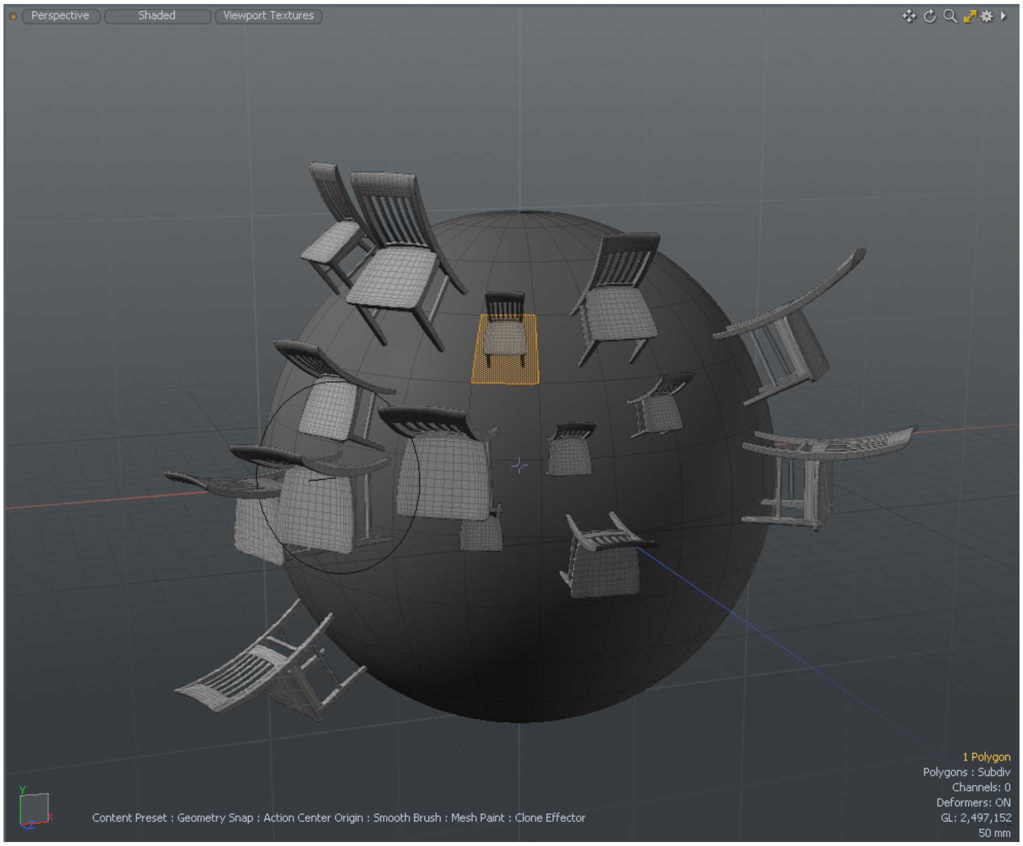
   The chair is added to the surface of the chair.


Perspective
Shaded
Viewport Textures
1 Polygon
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 174,720
50 mm
Content Preset : Geometry Snap : Action Center Origin : Smooth Brush : Mesh Paint : Clone Effector

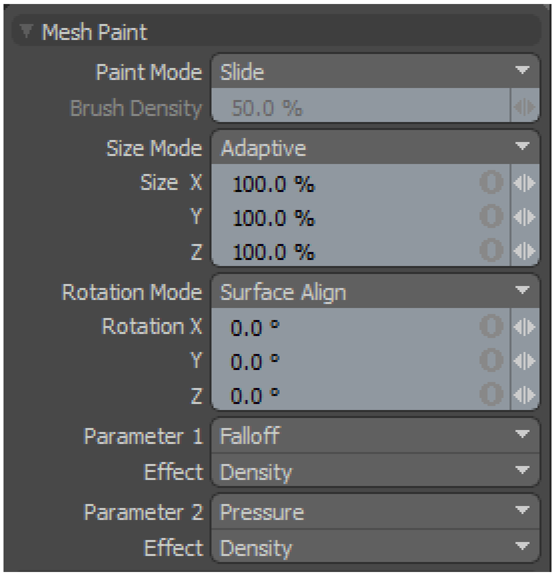
7. Right-click and drag over a target area on the Sphere's surface to change the brush size and then click again.

Tip: You can also change the **Smooth Brush** > **Size** value in the toolbox properties and then click on the sphere's surface.

8. Click and drag on the sphere's surface to position the next chair and repeat.


Perspective
Shaded
Viewport Textures
1 Polygon
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 2,497,152
50 mm
Content Preset : Geometry Snap : Action Center Origin : Smooth Brush : Mesh Paint : Clone Effector

Mesh Paint Options



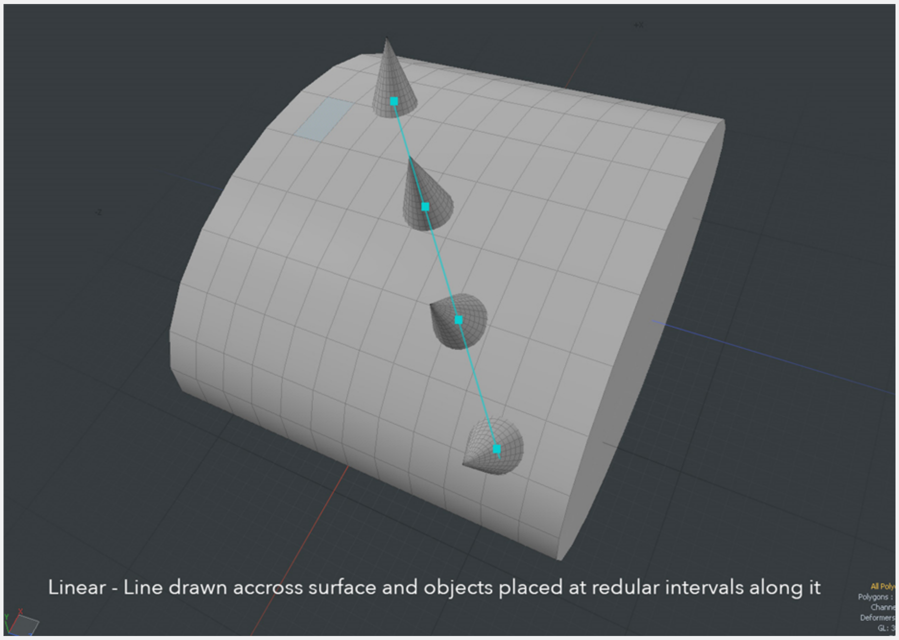
Mesh Paint

Paint Mode: Slide
Brush Density: 50.0 %
Size Mode: Adaptive
Size X: 100.0 %
Y: 100.0 %
Z: 100.0 %
Rotation Mode: Surface Align
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Parameter 1: Falloff
Effect: Density
Parameter 2: Pressure
Effect: Density

Paint Mode	There are several modes that control how Modo applies the background Mesh Item(s)

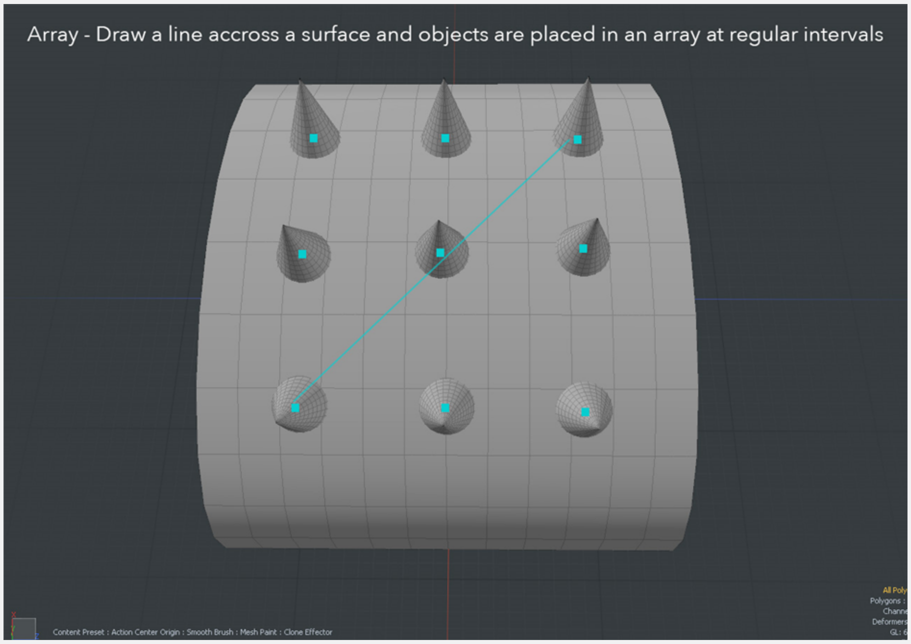
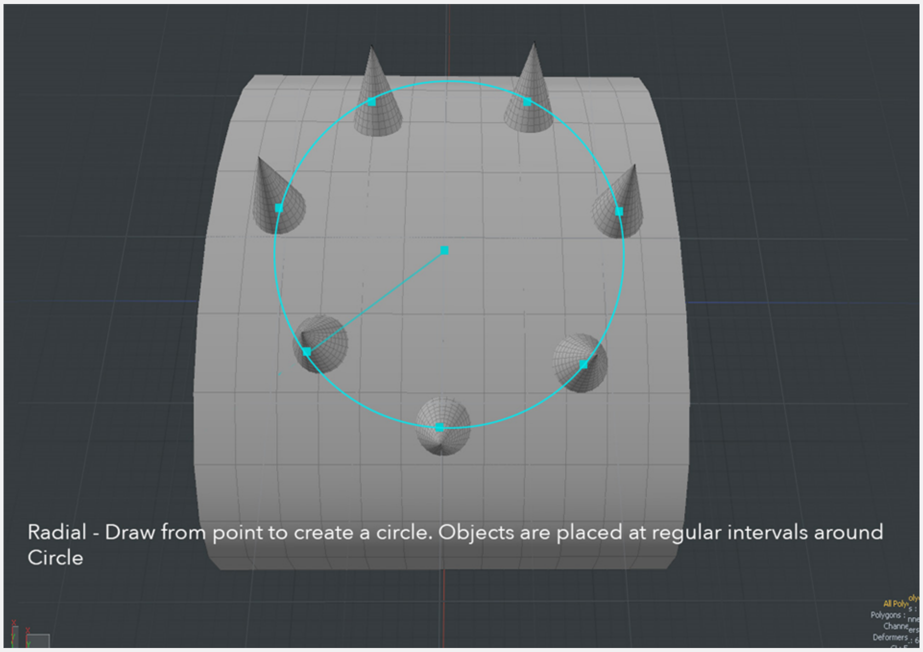
with each stroke.

- **Slide** - Drags a single item over the target surface and positions it where you release the mouse button. You can also press **Ctrl** before clicking to interactively set the scale of the duplicates.
- **Strokes** - Scatters multiple items along the brush stroke to clone many items across a surface with randomized placements.
- **Drop** - Places a single item at the intersection of the pointer (when you click) and the target surface. You can drag to interactively scale and rotate the item.
- **Linear** - Creates objects along a line. Click and drag to position. You can set the number of duplicates in the **Linear Count** option. The stroke is constrained the angle to 45 degree by holding the **Shift** key.



Linear - Line drawn accross surface and objects placed at redular intervals along it

- **Array** - Creates objects at the grid points. Click and drag to position. **Array Count X** is the number of duplicates along X. **Array Count Y** is the number of duplicats along Y. The stroke width and height is constrained to square by holding the **Shift** key.

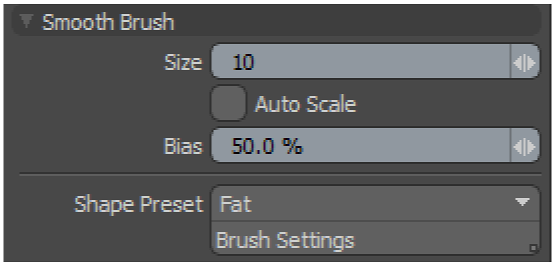
	 Array - Draw a line accross a surface and objects are placed in an array at regular intervals Content Preset : Action Center Origin : Smooth Brush : Mesh Paint : Clone Effector • **Radial** - Creates objects along the circle at the first click point. Click and drag to position. The radius of the circle is distance from the first click point and moving point. **Radial Count** is the number of duplication. The start of radial position is constrained the angle to 45 degree by holding the **Shift** key.  Radial - Draw from point to create a circle. Objects are placed at regular intervals around Circle
Size Mode	Determines the scale of the mesh duplicates on the surface.

	• **Uniform** - Retains the original size of the item when placed. You can adjust the scale with the X, Y, and Z values as a multiplier of the original size. (Modo ignores Item mode transforms on the source.) • **Adaptive** - Adapts the size of the item to the brush size. You can adjust the scale with the X, Y, and Z values as a multiplier of the generated size. (Modo ignores Item mode transforms on the source.) • **Random** - Scales each duplicate randomly as determined by the X, Y, and Z values. **0%** provides no variation; larger values increase the amount of variation. • **Adaptive + Random** - Combines the properties of these Size modes to add randomness to the adaptive scaling of cloned duplicates. • **Locked** - Locks the size of the placed mesh at the current size. You can then zoom the viewport without the brush being resized.
Size X, Y, Z	Changes the X, Y, and Z dimensions of the geometry on the target surface.
Rotation Mode	Determines the orientation of the placed mesh duplicates onto the surface. • **Uniform** - Retains the original orientation of the item when placed. You can adjust the rotation with the X, Y, and Z values as an offset of the original rotation. (Modo ignores Item mode transforms on the source.) • **Surface Align** - Aligns the target surface's normal facing direction with the up (Y+) direction of the cloned item. • **Random** - Rotates each duplicate randomly as determined by the X, Y, and Z values. **0°** provides no variation; larger values increase the amount of variation. • **Surface Align + Random** - Aligns the item's up direction to the target surface's normal facing direction and then randomizes it away from that as determined by the X, Y, and Z values. • **Surface Align + Tablet** - Aligns the item's up direction to the target surface's normal facing direction. You can then rotate items based on the stylus' orientation to the tablet's surface. (This requires a pen tablet with orientation support.) • **Surface Align + Tablet (no Pitch)** - Ignores **Pitch** rotation (the Z axis), but otherwise is the same as **Surface Align+Tablet**. • **Screen Align + Tablet** - Considers the orientation of the view on screen. You can then rotate items based on the stylus' orientation to the tablet's surface (This requires a pen tablet with orientation support.) • **Screen Align + Tablet (no Pitch)** - Ignores **Pitch** rotation (the Z axis), but otherwise is the same as **Screen Align+Tablet**.

Rotation X, Y, Z	Changes the rotation position of the geometry painted on the surface. The default is 0 degrees for X, Y, and Z.
Parameter 1/Parameter 2	Provides additional controls over the duplicated geometry by using a number of criteria. The parameters are the controlling factors; the effects of each determine what the relevant parameter controls. • **Falloff** - Uses a falloff. • **Pressure** - Uses the strength of the stylus' pressure against tablet. • **Slope** - Uses the slope (angle) of the target surface. • **Altitude** - Uses the altitude (height) of the target surface. • **1.0 - Slope** - Inverts the **Slope** parameter. • **1.0 - Altitude** - Inverts the **Altitude** parameter.
Effect	Determines the aspect controlled by **Parameter 1** or **Parameter 2**. • **(none)** - Affects nothing. This disables the **Parameter 1/2**. • **Density** - Controls the number of duplicates. • **Size** - Controls the scale of the duplicates. • **Rotation** - Controls the orientation of the duplicates. • **Size + Density** - Applies simultaneously both the **Size** and **Density** attributes.

Smooth Brush

This is one of the standard brushes used in Modo for painting and sculpting. See Smooth tool for details.



Smooth Brush

Size 10

☐ Auto Scale

Bias 50.0 %

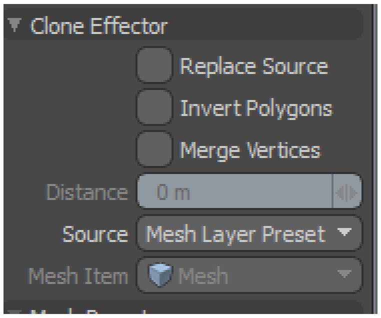
Shape Preset Fat

Brush Settings

Size	Controls the area across which Modo scatters the duplicates and their initial density when placing elements with the **Strokes** mode. Larger brushes (in relation to the surface) spread duplicates across a larger area. When the **Size** mode is set to **Adaptive**, the size of the brush also

	affects the size of the duplicated geometry.
Auto Scale	Automatically resizes the brush based on the zoom level.
Bias	Changes the curve adjustment to favor one side over another.
Shape Preset	Provides preset brush shapes to control the initial density of the duplicates.

Clone Effector



Clone Effector

☐ Replace Source

☐ Invert Polygons

☐ Merge Vertices

Distance: 0 m

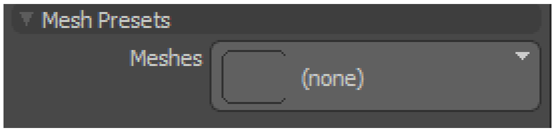
Source: Mesh Layer Preset

Mesh Item: Mesh

Replace Source	Replaces the source with a duplicate shape.
Invert Polygons	Reverses the normals of polygons
Merge Vertices	Measures the distance between vertices in the currently selected geometry to see if any fall within a specific distance threshold to one another and then merges them.
Distance	Specifies the distance to merge vertices.
Source	Determines the source mesh that Modo clones into the array. • **Active Mesh** - Uses the currently selected item(s) as a clone source. • **Specific Mesh**- Specifies a mesh item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset

	Browser. Click Mesh Item and select a mesh preset. • **Random Preset** - Uses the current selected preset at random.
Mesh Item	Specifies the mesh to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Mesh Presets



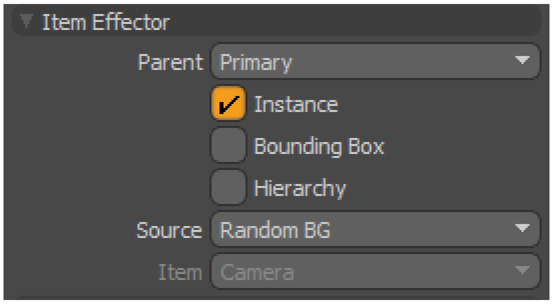
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**. For more information, see Preset Browser.

Mesh Paint Instance

The Mesh Paint Instance tool provides a method to directly apply a mesh instance onto the surface of a mesh layer. To activate the tool, open the **Duplicate** tab on the left panel, right-click the **Mesh Paint** tool and select **Mesh Paint Instance** from the list.

For more information, see Instances.

Mesh Paint Instance Options



Parent: Primary
☑ Instance
☐ Bounding Box
☐ Hierarchy
Source: Random BG
Item: Camera

In addition to the **Mesh Paint** and **Smooth Brush** properties described above, the following properties are available in the **Item Effector** section for the Mesh Paint Instance tool.

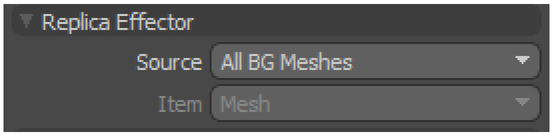
Parent	Determines how Modo treats instanced items in the **Item List**. • **Off** - Clones source item as additional instanced layers in the **Item List**. • **Hierarchy** - Creates a cascading array where each subsequent clone is a child of the previous clone. • **Root** - Creates instances as child items of the source mesh. • **Primary** - Creates Instances as children of a group locator that is parented to the source mesh.
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewport by pink wireframes. This is enabled by default.
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo clones into the array. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Click Mesh Item and select a mesh preset.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Mesh Paint Replica

The Mesh Paint Replica tool provides a method to directly apply a mesh replica onto the surface of a mesh layer. To activate the tool, open the **Duplicate** tab on the left panel, right-click the **Mesh Paint** tool and select **Mesh Paint Replica** from the list.

For more information, see Replicator Item.

Mesh Paint Replica Options



Replica Effector

Source: All BG Meshes

Item: Mesh

In addition to the **Mesh Paint** and **Smooth Brush** properties described above, the following properties are available in **Replica Effector** section for the Mesh Paint Replica tool.

For more information, see Replicators.

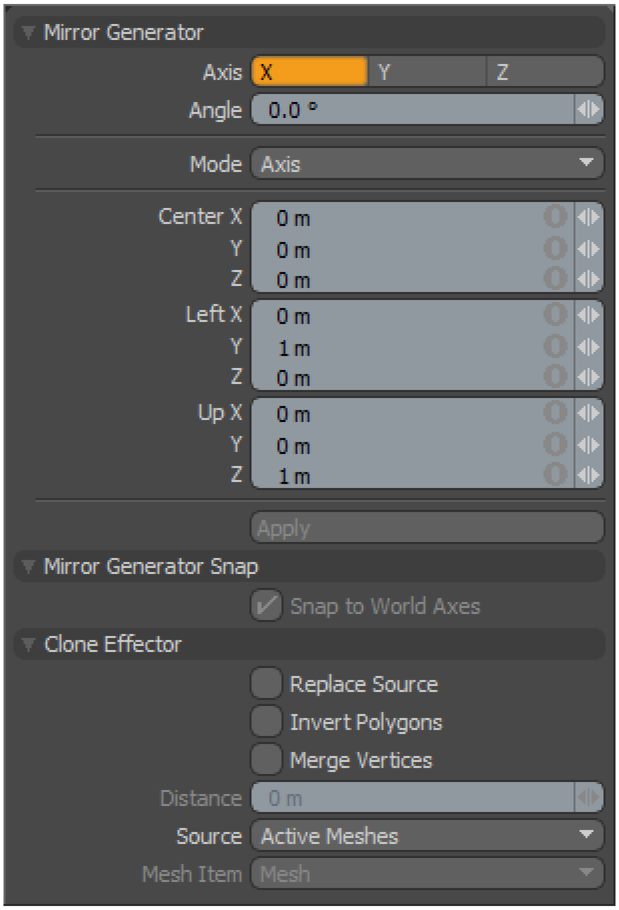
Source	Determines the source mesh that Modo clones into the array. • **Active Item** - Use the currently selected geometry as the clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG Meshes** - Combines all the currently visible background items and clones them as a single mesh. • **Mesh Layer Preset** - Clones a profile shape selected in the Preset Browser. Click Mesh Item and select a mesh preset.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Mirror

By using the Mirror tool, you can create a duplicate of the selected geometry in the active layer. When you activate the Mirror tool, Modo creates an axis across which it duplicates the selected geometry in a flipped (or mirrored) orientation to the original. With the tool active, you can interactively position the mirror plane to change the distance of the mirrored geometry or change the rotation of the plane to adjust the orientation of the duplicated geometry.

The **Mirror** tool is found in the **Model** layout **Toolbox**, under the **Duplicate** sub-tab. In addition, it is found in the **Topology** layout **Toolbox**, under the **Tools** sub-tab (**Instance Mirror**) . Alternatively, in the **Geometry** > **Duplicate** > **Mirror** menu.

Mirror Properties



Mirror Generator
Axis: X
Angle: 0.0 °
Mode: Axis
Center X: 0 m
Y: 0 m
Z: 0 m
Left X: 0 m
Y: 1 m
Z: 0 m
Up X: 0 m
Y: 0 m
Z: 1 m
Mirror Generator Snap
☑ Snap to World Axes
Clone Effector
☐ Replace Source
☐ Invert Polygons
☐ Merge Vertices
Distance: 0 m
Source: Active Meshes
Mesh Item: Mesh

The following properties are available for the Mirror tool.

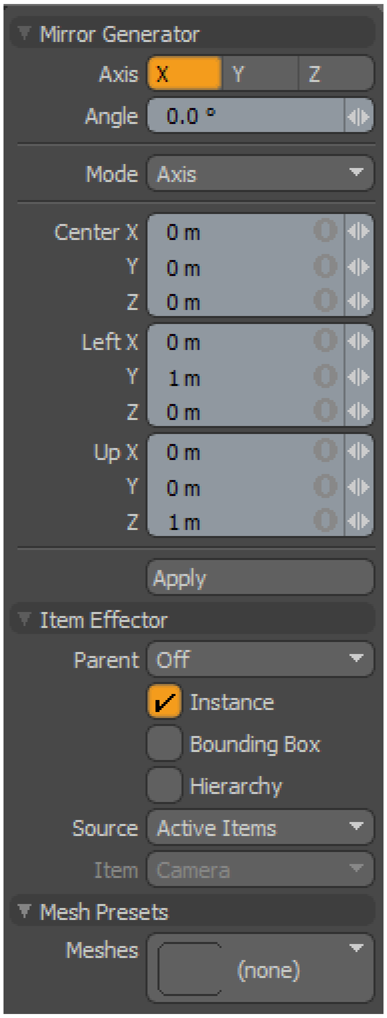
Mirror Generator	
Axis	Sets the axis direction (X, Y, or Z) for the mirror plane and, therefore, defines the center position across which Modo clones the geometry.
Angle	Determines the rotation of the mirror plane around the axis (in degrees). The default value of **180** degrees flips the geometry across the axis. By increasing or decreasing this value, you can skew the mirror plane to create an acute or obtuse angle between the original and new geometry.
Mode	Determines what handles Modo makes available to control the mirroring. • **Axis** - Keeps the mirror manipulator handle aligned to an upright axis. Use this for implementing a single-value rotation specified by the **Angle** value. • **Free Rotation** - Produces a second mirror manipulator handle for you to adjust an additional axis of rotation. • **Three Points** - Produces a triangular manipulator for you to adjust any degree of freedom in the angle and position of the mirroring plane.
Center X/Y/Z	Sets the position for the main mirror plane. You can specify this value numerically or by manipulating the Mirror tool handle. To adjust the position, drag the center of the handle.
Left X/Y/Z	Rotates the mirroring plane as a secondary value. You can specify this value numerically or by manipulating the Mirror tool handle.
Up X/Y/Z	Provides an additional degree of rotation of the mirror plane in **Free Rotation** or **Three Points** mode.
Mirror Generator Snap	
Snap to World Axis	When snapping and this option is enabled, snaps the handles to the axis' center positions. (To enable or disable snapping, press the **X** key. See Applying Snapping for details.)
Clone Effector	
Replace Source	Removes the initial geometry and leaves only the mirrored results.

Mirror Generator	
Invert Polygons	Flips the polygons' normals. This is useful when mirroring across a negative plane that inverts the directions of the polygons' faces.
Merge Vertices	When enabled, attempts to merge any vertices that overlap between the original and new geometry.
Distance	Determines the threshold for merging points from the new and original geometry when you enable **Merge Vertices**.
Source	Determines which geometry Modo mirrors. Generally, this is the default **Active Mesh**. Other options may be useful when cloning multiple items (such as with the Scatter Clone tool).

Instance Mirror

The Instance Mirror tool works similarly to the Mirror tool, but instead of producing actual geometry, it creates instances of the original item. To activate the Instance Mirror tool, right-click on the **Mirror** icon in the **Duplicate** sub-tab of the modeling toolbox, and select **Instance Mirror** from the list.

Instance Mirror Properties



Axis X Y Z

Angle 0.0 °

Mode Axis

Center X 0 m
Y 0 m
Z 0 m

Left X 0 m
Y 1 m
Z 0 m

Up X 0 m
Y 0 m
Z 1 m

Parent Off

☑ Instance

☐ Bounding Box

☐ Hierarchy

Source Active Items

Item Camera

Meshes (none)

Mirror Generator	
Axis	Sets the axis direction (X, Y, or Z) for the mirror plane and, therefore, defines the center position across which Modo clones the geometry.
Angle	Determines the rotation of the mirror plane around the axis (in degrees). The default value of **180** degrees flips the geometry across the axis. By increasing or decreasing this value, you can skew the mirror plane to create an acute or obtuse angle between the original and new geometry.
Mode	Determines what handles Modo makes available to control the

Mirror Generator	
	mirroring. • **Axis** - Keeps the mirror manipulator handle aligned to an upright axis. Use this for implementing a single-value rotation specified by the **Angle** value. • **Free Rotation** - Produces a second mirror manipulator handle for you to adjust an additional axis of rotation. • **Three Points** - Produces a triangular manipulator for you to adjust any degree of freedom in the angle and position of the mirroring plane.
Center X/Y/Z	Sets the position for the main mirror plane. You can specify this value numerically or by manipulating the Mirror tool handle. To adjust the position, drag the center of the handle.
Left X/Y/Z	Rotates the mirroring plane as a secondary value. You can specify this value numerically or by manipulating the Mirror tool handle.
Up X/Y/Z	Provides an additional degree of rotation of the mirror plane in **Free Rotation** or **Three Points** mode.
Mirror Generator Snap	
Snap to World Axis	When snapping and this option are enabled, snaps the handles to the axis' center positions. (To enable or disable snapping, press the **X** key. See the Applying Snapping topic for details.)
Clone Effector	
Replace Source	Removes the initial geometry and leaves only the mirrored results.
Invert Polygons	Flips the polygons' normals. This is useful when mirroring across a negative plane that inverts the directions of the polygons' faces.
Merge Vertices	When enabled, attempts to merge any vertices that overlap between the original and new geometry.
Distance	Determines the threshold for merging points from the new and original geometry when you enable **Merge Vertices**.
Source	Determines which geometry Modo mirrors. Generally, this is the default

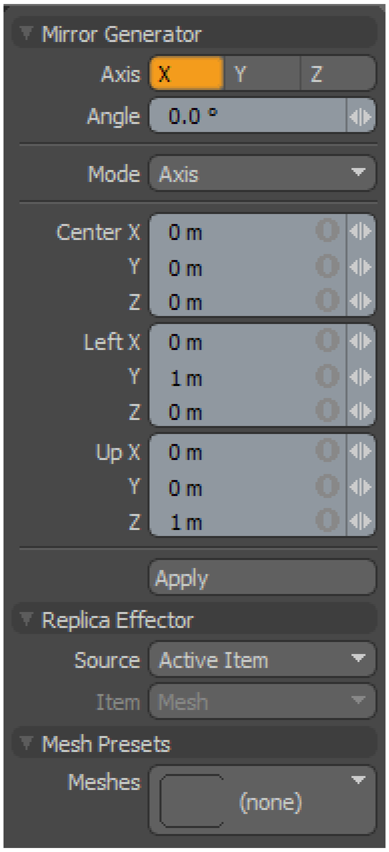
Mirror Generator	
	Active Mesh. Other options may be useful when cloning multiple items (such as with the Scatter Clone tool).
Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**. • **Off** - Clones source item as additional instanced layers in the **Item List**. • **Hierarchy** - Creates a cascading array where each subsequent clone is a child of the previous clone. • **Root** - Creates instances as child items of the source mesh. • **Primary** - Creates Instances as children of a group locator that is parented to the source mesh.
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewports by pink wireframes. This is enabled by default.
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo mirrors. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.

Mirror Generator	
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Replica Mirror

The Replica Mirror tool works similarly to the Mirror tool, but instead of producing actual geometry, it creates replicas of the source item. To activate the Replica Mirror tool, right-click on the **Mirror** icon and select **Replica Mirror** from the list.

Replica Mirror Properties



Axis X
Angle 0.0 °
Mode Axis
Center X 0 m
Y 0 m
Z 0 m
Left X 0 m
Y 1 m
Z 0 m
Up X 0 m
Y 0 m
Z 1 m
Source Active Item
Item Mesh
Meshes (none)

Mirror Generator	
Axis	Sets the axis direction (X, Y, or Z) for the mirror plane and, therefore,

Mirror Generator	
	defines the center position across which Modo clones the geometry.
Angle	Determines the rotation of the mirror plane around the axis (in degrees). The default value of **180** degrees flips the geometry across the axis. By increasing or decreasing this value, you can skew the mirror plane to create an acute or obtuse angle between the original and new geometry.
Mode	Determines what handles Modo makes available to control the mirroring. • **Axis** - Keeps the mirror manipulator handle aligned to an upright axis. Use this for implementing a single-value rotation specified by the **Angle** value. • **Free Rotation** - Produces a second mirror manipulator handle for you to adjust an additional axis of rotation. • **Three Points** - Produces a triangular manipulator for you to adjust any degree of freedom in the angle and position of the mirroring plane.
Center X/Y/Z	Sets the position for the main mirror plane. You can specify this value numerically or by manipulating the Mirror tool handle. To adjust the position, drag the center of the handle.
Left X/Y/Z	Rotates the mirroring plane as a secondary value. You can specify this value numerically or by manipulating the Mirror tool handle.
Up X/Y/Z	Provides an additional degree of rotation of the mirror plane in **Free Rotation** or **Three Points** mode.
Mirror Generator Snap	
Snap to World Axis	When snapping and this option are enabled, snaps the handles to the axis' center positions. (To enable or disable snapping, press the **X** key. See the Applying Snapping topic for details.)
Clone Effector	
Replace Source	Removes the initial geometry and leaves only the mirrored results.
Invert Polygons	Flips the polygons' normals. This is useful when mirroring across a

Mirror Generator	
	negative plane that inverts the directions of the polygons' faces.
Merge Vertices	When enabled, attempts to merge any vertices that overlap between the original and new geometry.
Distance	Determines the threshold for merging points from the new and original geometry when you enable **Merge Vertices**.
Source	Determines which geometry Modo mirrors. Generally, this is the default **Active Mesh**. Other options may be useful when cloning multiple items (such as with the Scatter Clone tool).
Replica Effector	
Source	Determines the source mesh that Modo mirrors. • **Active Item** - Use the currently selected geometry as the clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Preset Shape** - Clones a profile shape selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a profile before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Mirror in the Procedural

Mirror is also available as a procedural mesh operation. It is a combination of the Mirror Generator and the Clone Effector sub-tools. You can add the tool from the **Mesh Ops** tab, on the right-side of the **Model** layout.

To add the Mirror mesh operation:

1. Click the **Add Operator** button, and navigate to **Mesh Operations** > **Duplicate** > **Mirror**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

The Mirror mesh operation duplicates the geometry below it in the list on the selected axis.

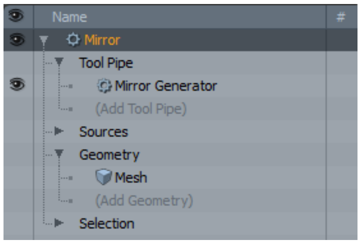
You can adjust the axis and other settings using the Mirror Generator sub-tool's **Properties** panel in the lower-right. To select the Mirror Generator sub-tool:

1. In the **Mesh Operations** list, expand the Mirror item by clicking the small arrow in front of it.
2. Expand the **Tool Pipe** and click Mirror Generator.

   The Mirror Generator's properties are displayed in the **Properties** panel in the lower-right of the interface. The Mirror mesh operation has the same properties as the traditional Mirror tool. See Mirror Properties.

Note: When **Mode** is set to **Axis**, the **Left** and **Up** fields don't work. To use these fields, set **Mode** to **Free Rotation** or **Three Points**.

The Mirror mesh operation uses the following inputs:

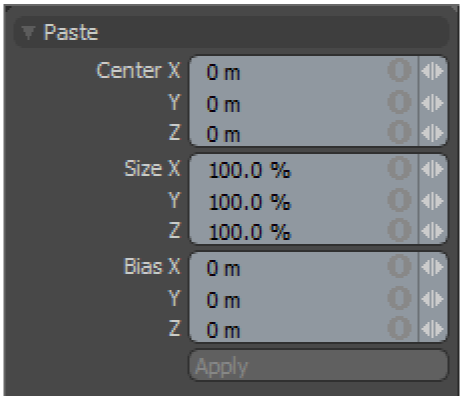


- **Tool Pipe** - Allows you to add sub-tools to the operation. For the Mirror tool, Mirror Generator is added by default.
- **Sources** - Allows you to specify the geometry you want to duplicate.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.
- **Selection** - Allows you to select the elements to which you want to apply the tool.

Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

Paste Tool

With the Paste tool you can quickly add copies of any geometry currently in the Clipboard. Activating the tool initially creates a copy of the geometry in the Clipboard. You can edit the copied material with the standard tool handles for position (**Center**) and scale (**Size**). Additionally, the Paste tool has options for shifting the mesh inside the handles.



Center X 0 m
Y 0 m
Z 0 m
Size X 100.0 %
Y 100.0 %
Z 100.0 %
Bias X 0 m
Y 0 m
Z 0 m

Paste	
Center X/Y/Z	Indicates the precise numeric placement of the center position of the Paste tool handles. You can interactively set the center position by dragging the center handle in the 3D edit viewport, or you can type the values directly in the **Properties** panel.
Size X/Y/Z	Scales the original mesh from the Clipboard. You can interactively set the center position by dragging the handles in the 3D edit viewport, or you can type the values directly in the **Properties** panel. A percentage value of **100%** equals the original size of the geometry in the Clipboard.
Bias X/Y/Z	Shifts the mesh inside the tool handles' cage. This is useful when you want to paste a mesh onto existing geometry (such as pasting rivets into a mechanical model). By default, Modo centers the pasted mesh on the tool handles. If you paste a mesh with the center handle constrained to the existing mesh, the pasted mesh is inset into the existing geometry. By adjusting the **Bias** values, you can shift the mesh so that it rests on the center handle rather than being centered on it.

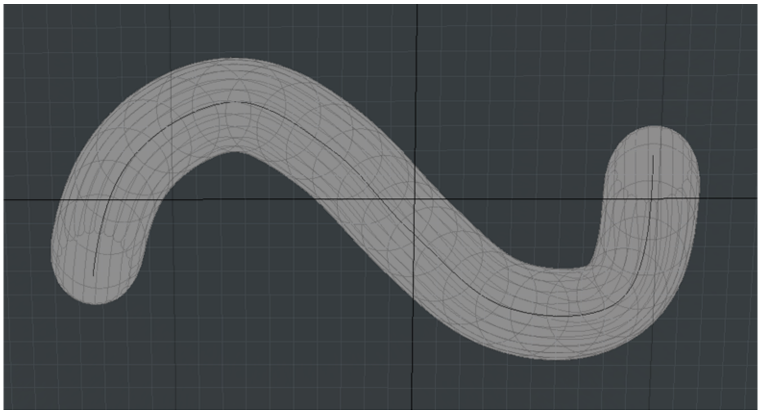
Path Segment Generator

The Path Segment Generator sub-tool is a procedural tool that allows you to create particles along a curve. You can use this together with other tools. For example, combining it with the Path Generator and the Sweep Effector tools provides the functionality of the Curve Extrude.

Using Path Segment Generator

To demonstrate the use of the Path Segment Generator, let's use the above Curve Extrude example, and combine it with a Sweep Effector and a Path Generator. We'll create a sphere and place instances of it along a curve, like the image below.

> **Tip:** The combination of these three tools is available as the procedural Curve Extrude tool. For more information, see Curve Extrude in the Procedural.



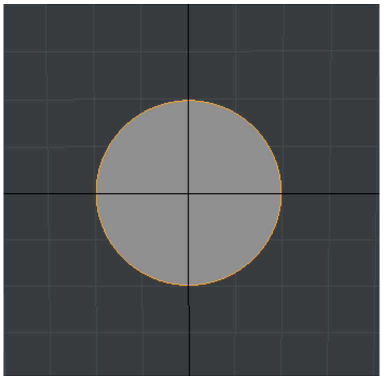
1. In the **Mesh Operations** list, create a procedural sphere by clicking the **Add Operator** button, then under **Mesh Operations** > **Create**, double-click **Sphere**.

> **Tip:** Alternatively, you can enter "sphere" in the search bar at the top of the Preset Browser to quickly find the operation.

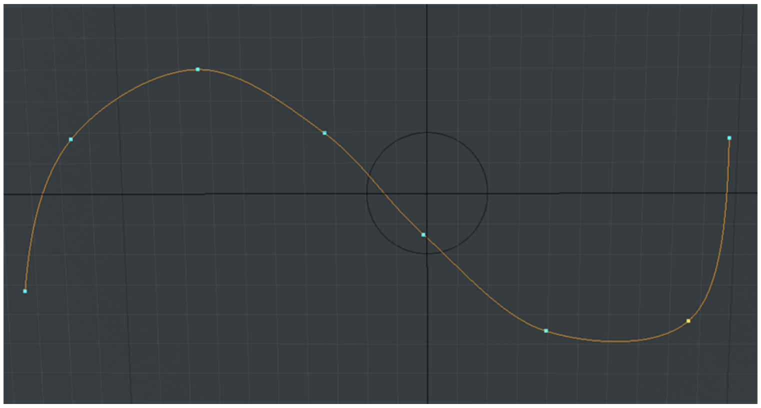
In this example we use the following parameters for the sphere:

- **Radius X**: 100 mm
- **Radius Y**: 0 mm
- **Radius Z**: 100 mm.

You can edit them in the sphere's **Properties** panel below the **Mesh Operations** list.

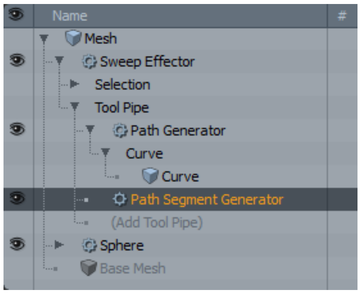


2. Add another mesh layer by clicking the **Add Operator** button, then under **Mesh Operations** > **Other**, double-click **Mesh**.
3. Add a curve by from the Modeling toolbox on the left side of the interface: Click the icon to activate the tool, then start clicking down control points for the curve in the 3D viewport.



4. In the original mesh layer, add the Sweep Effector tool to the **Mesh Operations** list. In the Preset Browser, you can find it under **Mesh Operations** > **Duplicate**.
5. Expand the Sweep Effector in the **Mesh Operations** list to reveal its Tool Pipe, then click **(Add Tool Pipe)**, and add a Path Generator and a Path Segment Generator.
6. Expand the Path Generator to reveal its **Curve** input, then click **(Add Curve)**, and double-click the mesh layer that contains your curve.

   Modo uses the sphere to walk along the curve. At this point, the **Mesh Operations** list looks like this:


Name
#
Mesh
Sweep Effector
Selection
Tool Pipe
Path Generator
Curve
Curve
Path Segment Generator
(Add Tool Pipe)
Sphere
Base Mesh

The Path Segment Generator has the same properties as the **PathSteps Generator** component of the Curve Extrude tool. For more information, see the Curve Extrude topic.

Pen Clone

The Pen Clone tool is useful for creating linear paths that have perfectly straight line segments between each user-defined point. You can clone geometry or items along this path based on the settings of the Clone effector. Modo duplicates each cloned item as a straight mesh and, thereby, increases the polygon count within the scene file. You can also clone items as instances or as Replicator Items, which do not increase scene geometry.

To activate the Pen Clone tool, choose **Geometry > Duplicate > Pen Clone**. You may also select the tool directly in the **Duplicate** sub-tab of the modeling toolbox in the **Model** and **Model Quad** interface tabs. If you click and hold the **Curve Clone** button, you can select the alternate tool options.

To use the tool, follow these steps:

1. Click the **Pen Clone tool** icon.
2. Click in the 3D viewport to create path points at the intersection of the viewport and the Work Plane.

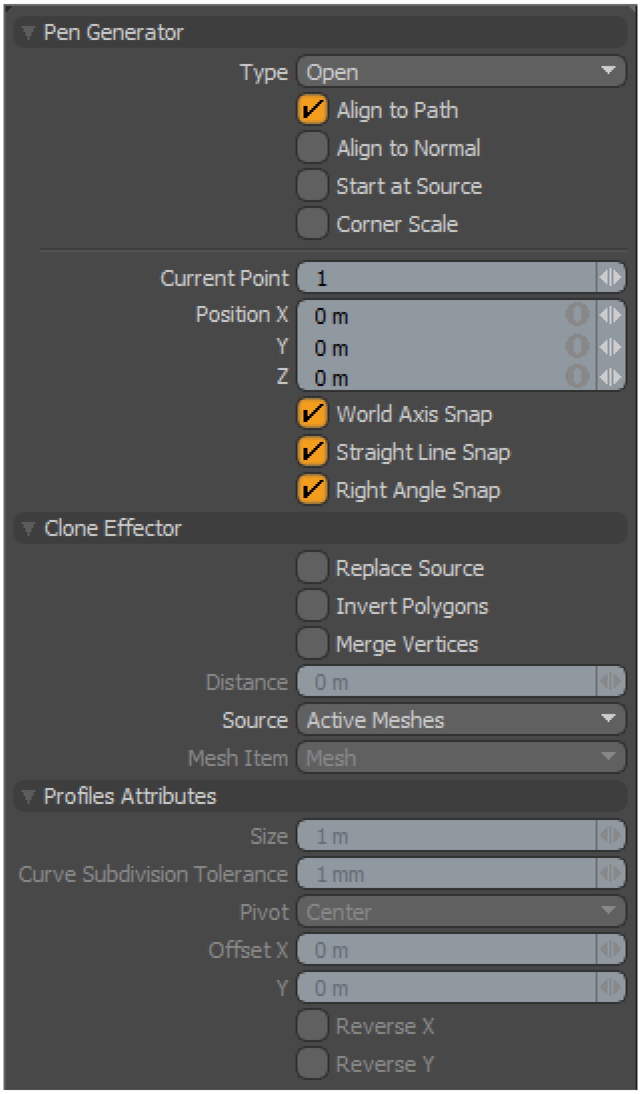
   Modo creates lines between each path point as you add subsequent points.

You can edit selected path points by specifying different values in the **Position** fields.

1. To select a particular path, move the pointer over the point until it becomes yellow.
2. Next, drag the point to change its position.

   You can also add points between existing positions by selecting a point and then clicking away from the selected point to add the intermediate point.

After pressing the **spacebar** key, the 'pen' feature drops and the 'clone' feature is in effect. When this occurs, you can no longer edit the path.



Pen Generator

Type: Open

☑ Align to Path

☐ Align to Normal

☐ Start at Source

☐ Corner Scale

Current Point: 1

Position X: 0 m

Y: 0 m

Z: 0 m

☑ World Axis Snap

☑ Straight Line Snap

☑ Right Angle Snap

Clone Effector

☐ Replace Source

☐ Invert Polygons

☐ Merge Vertices

Distance: 0 m

Source: Active Meshes

Mesh Item: Mesh

Profiles Attributes

Size: 1 m

Curve Subdivision Tolerance: 1 mm

Pivot: Center

Offset X: 0 m

Y: 0 m

☐ Reverse X

☐ Reverse Y

Pen Generator	
Type	Determines the termination of the path. • **Open** - Remains open as you create the path. • **Closed** - Draws a line between the first and last points to make the path a closed loop.
Align to Path	When enabled, Modo rotates the cloned items to align along the path's directional vector. Modo rotates the objects to match the direction of the path at the clone's position.
Align to Normal	When enabled, Modo tries to align the direction of the source objects to a straight up position before cloning the source item(s). If the item is

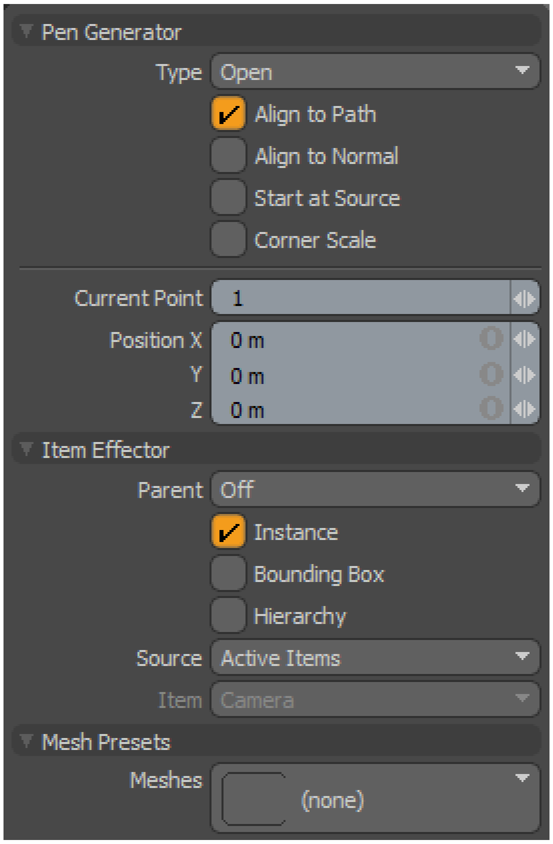
Pen Generator	
	already upright when cloned, this option has no effect.
Start at Source	When enabled, Modo positions cloned items based on their center point. When disabled, attempts to place the base of the cloned geometry at each path point. This is useful when cloning items in the same layer.
Corner Scale	When enabled, Modo scales corners automatically to make the path have a consistent thickness. (This option is also available for cloning items should you want clones to match a path created previously with the Pen tool.)
Current Point	Displays a number for the currently selected path point. You can select any point along the path by clicking directly on the point's handle when yellow or specify a point manually by adjusting the number.
Position X/Y/Z	Determines the position in 3D space where the currently selected point is. To move the currently selected point, you can type a new position or drag the point in the 3D viewport.
World Axis Snap	When enabled, attempts to snap a path's line segments to 90° angles relative to the World Axes.
Straight Line Snap	When enabled, attempts to snap a path's line segments to straight lines aligned with the previous two points in the path.
Right Angle Snap	When enabled, attempts to snap a path's line segments to 90° angles relative to the previous path segment.
Clone Effector	
Replace Source	When enabled, removes the source geometry and positions it at the first point in the path.
Invert Polygons	When enabled, inverts the normal direction of the cloned geometry.
Merge Vertices	When enabled, merges coinciding vertices into a single vertex.
Distance	Determines the maximum distance between vertices where merging occurs if you select **Merge Vertices**. Vertices outside this distance remain unchanged.

Pen Generator	
Source	Determines the mesh that Modo clones along the path. • **Active Meshes** - Uses the currently selected geometry as a clone source. • **Specific Mesh** - Uses a specific Mesh Item as the clone source as indicated in the **Mesh Item** option. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Preset Shape** - Clones a profile shape selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a profile before activating the tool.
Mesh Item	Specifies the mesh to be cloned. This is only relevant if you set **Source** to **Specific mesh**.
Profile Attributes - When you define the **Source** as **Preset Shape**, the attributes become active for you to specify the size and position of the cloned preset shape.	
Size	Determines the size of the cloned preset shape.
Curve Subdivision Tolerance	Determines the fidelity of how Modo converts the preset shape's curve into a polygon.
Pivot	Determines the position of the cloned preset shape in relation a point of the path.
Offset X/Y	Provides finer control over the preset shape's position in relation to the path itself. Modo adds this value to the **Pivot** position.
Reverse X/Reverse Y	Flips the preset shape in either the **X** or **Y** direction.

Pen Instance

This tool works like the Pen Clone tool: you can create a linear path with straight line segments between each user-defined point, and duplicate your geometry along the path. However, instead of producing actual geometry, the Pen Instance tool creates instances of the source item along the path.

You can activate the tool in the **Model** layout, under the **Duplicate** sub-tab. Click the **Curve Clone** icon and select **Pen Instance** from the list.



Pen Generator
Type: Open
☑ Align to Path
☐ Align to Normal
☐ Start at Source
☐ Corner Scale
Current Point: 1
Position X: 0 m
Y: 0 m
Z: 0 m
Item Effector
Parent: Off
☑ Instance
☐ Bounding Box
☐ Hierarchy
Source: Active Items
Item: Camera
Mesh Presets
Meshes: (none)

In addition to the Pen Generator properties described above, the following parameters are available for the Pen Instance tool.

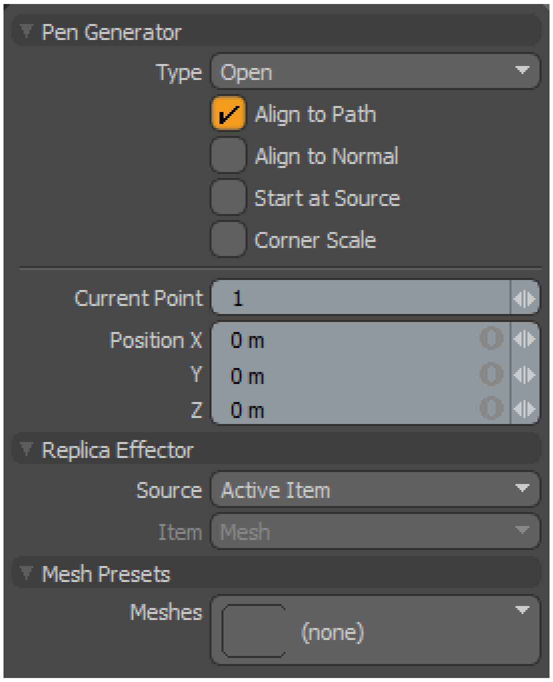
Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**. • **Off** - Clones source item as additional instanced layers in the **Item List**. • **Hierarchy** - Creates a cascading array where each subsequent clone is a child of the previous clone. • **Root** - Creates instances as child items of the source mesh. • **Primary** - Creates Instances as children of a group locator that is parented to the source mesh.
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source

Item Effector	
	layer and are represented in 3D Viewports by pink wireframes. This is enabled by default.
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, Modo clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo clones along the path. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Pen Replicate

This tool works like the Pen Clone tool, but instead of producing actual geometry, it creates replicas of the source item.

To activate the tool, click the **Curve Clone** icon and select **Pen Replicate** from the list.



Pen Generator

Type: Open

☑ Align to Path

☐ Align to Normal

☐ Start at Source

☐ Corner Scale

Current Point: 1

Position X: 0 m

Y: 0 m

Z: 0 m

Replica Effector

Source: Active Item

Item: Mesh

Mesh Presets

Meshes: (none)

In addition to the Pen Generator properties for the Pen Clone tool, you can manipulate the following properties of the Pen Replicate tool.

Replica Effector	
Source	Determines the source mesh that Modo clones along the path. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Transform Pen

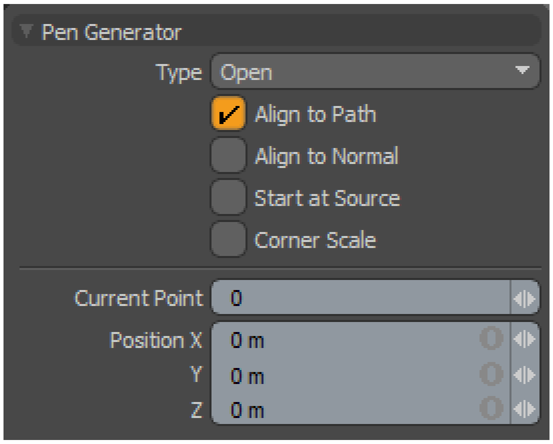
The **Transform Pen** tool allows you to edit the position of your existing items. For example, if you have multiple lights, you can distribute them in your scene with this tool.

To activate the tool, click the **Curve Clone** icon and select **Transform Pen** from the list.

To re-arrange your items using the Transform Pen, follow these steps.

1. Select all the items that you would like to arrange.
2. Activate the Transform Pen tool.
3. In the 3D viewport, click to the locations you would like to place your items.

   As you click, Modo relocates your items to the selected positions in the order they were initially selected.
4. When you are ready with placing your items, press **Q** to drop the tool.



Pen Generator

Type: Open

☑ Align to Path

☐ Align to Normal

☐ Start at Source

☐ Corner Scale

Current Point: 0

Position X: 0 m

Y: 0 m

Z: 0 m

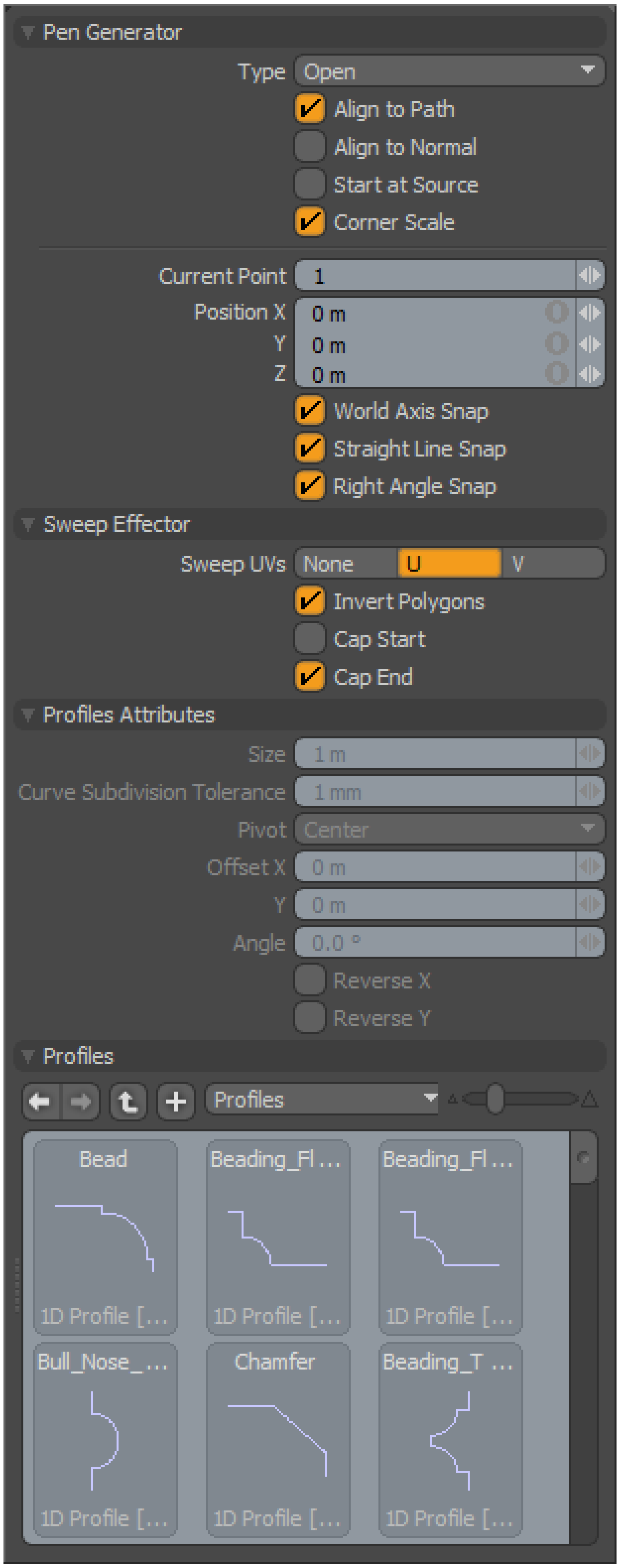
The following properties are available for this tool.

Pen Generator	
Type	Determines the termination of the path. • **Open** - Remains open as you create the path. • **Closed** - Draws a line between the first and last points to make the path a closed loop.
Align to Path	When enabled, Modo rotates the cloned items to align along the path's directional vector. Modo rotates the objects to match the direction of the path at the clone's position.

Pen Generator	
Align to Normal	When enabled, Modo tries to align the direction of the source objects to a straight up position before cloning the source item(s). If the item is already upright when cloned, this option has no effect.
Start at Source	When enabled, Modo positions cloned items based on their center point. When disabled, attempts to place the base of the cloned geometry at each path point. This is useful when cloning items in the same layer.
Corner Scale	When enabled, Modo scales corners automatically to make the path have a consistent thickness. (This option is also available for cloning items should you want clones to match a path created previously with the Pen tool.)
Current Point	Displays a number for the currently selected path point. You can select any point along the path by clicking directly on the point's handle when yellow or specify a point manually by adjusting the number.
Position X/Y/Z	Determines the position in 3D space where the currently selected point is. To move the currently selected point, you can type a new position or drag the point in the 3D viewport.

Pen Extrude

With the Pen Extrude tool you can quickly extrude a geometry selection along the length of a pen path where each knot creates a new segment. To use, select the tool and make it interactive by clicking in the viewport.



Pen Generator

Type: Open

☑ Align to Path

☐ Align to Normal

☐ Start at Source

☑ Corner Scale

Current Point: 1

Position X: 0 m

Y: 0 m

Z: 0 m

☑ World Axis Snap

☑ Straight Line Snap

☑ Right Angle Snap

Sweep Effector

Sweep UVs: None [✓ U] V

☑ Invert Polygons

☐ Cap Start

☑ Cap End

Profiles Attributes

Size: 1 m

Curve Subdivision Tolerance: 1 mm

Pivot: Center

Offset X: 0 m

Y: 0 m

Angle: 0.0 °

☐ Reverse X

☐ Reverse Y

Profiles

Profiles
Bead
1D Profile [...
Beading_Fl ...
1D Profile [...
Beading_Fl ...
1D Profile [...
Bull_Nose_ ...
1D Profile [...
Chamfer
1D Profile [...
Beading_T ...
1D Profile [...

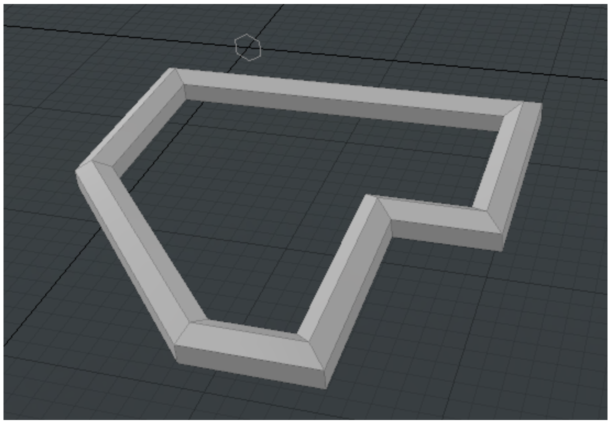
Pen Generator	
Type	Determines the termination of the path. • **Open** - Remains open as you create the path. • **Closed** - Generates an extra path between the first and last knots to make a closed loop.
Align to Path	When enabled, aligns segment edges with the curve's direction vector to produce a consistent thickness.
Align to Normal	Aligns the extruded shape to the normal direction of the initially selected polygon.
Start at Source	When enabled, uses the curve as a relative guide based on the original geometry as the true starting location. When disabled, may cause the newly created geometry to move away from the intended original location. (An extrusion may have its initial control knot imperfectly aligned with the geometry to be extruded.)
Corner Scale	When enabled, scales corners automatically to maintain the proper thickness over the extrusion's length.
Current Point	Displays a number for the currently selected vertex. Because the tool may have created several vertices, you may want to select one numerically. Modo numbers the vertices in sequence as you generate them with the tool. You can select a different vertex to edit its the **Position** values.
Position X/Y/Z	Determines the position in 3D space where the currently selected vertex is. To move the currently selected vertex, you can type a new position.
World Axis Snap	When enabled, snaps the pen tool handles along the World XYZ axes once you create more than one vertex.
Straight Line Snap	Tries to snap the handle to a position for creating a straight line between the next and previous vertices in the polygon being created.
Right Angle Snap	Snaps when the angle created by the current, previous, and next vertices form a 90-degree angle.

Pen Generator	
Sweep Effector	
Sweep UVs	Specifies whether to use the **U** or **V** axis when placing UV data for newly created geometry.
Invert Polygons	Reverses the normal facing direction of the polygons generated by the extrusion.
Cap Start/Cap End	Forces the back end of the extruded geometry to be capped or closed.
Profile Attributes	
Size	Specifies the size of the 1D or 2D profile to sweep. If the size is **0.0**, Modo uses the real size.
Curve Subdivision Tolerance	Controls the detail tolerance of the conversion from the original curve to the resulting polygon extrusion. This indicates the minimum angle used when Modo subdivides the Bezier curve to a string of two-point polygons for extrusion.
Pivot	Specifies the pivot position of the profile.
Offset X/Y	Specifies the offset amounts on the profile space.
Reverse X/Reverse Y	Reverses the selected profile related to the horizontal or vertical axis.
Profiles	Modo shows a mini Preset Browser for viewing various profiles. This works the same as the standard Preset Browser. Selecting a profile modulates the extrusion's width with the selected shape over its length. To return to the original polygon selection for the extrusion, press **Ctrl** as you click the currently selected profile.

Pen Generator

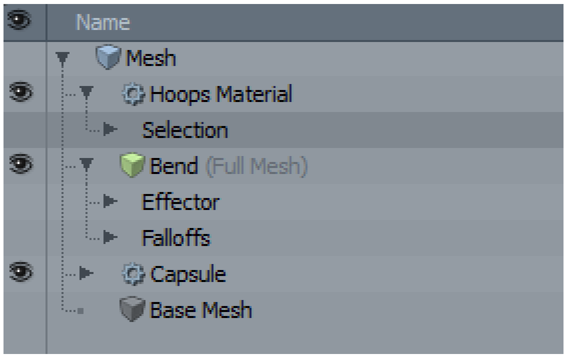
The **Pen Generator** tool allows an existing polyline to provide a path for other tools to walk and evaluate. This can be useful for things like extruding or transforming shapes along a polyline and controlling positions of elements along a polyline.

Note: The difference between the Pen Generator and the Path and Path Segment Generator tools is that the Pen Generator tool provides both the path and the particles to place along the path.



The Pen Generator has a single input for a mesh item containing a polyline to use as a path. The Pen Generator must be added to a Sweep Effector, which draws the resulting geometry. The Sweep Effector has an input for the mesh that is to be drawn along the polyline. This source mesh can be of any type.

The screen capture below shows the **Mesh Ops** tab for the image above. In this case, the swept mesh is a hexagon (made using the N-Sided primitive).



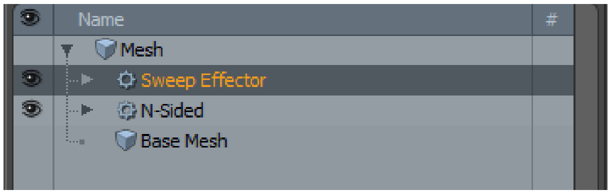
Name
- Mesh
  - Hoops Material
    - Selection
  - Bend (Full Mesh)
    - Effector
    - Falloffs
  - Capsule
  - Base Mesh

Using the Pen Generator

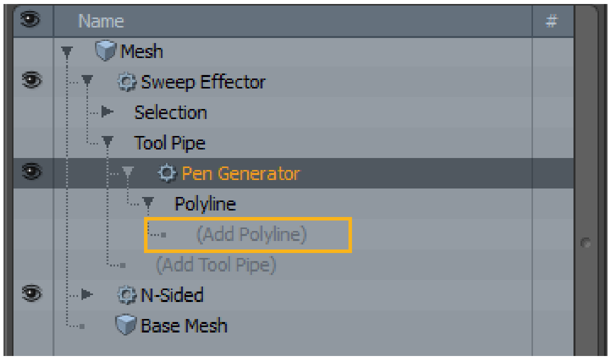
The following steps show how to use the Mesh Operations List to use the Pen Generator. In this example, the polyline is drawn using the Pen tool in a new mesh. The steps below start from scratch and build the mesh pictured above

Note: If you already have a mesh in the stack to use as an input to the Sweep Effector, skip to step 3.

1. Add a procedural mesh to the stack. In this case, let's add an N-Sided primitive by clicking **Add Operator** in the Mesh Operations List, and double-clicking **N-Sided** under **Mesh Operations** > **Create** > **Curves**.
2. In the **Properties** panel, set **Sides** to **6** to make a hexagon.
3. Click **Add Operator** again to open the Preset Browser.
4. Under **Mesh Operations** > **Duplicate**, double-click **Sweep Effector**. This adds a Sweep Effector to the stack.

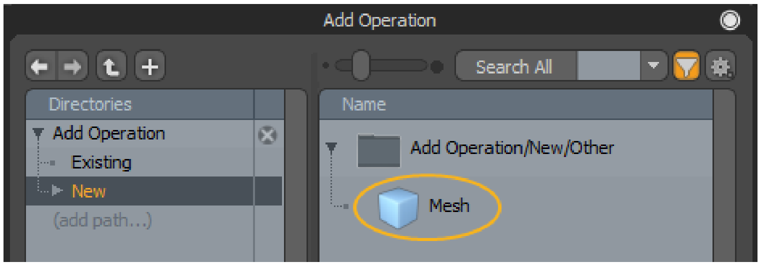

Name
#
Mesh
Sweep Effector
N-Sided
Base Mesh

5. In the stack, expand **Sweep Effector** to reveal the **Tool Pipe**, and click **(Add Tool Pipe)**.
6. Double-click on **Pen Generator** to add it the Sweep Effector's tool pipe.
7. In the stack, expand **Pen Generator** and expand **Polyline**, click **(Add Polyline)**.


Name
#
Mesh
Sweep Effector
Selection
Tool Pipe
Pen Generator
Polyline
(Add Polyline)
(Add Tool Pipe)
N-Sided
Base Mesh

This opens the **Preset Browser**.

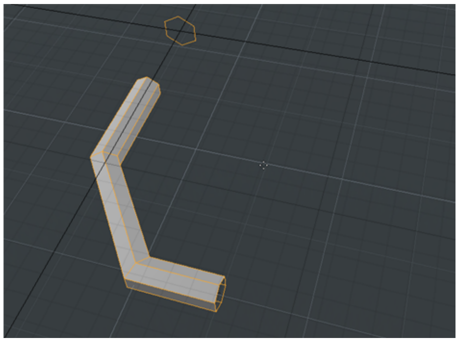
8. Let's add a new mesh to hold the polyline. In the **Preset Browser**, ensure the **New** category is selected, and double-click on **Mesh** in the right-hand column.


Add Operation
Search All
Directories
Add Operation
Existing
New
(add path...)
Name
Add Operation/New/Other
Mesh

9. You can now draw the polyline in the new mesh. Select the Pen tool (in the **Basic** sub-tab of the **Model** tools) and set the **Type** to **Polyline** in the tool's properties.

Note: This Pen Generator will not work if you do not set the Pen tool type to **Polyline**.

10. Draw the polyline by clicking in the 3D viewport. The source polygon is swept along the path.



Tip: If you want to draw along the ground plane, it helps to set the view to **Top** in the viewport.

Pen Generator Properties

You can customize the behavior of the Pen Generator using the properties below.

Pen Generator	
Polyline Index	Selects a polyline to use as a path from the input mesh, using its index value.
Align to Path	When enabled, Modo rotates the particle angle to align it with the path's directional vector. (An effector tool rotates the objects to match the direction of the path at the particle's position.)
Align to Normal	When enabled, Modo tries to align the direction of the source objects to a straight up position before generating the particle orientation.
Start at Source	When enabled, Modo positions particles based on their center point. When disabled, Modo attempts to place the base of the target geometry at each path point.
Corner Scale	When enabled, Modo scales corners automatically so that the path has a constant thickness.

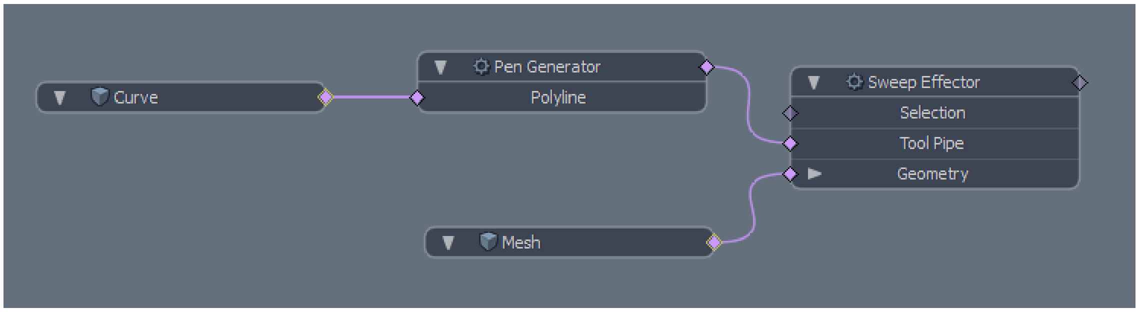
Pen Generator Procedural

The Pen Generator has the following procedural sub-tool:

- **Polyline** - This references a mesh containing a polyline that is used to describe the path.

Pen Generator Schematic

The image below shows an example of the Pen Generator when used with a Sweep Effector.



Procedural Clone Effector

The procedural version of the Clone Effector is a sub-tool, which allows you to duplicate source meshes at the particle positions provided by a tool generator connected to its Tool Pipe.

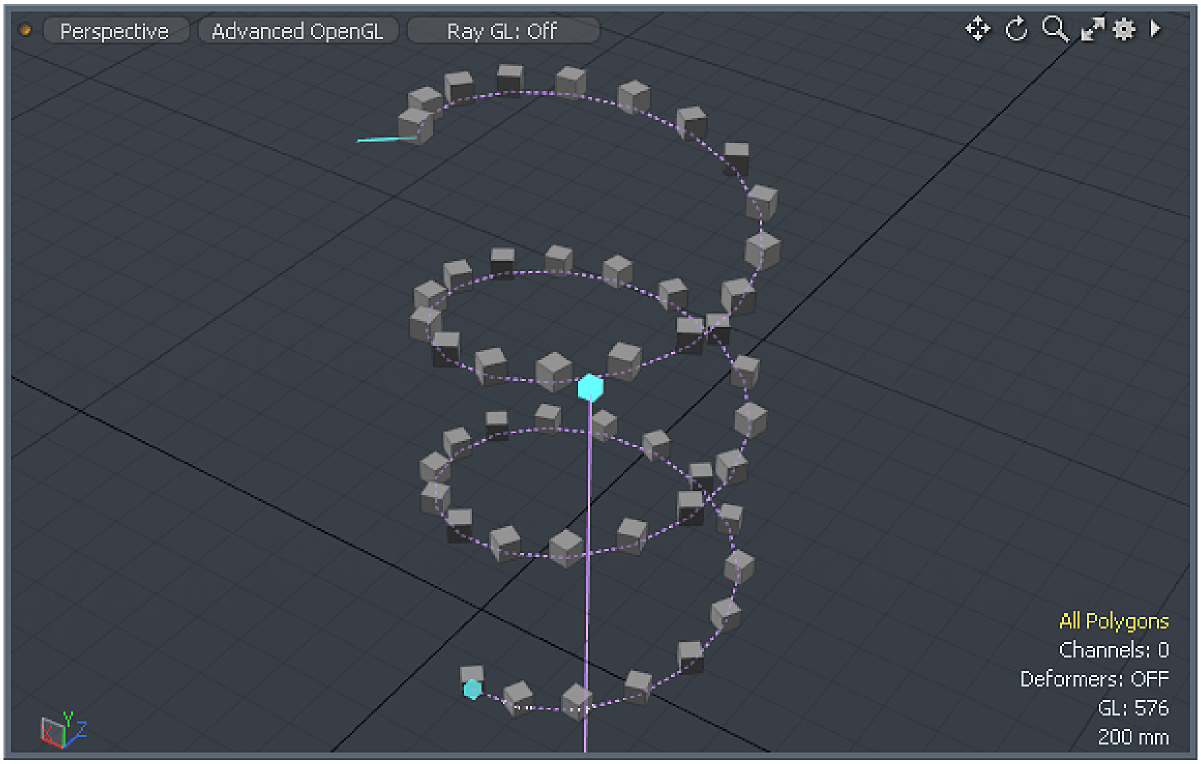
This sub-tool is used as a part of Clone, Array, Radial Clone, and Curve Clone procedural tool operators.

You can add the tool in the **Mesh Operations** list, by clicking **Add Operator**.

The Clone Effector has the following input options:

- **Tool Pipe** - Allows you to add sub-tools to the operation. For example, you can add an Array Generator operation to create geometry on 3D grid positions.
- **Sources** - Allows you to specify the geometry you want to duplicate.
- **Selection** - Allows you to select the elements to which you want to apply the tool.

Radial Array



The Radial Array tool recursively clones any selected geometry into a circular or helical pattern. With the tool's **Properties** panel, you have precise control over the count, placement, angles, and spiral offset distance.

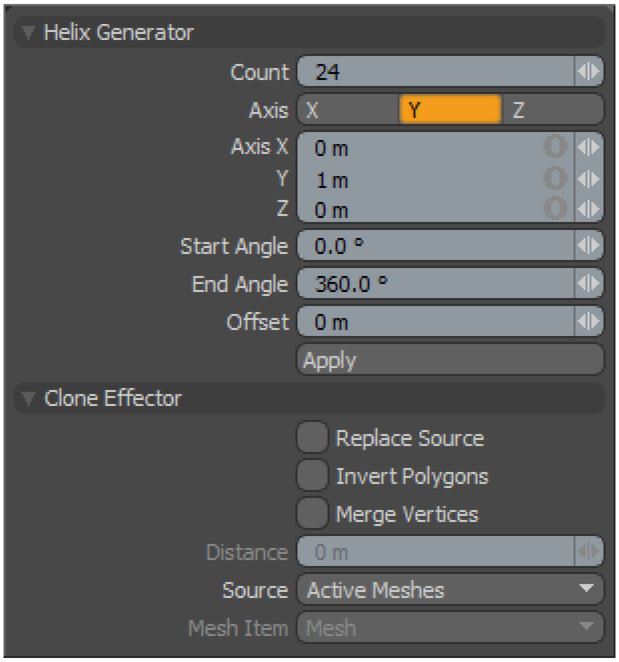
Usage

Before creating the array, select the target geometry for duplication. Component selections (such as vertices, edges, or polygons) clone into their own respective layers; item selections generate multiple additional item layers with their transform values set appropriately.

To activate the tool, with an item selected, click the **Radial Array** button in the modeling layout toolbox in the **Duplication** sub-tab. In this state, you can enter values in the tool's **Properties** panel and click **Apply** to generate the array. Additionally, when the tool is active, you can click in the 3D viewport to interactively adjust the values. The first click defines the center of rotation, which is at the intersection of the pointer and the current Work Plane. Alternatively, to add a procedural Radial Array, in the **Mesh Operations** list, click the **Add Operator** button, and navigate to **Mesh Operations** > **Duplicate** > **Radical Array**, or enter its name in the search bar. For more information, see Radial Array in the Procedural.

The helix angle control surrounds the center line that defines the **Axis** angle. Drag over the blue cube handle to adjust the **Start Angle** and drag to adjust the **End Angle**. You can use the blue cubes at the end of the line to adjust the axis angle for arrays that don't align with any of the three axes. You can also click again away from the handles to reposition the center location. While the tool is active, you can manually enter values in the **Properties** panel and see the changes in the 3D viewport.

Radial Array Properties



Helix Generator

Count 24

Axis X [✓ Y] Z

Axis X 0 m

Y 1 m

Z 0 m

Start Angle 0.0 °

End Angle 360.0 °

Offset 0 m

Apply

Clone Effector

☐ Replace Source

☐ Invert Polygons

☐ Merge Vertices

Distance 0 m

Source Active Meshes

Mesh Item Mesh

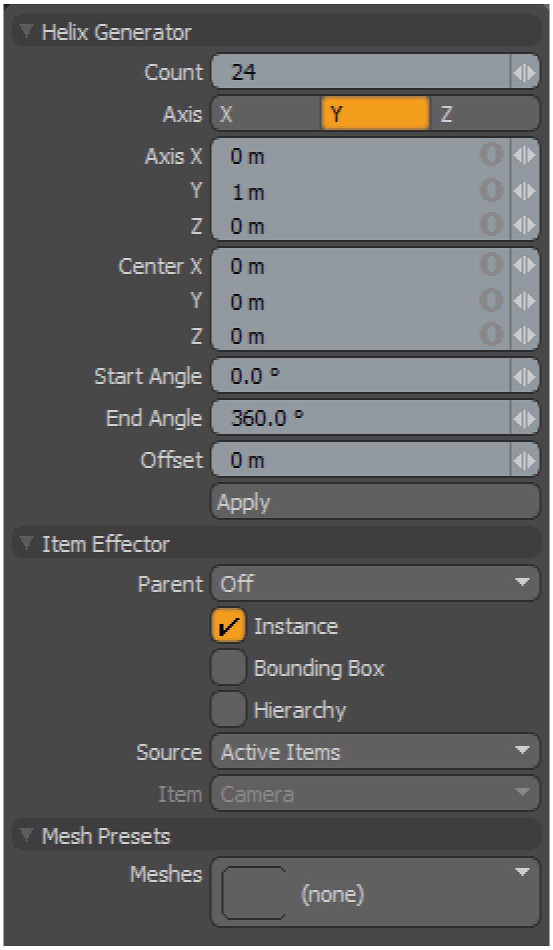
Helix Generator	
Count	Determines the total number of clones created.
Axis	Determines along which axis Modo creates the circular array.
Axis X/Y/Z	Determines the location of the tool's **Axis** control handles. If you set all the values to **1 m**, Modo sets one end at (-1,-1,-1) and the other end at (1,1,1) to create an skewed-off axis plane for the array.
Start Angle	Determines the rotation placement for the first cloned geometry as a number of degrees offset from the initial **0** degrees.
End Angle	Determines the rotation placement for the last cloned element as a number of degrees offset from **0** degrees.

Helix Generator	
	Note: By default, the **Start Angle** and **End Angle** are **0** to **360** degrees. This creates an array that completes a full circle. By changing these two values, you can create an array that revolves many times or one that leaves one side of the circle open. You may want to set the **End Angle** value greater than **360** degrees when combined with a non-zero **Offset** value to create a spiral.
Offset	Moves each successive clone a user-defined distance from the previous clone along the tools defined **Axis**. This is useful when combined with an **End Angle** greater than **360** degrees to create a spiral.
Replace Source	When enabled, replaces the original mesh element(s) being cloned by the tool with an additional clone at the **0**-degree position.

Instance Radial Array

This tool is similar to the Radial Array tool, but rather than creating geometry, it creates instances of the source in a circular or helical pattern. You can activate the tool by right-clicking on the **Radial Array** icon and selecting **Instance Radial Array** from the list.

Instance Radial Array Properties



Helix Generator
Count 24
Axis X Y Z
Axis X 0 m
Y 1 m
Z 0 m
Center X 0 m
Y 0 m
Z 0 m
Start Angle 0.0 °
End Angle 360.0 °
Offset 0 m

Item Effector
Parent Off
☑ Instance
☐ Bounding Box
☐ Hierarchy
Source Active Items
Item Camera
Mesh Presets
Meshes (none)

In addition to the **Helix Generator** properties described above, the following parameters are available for the Instance Radial Array tool.

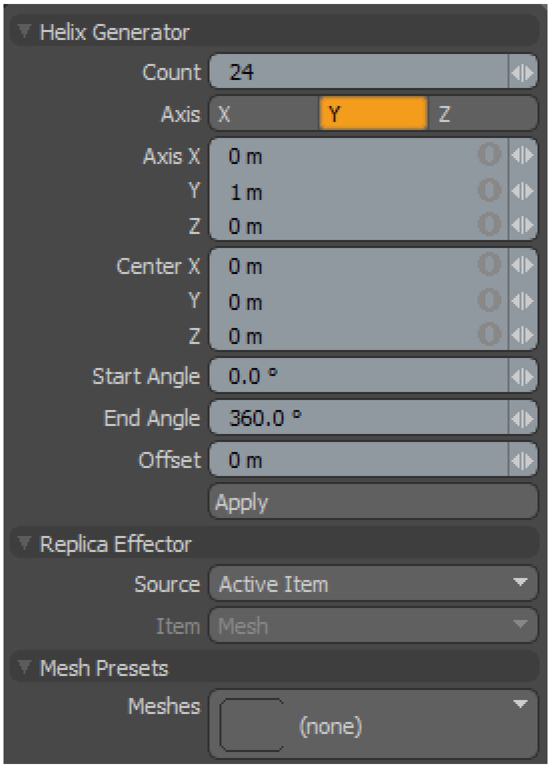
Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**. • **Off** - Clones source item as additional instanced layers in the **Item List**. • **Hierarchy** - Creates a cascading array where each subsequent clone is a child of the previous clone. • **Root** - Creates instances as child items of the source mesh. • **Primary** - Creates Instances as children of a group locator that is parented to the source mesh.

Item Effector	
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewports by pink wireframes. (This is enabled by default.)
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo clones into the array. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Replica Radial Array

This tool is similar to the Radial Array tool, but rather than creating geometry, it creates replicas of the source in a circular or helical pattern. You can activate the tool by right-clicking on the **Radial Array** icon and selecting **Replica Radial Array** from the list.

Replica Radial Array Properties


Helix Generator
Count 24
Axis X Y Z
Axis X 0 m
Y 1 m
Z 0 m
Center X 0 m
Y 0 m
Z 0 m
Start Angle 0.0 °
End Angle 360.0 °
Offset 0 m
Apply
Replica Effector
Source Active Item
Item Mesh
Mesh Presets
Meshes (none)

In addition to the **Helix Generator** properties described above, the following parameters are available for the Replica Radial Array tool.

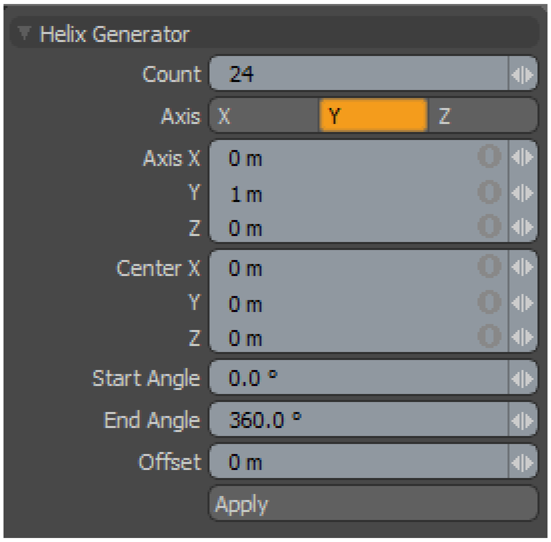
Replica Effector	
Source	Determines the source mesh that Modo clones into the array. • **Active Item** - Use the currently selected geometry as the clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Preset Shape** - Clones a profile shape selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a profile before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.

Replica Effector	
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Transform Radial Array

The Transform Radial Array tool creates arrays of existing items in a circular or helical pattern. You can activate the tool by right-clicking on the **Radial Array** icon and selecting **Transform Radial Array** from the list.

Transform Radial Array Properties



Count 24
Axis X [✓ Y] Z
Axis X 0 m
Y 1 m
Z 0 m
Center X 0 m
Y 0 m
Z 0 m
Start Angle 0.0 °
End Angle 360.0 °
Offset 0 m

Helix Generator	
Count	Determines the total number of clones created.
Axis	Determines along which axis Modo creates the circular array.
Axis X/Y/Z	Determines the location of the tool's **Axis** control handles. If you set all the values to **1 m**, Modo sets one end at (-1,-1,-1) and the other end at (1,1,1) to create an skewed-off axis plane for the array.
Start Angle	Determines the rotation placement for the first cloned geometry as a number of

Helix Generator	
	degrees offset from the initial **0** degrees.
End Angle	Determines the rotation placement for the last cloned element as a number of degrees offset from **0** degrees. **Note:** By default, the **Start Angle** and **End Angle** are **0** to **360** degrees. This creates an array that completes a full circle. By changing these two values, you can create an array that revolves many times or one that leaves one side of the circle open. You may want to set the **End Angle** value greater than **360** degrees when combined with a non-zero **Offset** value to create a spiral.
Offset	Moves each successive clone a user-defined distance from the previous clone along the tools defined **Axis**. This is useful when combined with an **End Angle** greater than **360** degrees to create a spiral.

Radial Array in the Procedural

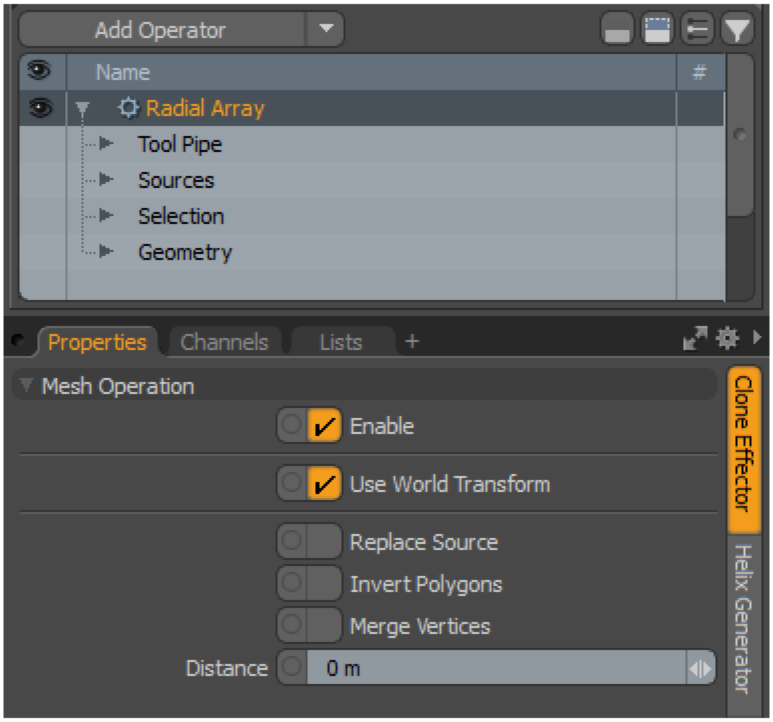
Radial Array is also available as a procedural mesh operation. It is a combination of the Clone Effector and the Helix Generator sub-tools. You can find the tool in the **Mesh Ops** tab, on the right-side of the **Model** layout.

Here's how you can add the Radial Array mesh operation:

1. In the **Mesh Ops** tab, click the **Add Operator** button, and navigate to **Mesh Operations** > **Duplicate** > **Radial Array**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

   The Radial Array mesh operation duplicates the geometry below it in the list. You can adjust the operation using its **Properties** panel in the lower-right.

You can expand the Radial Array item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses.


Add Operator
Name
#
Radial Array
Tool Pipe
Sources
Selection
Geometry
Properties
Channels
Lists
+
Mesh Operation
Clone Effector
Helix Generator

☑ Enable
☑ Use World Transform
☐ Replace Source
☐ Invert Polygons
☐ Merge Vertices
Distance 0 m

Radial Array can use the following inputs:

- **Tool Pipe** - Allows you to add sub-tools to the operation. For the Radial Array tool, a Helix Generator is added by default and the options are displayed in the Properties tab.
- **Sources** - Allows you to specify the geometry you want to duplicate.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.
- **Selection** - Allows you to select the elements to which you want to apply the tool.

Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

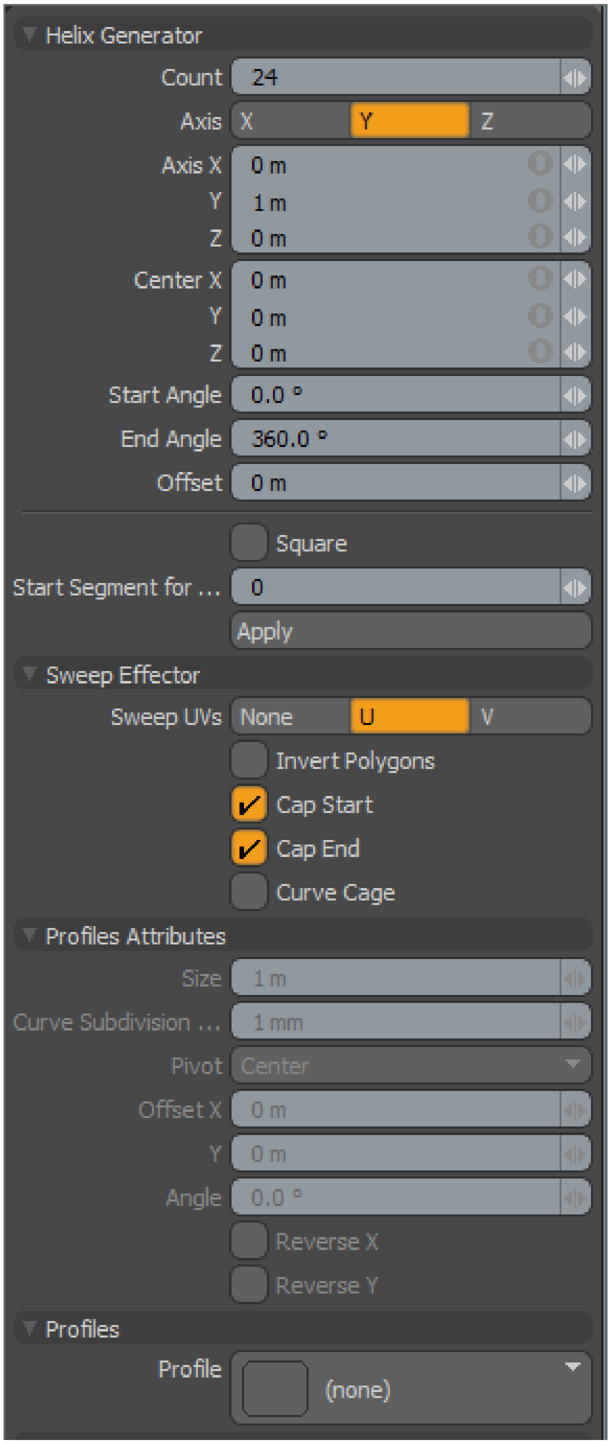
The Radial Array mesh operation has the same properties as the traditional Radial Array tool. See Radial Array Properties.

Radial Sweep

The Radial Sweep tool creates geometry by revolving the selection around a central point. The geometry can be a Modo curve profile or a polygon silhouette shape. You can use this tool to create simple round objects such as bottles, plates, or goblets. The Radial Sweep tool is also useful for creating objects such as springs by using the **Offset** control to not only spin the selected geometry while sweeping it but also move the new elements along the tool's axis. You may not receive satisfactory results when sweeping closed shapes such as cubes and spheres.

Activate the Radical Sweep tool by clicking **Radical Sweep** in the Duplicate sub-tab of the modeling toolbox.

Radial Sweep Properties



Helix Generator

Count: 24

Axis: X | Y | Z (Y selected)

Axis X: 0 m

Y: 1 m

Z: 0 m

Center X: 0 m

Y: 0 m

Z: 0 m

Start Angle: 0.0 °

End Angle: 360.0 °

Offset: 0 m

☐ Square

Start Segment for ...: 0

Apply

Sweep Effector

Sweep UVs: None | U | V (U selected)

☐ Invert Polygons

☑ Cap Start

☑ Cap End

☐ Curve Cage

Profiles Attributes

Size: 1 m

Curve Subdivision ...: 1 mm

Pivot: Center

Offset X: 0 m

Y: 0 m

Angle: 0.0 °

☐ Reverse X

☐ Reverse Y

Profiles

Profile: (none)

Helix Generator	
Count	Determines the number of new segments generated by the tool.
Axis	Determines along which axis to sweep the geometry.
Axis X/Y/Z	Determines the location of the tool's **Axis** control handles. If you set all the values to **1 m**, Modo sets one end at (-1,-1,-1) and the other end at (1,1,1) to create an skewed-off axis plane for the array.
Start Angle	Determines the rotational placement first lathed geometry.
End Angle	Determines the rotational placement for the last lathed geometry. **Note:** By default, the **Start Angle** and **End Angle** are **0** to **360** degrees. This creates an array that completes a circle. By changing these two values, you can create an array that revolves many times or one that leaves one side of the circle open. You may want to set the **End Angle** value greater than **360** degrees when combined with a non-zero **Offset** value to create a spiral.
Offset	Moves each successive segment a user-defined distance from the previous segment along the tool's axis. This is useful when combined with an **End Angle** greater than **360** degrees to create a spiral for items such as a spring or a telephone cord.
Square	Enables **Square** mode. The segments to sweep are aligned along a square.
Start Segment for ...	Specifies the offset of the start segment for **Square** mode.
Sweep Effector	
Sweep UVs	Determines along which UV axis (**U** or **V**) to span the newly created geometry.
Invert Polygons	Flips the polygons created by the tool. In some circumstances lathing geometry turns inside out. In this case, enable **Invert Polygons** to have the geometry face the correct direction.
Cap Start	Creates a cap on the start of the lathe to avoid a hole in the mesh.
Cap End	Creates a cap on the end of the lathe.

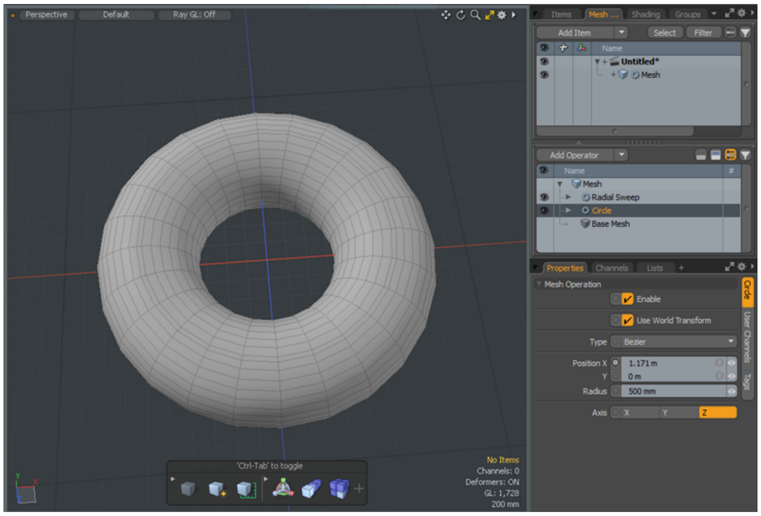
Helix Generator	
	Note: The **Cap** options are only useful when the start and end angles do not result in a complete 360 rotation. In this case, the start and end segments are joined together.
Profile Attributes - If you wish, you can have Modo sweep a preset profile as readily as selected geometry. Select a profile in the **Profiles** section of the tool's **Properties** panel.	
Size	Determines the size of the 1D or 2D profile to sweep. If the size is **0.0**, Modo uses the actual size.
Curve Subdivision Tolerance	Controls the detail tolerance of the conversion from the original curve to the resulting sweep. This is the minimum angle Modo uses when it subdivides the Bezier curve into a string of two-point polygons for extrusion.
Pivot	Specifies the pivot position of the profile.
Offset X/Offset Y	Specifies the distance of the profile space. Modo adds this value to the **Pivot** position.
Reverse X/Reverse Y	Flips the selected profile in the horizontal or vertical direction.

Radial Sweep in the Procedural

The Radial Sweep tool is also available as a procedural mesh operation. You can find the tool in the **Mesh Ops** tab. In the **Model** layout, the tab is on the right side of the interface, next to the **Item List**.

To add the Radial Sweep mesh operation:

1. Click the **Add Operator** button, expand **Mesh Operations** and double-click **Circle**, or enter its name in the search bar.
2. Click the **Add Operator** button, expand **Mesh Operations** > **Duplicate**, and double-click **Radial Sweep**, or enter its name in the search bar.
3. In the **Mesh Operations** list, select **Circle**, open the **Circle** tab in **Properties**, and set **Position X** to **1.171 m**.


Perspective
Default
Ray GL: Off
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Mesh
Add Operator
Name
Mesh
Radial Sweep
Circle
Base Mesh
Properties
Channels
Lists
Circle
User Channels
Tags
'Ctrl-Tab' to toggle
No Items
Channels: 0
Deformers: ON
GL: 1,728
200 mm

Mesh Operation

☑ Enable

☑ Use World Transform

Type: Bezier

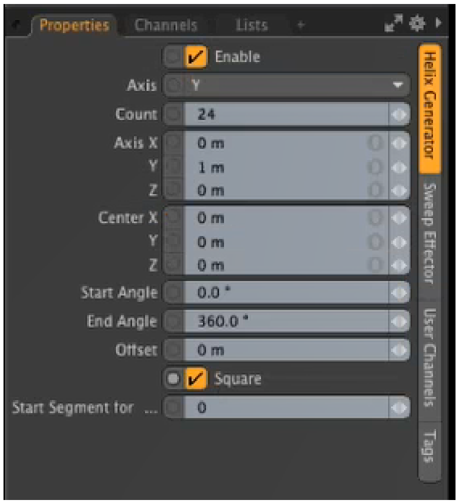
Position X: 1.171 m

Y: 0 m

Radius: 500 mm

Axis: X Y Z

4. In the **Mesh Operations** list, select the **Radial Sweep** item, open the **Helix Generator** tab, and enable **Square**.


Properties
Channels
Lists
Helix Generator
Sweep Effector
User Channels
Tags

☑ Enable

Axis: Y

Count: 24

Axis X: 0 m

Y: 1 m

Z: 0 m

Center X: 0 m

Y: 0 m

Z: 0 m

Start Angle: 0.0 °

End Angle: 360.0 °

Offset: 0 m

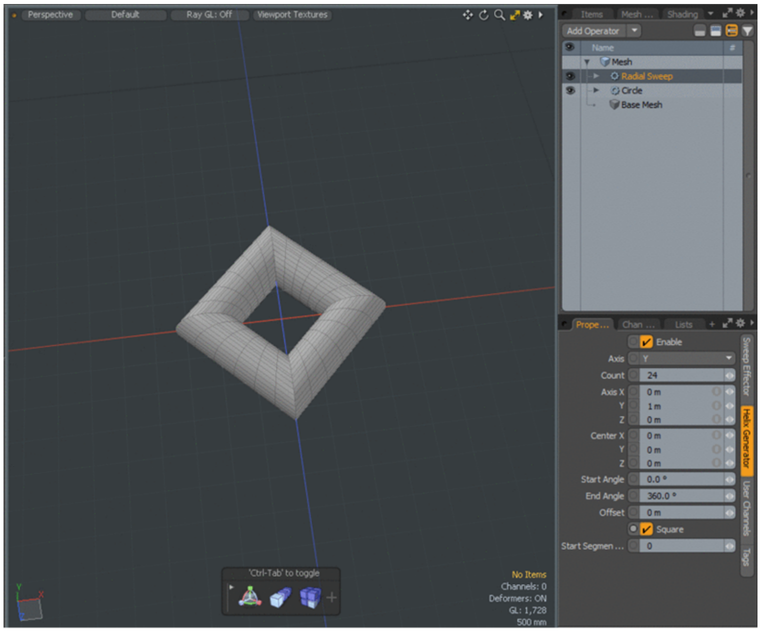
☑ Square

Start Segment for ...: 0

The circle is converted to a square.

5. Set **Start Segment for ...** to **3** and then increase the value to **5**, and then **7**.

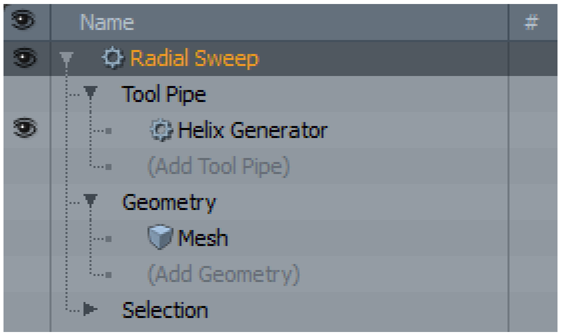
The offset for the start segment is updated in the 3D viewport.


Perspective
Default
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Add Operator
Name
Mesh
Radial Sweep
Circle
Base Mesh
Prope ...
Chan ...
Lists
Sweep Effector
Helix Generator
User Channels
Tags
'Ctrl-Tab' to toggle
No Items
Channels: 0
Deformers: ON
GL: 1,728
500 mm

Enable
Axis Y
Count 24
Axis X 0 m
Y 1 m
Z 0 m
Center X 0 m
Y 0 m
Z 0 m
Start Angle 0.0 °
End Angle 360.0 °
Offset 0 m
Square
Start Segmen ... 0

Radial Sweep Input Operations

Radial Sweep can use the following inputs:


Name
#
Radial Sweep
Tool Pipe
Helix Generator
(Add Tool Pipe)
Geometry
Mesh
(Add Geometry)
Selection

- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation. As you can see in the image above, when you add a Radial Sweep, Helix Generator is automatically added to its Tool Pipe. The procedural Helix Generator has the same properties as the traditional tool, described in Radial Sweep.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.
- **Selection** - Allows you to select the elements to which you want to apply the tool.

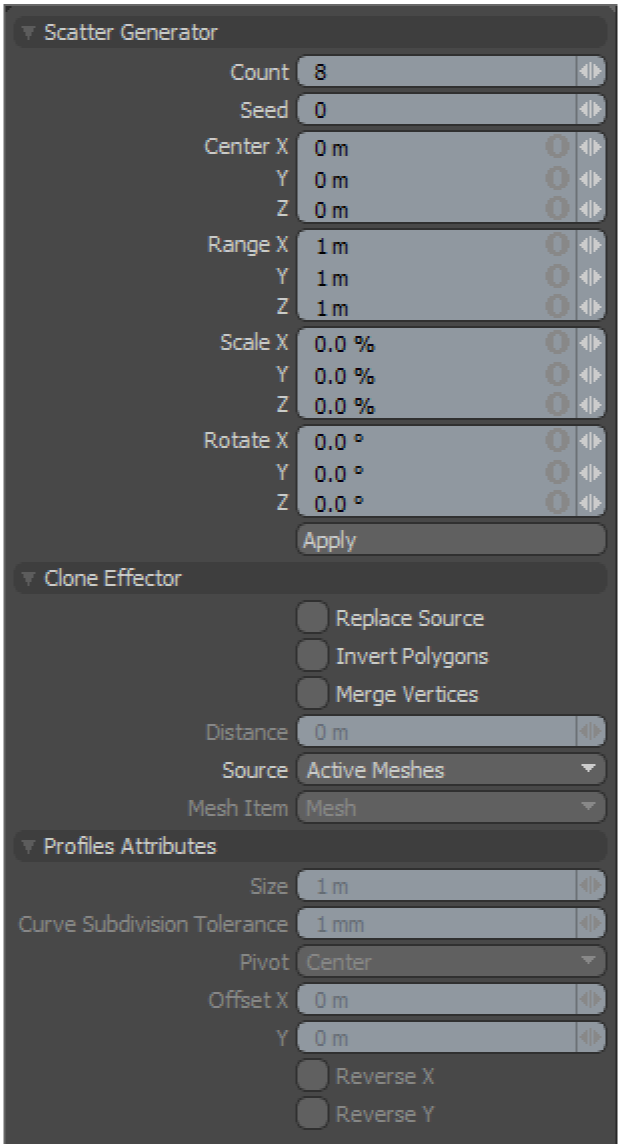
Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

The Radial Sweep mesh operation has the same properties as the traditional Radial Sweep tool. See Radial Sweep Properties.

Scatter Clone

The Scatter Clone tool generates a random point cloud onto which Modo places clones of the original geometry.

Scatter Clone Properties



Count 8
Seed 0
Center X 0 m
Y 0 m
Z 0 m
Range X 1 m
Y 1 m
Z 1 m
Scale X 0.0 %
Y 0.0 %
Z 0.0 %
Rotate X 0.0 °
Y 0.0 °
Z 0.0 °

☐ Replace Source
☐ Invert Polygons
☐ Merge Vertices
Distance 0 m
Source Active Meshes
Mesh Item Mesh

Size 1 m
Curve Subdivision Tolerance 1 mm
Pivot Center
Offset X 0 m
Y 0 m
☐ Reverse X
☐ Reverse Y

The following properties are available for the scatter clone tool.

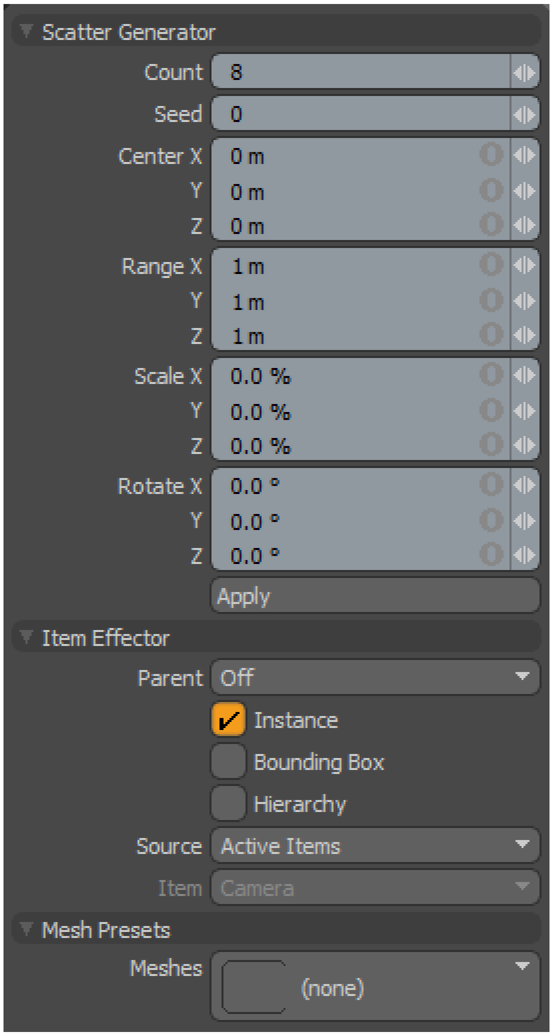
Scatter Clone	
Count	Specifies the total number of clones (including the source).
Center X/Y/Z	Determines the center position of the scatter clone effect.

Scatter Clone	
Range X/Y/Z	Defines the maximum distance on each axis that the clones can be from the center of the effect.
Scale X/Y/Z	Specifies the maximum amount of randomized fluctuation for any given clone.
Rotate X/Y/Z	Indicates the maximum amount of randomized rotation for any given clone.
Replace Source	When enabled, removes the original geometry and creates a replacement clone for it at the center of the scatter clone effect.

Instance Scatter

Similarly to Scatter Clone, the Instance Scatter tool generates a random point cloud onto which Modo places instances of the original geometry. To activate the tool, right-click the **Scatter Clone** icon and select **Instance Scatter** from the list.

Instance Scatter Properties



Scatter Generator

Count 8

Seed 0

Center X 0 m

Y 0 m

Z 0 m

Range X 1 m

Y 1 m

Z 1 m

Scale X 0.0 %

Y 0.0 %

Z 0.0 %

Rotate X 0.0 °

Y 0.0 °

Z 0.0 °

Item Effector

Parent Off

☑ Instance

☐ Bounding Box

☐ Hierarchy

Source Active Items

Item Camera

Mesh Presets

Meshes (none)

In addition to the **Scatter Generator** properties described above for the Scatter Clone tool, the following parameters can be set for the Instance Scatter tool.

Item Effector	
Parent	Determines how Modo treats instanced items in the **Item List**. • **Off** - Clones source item as additional instanced layers in the **Item List**. • **Hierarchy** - Creates a cascading array where each subsequent clone is a child of the previous clone. • **Root** - Creates instances as child items of the source mesh.

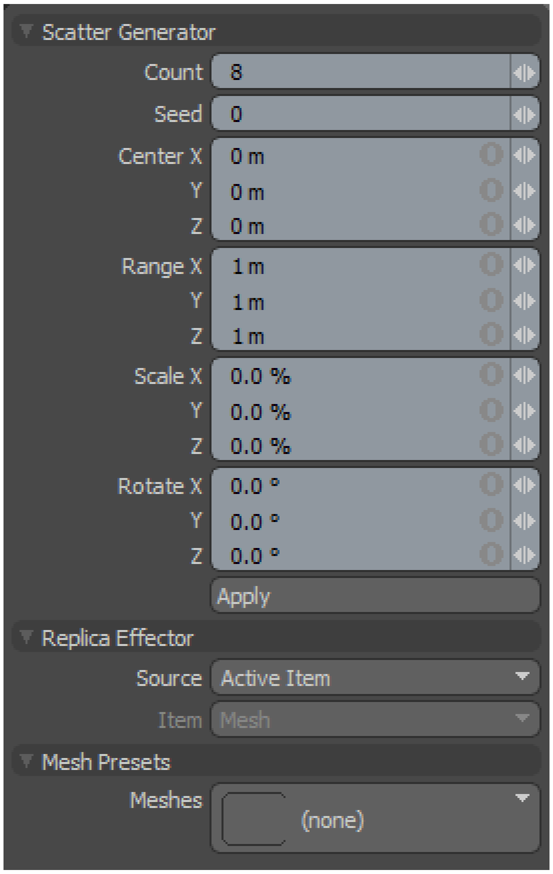
Item Effector	
	• **Primary** - Creates Instances as children of a group locator that is parented to the source mesh.
Instance	When enabled, each cloned item is an instance of the source mesh. Instances contain no physical geometry, but they reference a source layer and are represented in 3D Viewports by pink wireframes. (This is enabled by default.)
Bounding Box	When enabled, the resulting instances display as bounding boxes rather than wireframe meshes. This may provide a 3D viewport performance increase when Modo generates many instances.
Hierarchy	When enabled, clones any child items of the source geometry and retains their hierarchy through each subsequent clone.
Source	Determines the source mesh that Modo clones into the point cloud. • **Active Items** - Uses the currently selected item(s) as a clone source. • **Specific Item** - Specifies a Mesh Item as the clone source based on the **Mesh Item** information. • **All BG** - Combines all the currently visible background items and clones them as a single mesh. • **Random BG** - Selects a random background item from all the currently visible background items at each clone position. • **Mesh Layer Preset** - Clones a mesh preset selected in the Preset Browser. Open the Preset Browser (by pressing **F6**) and select a mesh preset before activating the tool.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Replica Scatter

Similarly to Scatter Clone, the Replica Scatter tool generates a random point cloud onto which Modo places replicas of the original geometry. To activate the tool, right-click the **Scatter Clone** icon and select **Replica**

Scatter from the list.

Replica Scatter Properties



Scatter Generator
Count 8
Seed 0
Center X 0 m
Y 0 m
Z 0 m
Range X 1 m
Y 1 m
Z 1 m
Scale X 0.0 %
Y 0.0 %
Z 0.0 %
Rotate X 0.0 °
Y 0.0 °
Z 0.0 °
Apply
Replica Effector
Source Active Item
Item Mesh
Mesh Presets
Meshes (none)

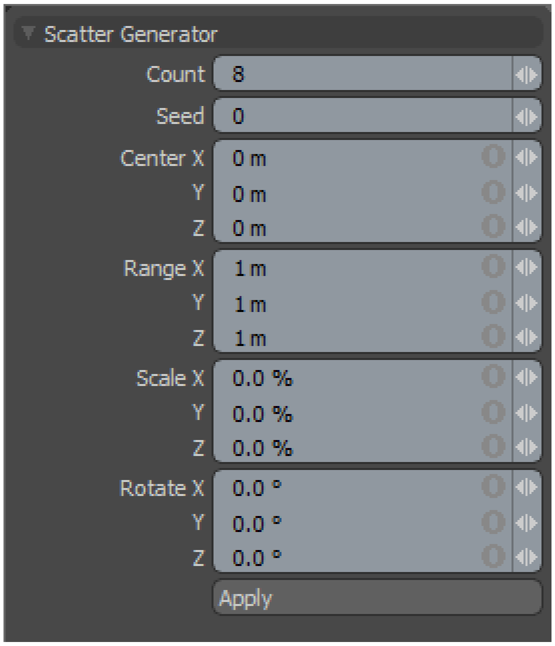
In addition to the **Scatter Generator** properties described above for the Scatter Clone tool, the following parameters can be set for the Replica Scatter tool.

Replica Effector	
Source	Determines the source mesh that Modo clones into the point cloud.
Item	Specifies the item to be cloned. This is only relevant if you set **Source** to **Specific Item**.
Mesh Presets	
Meshes	Opens the Preset Browser, where you can select a mesh preset to use as the **Source**.

Transform Scatter

To activate the tool, right-click the **Scatter Clone** icon and select **Transform Scatter**.

Transform Scatter Properties



Scatter Generator

Count 8

Seed 0

Center X 0 m

Y 0 m

Z 0 m

Range X 1 m

Y 1 m

Z 1 m

Scale X 0.0 %

Y 0.0 %

Z 0.0 %

Rotate X 0.0 °

Y 0.0 °

Z 0.0 °

The following properties are available for the Transform Scatter tool.

Scatter Clone	
Count	Specifies the total number of clones (including the source).
Center X/Y/Z	Determines the center position of the scatter clone effect.
Range X/Y/Z	Defines the maximum distance on each axis that the clones can be from the center of the effect.
Scale X/Y/Z	Specifies the maximum amount of randomized fluctuation for any given clone.
Rotate X/Y/Z	Indicates the maximum amount of randomized rotation for any given clone.

Scatter Clone	
Replace Source	When enabled, removes the original geometry and creates a replacement clone for it at the center of the scatter clone effect.

Scatter Clone in the Procedural

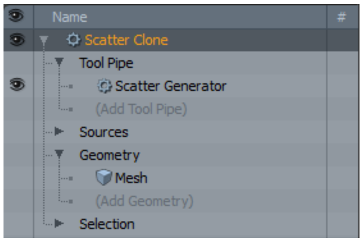
Scatter Clone is also available as a procedural mesh operation. It is a combination of the Clone Effector and Scatter Generator sub-tools. You can find the tool in the **Mesh Ops** tab on the right-side of the **Model** layout.

Here's how you can add the Scatter Clone mesh operation:

1. Click the **Add Operator** button, and navigate to **Mesh Operations** > **Duplicate** > **Scatter Clone**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

   The Scatter Clone mesh operation clones the geometry below it in the list. You can adjust the operation using its **Properties** panel in the lower-right.

You can expand the Scatter Clone item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses.



- **Tool Pipe** - Allows you to add sub-tools to the operation. For the Scatter Clone tool, Scatter Generator is added by default.
- **Sources** - Allows you to specify the geometry you want to duplicate.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.
- **Selection** - Allows you to select the elements to which you want to apply the tool.

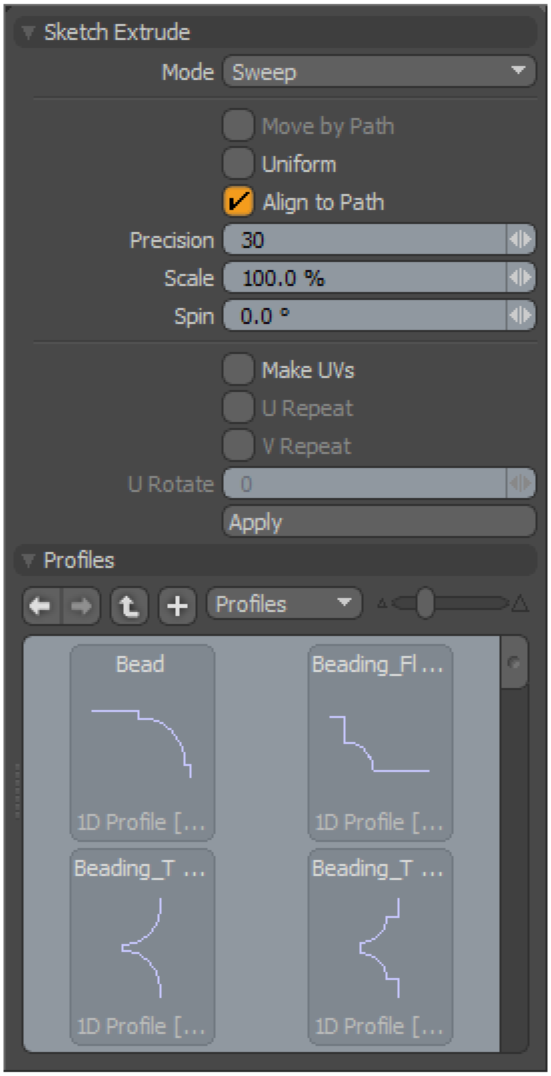
Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

The Scatter Clone mesh operation has the same properties as the traditional Scatter Clone tool. See Scatter Clone Properties.

Sketch Extrude

With the Sketch Extrude tool you can extrude selected polygons by drawing in the viewport.

To use this tool, select a polygon, and then click the Sketch Extrude tool on the **Polygon** tab of the Modo Tools toolbar. Click in the viewport to activate the tool, and then drag the handle to initiate the sketch. The handle turns to a yellow circle. The polygon extrusion follows the movement of the pointer as you draw in the viewport.



Sketch Extrude

Mode: Sweep

☐ Move by Path

☐ Uniform

☑ Align to Path

Precision: 30

Scale: 100.0 %

Spin: 0.0 °

☐ Make UVs

☐ U Repeat

☐ V Repeat

U Rotate: 0

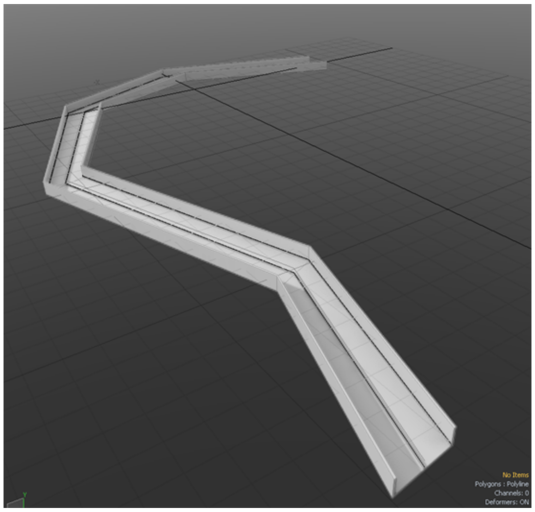
Apply
Profiles
Profiles
Bead
1D Profile [...
Beading_Fl ...
1D Profile [...
Beading_T ...
1D Profile [...
Beading_T ...
1D Profile [...

Sketch Extrude	
Modes	Provides a number of options for controlling the tool. • **Sweep** - Sketches the path for extruding the polygons when you drag any visible handles (cyan dots). • **Edit Path** - Reveals the control points on all user-created paths. Dragging these control points adjusts the sketched curves. • **Delete Knot** - Removes control points from the curve path when you click them. • **Uniform Spans** - Shows handles only at the end of each path. Clicking any handle adjusts the control knots along the path such that they are distributed uniformly along the path. • **Straight** - Straightens a path after you click the end handle.

Sketch Extrude	
Move by Path	When enabled, moves the points that are between the selected control point and the end of the path when dragging any control point. This option is only available after selecting **Edit Path**.
Uniform	When enabled, constrains all newly generated geometry to be uniformly distributed between the original polygon's position and the end of the path.
Align to Path	When enabled, rotates the geometry that Modo creates along the path such that it is aligned to the curve. When disabled, all new polygons have the same World Rotation as the original polygon.
Precision	Determines how often control points should be created when you interactively sketch the path. This number represents a screen pixel count. For example, if you set the value to **30** (default value,) Modo creates a new control span each time the pointer travels 30 pixels on screen during the sketch.
Scale	Varies the thickness of the generated geometry along the path from the original polygonal location to the control handle at the end of each path.
Spin	Rotates each generated span along the path to provide a twisting effect along the sketched curve.
Make UVs	Controls how Modo generates UVs for the new polygons in the mesh. With the default option, Modo attempts to create the UVs inside the existing UV space amongst the corresponding geometry. Selecting **U Repeat** or **V Repeat** overlays each new segment over the last along the selected **U** or **V** axis.
Profiles	Provides a list (essentially a mini Preset Browser) for viewing various profiles and selecting one. This works the same as the standard Preset Browser. Selecting any profile modulates the extrusion width with the selected shape over its length. To return to the original polygon selection for the extrusion, press **Ctrl** as you click the selected profile.

Sweep Effector

The procedural **Sweep Effector** draws geometry along a curve or line that is generated by tools such as the Pen Generator or Path Generator. The image below shows the Sweep Effector used to draw a curve generated by the **Pen Generator**. (In this case, the source geometry is a Profile from the Content Preset.)



Note: A non-procedural version of the **Sweep Effector** is integral to tools such as the Radial Sweep and Curve Extrude. The procedural **Sweep Effector** allows you to emulate these tools in the Mesh Operations List.

Using the Sweep Effector

You can see step-by-step tutorials using the **Sweep Effector** in the topics for the Content Preset and Pen Generator.

Sweep Effector Procedural

The Sweep Effector has the following inputs:

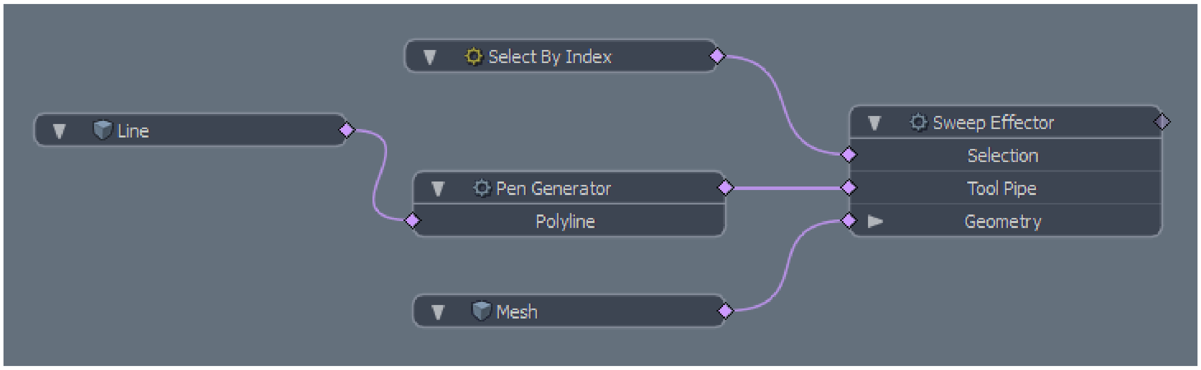
- **Selection** - Allows you to select part of the source geometry to be used in the sweep. Without a selection, the entire source geometry is used.
- **Tool Pipe** - You can add one or more sub-tools as input to the **Sweep Effector**. Click **(Add Tool Pipe)** to display the list of compatible sub-tools in the Procedural Preset Browser.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Sweep Effector Schematic

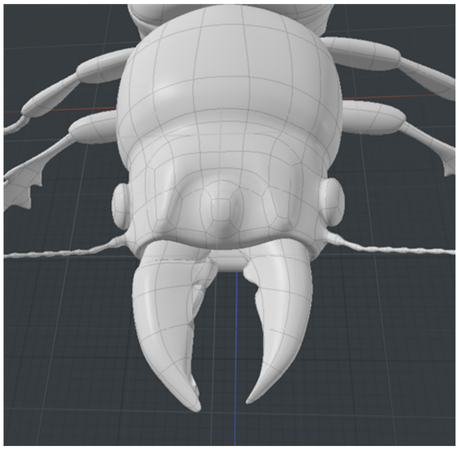
The image below shows an example in which a selected part of a mesh is swept along a line that is generated by a **Pen Generator**.

Note: To sweep the whole mesh, leave the **Selection** input empty.

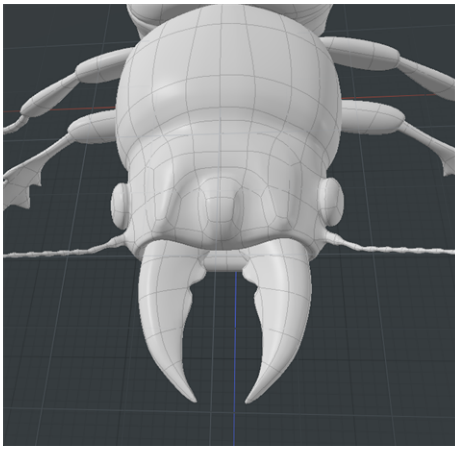

Select By Index
Line
Sweep Effector
Selection
Tool Pipe
Pen Generator
Polyline
Geometry
Mesh

Symmetrize

The Symmetrize tool deletes half of the mesh and mirrors the opposite half in its place. This is useful when you accidentally lose symmetry by editing one side of your mesh item with the Symmetry Tool turned off.



Before applying Symmetrize:
The left mandible is thicker than the right one



After applying Symmetrize:
The mandibles are symmetrical

This tools works as a procedural modeling operator, a schematic operator, and as a direct modeling tool.

For more information, see Modeling Techniques.

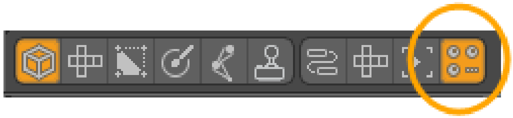
- For direct modeling: Using the **Model** layout, open the **Polygons** tab on the left panel, and click **Symmetrize**.
- For procedural modeling: Using the **Model** layout or **Setup** layout, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Polygons** > **Symmetrize**.
- For schematic modeling: On the top-left corner of the interface, click the schematic palette icon, click **Add...**, and double-click **Mesh Operations** > **Polygons** > **Symmetrize**.

Applying a Direct Modeling Symmetrize Tool

The direct modeling **Symmetrize** tool allows you to quickly correct symmetry problems. For example, if you have beveled one side of your model and it is now non-symmetrical, you can run **Symmetrize** tool to correct the positional changes and geometry updates. To gain more control on how the symmetry is applied, you can also choose the axis to use, select if you want to apply a direction of negative to positive or positive to negative on the specified axis, and change other properties.

1. On the top-left corner of the interface, click the Presets palette icon.

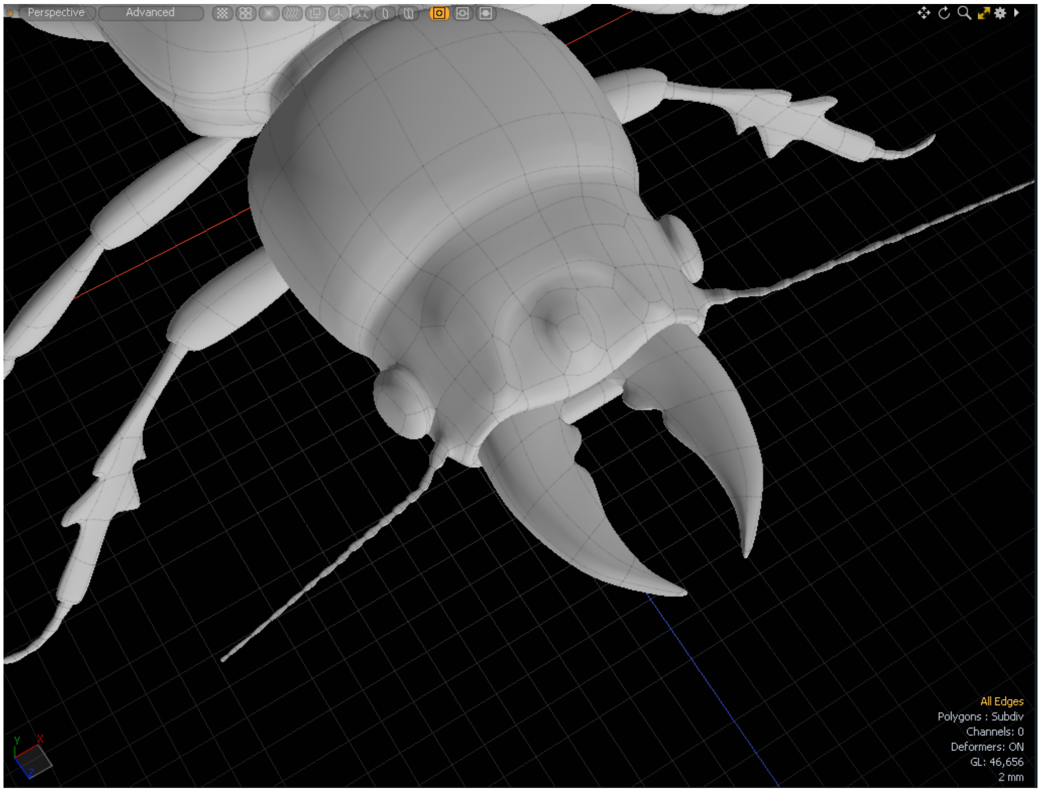
   The button displays in orange when activated.



2. On the left column, expand the **Cloud Assets** > **Meshes** > **Animals** directory, and double-click **Beetle 01**.

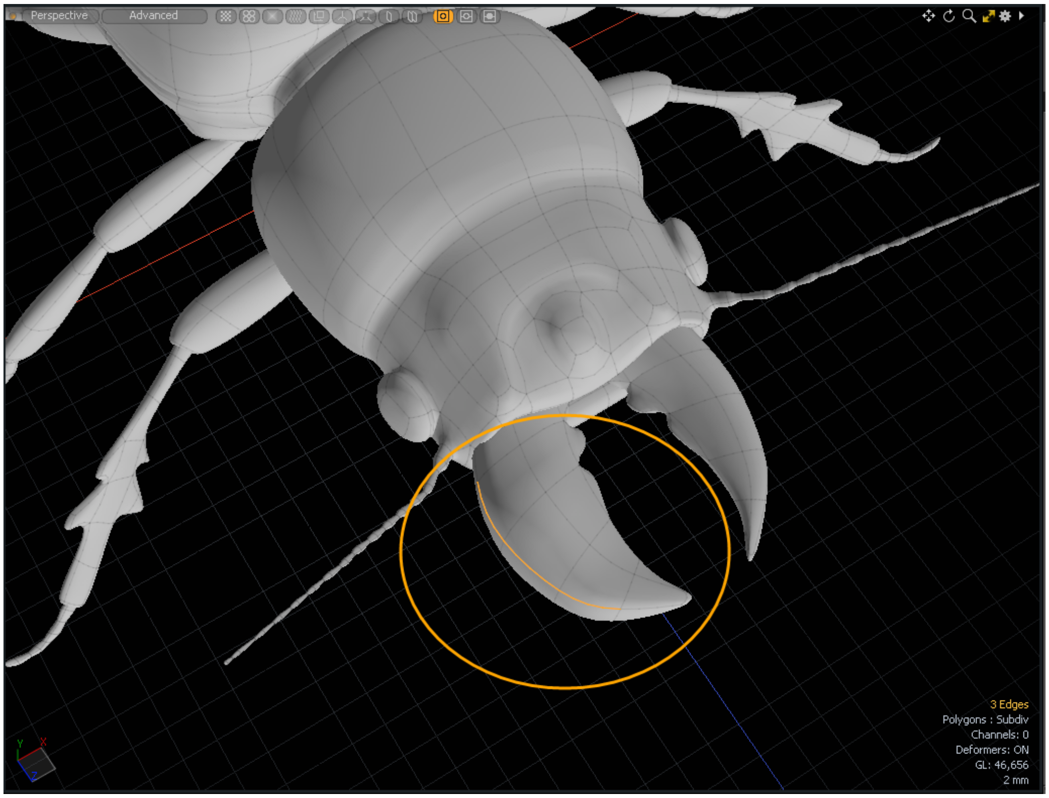
Note: If you have already downloaded this asset to your local drive, open the **Assets** > **Animals** directory and double-click **Beetle 01**.

3. Close the **Presets** window.
4. Use your middle-mouse wheel to zoom in and press **A** to center the mesh item in the 3D viewport.
5. Activate **Edges** selection mode and **Shift**+click on the outer edges of the left mandible of the beetle.

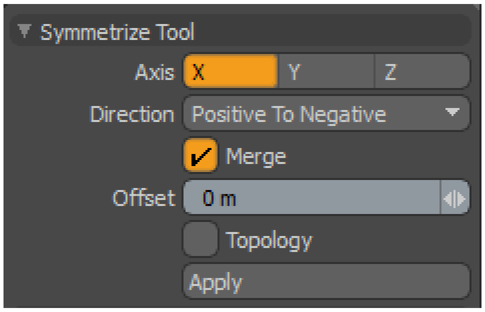

Perspective
Advanced
All Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 46,656
2 mm

6. On the left panel, open the **Deform** tab, click **Push**, and drag along the **Z** axis in the 3D viewport.

   The left mandible is now wider than the right mandible on the beetle.


Perspective
Advanced
3 Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 46,656
2 mm

7. Press the **Spacebar** twice to drop the Push tool.
8. On the left panel, open the **Polygon** tab, and click **Symmetrize**.
9. On the bottom of the left panel, set **Axis** to **X**, **Direction** to **Positive to Negative**, enable **Merge**, set **Offset** to **0**, and click **Apply**.



▼ Symmetrize Tool

Axis: X | Y | Z

Direction: Positive To Negative

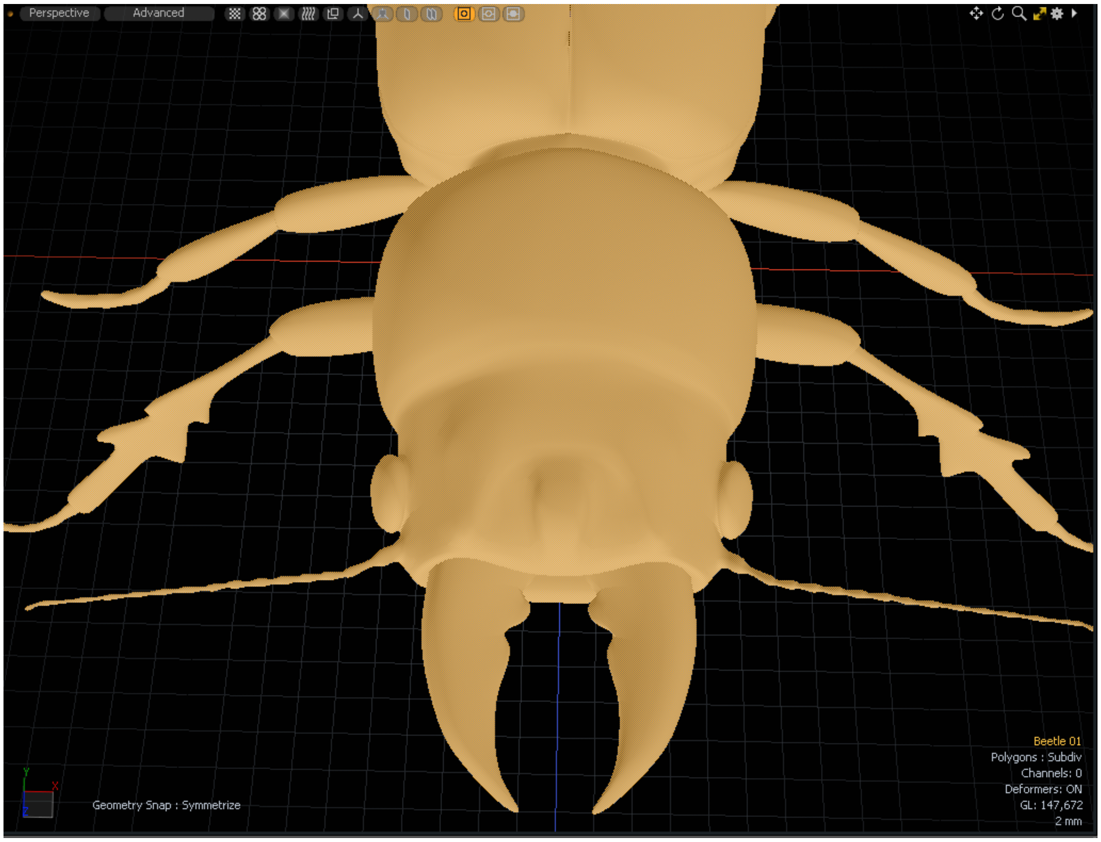
☑ Merge

Offset: 0 m

☐ Topology

Apply

Symmetrize is applied to the beetle and both mandibles are the same size.


Perspective
Advanced
Beetle 01
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 147,672
2 mm
Geometry Snap : Symmetrize

Applying a Procedural Symmetrize Tool

1. Using the **Model** layout, on the right panel open the **Mesh Ops** tab, and click **Add Operator**.
2. Navigate to **Mesh Operations** > **Polygon** and double-click **Symmetrize.**

Note: In the **Mesh Operations** list, place the Symmetrize mesh operation above the mesh you want to modify.

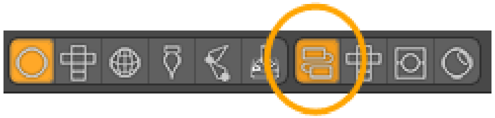
When added to the list, Symmetrize automatically copies the positive half of the selected mesh over the negative half. You can reverse the direction and edit other settings in the mesh operation's **Properties** tab. For more information, see Symmetrize Tool Properties.

Applying a Schematic Symmetrize Tool

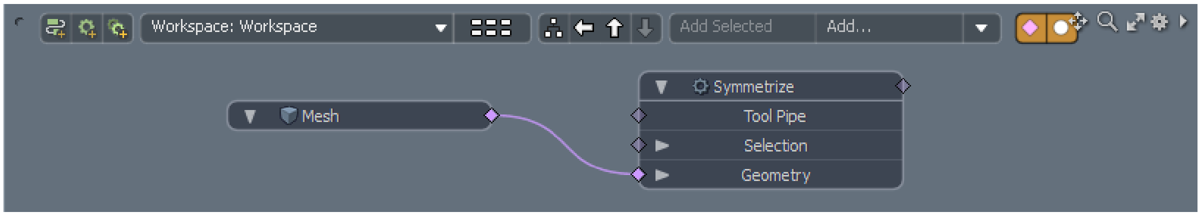
The Symmetrize node requires the same inputs as the procedural version.

1. On the top-left corner of the interface, click the schematic palette icon.

   When activated, the button has an orange background.



2. Drag-and-drop a mesh item from the **Items** list onto the Schematic viewport.
3. Click **Add...**, and double-click **Mesh Operations** > **Polygons** > **Symmetrize**.


Workspace: Workspace
Add Selected
Add...
Mesh
Symmetrize
Tool Pipe
Selection
Geometry

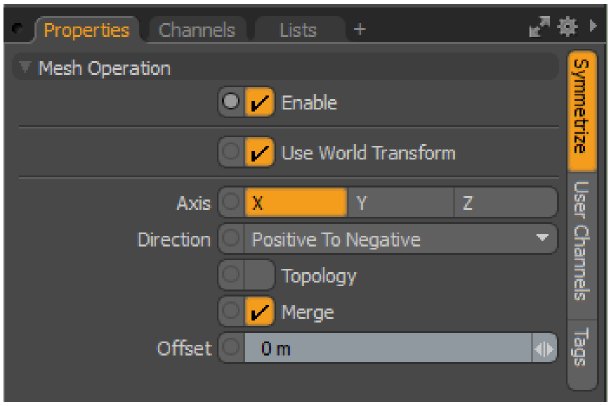
The Symmetrize mesh operation has the following inputs:

- **Selection** - Allows you to select the elements to which you want to apply the tool. For more information, see Selection Operations.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list.

For information about **Symmetrize** tool properties, see Symmetrize.

Symmetrize Tool Properties

The direct modeling **Symmetrize** tool displays properties on the left panel. Both the procedural and schematic modeling **Symmetrize** tool displays properties on the right panel. Some properties may vary.


Properties
Channels
Lists
+
Symmetrize
User Channels
Tags

Mesh Operation

☑ Enable

☑ Use World Transform

Axis: X | Y | Z (X selected)

Direction: Positive To Negative

☐ Topology

☑ Merge

Offset: 0 m

Enable	Activates and deactivates the operation.
Use World	When enabled, the mesh operation maintains the world transforms on the input layer,

Transform	transforming any geometry inputs by the transform matrix of the input item.
Axis	Defines the axis along which to symmetrize.
Direction	Specifies the direction of the operation. The following options are available: • **Positive To Negative**: Deletes polygons on the negative side and mirrors the remaining polygons on the positive side. • **Negative To Positive**: Deletes polygons on the positive side and mirrors the remaining polygons on the negative side.
Topology	Mirrors vertex positions for topologically symmetrical vertex pairs. This does not remove half side polygons, but it moves symmetrical vertex positions based on polygon connectivity.
Merge	When enabled, merges vertices on symmetry center. **Note:** **Merge** does not work when **Topology** is enabled.
Offset	Offset distance of the center along the symmetry axis.

Deforming Items

With Modo's various deform tools, you can produce soft, organic deformations (such as with the Soft Move and Sculpt tools) or hard, mechanical forms (such as with the Shear and Bend tools). Many of Modo's deforming tools are hybrid tools that combine functions of action centers, falloffs, and transforms into a single tool. These tools offer a wide array of options to contort and otherwise shape the geometry into any form you wish. With the Toolpipe function, you can make your own tools from a combination of functions. For information on using a specific tool, click a link below to see that tool's topic.

Deform		
	Element Move	Allows you to quickly edit individual component elements of a mesh.
	Element Push	Allows you to push individual component elements of a mesh along their normals.

Deform		
	Flex	Provides a natural, organic bending-type deformation.
	Soft Move	A tool preset that consists of the Transform tool with Radial Falloff. With this tool preset, you can select vertices, edges, or polygons and move them.
	Soft Drag	A tool preset that consists of the Move tool combined with the Screen Falloff. It allows you to select vertices, edges, or polygons, and then drag them to affect their neighbors.
	Shear	Combines the Transform tool with a Linear falloff.
	Smooth	Reduces the jaggedness or roughness of a surface.
	Jitter	Repositions selected items or components based on a seed number and a range value for the X, Y, and Z axes.
	Quantize	Evaluates the contour of the selected geometry and shuffles the vertices to align to a three-dimensional grid.
	Symmetry Tool	Allows you to interactively correct minor symmetry errors.
	Soft Selection Move	Moves the selected geometry rigidly, while gradually attenuating the translation outside the selected area, defined by the Soft Selection Radius.
	Soft Selection Rotate	Rotates the selected geometry rigidly, while gradually attenuating the rotation amount outside of the selected area as defined by the Soft Selection Radius.
	Soft Selection Scale	Scales or re-sizes the selected geometry rigidly while gradually attenuating the scale amount outside the selected area defined by the Soft Selection Radius.
	Soft Selection Transform	Allows you to simultaneously move, rotate, and scale the selected geometry rigidly while gradually attenuating the translation outside selected area

Deform		
		defined by the Soft Selection Radius.
	Linear Align	Aligns selected vertex or UV positions along a line.
	Radial Align	Moves selected vertex or UV positions along a circle or polygon.
	Quick Align	Aligns geometry to another piece of geometry interactively in the 3D viewport.
	Ground Align	Aligns the selected geometry down to the ground.
	UV Transform	Transforms a source mesh's vertices to the world position of the surface of a target mesh, using a UV map on the target mesh.
	Path Generator	Uses an existing curve to provide a path for other tools to walk and evaluate.
Rotate Deform		
	Twist	Combines the Axis Rotate tool with a Linear Falloff.
	Bend	Deforms selected geometry in a curved manner. For example, bending a straight cylinder into an arc.
	Vortex	Combines the Axis Rotate tool with the Cylindrical Falloff.
	Swirl	Combines the Axis Rotate tool with a Radial Falloff.
Scale Deform		
	Push	Moves the selected geometry along the selected item's normals.
	Sculpt	Allows you to shape polygons with a brush-like paradigm to smoothly deform a surface.

Deform		
	Taper	Combines the Transform tool with a Linear Falloff.
	Bulge	Combines the Transform Scale tool with a Radial Falloff to produce a swelling effect on the target object or surface.
	Flare	Combines the Push tool with a Linear Falloff.

Deform Tools

Learn about the tools used to deform geometry in Modo.

Element Move

With the Element Move tool you can quickly edit individual component elements of a mesh (vertices, edges, or polygons).

Using Element Move

You can find the tool in the **Model** layout's **Deform** sub-tab, or in the menu bar, under **Edit** > **Deform** > **Element Move Tool**.

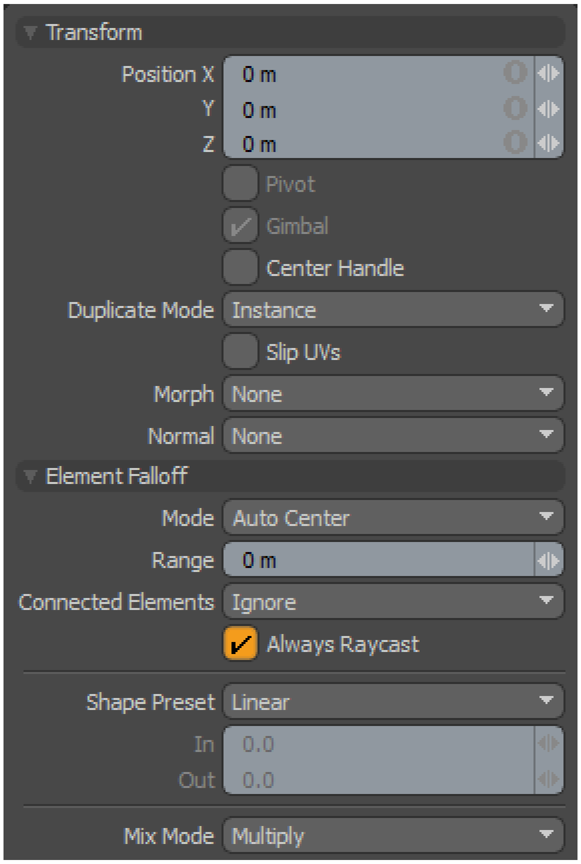
1. Activate the tool by clicking the **Element Move** button, or by pressing **T** on the keyboard.

   As you move the pointer in the 3D viewport, selectable elements are highlighted
2. Click and drag on a vertex, edge, or polygon to move it. Alternatively, click on an element to display the transform handles, and use those to move it along an axis.

There are two modes for the tool: Manual and Automatic. By default, the tool is in Automatic mode, and Modo determines the element type to be edited when you click. For example, if you click an edge, you edit the edge; if you click a polygon, you edit the polygon. With this tool in Automatic mode, you can edit components of the mesh independently of the current selection state.

The Manual mode forces the tool to act only on the component type defined by the current global selection mask. For example, if the selection mode is set to Vertex and the tool is in Manual mode, you can only edit vertices.

The Element Move tool also has an **Element Falloff** value, which creates a spherical area of influence around the element under the pointer. You can set this value in the tool's **Properties** panel for the tool, or by right-clicking and dragging in the 3D viewport. Modo represents this falloff in the 3D view by a yellow wireframe sphere for you to interactively adjust the region.



Transform

Position X 0 m

Y 0 m

Z 0 m

☐ Pivot

☑ Gimbal

☐ Center Handle

Duplicate Mode: Instance

☐ Slip UVs

Morph: None

Normal: None

Element Falloff

Mode: Auto Center

Range: 0 m

Connected Elements: Ignore

☑ Always Raycast

Shape Preset: Linear

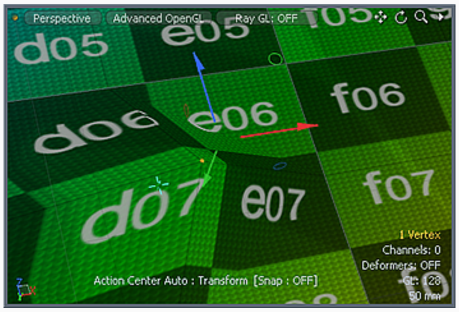
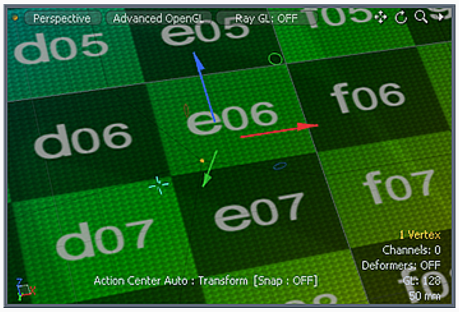
In 0.0

Out 0.0

Mix Mode: Multiply

The following properties are available for the Element Move tool:

Transform	
Position X/Y/Z	Displays the current offset distance. When adjusting an object interactively in the viewport, use these values as feedback for moving the element. You may specify the **X**, **Y**, and **Z** values to apply a specific offset distance (based on the object's position when you activated the tool).

Transform	
Pivot	Disabled for the Element Move tool.
Gimbal	Disabled for the Element Move tool.
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f06 d06 e06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping)  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f06 d06 e06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions.

Transform	
	Note: In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Element Falloff	
Mode	Defines the automatic center that the transform offsets from when applied as well as the element type selected under the pointer. **Auto** is the default for the tool to select whatever element the pointer is over. It positions the center where the pointer intersects the surface it is modifying. Other options give you finer control.
Range	Specifies a soft falloff region that attenuates the transform over the range's distance.
Connected Elements	Defines what connected geometry does when you transform an element. • **Ignore** - Ignores all connected elements and only moves the specified element. • **Use Connectivity** - Affects only single-surface or connected elements. Ignores unconnected elements within range. • **Rigid Connections** - Moves all connected elements equally the specified distance. • **Edge Loops** - Moves connected loops (all the connected polygons, edges, and vertices in a single row).
Always Raycast	When enabled, the tool can affect polygons in **Wireframe** viewport mode. When disabled, only vertices and edges can be edited in **Wireframe** mode.
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the In and Out values.
In/Out	Determines the strength of the falloff nearer to the start or end position.

Transform	
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Element Push

With the Element Push tool you can push individual component elements of a mesh (vertices, edges, or polygons) along their normals.

Using Element Push

You can find the tool in the **Model** layout's **Deform** sub-tab, or in the menu bar, under **Edit** > **Deform** > **Element Push Tool**.

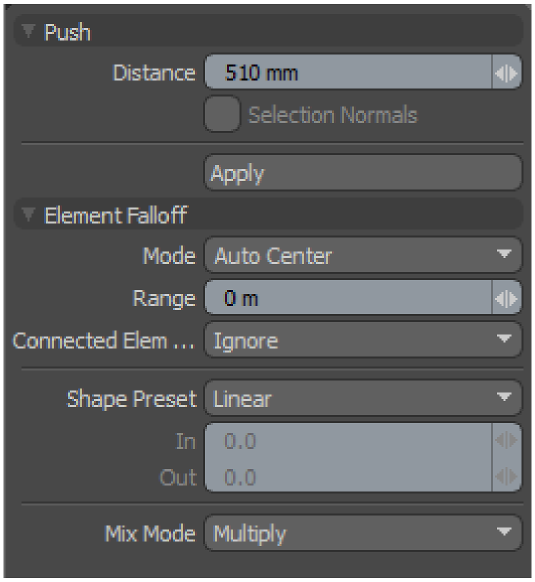
1. Activate the tool by clicking the **Element Push** button.

   As you move the pointer in the 3D viewport, selectable elements are highlighted
2. Click and drag on a vertex, edge, or polygon to push it. Alternatively, click on an element to display the handles, and use those to push it along an axis.

By default, the tool is in **Auto** mode, and Modo determines the element type to be edited when you click. For example, if you click an edge, you edit the edge; if you click a polygon, you edit the polygon. With this tool in **Auto** mode, you can edit components of the mesh independently of the current selection state. You can change the mode in the tool's **Properties** panel.

The Element Push tool also has an **Element Falloff** value, which creates a spherical area of influence around the element under the pointer. You can set this value in the tool's **Properties** panel, or by right-clicking and dragging in the 3D viewport. Modo represents this falloff in the 3D view by a yellow wireframe sphere for you to interactively adjust the region.

Element Push Properties


Push
Distance
510 mm
Selection Normals
Apply
Element Falloff
Mode
Auto Center
Range
0 m
Connected Elem ...
Ignore
Shape Preset
Linear
In
0.0
Out
0.0
Mix Mode
Multiply

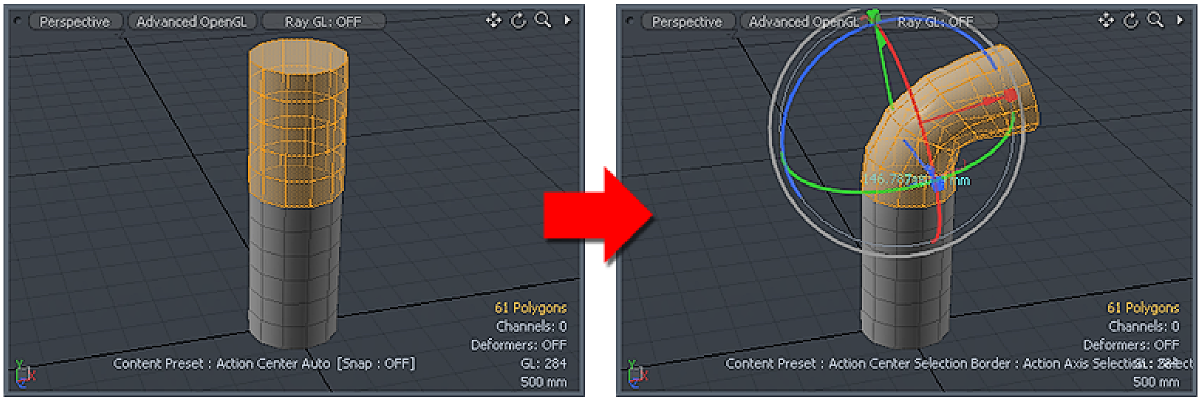
The following properties are available for the Element Push tool:

Push	
Distance	The offset distance of the push.
Selection Normals	When enabled, Modo uses the normals computed from selections. **Note:** This option is only available in **Polygons** selection mode.
Element Falloff	
Mode	Defines the automatic center that the push offsets from when applied, as well as the element type selected under the pointer. **Auto** is the default for the tool to select whatever element the pointer is over. It positions the center where the pointer intersects the surface it is modifying. Other options give you finer control.
Range	Specifies a soft falloff region that attenuates the transform over the range's distance.
Connected Elements	Defines what connected geometry does when you push an element. • **Ignore** - Ignores all connected elements and only pushes the specified element. • **Use Connectivity** - Affects only single-surface or connected elements. Ignores unconnected elements within range.

Push	
	• **Rigid Connections** - Pushes all connected elements equally the specified distance. • **Edge Loops** - Pushes connected loops (all the connected polygons, edges, and vertices in a single row).
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine-tunes the strength of the falloff based on the **In** and **Out** values.
Mix Mode	Defines how each falloff interacts with the other(s), when there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Flex

The Flex tool leverages the Selection Border action center and a selection-based falloff. Modo combines these with the standard Transform tool to create a tool that provides a more natural, organic bending-type deformation.


Perspective
Advanced OpenGL
Ray GL: OFF
61 Polygons
Channels: 0
Deformers: OFF
GL: 284
500 mm
Content Preset : Action Center Auto [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
61 Polygons
Channels: 0
Deformers: OFF
500 mm
Content Preset : Action Center Selection Border : Action Axis Selection

Using the Flex tool

You can find the tool in the **Model** layout's **Deform** sub-tab, or in the menu bar, under **Edit** > **Deform** > **Flex Tool**.

1. In the 3D viewport, select the elements you want to deform.

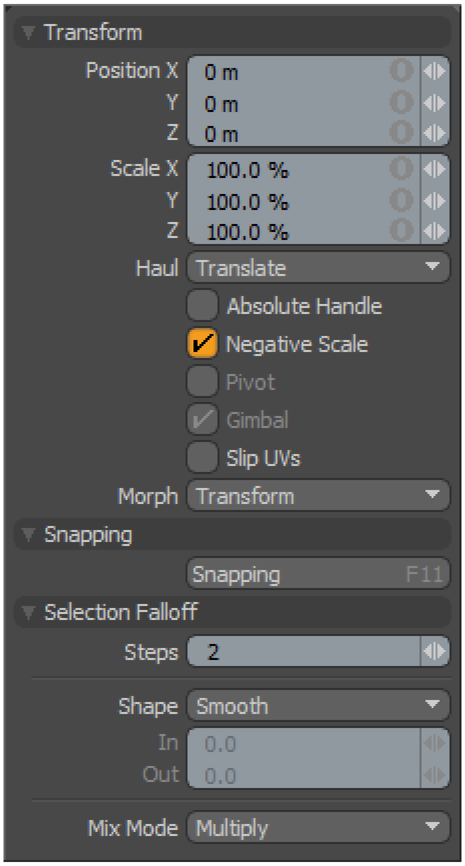
2. Click the **Flex** button, or press **Shift**+**Y** on the keyboard to activate the tool.

   The standard transform widget appears in the 3D viewport, and you can directly manipulate the currently-selected elements with Move, Rotate, and Scale transforms.

> **Tip:** If you use Flex with the **Select Closed Loop** command, you can pose meshes in a simple way.

Modo applies the amount of transform to the selection based on the **Steps** value. With the default value of **2**, Modo creates a smooth falloff starting at the boundary of the selection.

The tool's **Properties** panel includes direct numeric input of all transform values, as well as a **Haul** list to determine the hauling action, an **Absolute Handle** checkbox, a **Negative Scale** checkbox, and options for controlling the Selection falloff.



Transform

Position X 0 m

Y 0 m

Z 0 m

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

Haul Translate

☐ Absolute Handle

☑ Negative Scale

☐ Pivot

☑ Gimbal

☐ Slip UVs

Morph Transform

Snapping

Snapping F11

Selection Falloff

Steps 2

Shape Smooth

In 0.0

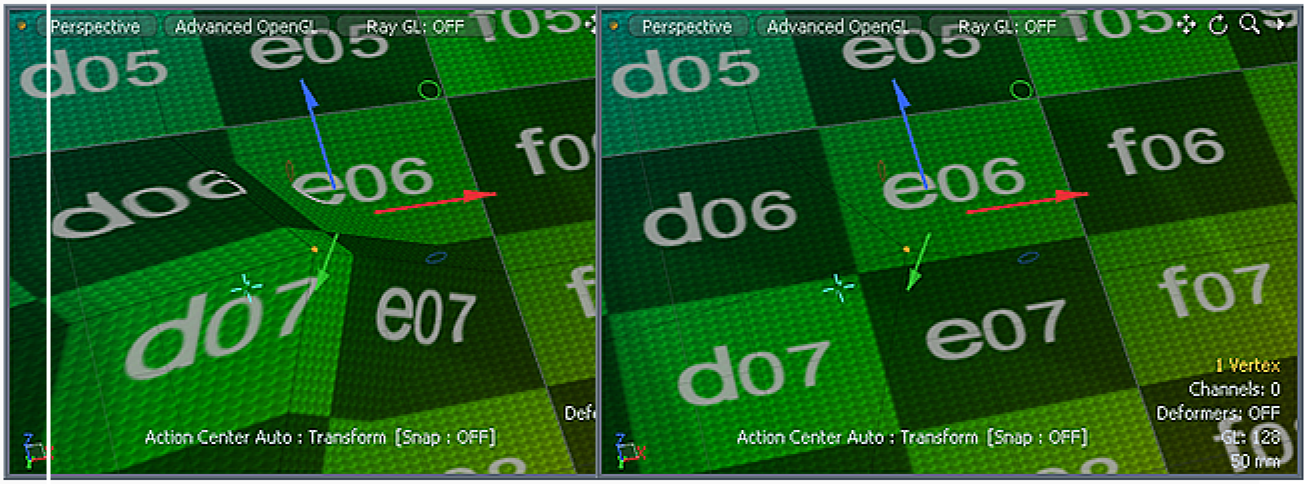
Out 0.0

Mix Mode Multiply

The following options are available for the Flex tool:

Transform	
Position	Displays the current offset distance. When adjusting an object interactively in the viewport,

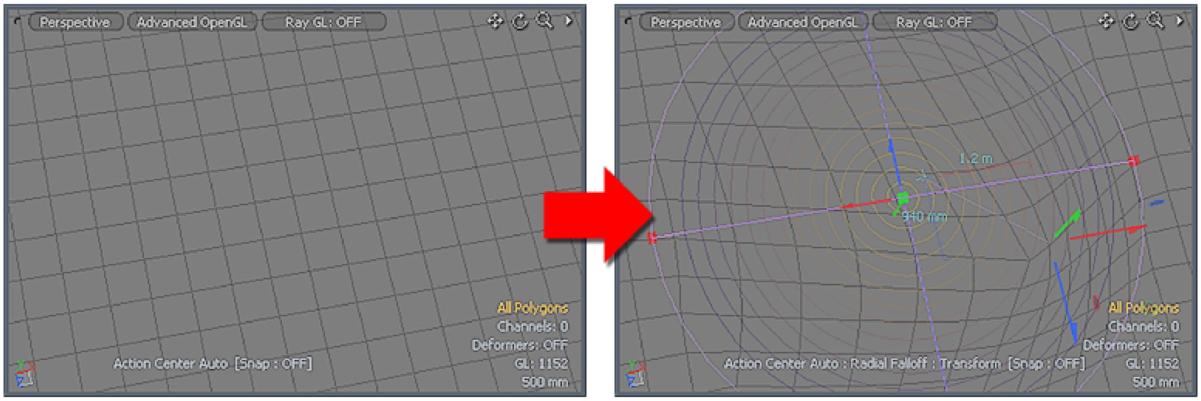
Transform	
X/Y/Z	use these values as feedback for moving the element. You may specify the **X**, **Y**, and **Z** values to apply a specific offset distance (based on the object's position when you activated the tool).
Scale X/Y/Z	Applies specific scaling values. When adjusting an object interactively in the viewport, the **Scale** values display the current scale factor applied as feedback for scaling the element.
Haul	Determines how Modo interprets dragging actions in a viewport (when not using a tool's handles). You have three options available: **Translate** (the default, to move the selection), **Rotate** (to rotate the selection), and **Scale** (to re-size the selection).
Absolute Handle	When enabled, applies an additional function to the tool handles for you to scale the handle, itself, without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.
Negative Scale	When enabled, applies negative scale values, if appropriate, when you manipulate the tool handles in the viewport. When disabled, Modo does not automatically change the values if they are negative. You can always input negative values directly.
Position Only	(Only available in **Items mode**) When enabled, only changes the position of the selected items; the orientaion is left unchanged.
Pivot	Disabled for the Flex tool.
Gimbal	Disabled for the Flex tool.
Child Compensate	(Only available in **Items mode**) When enabled, ignores any transform values for child items. When disabled, passes along transform values to any child items of the current selection.
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry.

Transform	
	 Perspective Advanced OpenGL Ray GL: OFF Action Center Auto : Transform [Snap : OFF] 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm **Slip UVs** function disabled (notice the texture warping) **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Selection Falloff	

Transform	
Steps	Determines the extent of the falloff range in that each step represents a loop of polygons or a ring of edge sections. The larger the **Steps** value, the further the falloff attenuates.
Shape	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Soft Move

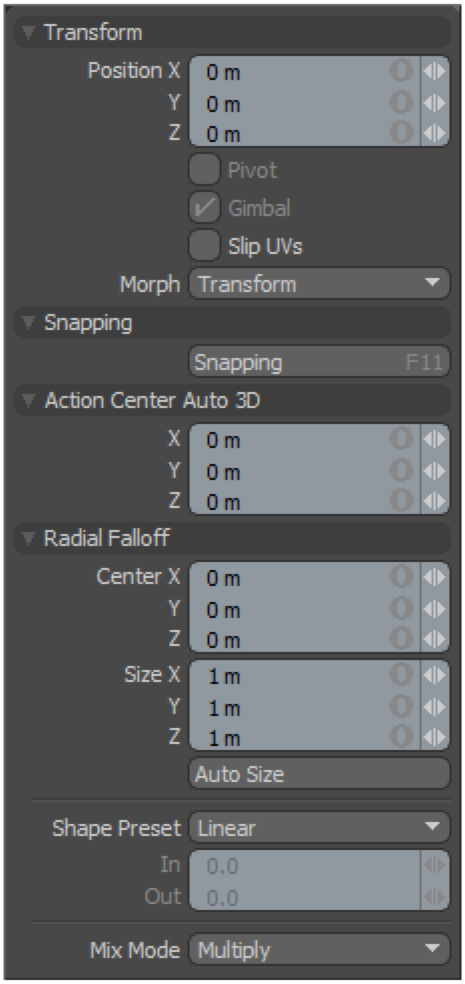
Soft Move is a tool preset that consists of the Transform tool with Radial falloff. With this tool preset, you can select vertices, edges, or polygons and move them.


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 1152
500 mm
Action Center Auto [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
1.2 m
910 mm
All Polygons
Channels: 0
Deformers: OFF
GL: 1152
500 mm
Action Center Auto : Radial Falloff : Transform [Snap : OFF]

Using Soft Move

1. You can find the tool in the **Model** layout's **Deform** sub-tab, or in the menu bar, under **Edit** > **Deform** > **Soft Drag**.
2. Click in the 3D viewport to display the transform handles and the Radial Falloff.
3. Adjust the handles by clicking and dragging in the viewport.

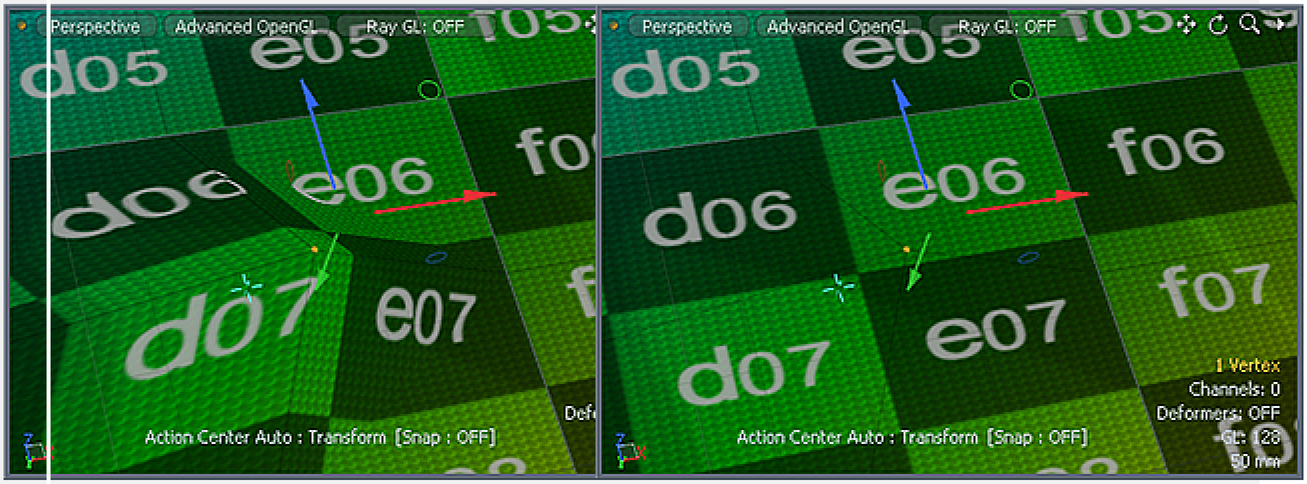
Modo attenuates the amount of movement to neighboring items based on the size of the Radial Falloff. This produces a soft, organic deformation. You can define the Radial Falloff interactively.



Position X 0 m
Y 0 m
Z 0 m
☐ Pivot
☑ Gimbal
☐ Slip UVs
Morph Transform
Snapping F11
X 0 m
Y 0 m
Z 0 m
Center X 0 m
Y 0 m
Z 0 m
Size X 1 m
Y 1 m
Z 1 m
Auto Size
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply

The following options are available for Soft Move:

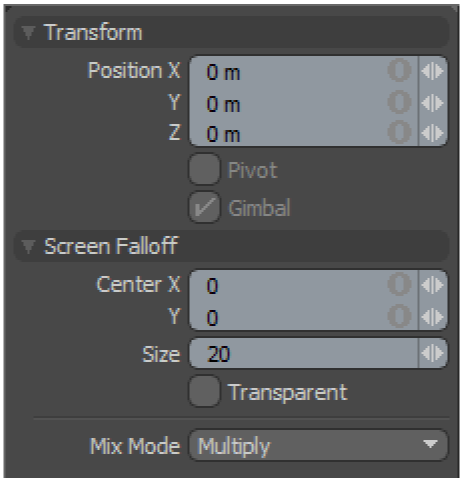
Transform	
Position X/Y/Z	Applies specific offset distances (calculated from the position the object was in when you activated the tool). When adjusting an object interactively in the viewport, the **Position** values display the current offset distance.
Pivot	Disabled for the Soft Move tool.
Gimbal	Disabled for the Soft Move tool.
Child Compensa	(Only available in **Items mode**) When enabled, ignores any transform values for child items. When disabled, passes along transform values to any child items of the current selection.

Transform	
te	
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry.  Perspective Advanced OpenGL Ray GL: OFF Action Center Auto : Transform [Snap : OFF] 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm **Slip UVs** function disabled (notice the texture warping) **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions.

Transform	
	Note: In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Radial Falloff	
Center X/Y/Z	Defines the center of influence, which is where the strength of the falloff is the greatest (**100%**). The strength of the falloff attenuates toward the outer bounds of the spherical volume. The area outside the volume receives no tool influence (**0%**).
Size X/Y/Z	Defines the radius of a perfect circle from the **Center** position and determines the outer area of the falloff where there is no effect.
Auto Size	Adjusts the falloff's **Center** and **Size** values automatically to match the bounding box of the current selection.
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Soft Drag

Soft Drag is a tool preset that consists of the Move tool combined with the Screen Falloff.



Transform

Position X 0 m
Y 0 m
Z 0 m

☐ Pivot

☑ Gimbal

Screen Falloff

Center X 0
Y 0
Size 20

☐ Transparent

Mix Mode Multiply

Using Soft Drag

You can access Soft Drag from the **Deform** tab of the Modo Tools toolbar or by choosing **Edit** > **Deform** > **Soft Drag**.

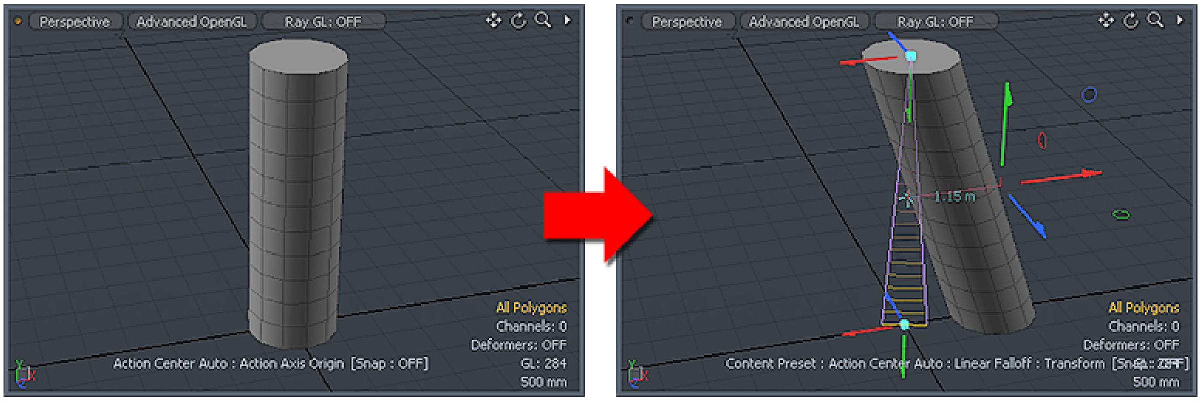
Activate the tool, then click and drag in the 3D viewport to drag out the elements you need. You can select vertices, edges, or polygons, and then drag them to affect their neighbors to a greater or lesser degree based on the Screen Falloff.

You can use the right-mouse button to define the Screen Falloff's circle.

Note: Enable **Transparent** in the **Screen Falloff** section for the Soft Drag selection to happen through the mesh. With **Transparent** disabled, this tool preset only affects geometry facing the view.

Shear

The Shear tool combines the Transform tool with a Linear falloff.


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 284
500 mm
Action Center Auto : Action Axis Origin [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
1.15 m
All Polygons
Channels: 0
Deformers: OFF
Content Preset : Action Center Auto : Linear Falloff : Transform [Snap : OFF]
500 mm

Using Shear

You can find the tool in the **Deform** sub-tab of the **Model** layout, on the left side of the screen. You can also activate the tool in the menu bar, under **Edit** > **Deform**.

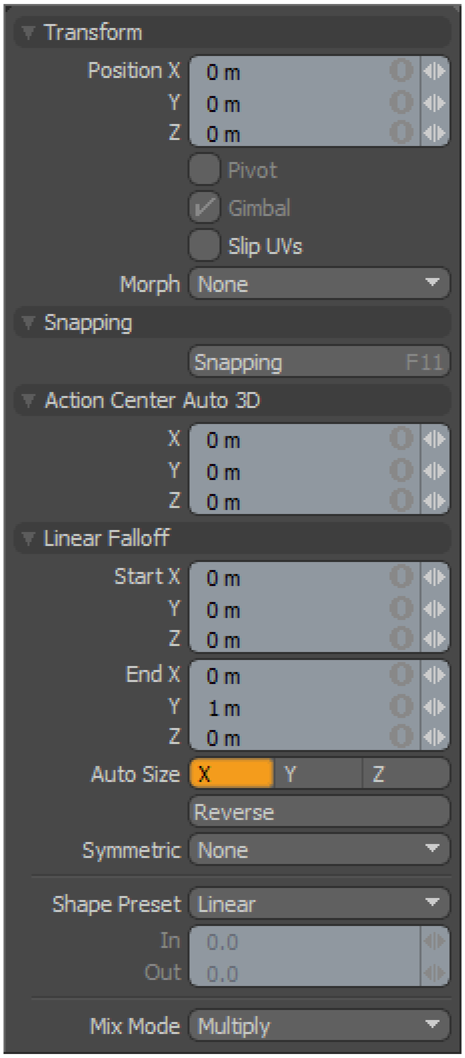
1. Activate the tool, then click in the 3D viewport.

   Modo sets the Linear Falloff perpendicular to the pointer's position and automatically scales to match the currently-selected geometry. The transform handles also appear in the viewport

Tip: To precisely place the falloff, you can either use the handles to manipulate the falloff after it has been auto-sized, or right-click and drag away from the handles to reset the falloff and drag it out interactively.

2. Drag on the handles to move the geometry.

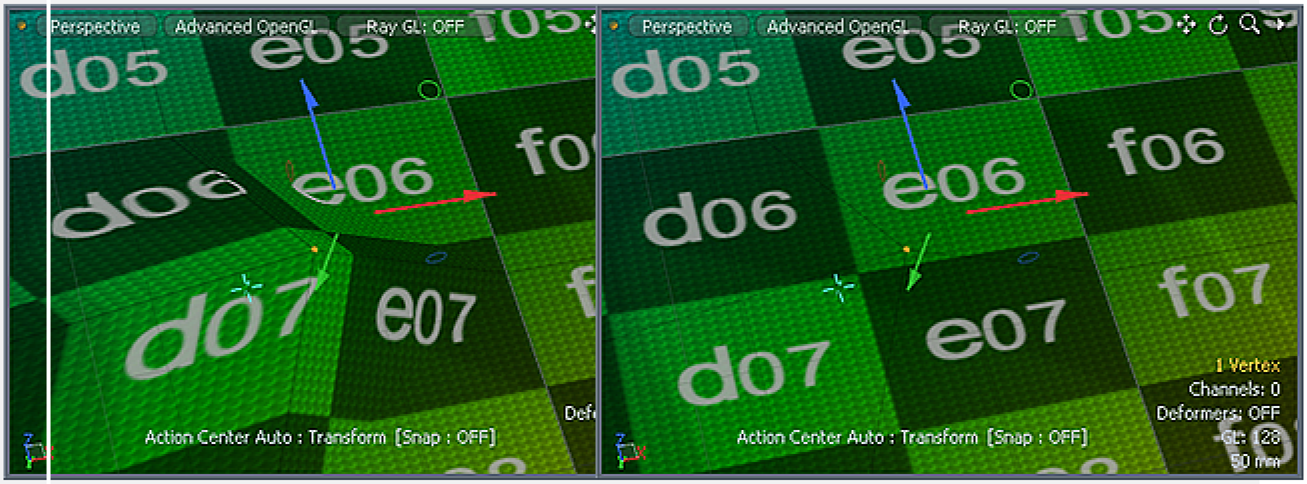
   The Linear Falloff modulates the result of the Move operation.



Position X 0 m
Y 0 m
Z 0 m
☐ Pivot
☑ Gimbal
☐ Slip UVs
Morph None
X 0 m
Y 0 m
Z 0 m
Start X 0 m
Y 0 m
Z 0 m
End X 0 m
Y 1 m
Z 0 m
Auto Size X Y Z
Symmetric None
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply

The following options are available for the Shear tool:

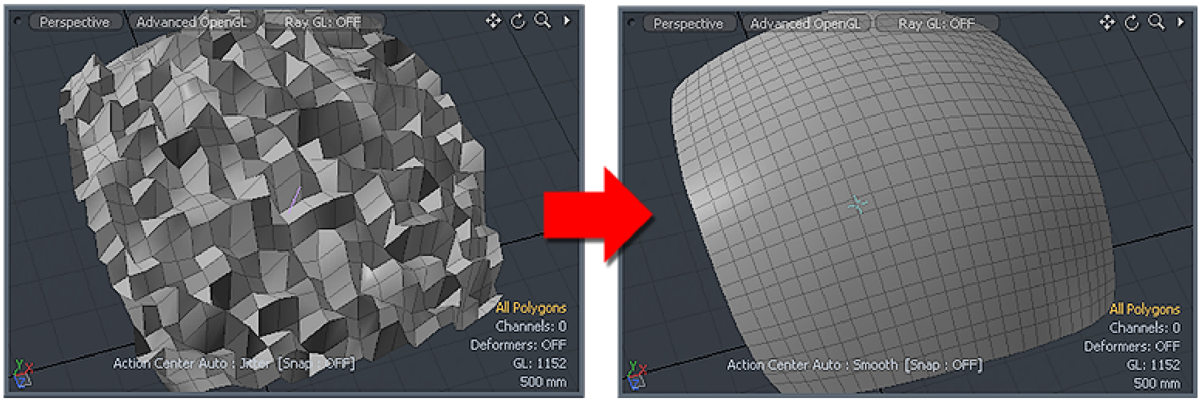
Transform	
Position X/Y/Z	Applies specific offset distance values (calculated from the position the object was in when you activated the tool). When you adjust an object interactively in the viewport, Modo displays the current offset distance to give you useful feedback.
Pivot	Disabled for the Shear tool.
Gimbal	Disabled for the Shear tool.
Child Compensate	(Only available in **Items mode**) When enabled, ignores any transform values for child items. When disabled, passes along transform values to any child items of the current selection.

Transform	
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry.  Perspective Advanced OpenGL Ray GL: OFF Action Center Auto : Transform [Snap : OFF] Perspective Advanced OpenGL Ray GL: OFF Action Center Auto : Transform [Snap : OFF] 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm **Slip UVs** function disabled (notice the texture warping) **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions.

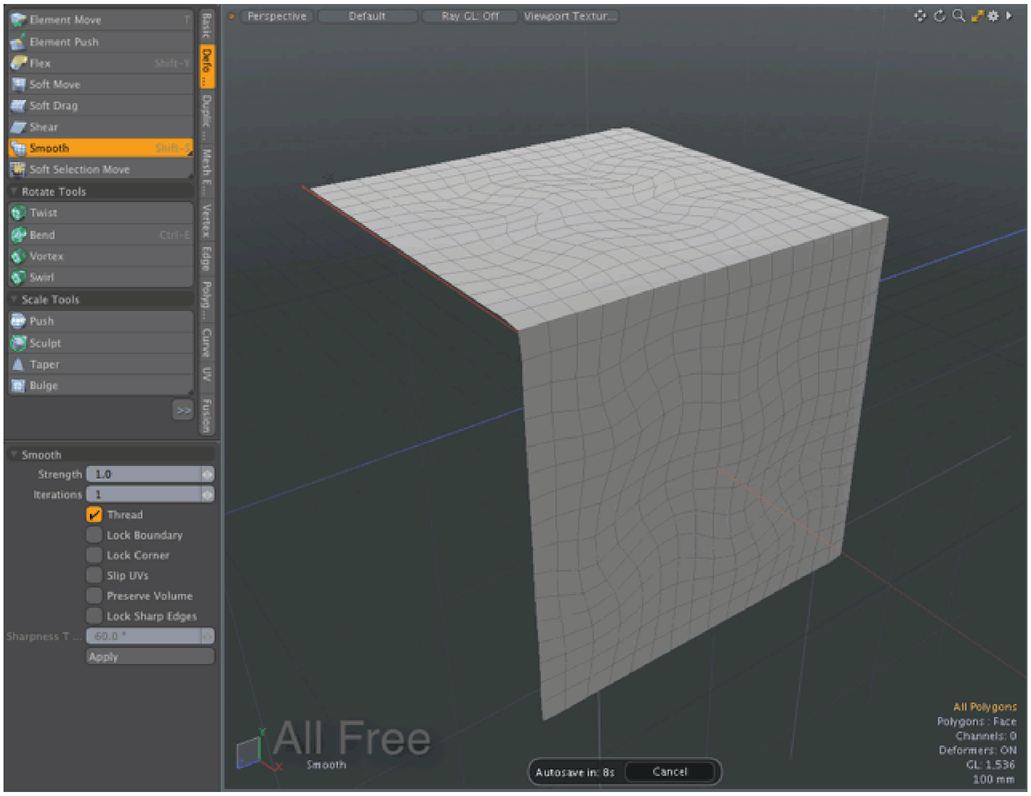
Transform	
	Note: In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Linear Falloff	
Start X/Y/Z	Defines the starting position of the falloff at a specific location. This point and everything beyond it receives the maximum amount of influence attenuating toward the **End** position.
End X/Y/Z	Defines the end position of the falloff at a specific location. This point and everything beyond it receives no influence from the falloff.
Auto Size	Scales the **Start** and **End** points of the falloff automatically to match the bounding box size of the selected elements along one of the three axes: **X**, **Y**, or **Z**.
Reverse	Inverts the **Start** and **End** points to reverse the influence of the falloff.
Symmetric	Mirrors the influence area of the falloff automatically. • **None** - Disables the symmetric function. • **Start** - Mirrors the influence of the falloff symmetrically across the **Start** position. • **End** - Mirrors the influence of the falloff symmetrically across the **End** position.
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Smooth

The **Smooth** tool reduces the jaggedness or roughness of a surface.


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 1152
500 mm
Action Center Auto : Jitter [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 1152
500 mm
Action Center Auto : Smooth [Snap : OFF]

In addition, the **Smooth** tool allows you to smooth positions along boundary edges and corner vertices.


Element Move
Element Push
Flex
Soft Move
Soft Drag
Shear
Smooth
Soft Selection Move
Rotate Tools
Twist
Bend
Vortex
Swirl
Scale Tools
Push
Sculpt
Taper
Bulge
Basic
Defo...
Duplic...
Mesh E...
Vertex
Edge
Polyg...
Curve
UV
Fusion
Perspective
Default
Ray GL: Off
Viewport Textur...
All Free
Smooth
Autosave in: 8s
Cancel
All Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 1,536
100 mm

Smooth

Strength 1.0

Iterations 1

☑ Thread

☐ Lock Boundary

☐ Lock Corner

☐ Slip UVs

☐ Preserve Volume

☐ Lock Sharp Edges

Sharpness T... 60.0 °

Apply

The tool operates in the 3D and UV viewports, and is good for smoothing overall topology, or for smoothing specific areas of a UV map.

Note: Jagged surfaces are also known as jittered polygons.

This tool also uses any active falloffs (including a Background Constraint) for you to target specific areas for smoothing.

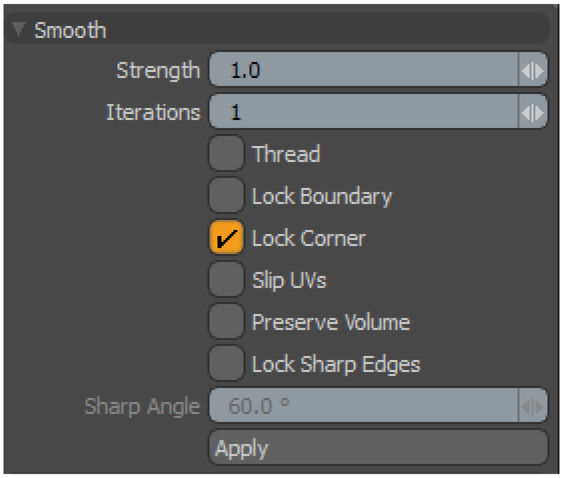
The **Smooth** tool is found in a number of places in the interface:

- **Model** layout: In the Toolbox on the left panel, under the **Deform** sub-tab. It is also an option in the **Jitter** and **Quantize** tools.
- **Topology** layout: In the Toolbox on the left panel, under the **Tools** sub-tab.
- **Edit** menu: Click **Edit** > **Deform** > **Smooth**.
- **Procedural Mesh Operation**: Open the **Mesh Operations** tab on the right panel, click **Add Operator**, expand the **Deform** category, and click **Smooth**. For more information, see .Smooth.

Smooth Tool Properties

The following properties are available for the **Smooth** tool:

Note: Property options vary between the direct modeling and the procedural versions of the tool.



Smooth

Strength 1.0

Iterations 1

☐ Thread

☐ Lock Boundary

☑ Lock Corner

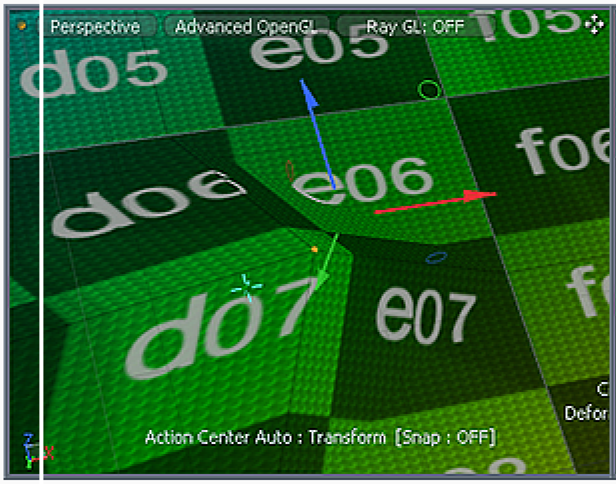
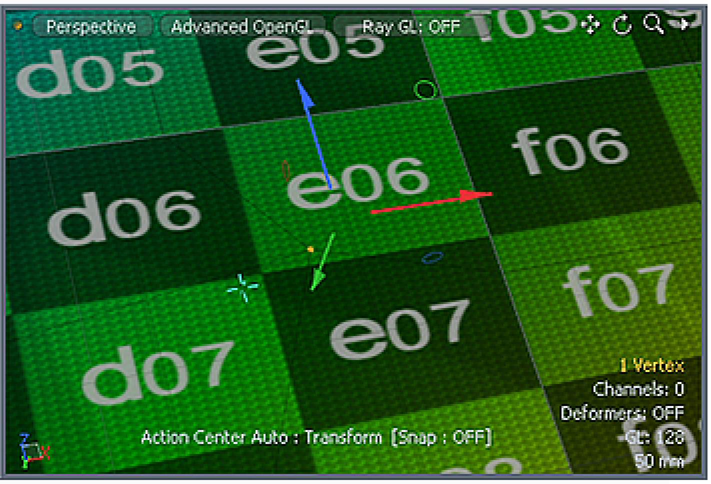
☐ Slip UVs

☐ Preserve Volume

☐ Lock Sharp Edges

Sharp Angle 60.0 °

Option	Description
Smooth	
Strength	Controls the amount of smoothing applied to the model.

Option	Description
Smooth	
Iterations	Determines the number of times that Modo applies the smoothing. A combination of both **Strength** and **Iterations** controls the overall smoothing of the model.
Thread	Provides faster feedback on the smoothing results, but it can reduce overall system interactivity. It may be useful to enable this option on complex models even though the smoothing calculations are processor intensive.
Lock Boundary	Locks the vertex positions along boundary edges.
Lock Corner	Locks the vertex positions at a corner vertex with a valence of 2. The corner vertex is shared by a single polygon and it is on the boundary.
Slip UVs	Checks to stop edits to the geometry changing the existing UV map. UV values are generally fixed to specific vertices, therefore further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this, enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry.  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping)  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Preserve	Allows you to maintain some of the definition of the original object by preserving object

Option	Description
Smooth	
Volume	volume. This is done by constraining smoothing points to the original object's surface.
Lock Sharp Edges	Locks sharp edges.
Sharp Angle	Calculates the threshold angle to determine which sharp edges to lock. This is the angle between normal vectors from two face polygons which shared edges. The edges, which have a larger angle than the Sharp Angle, are locked.

See Applying Snapping and Specifying Action Centers and Falloffs for further control options.

Smooth Tool Procedural Mesh Operation

The **Smooth** tool is available as a procedural operation for use in the Mesh Operations List. In the Procedural Preset Browser, you can find this tool under **Procedural** > **Deform**.

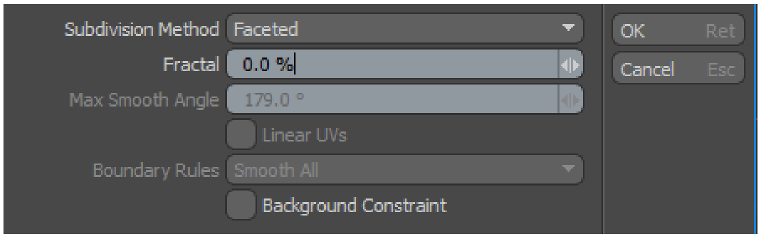
The following is an example of applying the procedural **Smooth** tool to a plane with a jittered surface.

Creating a jittered plane:

1. Open the **Model** layout.
2. In the toolbox, open the **Basic** tab, hold **Shift** and click **Plane**.

   The Plane is added to your scene.
3. Under the Switcher bar, click **Polygons**.
4. With the Plane selected, click **Shift** + **D.**

   The Subdivide Polygons dialog opens.
5. Set the Subdivision Method to **Faceted**. Repeat this step to create more subdivisions.



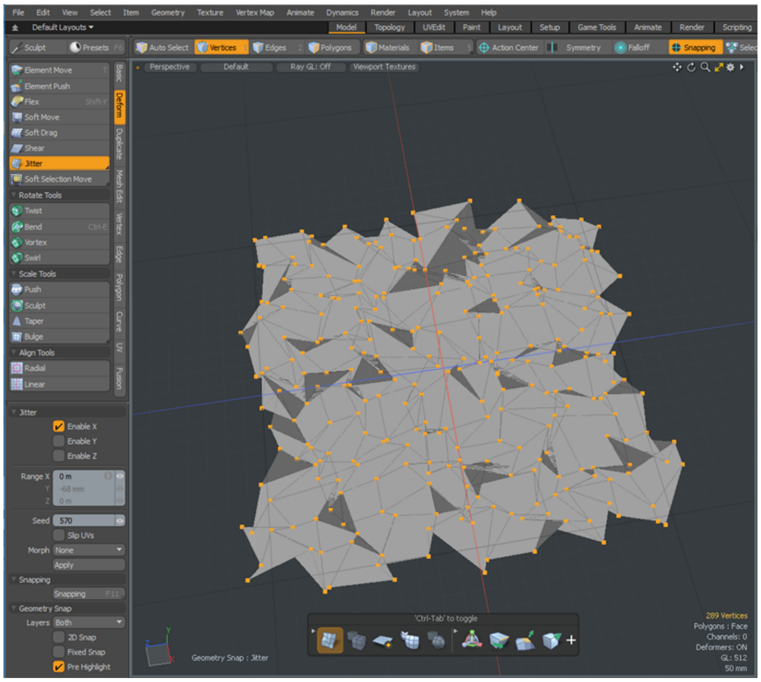
Subdivision Method Faceted
Fractal 0.0 %
Max Smooth Angle 179.0 °
☐ Linear UVs
Boundary Rules Smooth All
☐ Background Constraint

6. Under the Switcher bar, click **Vertices**, right-click and drag to select all the vertices.
7. On the Toolbox panel, open the **Deform** tab, and click the **Jitter** button.

Tip: If the Jitter tool is not visible, click and hold the **Smooth/Quantize** button to reveal the additional options, then select **Jitter**.

8. With the tool active, click and drag in the viewport to interactively apply the jitter, or type the **Range** values in the tool's **Properties** panel, and then click **Apply**.

   For more information about the Jitter tool properties, see Jitter.



Jitter

☑ Enable X

☐ Enable Y

☐ Enable Z

Range X 0 m

Y -68 mm

Z 0 m

Seed 570

☐ Slip UVs

Morph None

Apply

Snapping

Snapping F11

Geometry Snap

Layers Both

☐ 2D Snap

☐ Fixed Snap

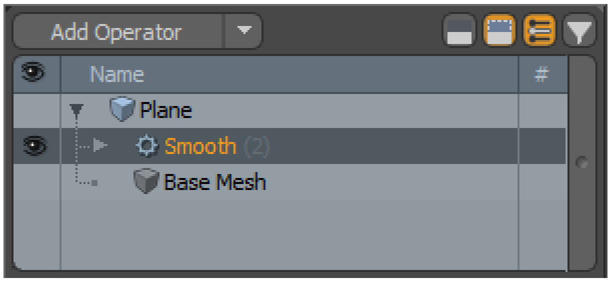
☑ Pre Highlight

Element Move
Element Push
Flex
Soft Move
Soft Drag
Shear
Jitter
Soft Selection Move
Rotate Tools
Twist
Bend
Vortex
Swirl
Scale Tools
Push
Sculpt
Taper
Bulge
Align Tools
Radial
Linear
Perspective
Default
Ray GL: Off
Viewport Textures
'Ctrl-Tab' to toggle
Geometry Snap : Jitter
289 Vertices

Applying a procedural Smooth operation:

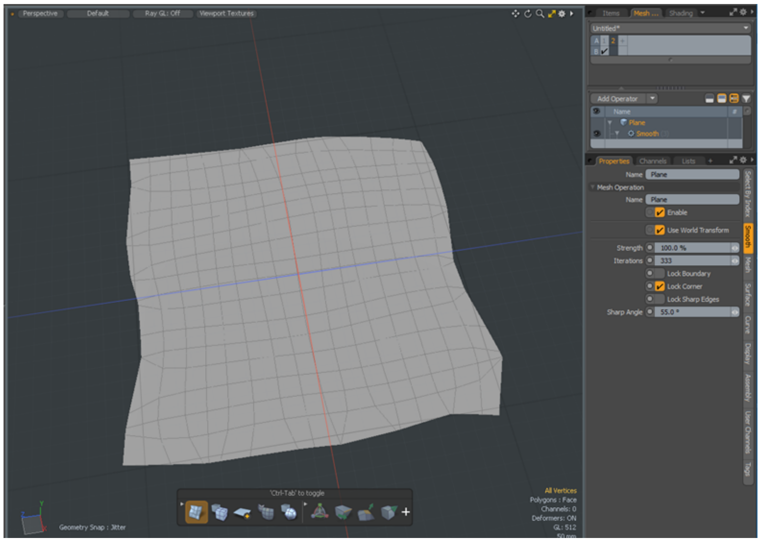
1. With the Plane item selected, open the **Mesh Ops** tab on the right panel.
2. Click **Add Operator**, select **Mesh Operations** > **Deform**, and double-click **Smooth**.

   The Smooth tool is listed below the Plane in the Mesh Ops tab.



Add Operator
Name
#
Plane
Smooth (2)
Base Mesh

3. Hold **Shift**, click the **Plane** item, and open the **Smooth** tab in the right panel.
4. Adjust the properties as required.


Perspective
Default
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Add Operator
Name
Plane
Smooth
Properties
Channels
Lists
Select By Index
Smooth
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags
'Ctrl-Tab' to toggle
Geometry Snap : Jitter
All Vertices
Polygons : Face
Channels: 0
Deformers: ON
GL: 512

Name: Plane

Mesh Operation

Name: Plane

☑ Enable

☑ Use World Transform

Strength: 100.0 %

Iterations: 333

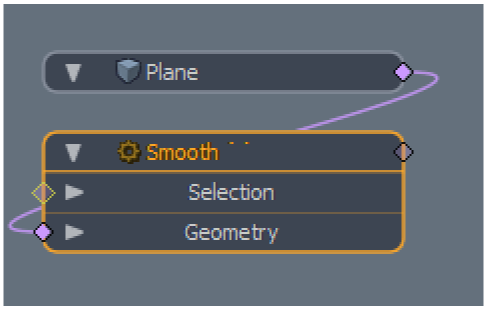
☐ Lock Boundary

☑ Lock Corner

☐ Lock Sharp Edges

Sharp Angle: 55.0 °

When working in the **Setup** layout, the **Smooth** tool has two parameters you can set up.


Plane
Smooth
Selection
Geometry

- **Selection** - Lets you restrict the smoothing to a sub-selection of mesh elements. See Procedural Selection for more details.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

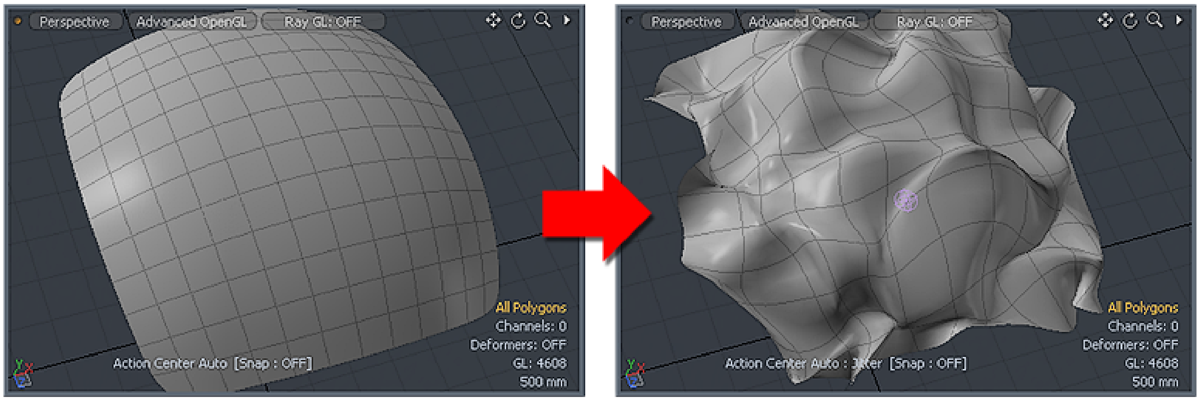
The procedural version of the Smooth tool has the following properties:

Option	Description
Enable	Toggles the Smooth tool off and on.
Strength	Controls the amount of smoothing applied to the mesh.
Iterations	Determines the number of times that Modo applies the smoothing. A combination of both **Strength** and **Iterations** controls the overall smoothing of the mesh.

Option	Description
Lock Boundary	Locks the vertex positions along boundary edges.
Lock Corner	Locks the vertex positions at a corner vertex with a valence of 2. The corner vertex is shared by a single polygon and it is on the boundary.
Lock Sharp Edges	Locks sharp edges.
Sharp Angle	Calculates the threshold angle to determine which sharp edges to lock. This is the angle between normal vectors from two face polygons that share edges. The edges, which have a larger angle than the Sharp Angle, are locked.
Preserve Volume	Allows you to maintain some of the definition of the original object by preserving object volume. This is done by constraining smoothing points to the original object's surface.

Jitter

The **Jitter** tool repositions selected items or components based on a seed number and a range value for the X, Y, and Z axes. It applies a random offset to each vertex or item position for a more random, organic appearance.

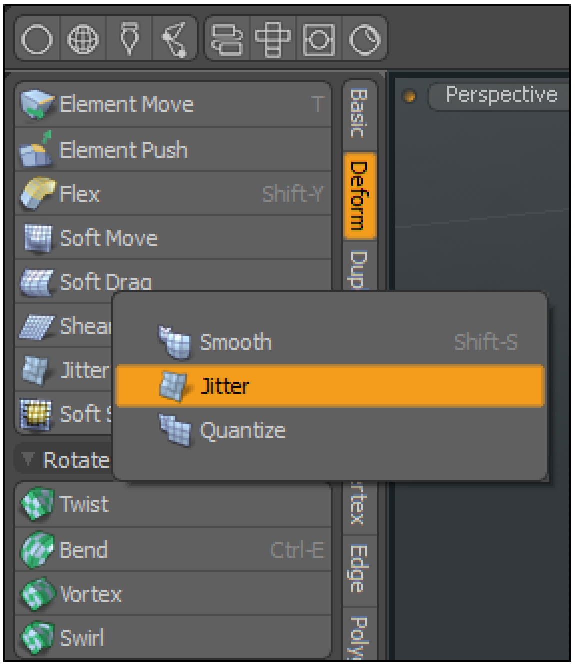

Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 4608
500 mm
Action Center Auto [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 4608
500 mm
Action Center Auto : Jitter [Snap : OFF]

Activating the Jitter Tool

Direct Modeling:

- In the **Model** layout, on the left panel, open the **Deform** tab, and click **Jitter**.

Tip: You can find the tool in the Model layout, in the **Deform** sub-tab of the modeling toolbox (sometimes as a sub-option of the Smooth or Quantize tools).


Element Move T
Element Push
Flex Shift-Y
Soft Move
Soft Drag
Shear
Jitter
Soft
Rotate
Twist
Bend Ctrl-E
Vortex
Swirl
Basic
Deform
Dup
rtex
Edge
Poly
Perspective
Smooth Shift-S
Jitter
Quantize

Procedural Modeling:

- In the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Deform** > **Jitter**.
- In the **Schematic** view, click **Add...**, and double-click **Mesh Operations** > **Deform** > **Jitter**.

Applying Jitter

1. Activate the **Jitter** tool.
2. With the tool active, drag in the viewport to interactively apply the jitter, or type **Range** values in the tool's **Properties** panel.

   Changing the **Seed** value changes the random numbers to produce a variation on the results. This tool uses any active falloffs to target specific areas for the jitter. You can also randomly move, scale, and/or rotate connected selections in X, Y and Z, or U and V direction.

Direct Modeling Example

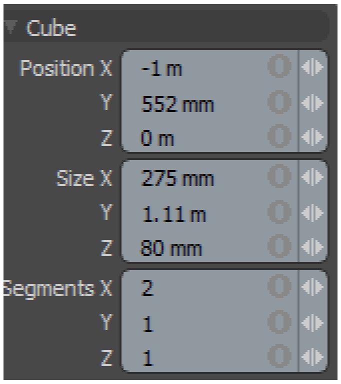
The following is an example of using the **Jitter** tool to create a fence with random spacing between each post. This is done by duplicating using the **Clone** tool. By default the **Jitter** is applied to the vertices. By enabling the **Rigid Translate**, **Rigid Rotate**, and **Rigid Scale** options, all the duplicates are treated as individual objects. Use these options to quickly create real geometry without having to use Replicators.

Once created, we can use this model to apply a texture image map with random jitter effects, which can be used in a Games Engine. To view a random texture on all duplicates, you can quickly view the results in the 3D viewport without having to render your scene.

To create a fence post:

1. Open the **Model** layout, under the **Basic** sub-tab in the Toolbox, and click **Cube**
2. In the properties on the left panel, set **Segment X** to **2** and then click-and-drag a shape in the 3D viewport.

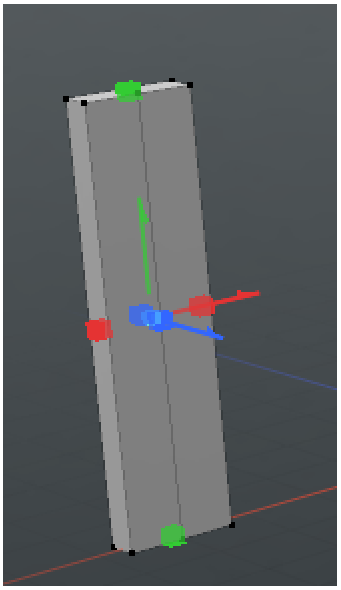
   The following properties were used in this example.



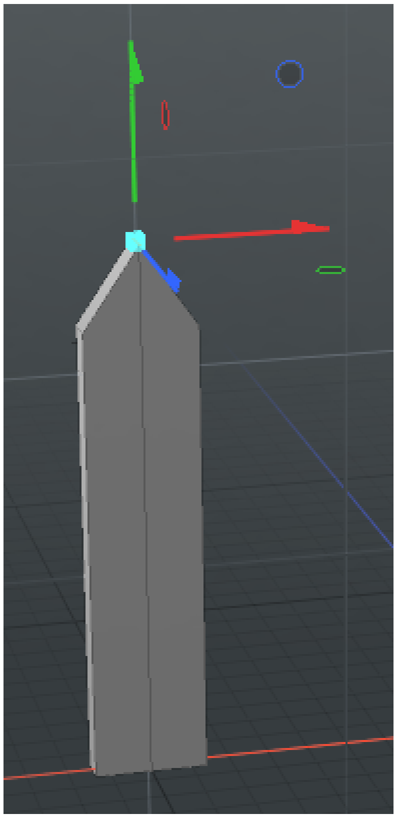
Cube

Position X	-1 m
Y	552 mm
Z	0 m
Size X	275 mm
Y	1.11 m
Z	80 mm
Segments X	2
Y	1
Z	1

Tip: Use the scale transform square handles to make the initial shape.



3. Press the spacebar to drop the tool.
4. Under the menu bar and above the 3D viewport, click **Edges**.
5. Click on the top center edge of the fence post to activate it and press **W** to activate the **Move** tool.
6. Drag the transform handle upwards to make a peaked fence post.

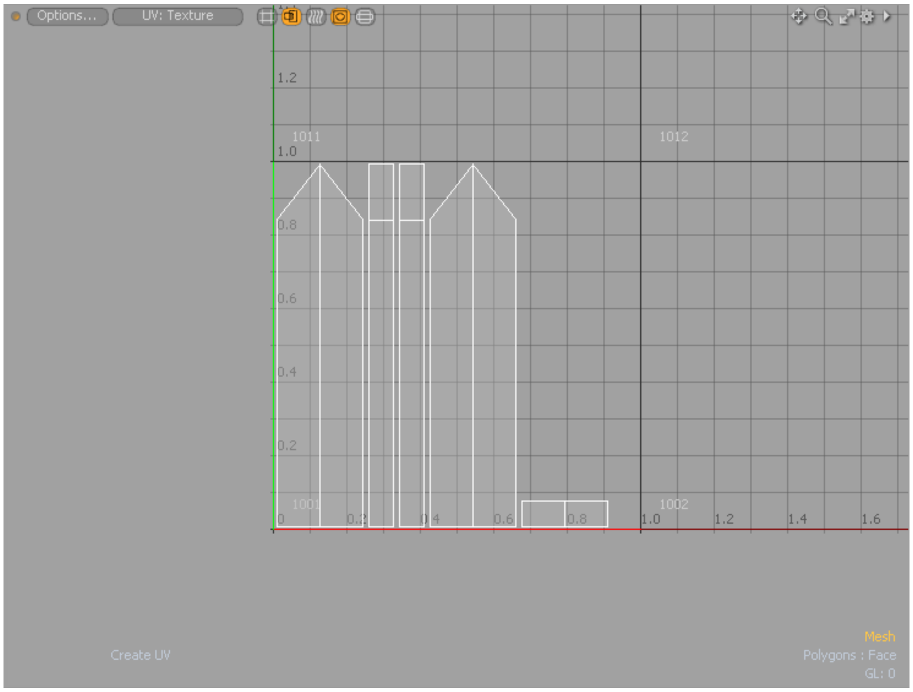


7. Press the **Spacebar** to drop the tool.

Create a new UV map:

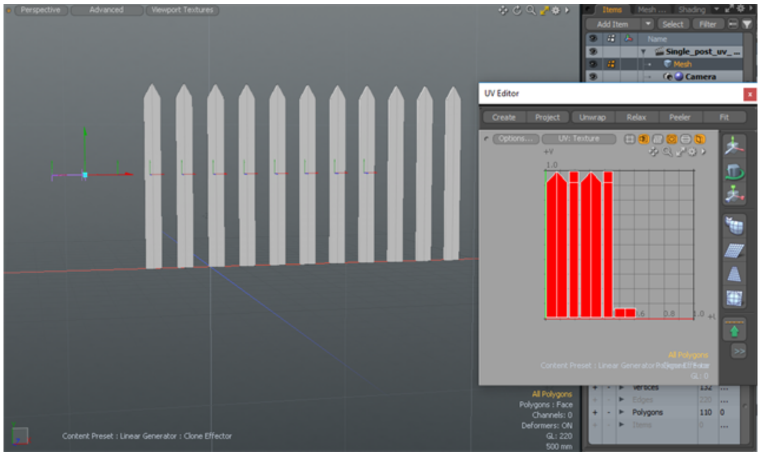
1. On the right panel, with the **Items** tab open, navigate to the bottom of the panel and open the **Lists** tab.
2. Expand **UV Maps**, right-click on **Texture**, and click **Clear**.
3. Open the **UV** layout, in the Toolbox on the left panel, click **Project**.
4. In the **Create UV** properties on the Toolbar, set the **Projection Type** to **Atlas**.

   The UV layout displays an update UV map, similar to the image below.


Options...
UV: Texture
1.2
1011
1012
1.0
0.8
0.6
0.4
0.2
1001
1002
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
Create UV
Mesh
Polygons : Face
GL: 0

To duplicate the fence post:

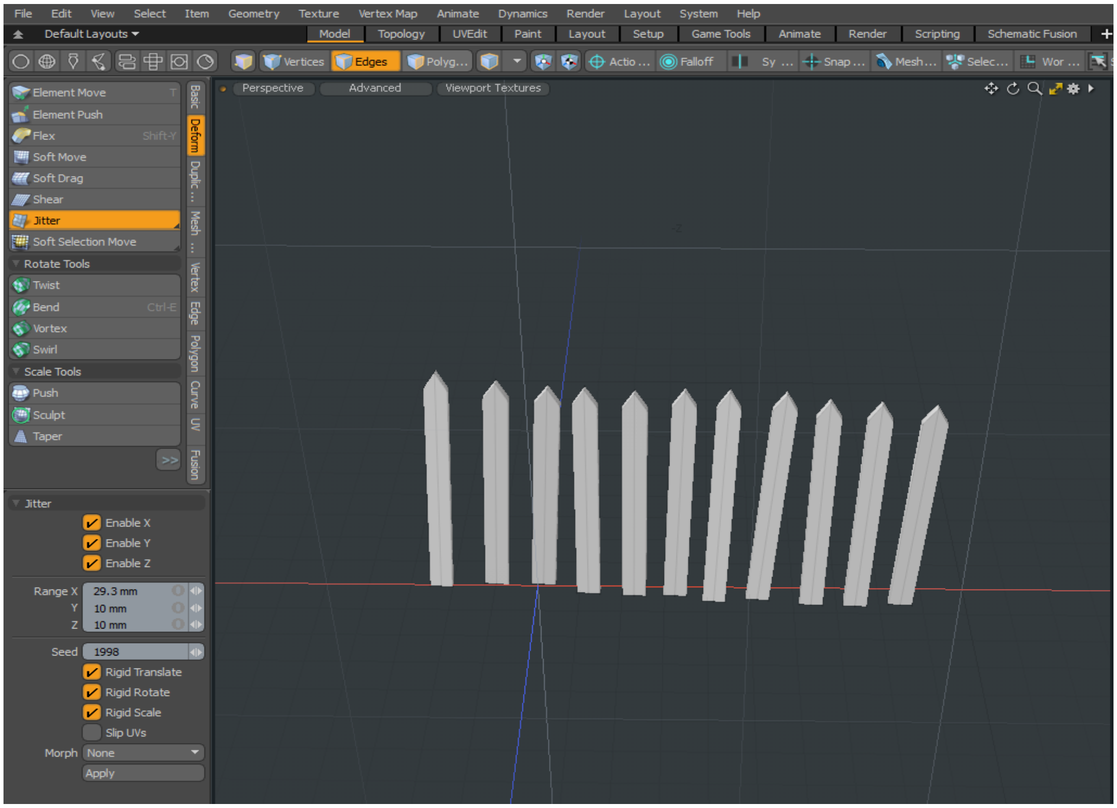
1. Under the menu bar and above the 3D viewport, click **Polygons**, and double-click on the fence post mesh.
2. Open the Model layout, in the Toolbox on the left panel, open the **Duplicate** sub-tab, and click **Clone**.
3. Set the **Number of Clones** to **10** and click-and-drag in the 3D viewport to position the duplicates.


Perspective
Advanced
Viewport Textures
Items
Add Item
Select
Filter
Single_post_uv_
Mesh
Camera
UV Editor
Create
Project
Unwrap
Relax
Peeler
Fit
Options...
UV: Texture
All Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 220
500 mm
Polygons
110
Content Preset : Linear Generator : Clone Effector

4. Press the **Spacebar** twice to drop the tool.

To add randomness to the fence:

1. In the **Items** tab, on the right panel, select the **Mesh** item.
2. Under the menu bar and above the 3D viewport, click **Edges**.
3. Lasso Select all of the fence posts.
4. On the Toolbar, open the **Deform** sub-tab and click **Jitter**.
5. Set the **Range** for the **X** , **Y**, and **Z** values to **29 mm**, **10 mm, 10 mm** respectively.
6. Enable **Rigid Translate**, **Rigid Rotate**, **Rigid Scale**, and click **Apply**.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Default Layouts
Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
Schematic Fusion
Vertices
Edges
Polyg...
Actio ...
Falloff
Sy ...
Snap ...
Mesh ...
Selec ...
Wor ...
Element Move
Element Push
Flex
Soft Move
Soft Drag
Shear
Jitter
Soft Selection Move
Rotate Tools
Twist
Bend
Vortex
Swirl
Scale Tools
Push
Sculpt
Taper
Basic
Deform
Duplic...
Mesh ...
Vertex
Edge
Polygon
Curve
UV
Fusion
Perspective
Advanced
Viewport Textures

Jitter

☑ Enable X

☑ Enable Y

☑ Enable Z

Range X 29.3 mm

Y 10 mm

Z 10 mm

Seed 1998

☑ Rigid Translate

☑ Rigid Rotate

☑ Rigid Scale

☐ Slip UVs

Morph None

Apply

The fence posts are now randomly placed.

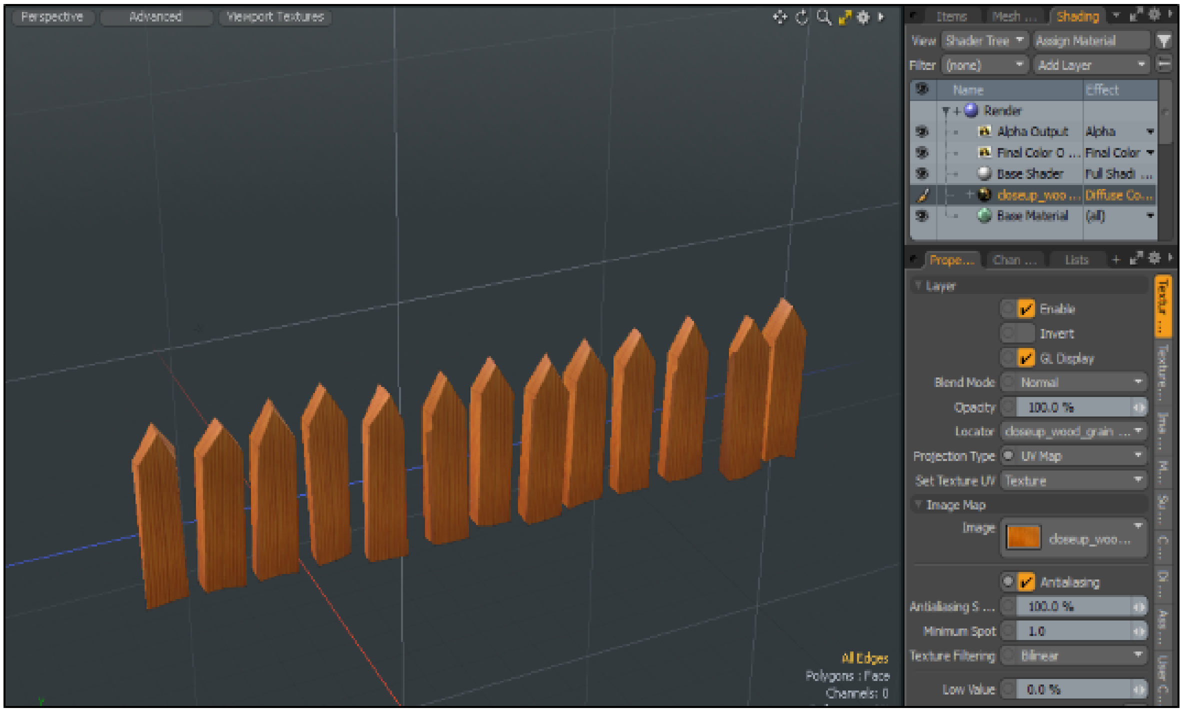
7. In the **List** tab, expand **UV Map**, right-click**Texture**, and select **Clear**.
8. On the left panel, open the **UV** tab, and click **Project**.

   A new set of UVs is created for all fence posts.

To create a random texture:

1. On the right panel, open the **Shading** tab, and select **Base Material**.
2. Click **Add Layer** > **Image Map** > **(load Image),** select a texture image, and click **Open**.

   The fence posts all have the a random texture image map applied.

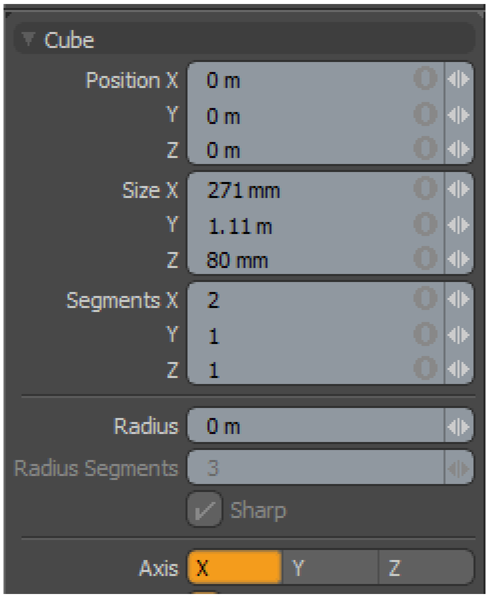

Perspective
Advanced
Viewport Textures
All Edges
Polygons : Face
Channels: 0
Items
Mesh ...
Shading
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color O ...
Final Color
Base Shader
Full Shadi ...
closeup_woo ...
Diffuse Co...
Base Material
(all)
Prope...
Chan ...
Lists
Layer
Image Map
Textur ...
Texture...
Ima ...
M...
Su...
C...
Di ...
Ass ...
User C...

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: closeup_wood_grain ...
Projection Type: UV Map
Set Texture UV: Texture
Image: closeup_woo...
☑ Antialiasing
Antialiasing S ...: 100.0 %
Minimum Spot: 1.0
Texture Filtering: Bilinear
Low Value: 0.0 %

Procedural Jitter Example

The following is an example of using the procedural **Jitter** tool to create a fence with random spacing between each post. By enabling the **Rigid Translate**, **Rigid Rotate**, and **Rigid Scale** options, all the duplicates are treated as individual objects. Use these options to quickly create real geometry without having to use Replicators. In addition, a Linear Falloff procedural mesh operation is used to demonstrate how to modulate the Jitter mesh operation.

1. In **Model** layout, on the left panel in the **Basic** sub-tab, click the Cube icon.
2. On the left panel, set the following properties and click **Apply**.



Cube

Position X 0 m
Y 0 m
Z 0 m

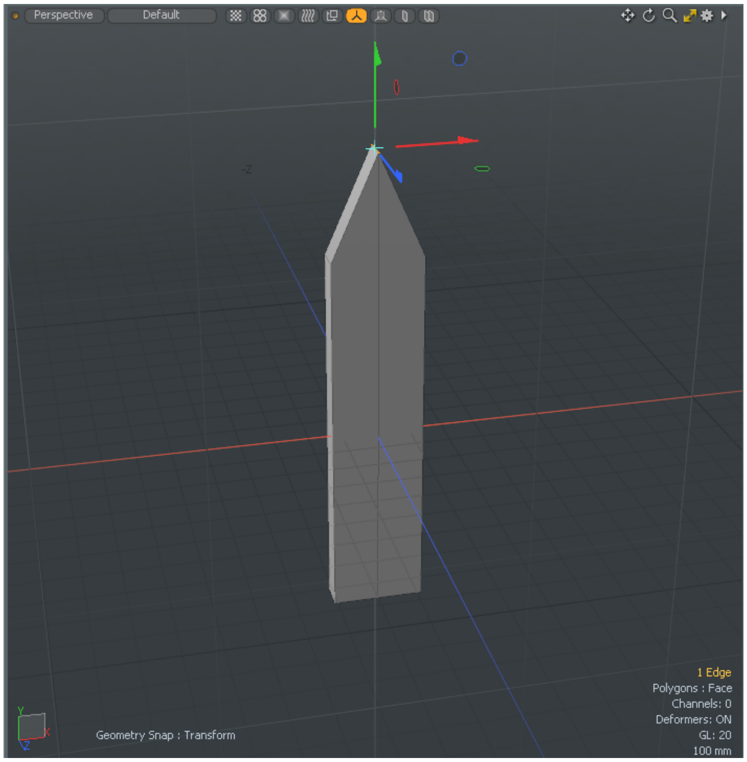
Size X 271 mm
Y 1.11 m
Z 80 mm

Segments X 2
Y 1
Z 1

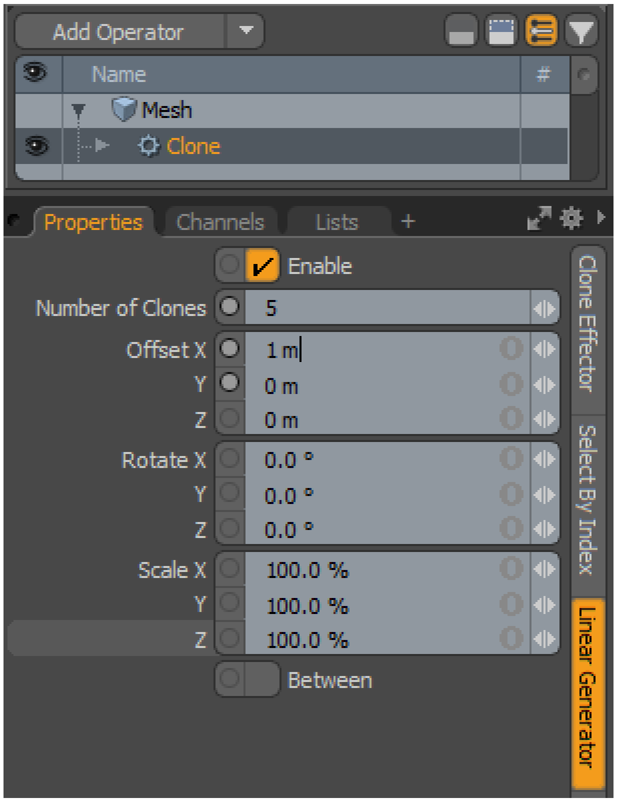
Radius 0 m
Radius Segments 3
Sharp

Axis X Y Z

3. Under the layout menus, click **Edges**, and select the top center edge.
4. Press **W** and drag the edge upwards to create a pointed fence post.


Perspective
Default
1 Edge
Polygons : Face
Channels: 0
Deformers: ON
GL: 20
100 mm
Geometry Snap : Transform

5. Press the **Spacebar** twice to drop the active tools.
6. Under the layout menus, click **Polygons**, and double-click on the fence post.
7. On the right panel, open the **Mesh Ops** tab, and click **Add Operator** > **Mesh Operations** > **Duplicate** > **Clone**.
8. On the right panel, open the **Linear Generator** tab and set **Number of Clones** to **5**, and **Offset X** to **1m**.


Add Operator
Name
#
Mesh
Clone
Properties
Channels
Lists
+
Clone Effector
Select By Index
Linear Generator

☑ Enable

Number of Clones 5

Offset X 1 m

Y 0 m

Z 0 m

Rotate X 0.0 °

Y 0.0 °

Z 0.0 °

Scale X 100.0 %

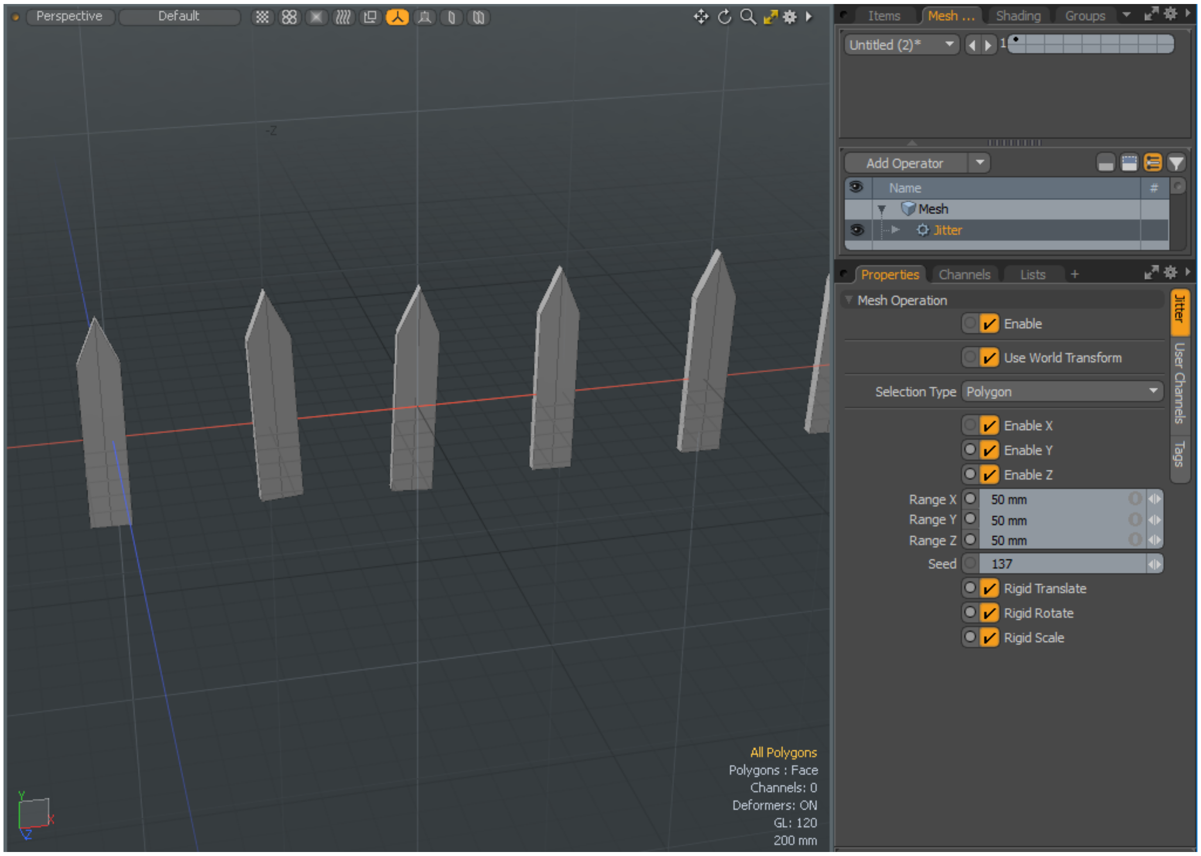
Y 100.0 %

Z 100.0 %

☐ Between

9. On the right panel, select the top mesh item, click **Add Operator**, and double-click **Mesh Operations** > **Deform** > **Jitter**.
10. Set the **Range X**, **Y**, and **Z** values to **50 mm** and enable **Rigid Translate**, **Rigid Rotate**, and **Rigid Scale**.

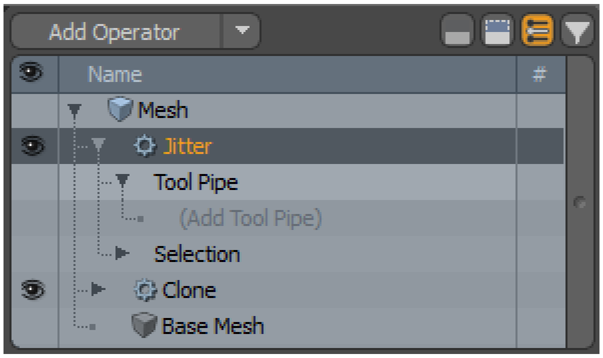
    The fence posts are randomly positioned.


Perspective
Default
Items
Mesh ...
Shading
Groups
Untitled (2)*
Add Operator
Name
#
Mesh
Jitter
Properties
Channels
Lists
Jitter
User Channels
Tags

Mesh Operation
☑ Enable
☑ Use World Transform
Selection Type: Polygon
☑ Enable X
☑ Enable Y
☑ Enable Z
Range X: 50 mm
Range Y: 50 mm
Range Z: 50 mm
Seed: 137
☑ Rigid Translate
☑ Rigid Rotate
☑ Rigid Scale

All Polygons
Polygons : Face
Channels: 0
Deformers: ON
GL: 120
200 mm

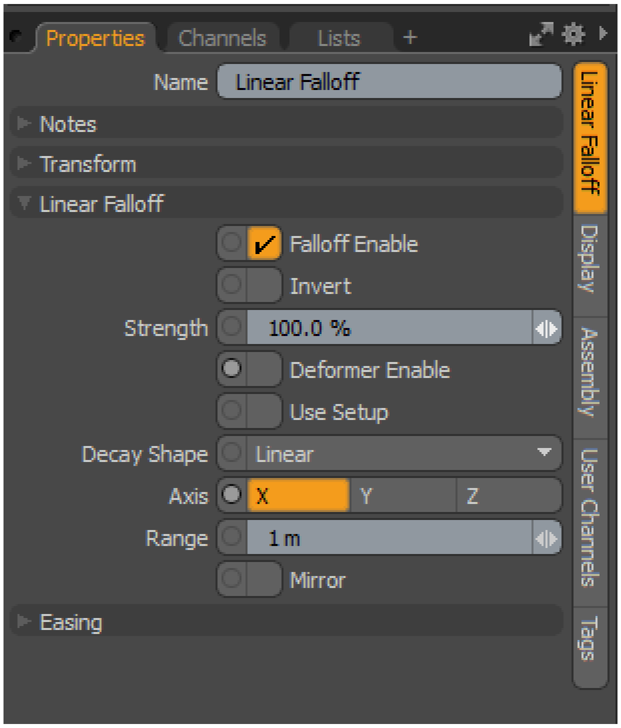
11. On the right panel, expand **Mesh** , **Jitter**, **Tool Pipe**, and click **(Add Tool Pipe)**.


Add Operator
Name
#
Mesh
Jitter
Tool Pipe
(Add Tool Pipe)
Selection
Clone
Base Mesh

12. At the top of the **Mesh Operations** dialog, type **Linear Falloff**, and double-click on its icon to add it to your scene.

    The Linear Falloff uses a straight line to define the falloff.

13. In the properties for the Linear Falloff, on the **Linear Falloff** tab, set **Axis** to **X**.


Properties
Channels
Lists
+
Notes
Transform
Linear Falloff
Easing
Linear Falloff
Display
Assembly
User Channels
Tags

Name: Linear Falloff

☑ Falloff Enable

☐ Invert

Strength: 100.0 %

☐ Deformer Enable

☐ Use Setup

Decay Shape: Linear

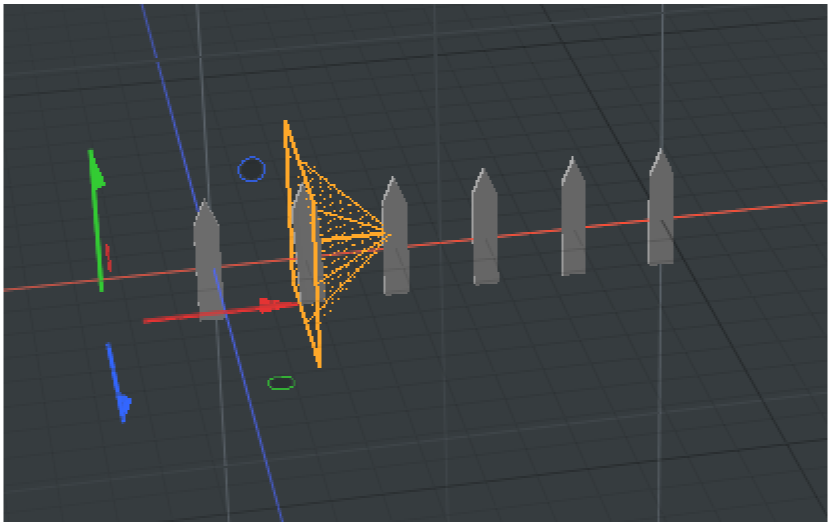
Axis: X | Y | Z

Range: 1 m

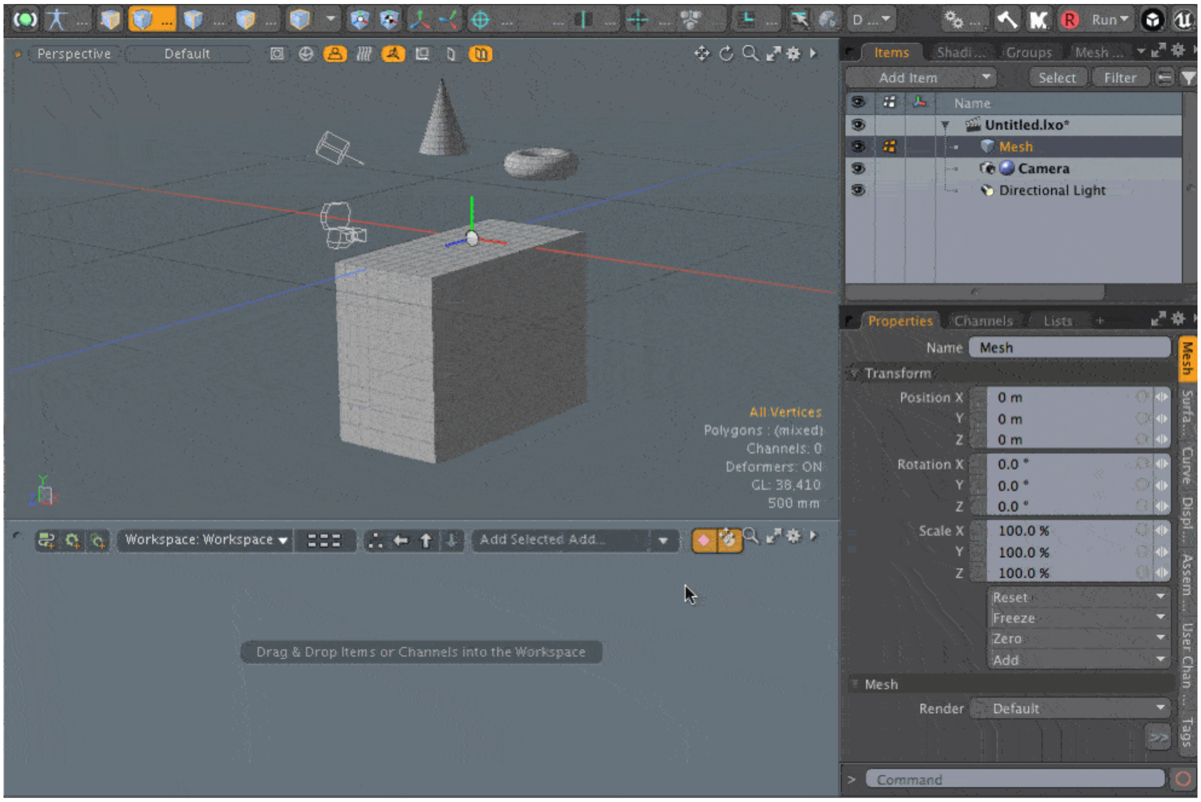
☐ Mirror

14. Press **W** and drag the red transfer handle through the series of fence posts.

    The Jitter properties are applied to each fence post as you drag the Linear Falloff over them.

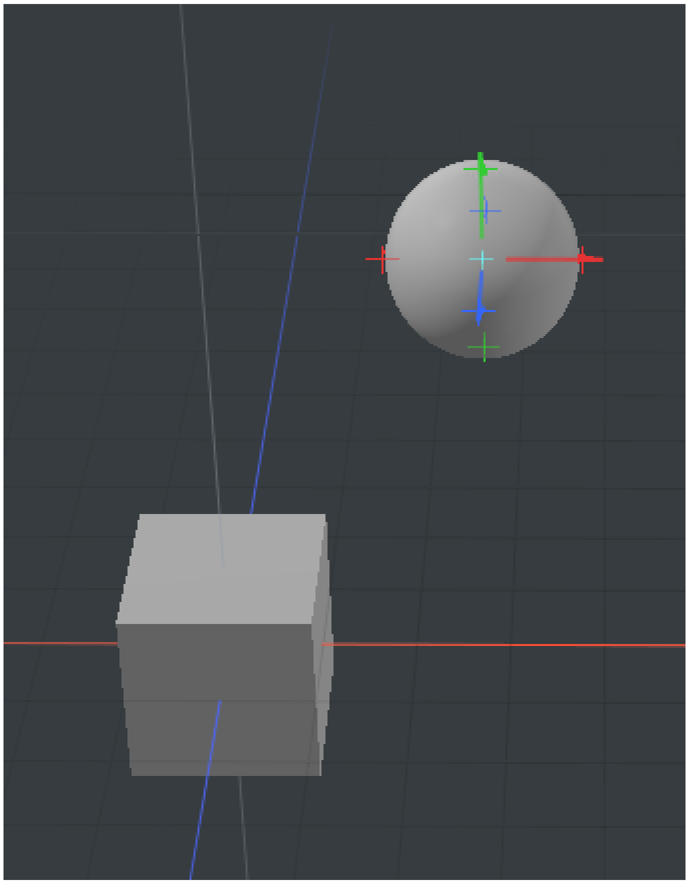


Schematic Jitter Example

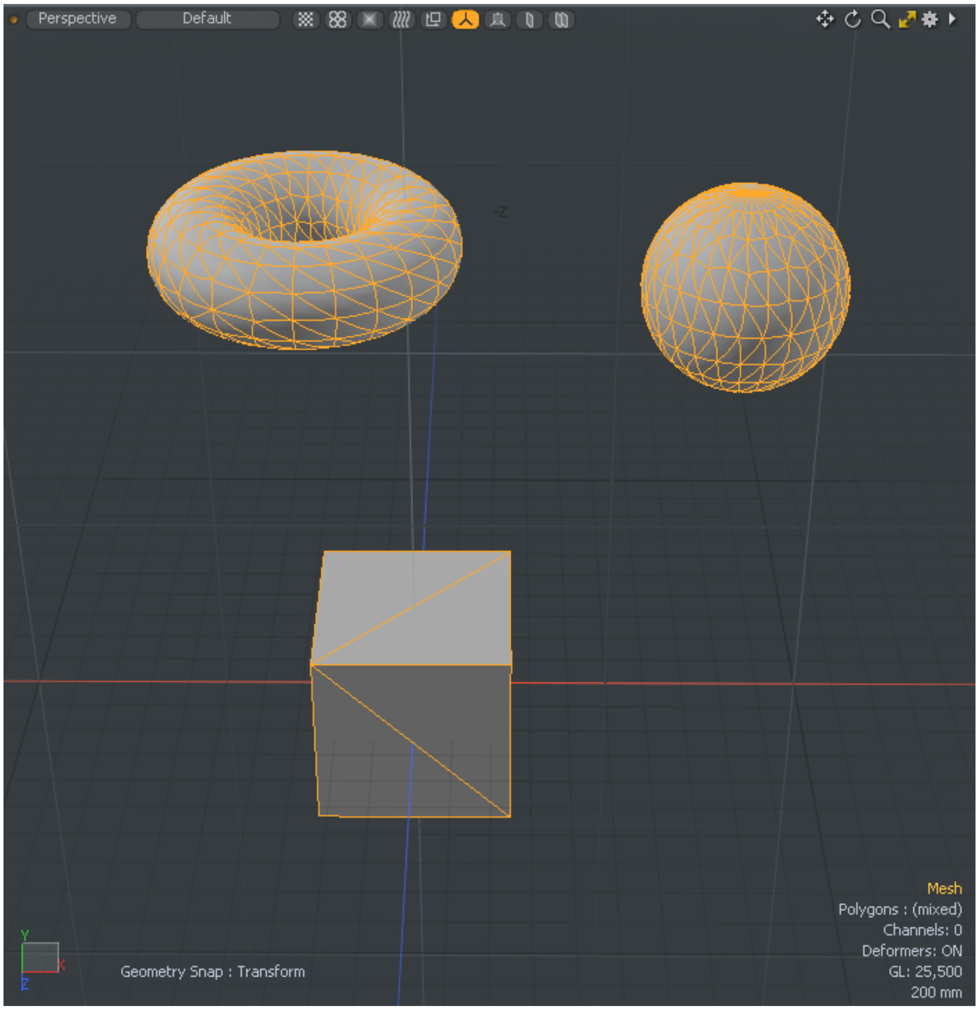

Perspective
Default
Items
Shadi...
Groups
Mesh ...
Add Item
Select
Filter
Name
Untitled.lxo*
Mesh
Camera
Directional Light
Properties
Channels
Lists
Name Mesh
Transform
Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Reset
Freeze
Zero
Add
Mesh
Render Default
Command
All Vertices
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 38,410
500 mm
Workspace: Workspace
Add Selected Add...
Drag & Drop Items or Channels into the Workspace

In this example, we are using a number of primitives to demonstrate how to apply the Jitter tool using the Schematic viewport.

1. Under the main menu, click **Model**.
2. On the left panel, on the **Basic** tab, click the **Cube** icon.
3. Right-click in the 3D viewport to position the cube, and then press the **Spacebar** twice to drop the tools.
4. On the left panel, on the **Basic** tab, click the **Sphere** icon.
5. In the 3D viewport, click-and-drag on the handles to reposition the mesh item, and press the **Spacebar** twice to drop the tools.



6. Repeat these steps to add a Torus primitive to your scene.

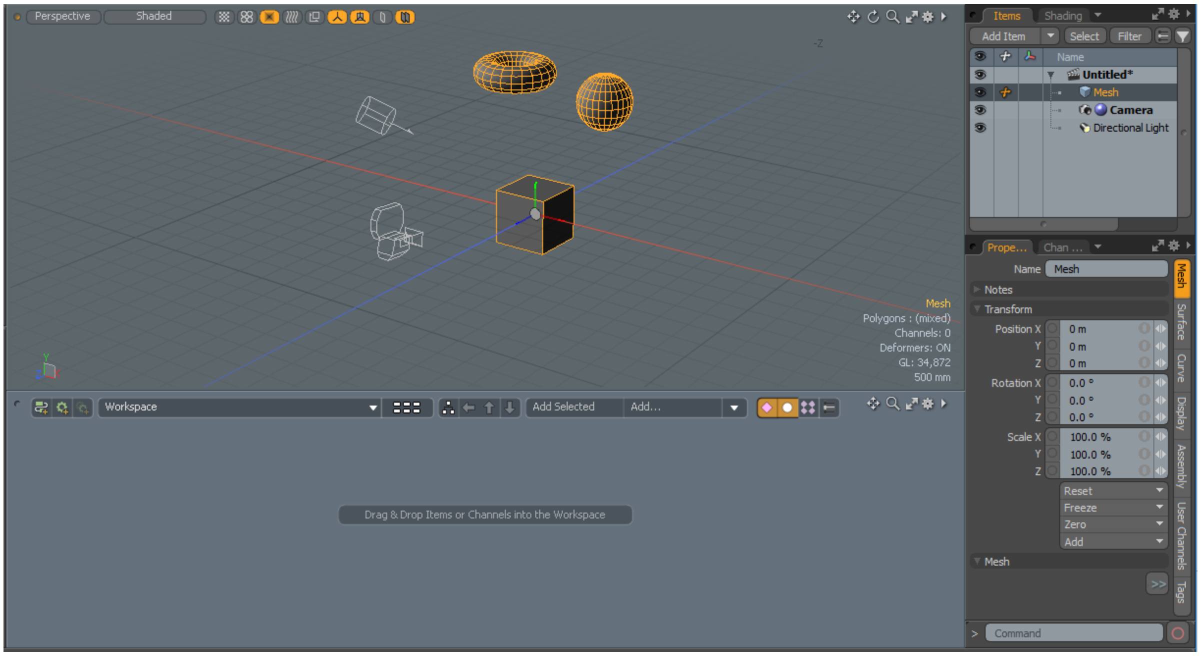

Perspective
Default
-Z
Y
X
Z

Geometry Snap : Transform

Mesh
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 25,500
200 mm

7. Under the main menu bar, click **Setup**.

   The Schematic viewport is displayed at the bottom of the layout.

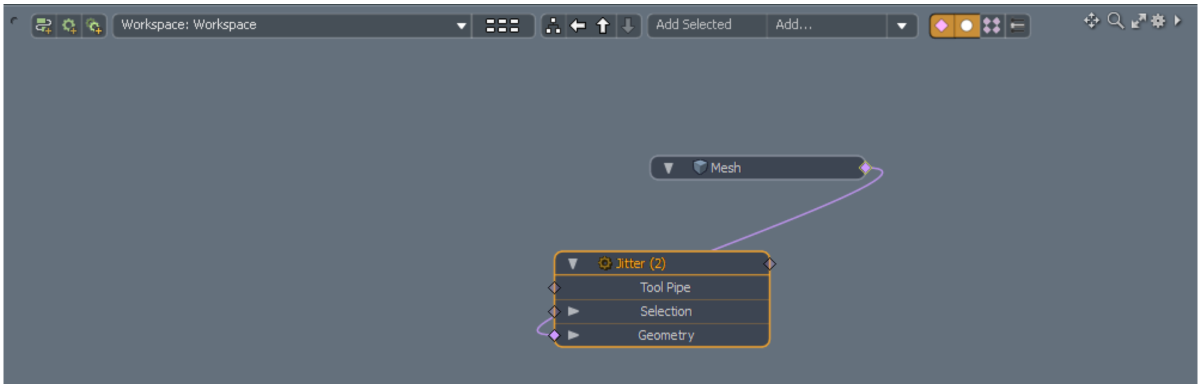

Perspective
Shaded
Items
Shading
Add Item
Select
Filter
Name
Untitled*
Mesh
Camera
Directional Light
Prope...
Chan ...
Name Mesh
Notes
Transform
Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Reset
Freeze
Zero
Add
Mesh
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags
Command
Mesh
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 34,872
500 mm
Workspace
Add Selected
Add...
Drag & Drop Items or Channels into the Workspace

8. On the right panel, open the **Items** tab, and drag your mesh item onto the Schematic viewport.

   The Mesh node displays in the Schematic viewport.

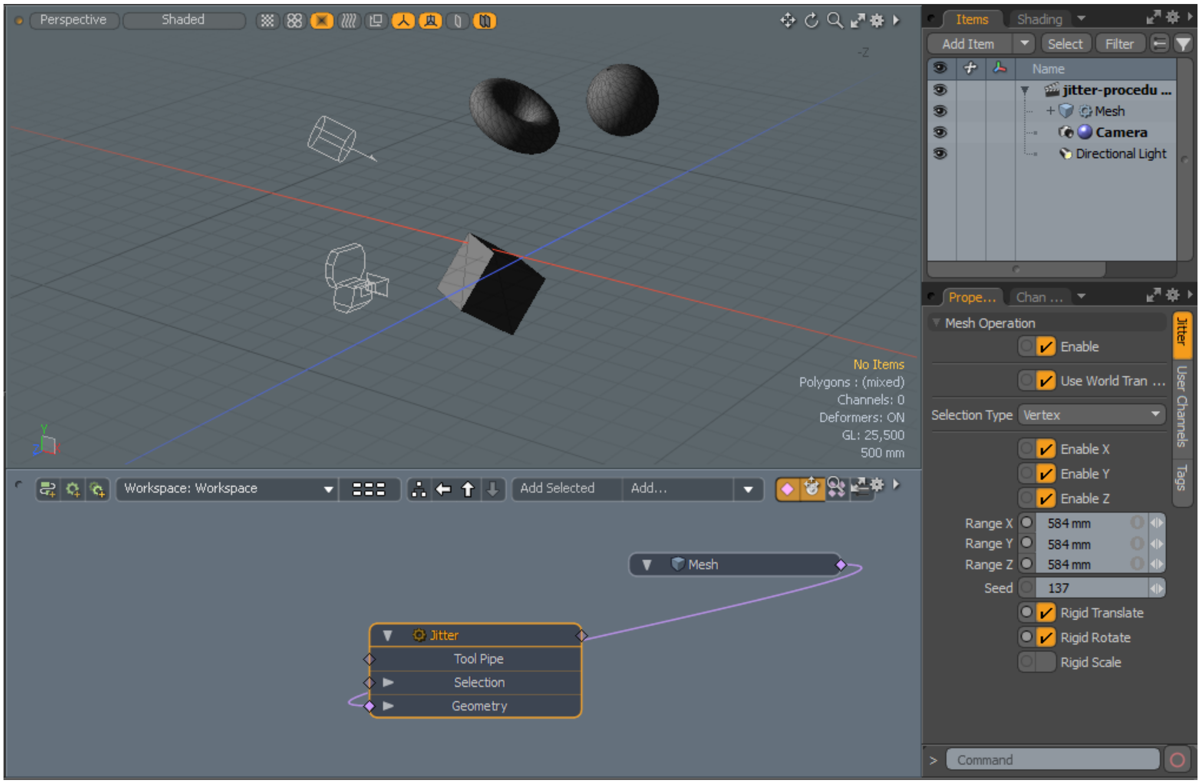
9. In the Schematic viewport, click **Add...**, and double-click **Mesh Operations** > **Deform** > **Jitter**.

   The Mesh node is automatically connected to the Jitter node.


Workspace: Workspace
Add Selected
Add...
Mesh
Jitter (2)
Tool Pipe
Selection
Geometry

10. On the right panel, change the **Range X**, **Y**, and **Z** values, and enable **Rigid Translate**, **Rigid Rotate**, and **Rigid Scale**.

    The mesh items are randomly positioned and scaled.


Perspective
Shaded
Items
Shading
Add Item
Select
Filter
Name
jitter-procedu ...
Mesh
Camera
Directional Light
Prope...
Chan ...
Jitter
User Channels
Tags
Workspace: Workspace
Add Selected
Add...
Mesh
Jitter
Tool Pipe
Selection
Geometry
Command

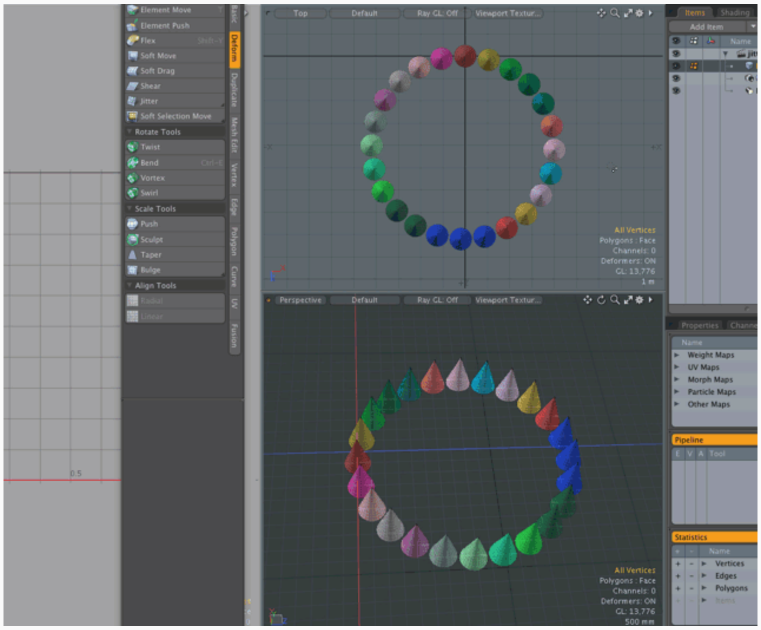
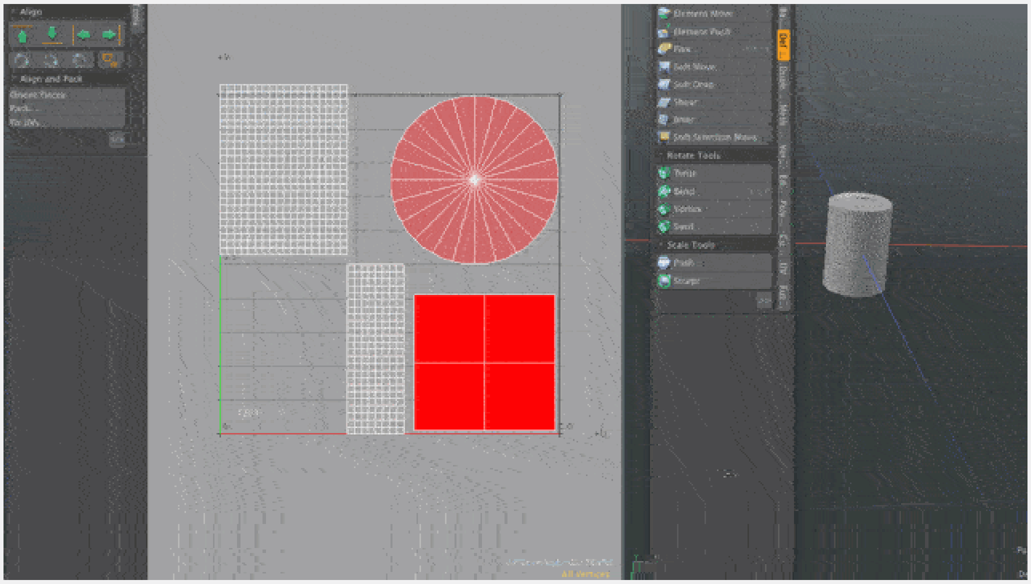
No Items
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 25,500
500 mm

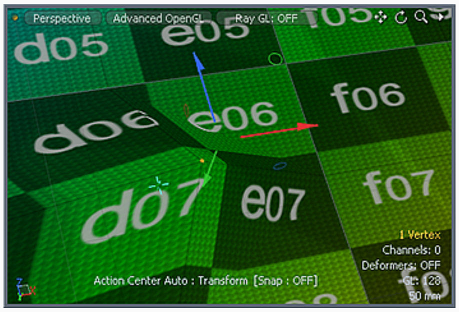
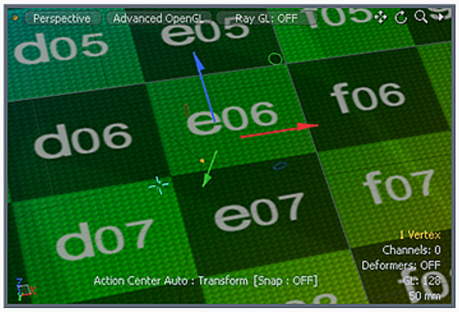
Mesh Operation
☑ Enable
☑ Use World Tran ...
Selection Type: Vertex
☑ Enable X
☑ Enable Y
☑ Enable Z
Range X: 584 mm
Range Y: 584 mm
Range Z: 584 mm
Seed: 137
☑ Rigid Translate
☑ Rigid Rotate
☐ Rigid Scale

Jitter Tool Properties

Enable	Toggles the tool off and on. This option is only available for a procedural modeling mesh operation.
Use World Transform	When enabled, the mesh operation maintains the world space on the input layer. This option is only available for a procedural modeling mesh operation.
Selection Type	Allows you to specify the selection type to apply Jitter to. • **Vertex** - Jitters the selected vertices and leaves the unselected vertices in your scene. • **Edge** - Jitters the selected edges and leaves the unselected edges in your scene. • **Polygon** - Jitters the selected polygon and leaves the unselected polygons in your scene.
Enable X/Enable Y/Enable Z	Enables the random jittering movement on a per axis basis for each axial direction.
Enable U / Enable V	When using the **UV Editor**, these options are displayed on the left panel. They

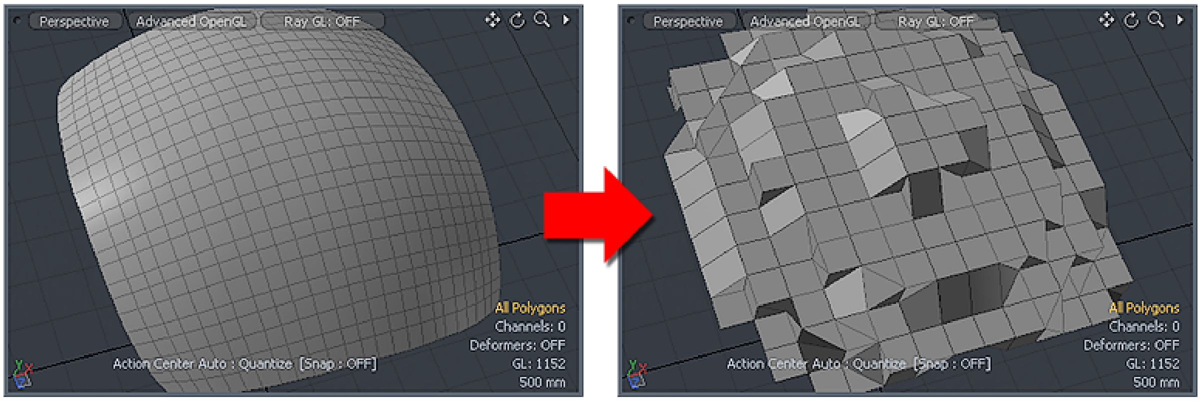
	enable the random jittering position on the U and V directions. **Note:** Range Z is not available in the UV Editor.
Range X/Y/Z	Controls the maximum random offset distance. Changing this value adjusts the strength of the jittering effect.
Seed	Provides the initial number for Modo to use when generating the random values. Each seed produces different random results.
Rigid Translate	Randomly move connected selections in X, Y and Z, or U and V direction. If you have a UV map, this option allows you to move each stacked island. **Note:** If using the direct modeling Jitter tool, the UI element displayed is **Translate**. If using the procedural mesh operation Jitter, the UI element displayed is **Rigid Translate**.
Rigid Rotate	Randomly rotate connected selections in X, Y and Z, or U and V direction. **Note:** If using the direct modeling Jitter tool, the UI element displayed is **Rotate**. If using the procedural mesh operation Jitter, the UI element displayed is **Rigid Rotate**.
Rigid Scale	Randomly scale connected selections in X, and Z, or U and V direction.

	 **Note:** If using the direct modeling Jitter tool, the UI element displayed is **Scale**. If using the procedural mesh operation Jitter, the UI element displayed is **Rigid Scale**.
Locked Stacked	When using the **UV Editor**, this option displays. It locks and unlocks stacked UV's so they can be randomized as a group or as individual UV islands. 

Slip UVs	When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping) >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.

Quantize

The Quantize tool evaluates the contour of the selected geometry and shuffles the vertices to align to a three-dimensional grid.


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 1152
500 mm
Action Center Auto : Quantize [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 1152
500 mm
Action Center Auto : Quantize [Snap : OFF]

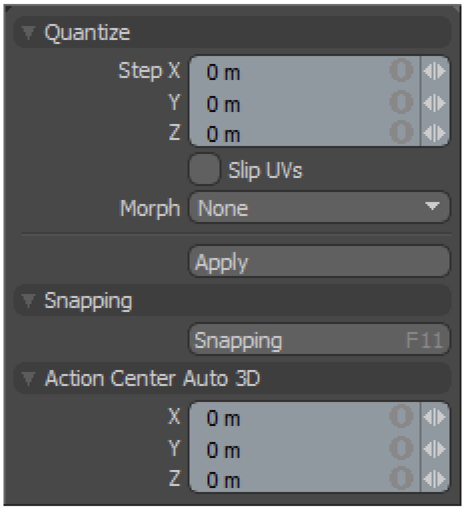
Using Quantize

You can find the tool in the **Model** layout, in the **Deform** sub-tab of the modeling toolbox, sometimes as a sub-option of the Smooth or Jitter tools.

1. Click the **Quantize** button. If it is not visible, click and hold the **Smooth/Jitter** button to reveal the additional options, then select **Quantize**.
2. Click and drag in the viewport, to interactively apply quantize, or edit the values in the tool's **Properties** panel.

   As you increase the **Step** value for each axis, Modo increases the size of each block (voxel) in the 3D grid. Modo moves each vertex to the closest matching point in the voxel.

The following options are available for the tool:



Quantize

Step X 0 m

Y 0 m

Z 0 m

☐ Slip UVs

Morph None

Apply

Snapping

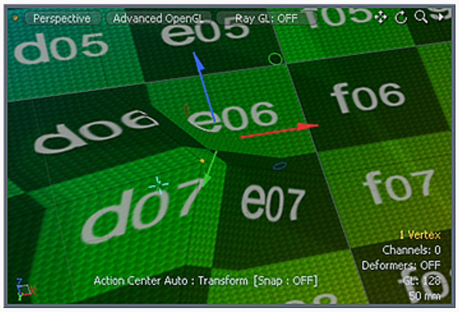
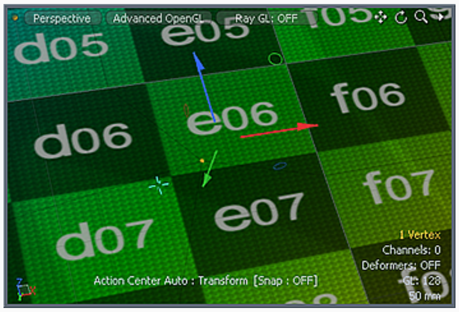
Snapping F11

Action Center Auto 3D

X 0 m

Y 0 m

Z 0 m

Quantize	
Step X/Y/Z	Defines the grid spacing for each axis.
Slip UVs	When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  **Slip UVs** function disabled (notice the texture warping) >  **Slip UVs** function enabled (texture remains even)
Morph	Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their

Quantize	
	source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Action Center Auto 3D	See the Specifying Action Centers and Falloffs topic for information.

Symmetry Tool

With the **Symmetry** tool you can interactively correct minor symmetry errors. Modo uses a distance tolerance to decide if vertices have a corresponding vertex on the opposite side of the axis of symmetry. If Modo locates a vertex within the tolerance amount, it moves that vertex to match its partner on the other side of the mesh.

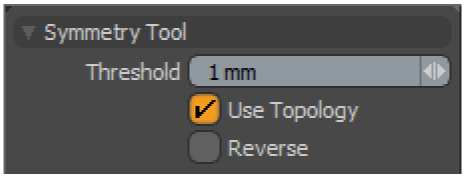
Note: Symmetry must be active to use this tool. You activate the **Symmetry** function by selecting an axis from the **Symmetry** list on Modo's **Modes** toolbar, or by choosing **Edit** > **Symmetry** > **axis to mirror across**. See the Applying Precision topic for detailed information.

Using the Symmetry Tool

1. Click the **Symmetry** button at the top of the viewport.
2. Add the geometry to your scene.
3. In the menu bar, click **Geometry** > **Symmetry Tool**.
4. Click in the viewport and drag to increase the threshold distance.

The **Symmetry Tool** properties are displayed in the left panel. Modo indicates non-symmetrical vertices with yellow squares that disappear as you increase the threshold. If all the indicated vertices (yellow squares) do not disappear, then re-apply the tool with the current threshold by middle-clicking. This may clean up remaining vertices.

Note: When enabled, **Use Topology** only works if some part of the model is symmetrical to start with. If this isn't the case then clear the **Use Topology** checkbox.



Symmetry Tool

Threshold 1 mm

☑ Use Topology

☐ Reverse

Threshold	Indicates the threshold value. You can interactively adjust the **Threshold** value by dragging in the 3D viewport. Alternatively, type a specific value.
Use Topology	When enabled, works recursively from a point that Modo knows to be symmetrical and assumes that similarly connected neighbors should also be symmetric instead of finding a corresponding vertex directly across an axis. This allows for a much more robust solution because there is less chance of considering one vertex and then finding the wrong corresponding vertex.
Reverse	This option determines the symmetry side to modify. When disabled, vertex positions on the negative side are synchronized with the corresponding vertices on the positive side. When enabled, the vertex positions on the positive side of the symmetry axis are moved.

Soft Selection Move

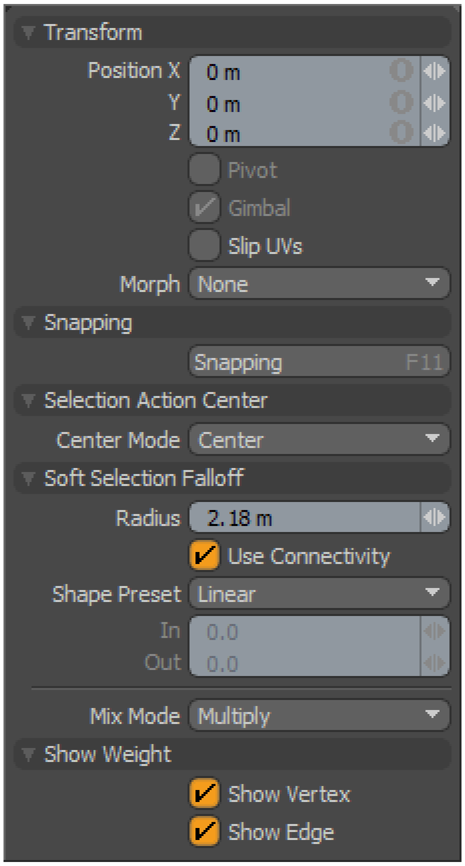
The Soft Selection Move tool is a deformation tool that moves or translates the selected geometry rigidly, while gradually attenuating the translation outside of the selected area defined by the **Radius** value for the Soft Selection falloff.

Using Soft Selection Move

You can find the tool in the Model layout, in the **Deform** sub-tab of the modeling toolbox (sometimes as a sub-tool of one of the other Soft Selection tools).

Click the **Soft Selection Move** button to activate the tool, then click in the viewport to reveal the transform handles. As you increase the radius, the softness of the transformation also increases.

Besides manipulating the handles in the 3D viewport, you can edit the values in the tool's **Properties** panel. The following options are available for the Soft Selection Move tool:



Transform

Position X 0 m
Y 0 m
Z 0 m
☐ Pivot
☑ Gimbal
☐ Slip UVs
Morph None

Snapping

Snapping F11

Selection Action Center

Center Mode Center

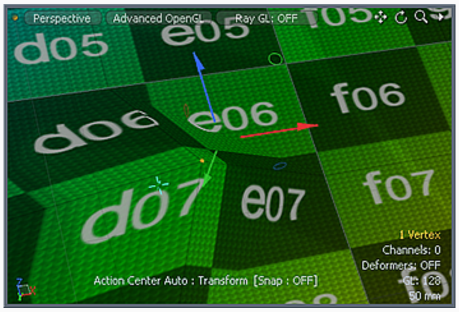
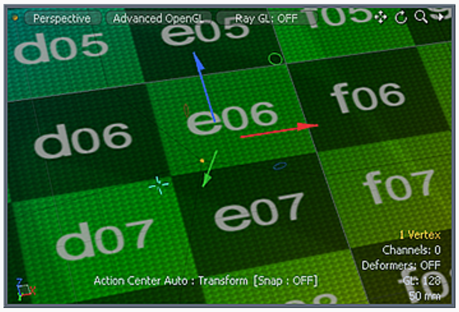
Soft Selection Falloff

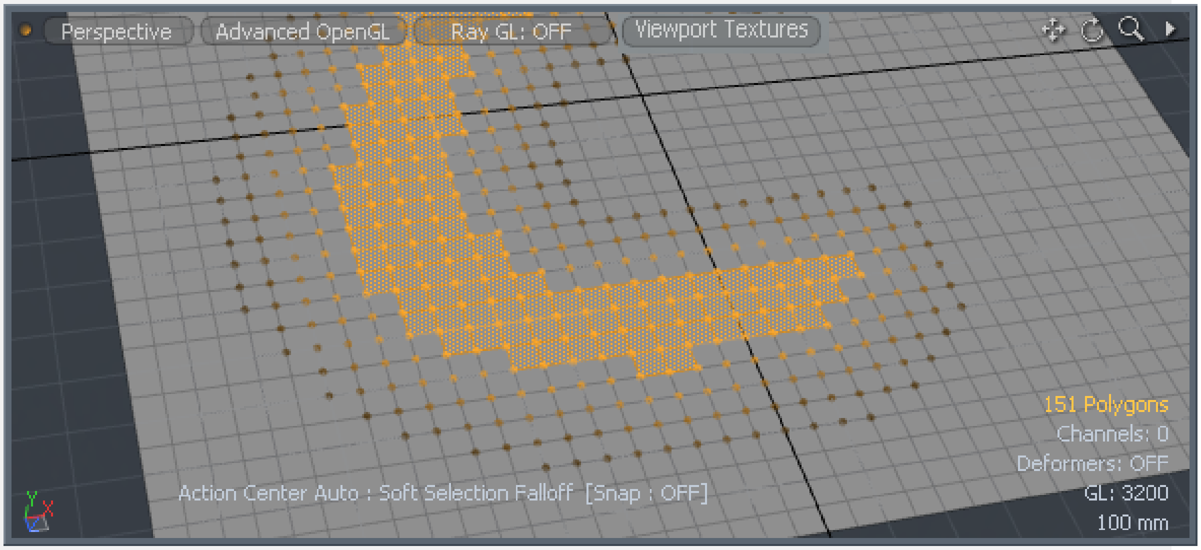
Radius 2.18 m
☑ Use Connectivity
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply

Show Weight

☑ Show Vertex
☑ Show Edge

Transform	
Position X/Y/Z	Displays the offset distance for each axis (calculated from the position the object was in when you activated the tool). When adjusting an object interactively in the viewport, the **Position** values indicate the current offset distance applied.
Pivot	Disabled for the Soft Selection Move tool.
Gimbal	Disabled for the Soft Selection Move tool.
Slip UVs	When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result,

Transform	
	you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping) >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Selection Action Center	See the Specifying Action Centers and Falloffs topic for information.

Transform	
Soft Selection Falloff	
Radius	Determines the range outside of the selected edge to attenuate the falloff.
Use Connectivity	When enabled, affects only single-surface, connected elements. Modo ignores unconnected elements within range.
Shape	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).
Show Weights	
Show Vertex/Show Edge	When enabled, displays indicators of vertices or edges for a visual reference about the Soft Selection falloff. The image below shows **Show Vertex** enabled:  Perspective Advanced OpenGL Ray GL: OFF Viewport Textures Action Center Auto : Soft Selection Falloff [Snap : OFF] 151 Polygons Channels: 0 Deformers: OFF GL: 3200 100 mm

Soft Selection Rotate

The Soft Selection Rotate tool is a deformation tool, which rotates the selected geometry rigidly, while gradually attenuating the rotation amount outside of the selected area as defined by the Soft Selection **Radius**.

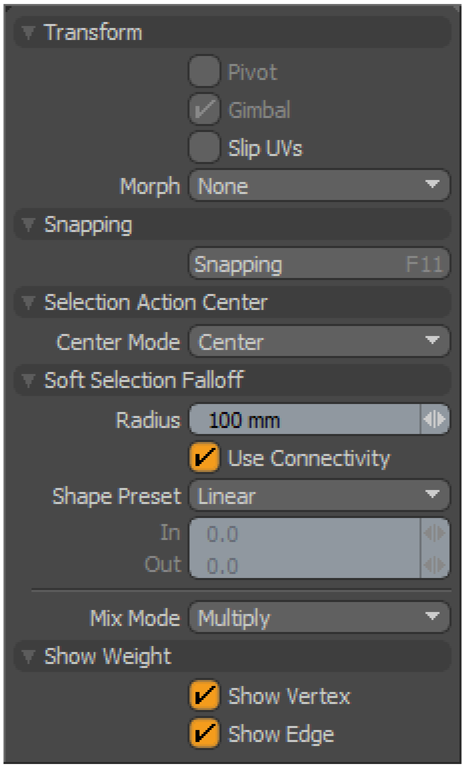
Using Soft Selection Rotate

You can find the Soft Selection Rotate tool in the **Deform** sub-tab of the modeling toolbox.

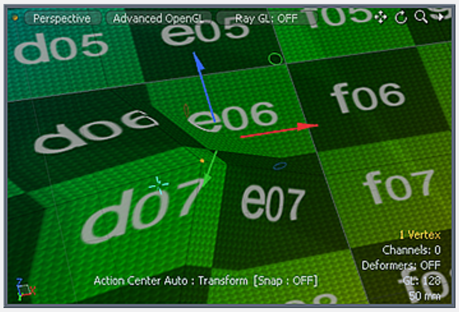
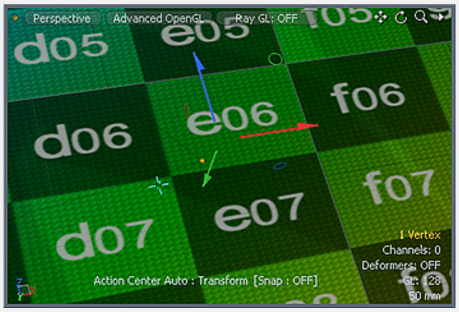
1. By default, it can be found as an additional option under the Soft Selection Move tool. To reveal it, click and hold the **Soft Selection Move** tool, then select **Soft Selection Rotate** from the list of additional tool.
2. With the tool active, click in the 3D viewport to reveal the tool handles.
3. Adjust the handles in the viewport, or edit the values in the tool's **Properties** panel.

   As the **Radius** value increases, so does the softness of the transformation.

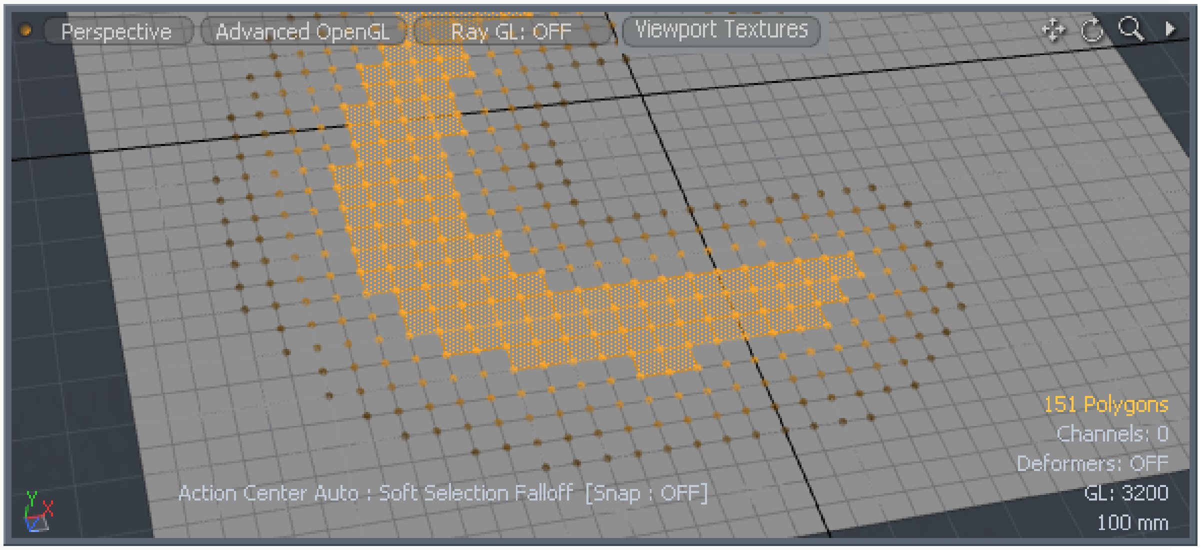
The following options are available for the tool:



Transform
☐ Pivot
☑ Gimbal
☐ Slip UVs
Morph None
Snapping
Snapping F11
Selection Action Center
Center Mode Center
Soft Selection Falloff
Radius 100 mm
☑ Use Connectivity
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply
Show Weight
☑ Show Vertex
☑ Show Edge

Transform	
Pivot	Disabled for the Soft Selection Rotate tool.
Gimbal	Disabled for the Soft Selection Rotate tool.
Slip UVs	When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping) >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions.

Transform	
	Note: In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Selection Action Center	See the Specifying Action Centers and Falloffs topic for information.
Soft Selection Falloff	
Radius	Determines the range outside of the selected edge to attenuate the falloff.
Use Connectivity	When enabled, affects only single-surface, connected elements. Modo ignores unconnected elements within range.
Shape	Controls the strength of the falloff's influence along the extent by using a shape preset. **Linear** - Attenuates the falloff evenly across its range. **Ease-In** - Strengthens the falloff toward the start position. **Ease-Out** - Strengthens the falloff toward the end position. **Smooth** - Strengthens the falloff toward the center of the falloff. **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).
Show Weights	
Show Vertex/Show	When enabled, displays indicators of vertices or edges for a visual reference about the Soft Selection falloff.

Transform	
Edge	The image below shows **Show Vertex** enabled:  Perspective Advanced OpenGL Ray GL: OFF Viewport Textures Action Center Auto : Soft Selection Falloff [Snap : OFF] 151 Polygons Channels: 0 Deformers: OFF GL: 3200 100 mm

Soft Selection Scale

the Soft Selection Scale tool is a deformation tool, which scales or re-sizes the selected geometry rigidly, while gradually attenuating the scale amount outside of the selected area defined by the Soft Selection **Radius**.

Using Soft Selection Scale

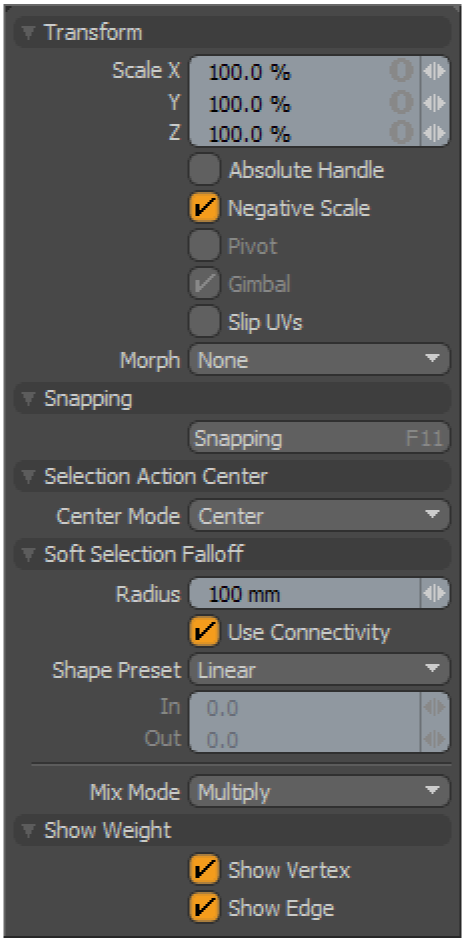
You can find the Soft Selection Scale tool in the **Deform** sub-tab of the modeling toolbox.

1. By default, it can be found as an additional option under the Soft Selection Move tool. To reveal it, click and hold the **Soft Selection Move** button, then select **Soft Selection Transform** from the list of additional tool.

   Tool handles appear in the 3D viewport.
2. Adjust the handles in the viewport, or edit the values in the tool's **Properties** panel.

   As the **Radius** value increases, so does the softness of the transformation.

The following options are available for the tool:



Transform

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

☐ Absolute Handle

☑ Negative Scale

☐ Pivot

☑ Gimbal

☐ Slip UVs

Morph None

Snapping

Snapping F11

Selection Action Center

Center Mode Center

Soft Selection Falloff

Radius 100 mm

☑ Use Connectivity

Shape Preset Linear

In 0.0

Out 0.0

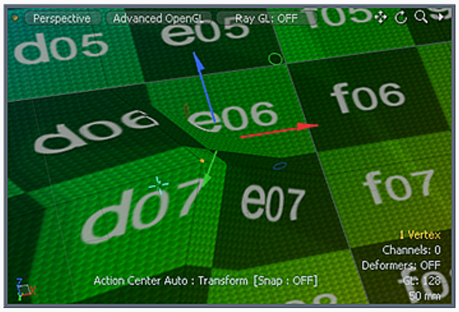
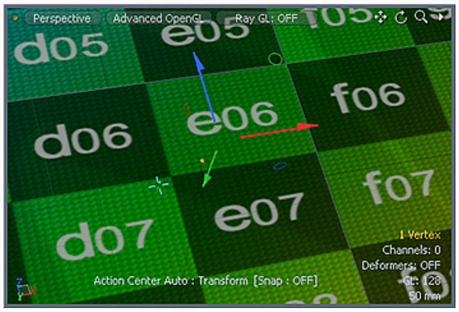
Mix Mode Multiply

Show Weight

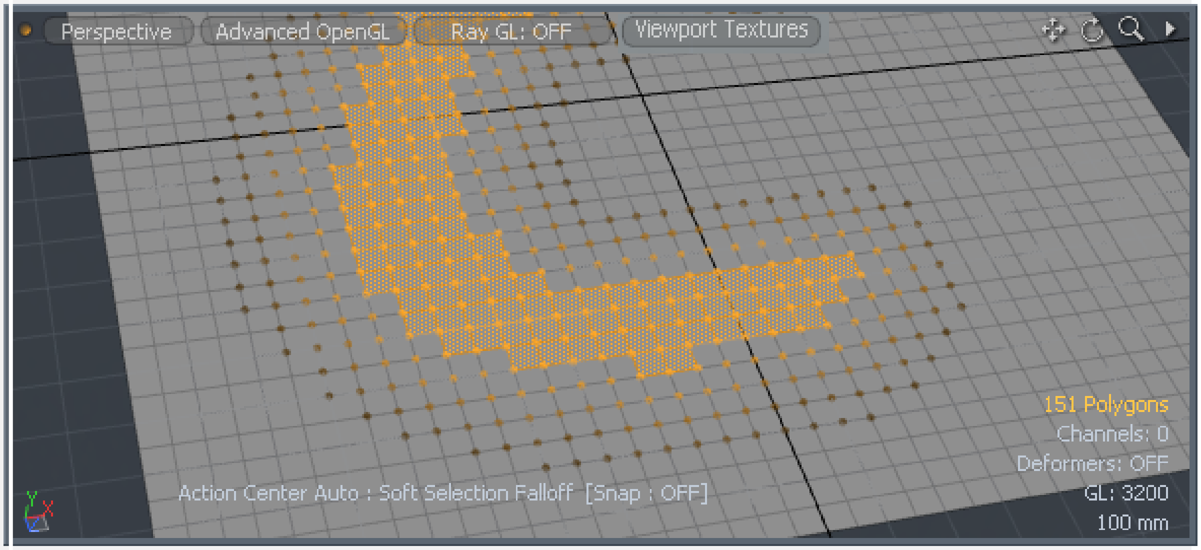
☑ Show Vertex

☑ Show Edge

Transform	
Scale X/Y/Z	Displays the scaling factor for each axis. When adjusting an object interactively in the viewport, the **Scale** values indicate the current scaling factor applied.
Absolute Handle	When enabled, applies an additional function to the tool handles for you to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.
Negative Scale	When enabled, applies negative scale values, if appropriate, when you manipulate the tool handles in the viewport. When disabled, Modo does not automatically change the values if they are negative. You can always input negative values directly.
Pivot	Disabled for the Soft Selection Scale tool.
Gimbal	Disabled for the Soft Selection Scale tool.
Slip UVs	When enabled, edits applied to the geometry do not change the existing UV map. UV

Transform	
	values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f06 d06 e06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping) >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f06 d06 e06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.

Transform	
Selection Action Center	See the Specifying Action Centers and Falloffs topic for information.
Soft Selection Falloff	
Radius	Determines the range outside of the selected edge to attenuate the falloff.
Use Connectivity	When enabled, affects only single-surface, connected elements. Modo ignores unconnected elements within range.
Shape	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).
Show Weights	
Show Vertex/Show Edge	When enabled, displays indicators of vertices or edges for a visual reference about the Soft Selection falloff. The image below shows **Show Vertex** enabled:

Transform	
	

Perspective
Advanced OpenGL
Ray GL: OFF
Viewport Textures
151 Polygons
Channels: 0
Deformers: OFF
GL: 3200
100 mm
Action Center Auto : Soft Selection Falloff [Snap : OFF]

Soft Selection Transform

The Soft Selection Transform tool is a deformation tool, which allows you to simultaneously move, rotate, and scale the selected geometry rigidly, while gradually attenuating the translation outside of the selected area defined by the Soft Selection **Radius**.

Using Soft Selection Transform

You can find the Soft Selection Transform tool in the **Deform** sub-tab of the modeling toolbox.

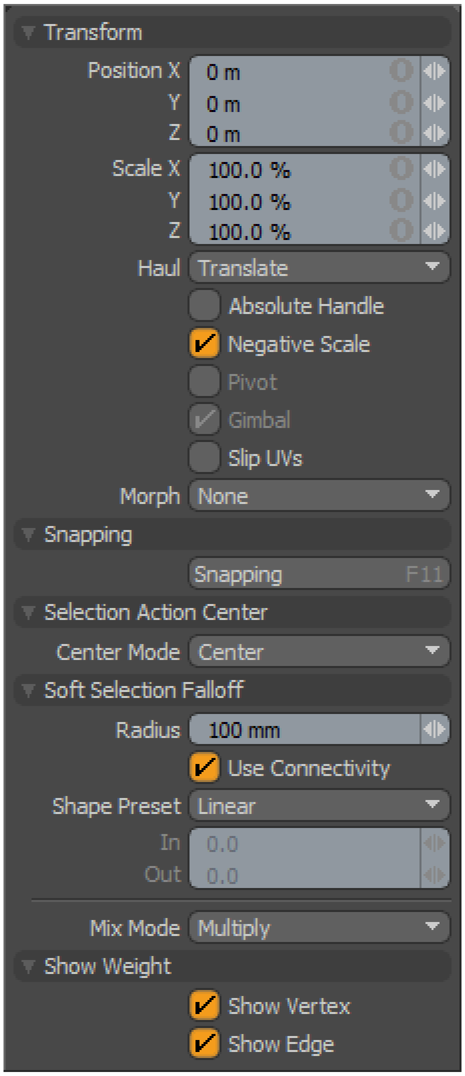
1. By default, it can be found as an additional option under the Soft Selection Move tool. To reveal it, click and hold the **Soft Selection Move** button, then select **Soft Selection Scale** from the list of additional tool.

   Tool handles appear in the 3D viewport.

2. Adjust the handles in the viewport, or edit the values in the tool's **Properties** panel.

   As the **Radius** value increases, so does the softness of the transformation.

The following options are available for the tool:



Transform

Position X 0 m

Y 0 m

Z 0 m

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

Haul Translate

☐ Absolute Handle

☑ Negative Scale

☐ Pivot

☑ Gimbal

☐ Slip UVs

Morph None

Snapping

Snapping F11

Selection Action Center

Center Mode Center

Soft Selection Falloff

Radius 100 mm

☑ Use Connectivity

Shape Preset Linear

In 0.0

Out 0.0

Mix Mode Multiply

Show Weight

☑ Show Vertex

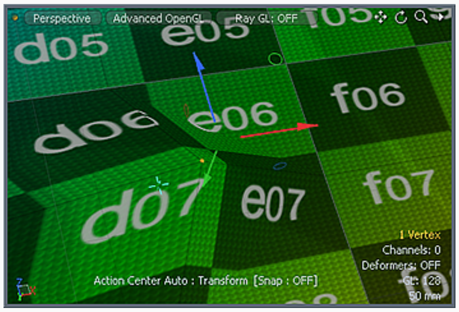
☑ Show Edge

Transform	
Position X/Y/Z	Displays the offset distance for each axis (calculated from the position the object was in when you activated the tool). When adjusting an object interactively in the viewport, the **Position** values indicate the current offset distance applied.
Scale X/Y/Z	Displays specific scaling values applied to the geometry. When adjusting an object interactively in the viewport, the **Scale** input values indicate the current scale factor applied.
Haul	Specifies what action dragging the pointer (when not dragging tool handles) does in the viewport. • **Translate** - Moves the selection. • **Rotate** - Rotates the selection.

<table>
<tr><th colspan="2">Transform</th></tr>
<tr><td></td><td>• Scale - Resizes the selection.</td></tr>
<tr><td>Absolute Handle</td><td>When enabled, applies an additional function to the tool handles for you to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.</td></tr>
<tr><td>Negative Scale</td><td>When enabled, applies negative scale values, if appropriate, when you manipulate the tool handles in the viewport. When disabled, Modo does not automatically change the values if they are negative. You can always input negative values directly.</td></tr>
<tr><td>Pivot</td><td>Disabled for the Soft Select Transform tool.</td></tr>
<tr><td>Gimbal</td><td>Disabled for the Soft Select Transform tool.</td></tr>
<tr><td>Slip UVs</td><td>(Only in Component Modes)

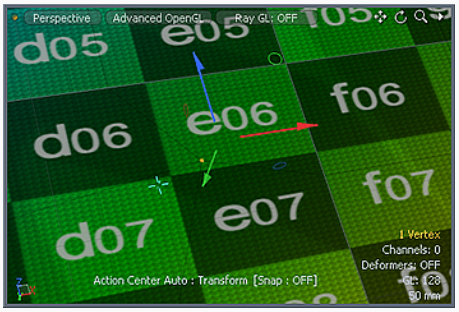
When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable Slip UVs so as to not disturb any existing UV mapping applied to the geometry.

>


Perspective
Advanced OpenGL
Ray GL: OFF
d05
e05
f05
d06
e06
f06
d07
e07
f07
1 Vertex
Channels: 0
Deformers: OFF
GL: 128
50 mm
Action Center Auto : Transform [Snap : OFF]

Slip UVs function disabled (notice the texture warping)

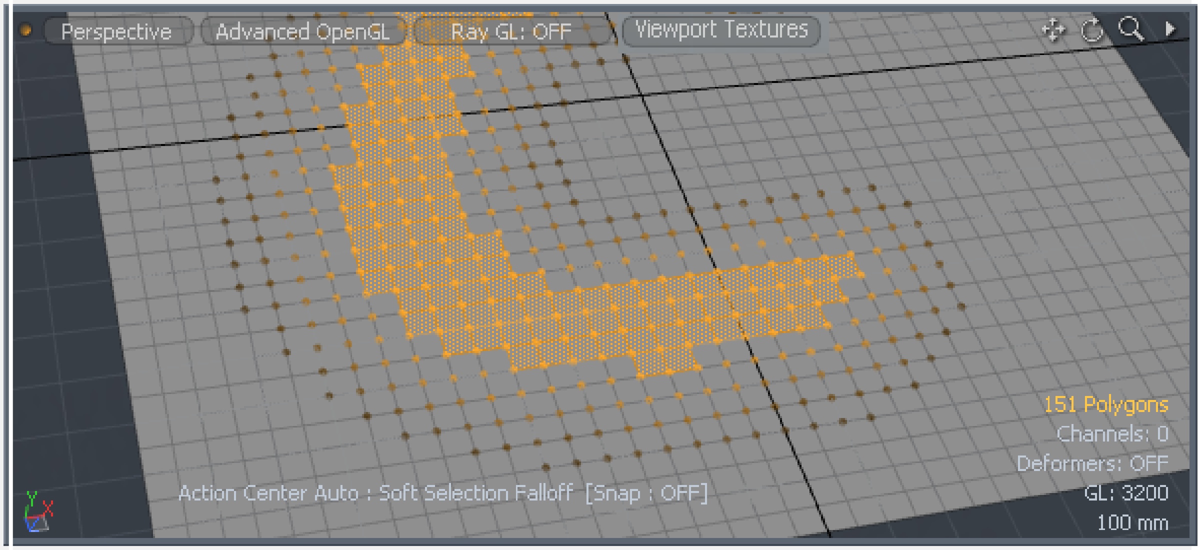
>


Perspective
Advanced OpenGL
Ray GL: OFF
d05
e05
f05
d06
e06
f06
d07
e07
f07
1 Vertex
Channels: 0
Deformers: OFF
GL: 128
50 mm
Action Center Auto : Transform [Snap : OFF]

Slip UVs function enabled (texture remains even)</td></tr>
<tr><td>Morph</td><td>(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms.

• None - Transforms selected (visible) morphs independent of their source, but does not</td></tr>
</table>

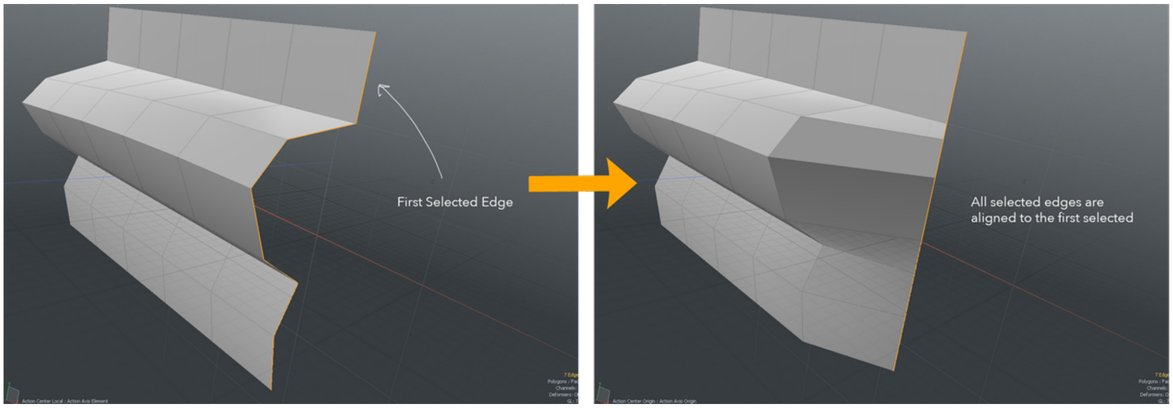
Transform	
	affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Selection Action Center	See the Specifying Action Centers and Falloffs topic for information.
Soft Selection Falloff	
Radius	Determines the range outside of the selected edge to attenuate the falloff.
Use Connectivity	When enabled, affects only single-surface, connected elements. Modo ignores unconnected elements within range.
Shape	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Transform	
Show Weights	
Show Vertex/Show Edge	When enabled, displays indicators of vertices or edges for a visual reference about the Soft Selection falloff. The image below shows **Show Vertex** enabled: 

Perspective
Advanced OpenGL
Ray GL: OFF
Viewport Textures
151 Polygons
Channels: 0
Deformers: OFF
GL: 3200
100 mm
Action Center Auto : Soft Selection Falloff [Snap : OFF]

Linear Align

The **Linear Align** tool aligns selected vertices, edges, polygons, or UV positions along a line. Modo finds the outer-most vertices from selected edges or groups of edges and then interpolates other points between the end vertices. In **Edges** selection mode, the tool connects the selected edges and then aligns the connecting edge groups.

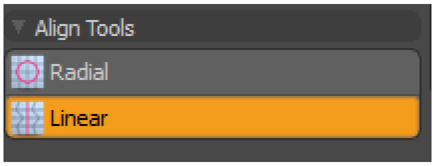

First Selected Edge
All selected edges are aligned to the first selected

You can align multiple edge groups at once, but edge groups cannot intersect or cross over each other. In **Vertices** mode, Modo tries to connect selected vertices along edges. If the selected vertices are not continuous, Modo uses the vertex selection order. In **Polygons** mode, Modo uses the boundary edges and connects them as it does in **Edges** mode. The tool always aligns middle positions between corners of the overall boundary edges.

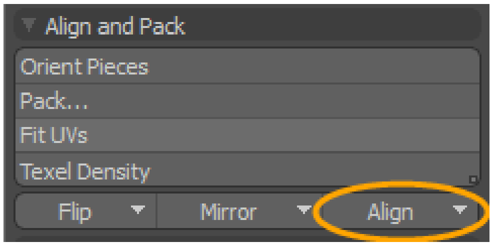
Activating the Linear Align Tool

Direct Modeling

- In the **Model** layout, open the **Deform** tab on the left panel, expand **Align Tools**, and click **Linear**.


Align Tools
Radial
Linear

- In the **UV** layout, on the left panel, under **Align and Pack**, click the down arrow for **Align** options and click **Linear**.

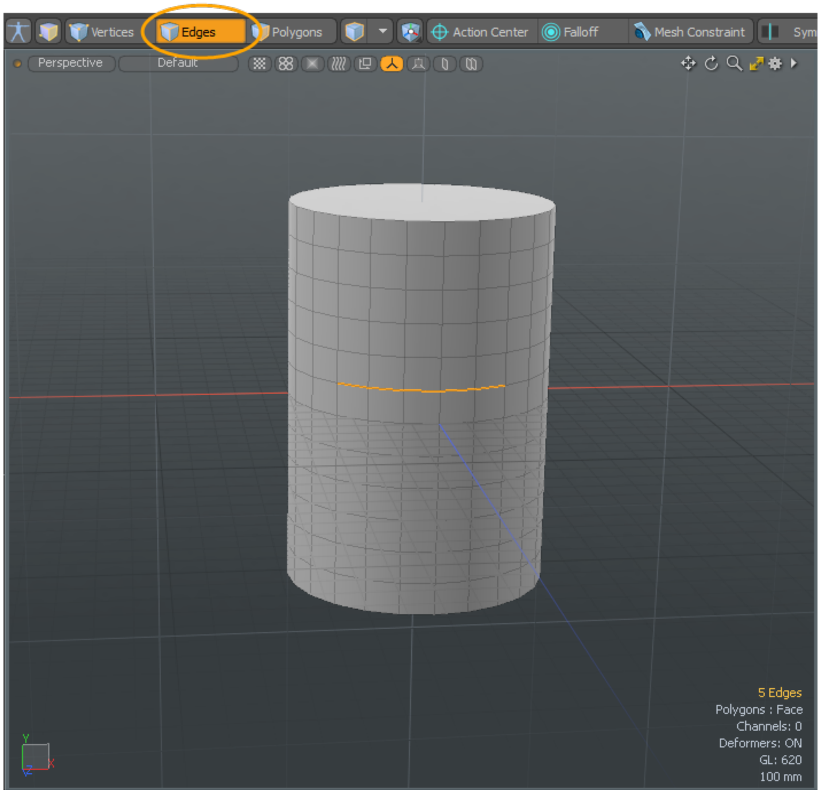

Align and Pack
Orient Pieces
Pack...
Fit UVs
Texel Density
Flip
Mirror
Align

Applying Linear Align

In this example, add a cylinder to your scene, create a selection of misaligned edges, and then apply the Linear Align tool. By enabling the **Screen Align** option, the Linear Align tool straightens selected edges in a line and keeps the depth position. Once done, click in the 3D viewport to apply the tool.

1. In the **Model** layout, open the **Basic** tab in the left panel, and **Ctrl/Cmd** + click **Cylinder**.

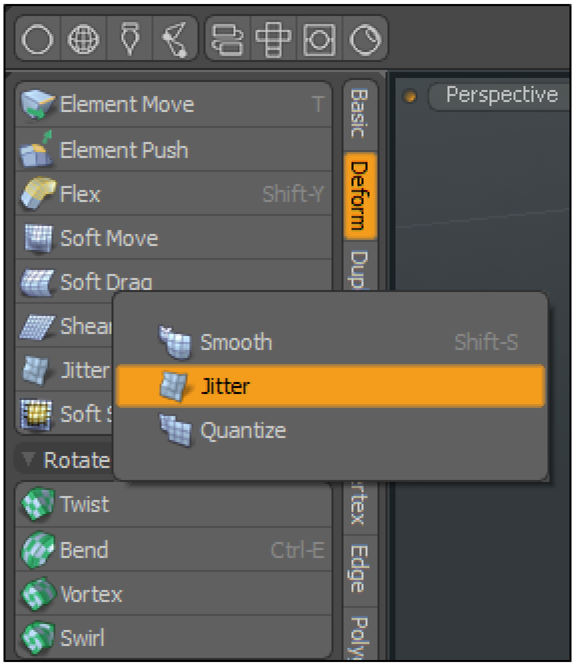
   The cylinder primitive is placed at the center origin of your scene.
2. Under the layout menu bar, click **Edges** to active the edges selection mode.
3. Press **Shift** + click on a number of consecutive edges.


Vertices
Edges
Polygons
Action Center
Falloff
Mesh Constraint
Sym
Perspective
Default

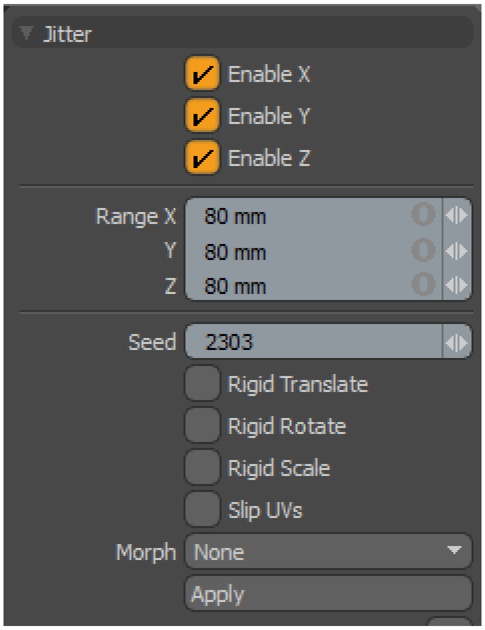
5 Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 620
100 mm

4. On the left panel, open the **Deform** tab, and click **Jitter**.

Tip: You can find the Jitter tool in the **Deform** sub-tab of the modeling toolbox (sometimes as a sub-option of the Smooth or Quantize tools).


Element Move T
Element Push
Flex Shift-Y
Soft Move
Soft Drag
Shear
Jitter
Soft
Rotate
Twist
Bend Ctrl-E
Vortex
Swirl
Smooth Shift-S
Jitter
Quantize
Basic
Deform
Dup
rtex
Edge
Poly
Perspective

5. Set the **Range** for **X**, **Y**, and **Z** to **80 mm**, the **Seed** value to **2303**, and click **Apply**.


Jitter

☑ Enable X

☑ Enable Y

☑ Enable Z

Range X 80 mm

Y 80 mm

Z 80 mm

Seed 2303

☐ Rigid Translate

☐ Rigid Rotate

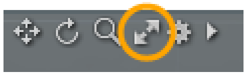
☐ Rigid Scale

☐ Slip UVs

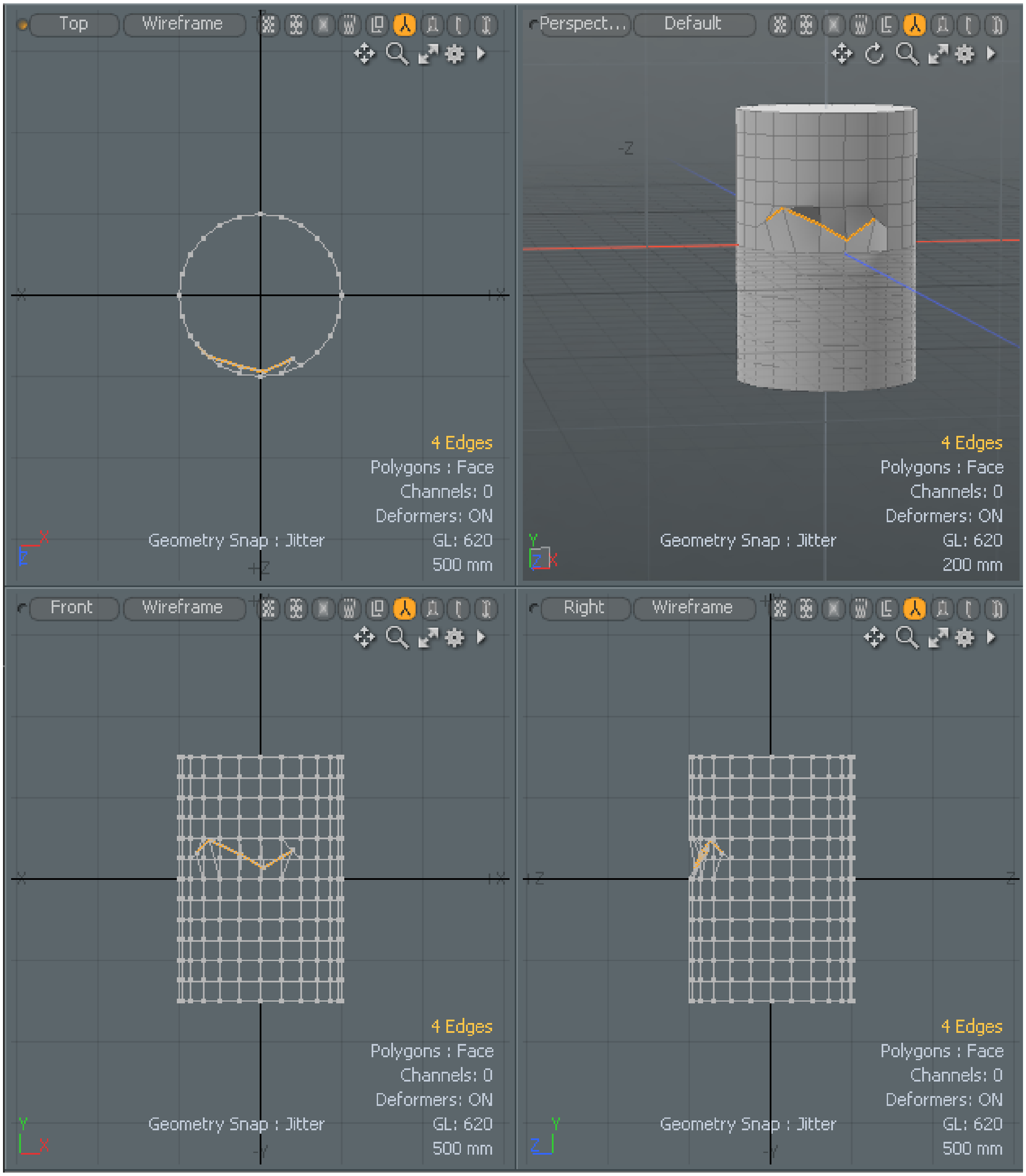
Morph None

Apply

6. In the 3D viewport, click the **Minimize/Maximize** arrow to view the mesh item in all four views (Top/Bottom, Perspective, Front/Back, and Right/Left views).

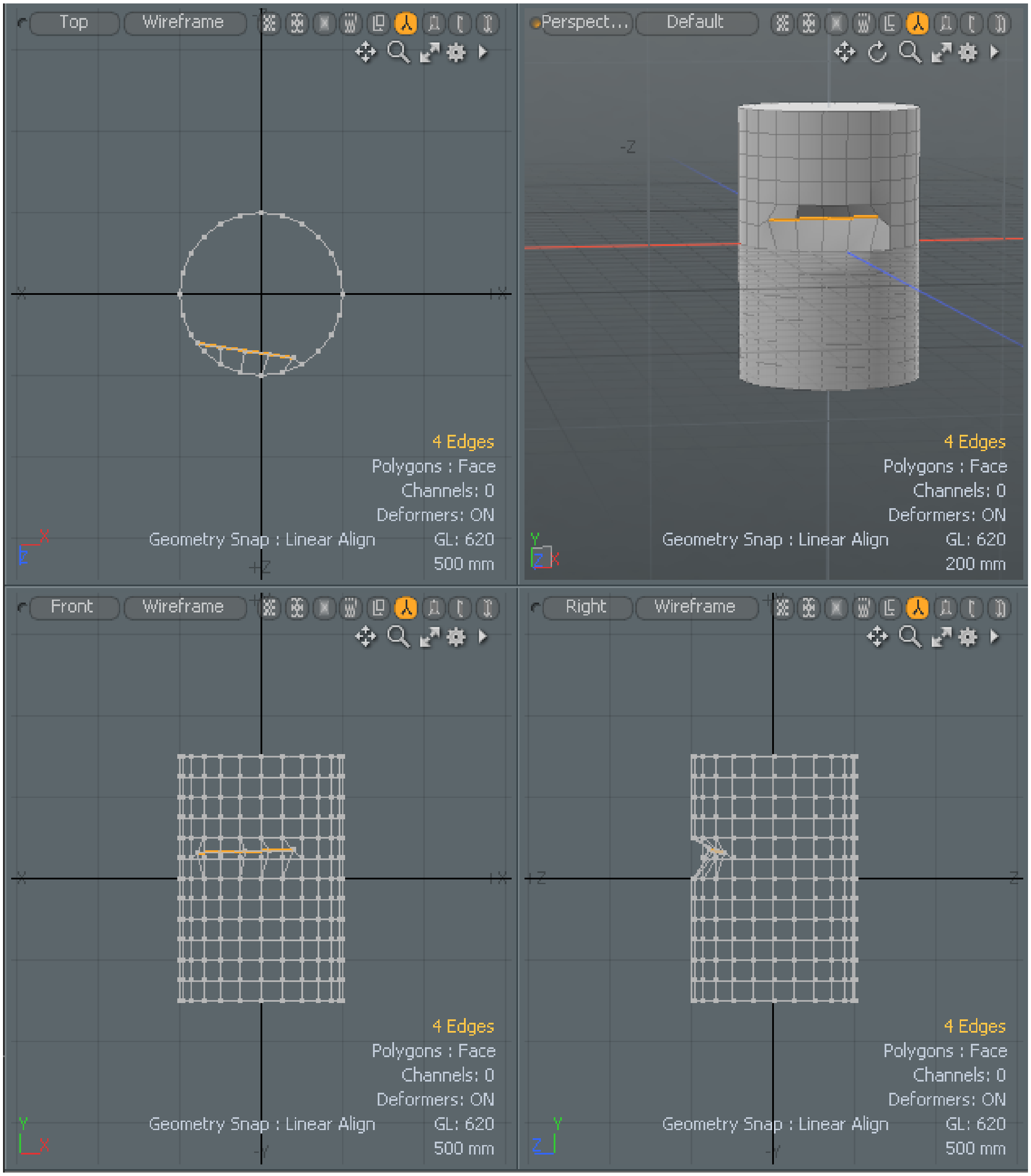


The selected edges are misaligned edges.


Top
Wireframe
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Jitter
GL: 620
500 mm
Perspect...
Default
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Jitter
GL: 620
200 mm
Front
Wireframe
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Jitter
GL: 620
500 mm
Right
Wireframe
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Jitter
GL: 620
500 mm

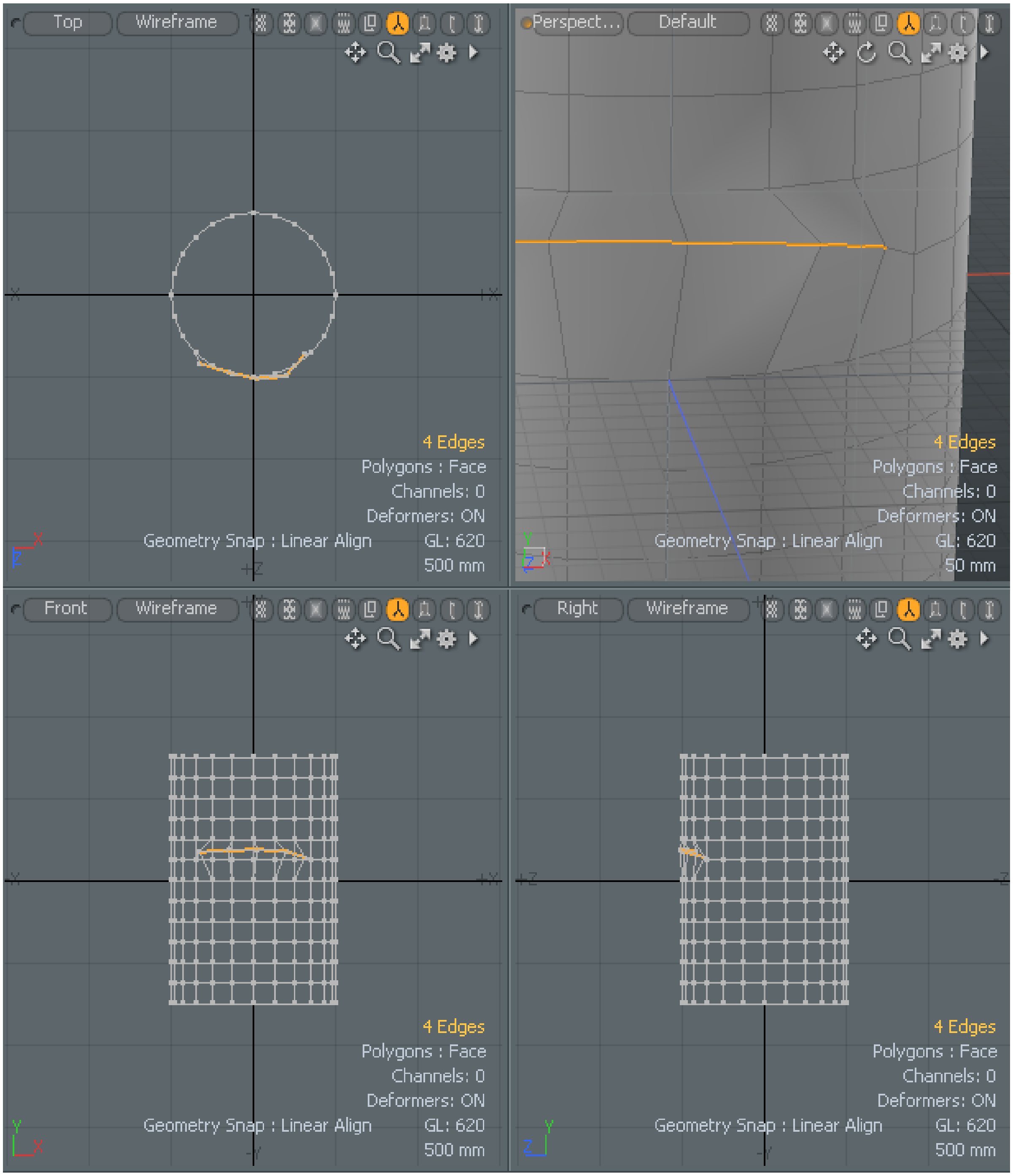
7. On the **Deform** tab on the left panel, expand **Align Tools**, and click **Linear**.
8. Click anywhere in the **Perspective** view or another view to apply the default settings.

   The misaligned edges are now aligned to the first selected edge.


Top
Wireframe
Perspect...
Default
Front
Wireframe
Right
Wireframe
-Z
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Linear Align
GL: 620
500 mm
200 mm

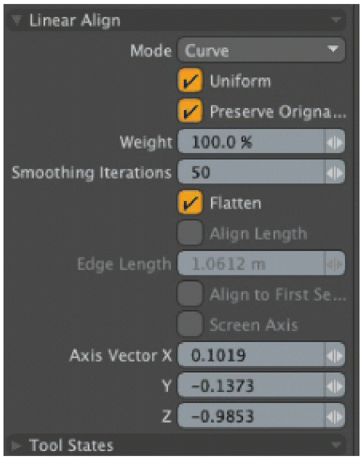
9. In the **Linear** properties on the left panel, enable **Screen Axis**.

The Linear Align tool straightens selected edges on screen view and keeps the depth position along screen space.


Top
Wireframe
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Linear Align
GL: 620
500 mm
Perspect...
Default
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Linear Align
GL: 620
50 mm
Front
Wireframe
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Linear Align
GL: 620
500 mm
Right
Wireframe
4 Edges
Polygons : Face
Channels: 0
Deformers: ON
Geometry Snap : Linear Align
GL: 620
500 mm

10. Change any of the other Linear Align Properties to tweak your results.

Linear Align Properties



Linear Align

Mode: Curve

☑ Uniform

☑ Preserve Origna...

Weight: 100.0 %

Smoothing Iterations: 50

☑ Flatten

☐ Align Length

Edge Length: 1.0612 m

☐ Align to First Se...

☐ Screen Axis

Axis Vector X: 0.1019

Y: -0.1373

Z: -0.9853

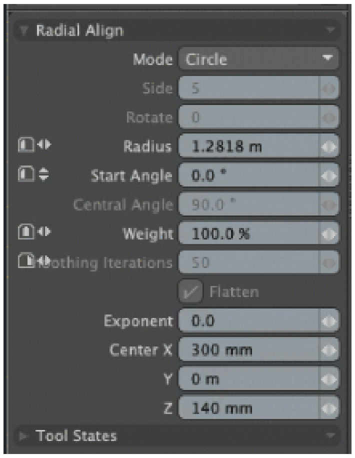
Tool States

Mode	Determines how the target selections are aligned. • **Line** - interpolates linearly between the end positions. • **Curve** - tries to fit a curve to the selected edges.
Uniform	When enabled, aligns points with the same spans; when disabled, Modo aligns the positions to the closest position along the line.
Preserve Original Curve	In Curve mode, this option fits the aligned curve points to the original curve shape as closely as possible.
Weight	Determines how to interpolate positions between the source (non-aligned) positions and the aligned positions. You can adjust this value interactively by dragging directly in the viewport. Additionally, the value for each position can be modulated by a falloff.
Smoothing Iterations	Specifies how many times Modo applies the smoothing operation. The **Linear Align** tool relaxes points inside of boundary edges only with selected polygons (in **Polygons** selection mode).
Flatten	Moves inner points to the circle plane before applying the smoothing process. This works only with selected closed polygons (in **Polygons** selection mode).
Align Length	Scales the edges or continuous selected open edges to the given length. This only works with selected open edges in **Edges** selection mode. The center of the scale is the averaged position from selected edges. The **Linear Align** tool uses the last selected edge as the initial length to re-size all others.

Edge Length	Determines the actual length of the resulting aligned segments when **Align Length** is enabled.
Align to First Selected	Aligns to the first selected vertex, edge, or polygon. The depth position of the selected edges along screen space is maintained. **Note:** When **Screen Align** is enabled this option is disabled automatically.
Screen Align	Straightens selected vertices, edges, or polygons on the screen view and keeps the depth position along screen space. The screen axis vector is stored in Axis Vector X, Y, and Z, which is set by clicking in the 3D viewport.
Axis Vector X, Y, and Z	Specifies the **X**, **Y**, and **Z** axis values to align the selected vertices, edges, or polygons.

Radial Align

Found in the **Deform** sub-tab of Modo's default toolbox, the **Radial Align** tool moves selected vertex or UV positions along a circle or polygon. Modo computes the center of the selected elements and the radius when you activate the tool based on the location of the first click in the viewport. You can interactively adjust the center of circle, the radius, and the angle offset after activation by adjusting the tool handles or by dragging in the viewport. In **Edges** selection mode, Modo connects selected edges and then aligns points in the connecting edge loop. In **Vertices** mode, Modo tries to connect selected vertices along edges. If the selected vertices are not continuous, then Modo uses the vertex selection order instead. In **Polygons** mode, Modo uses boundary edges and connects them like in **Edges** mode.



Radial Align
Mode Circle
Side 5
Rotate 0
Radius 1.2818 m
Start Angle 0.0 °
Central Angle 90.0 °
Weight 100.0 %
Smoothing Iterations 50
☑ Flatten
Exponent 0.0
Center X 300 mm
Y 0 m
Z 140 mm
Tool States

The following **Radial Align** options are available for **Radial Align**:

Radial Align	
Mode	Controls the alignment procedure. • **Circle** - adjusts the selected positions to align along a circle specified by the **Radius** value and the calculated center position. • **N-Sided** - aligns positions along a polygon specified by the **Side** and **Rotate** values. If the number of selected positions is less than the number of sides, Modo interpolates the remaining points between the corner points of the polygon.
Side	Defines the number of sides in a polygon when you select **N-Sided**.
Rotate	Offsets the start position by the defined number of degrees when you select **N-Sided**.
Radius	Sets the radius of the circle. Modo computes the initial radius of the selected elements when you activate the tool. You can adjust the radius interactively by using the **Radial Align** tool's handle.
Start Angle	Sets the start arch position.
Central Angle	Defines arch size by angle.
Weight	Determines how to interpolate positions between the source (non-aligned) positions and the aligned positions. You can adjust this value interactively by dragging directly in the viewport. Additionally, the value for each position can be modulated by a falloff.
Smoothing Iterations	Specifies how many times Modo applies the smoothing operation. The **Radial Align** tool relaxes points inside of boundary edges only with selected polygons (in **Polygons** selection mode).
Flatten	Moves inner points to the circle plane before applying the smoothing process. This works only with selected closed polygons (in **Polygons** selection mode).
Exponent	In **Circle** mode, modifying the Exponent value modulates the round shape. A positive value makes any round corners sharper and a negative value produces a concave shape.
Center X/Y/Z	Represents the center location. Modo automatically computes the center when you activate the tool. You can modify it interactively in the viewport by using the handle or by specifying values in the **Properties** panel.

Quick Align

With the Quick Align tool, you can align geometry interactively in the 3D viewport. The tool also allows you to flatten, move, and rotate the geometry you're aligning. Quick Align can be useful for straightening an edge by aligning it to another one.

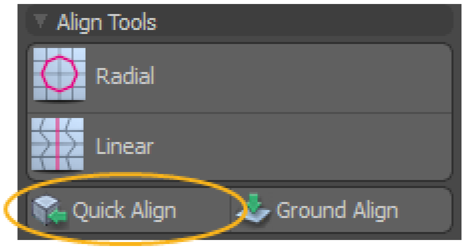
Note: Quick Align doesn't work with procedural geometry.

You can find the Quick Align tool in the **Deform** sub-tab of the Model toolbox, which is easily accessible from the left toolbar of the **Model** and **Modo** layouts.

To use the tool:

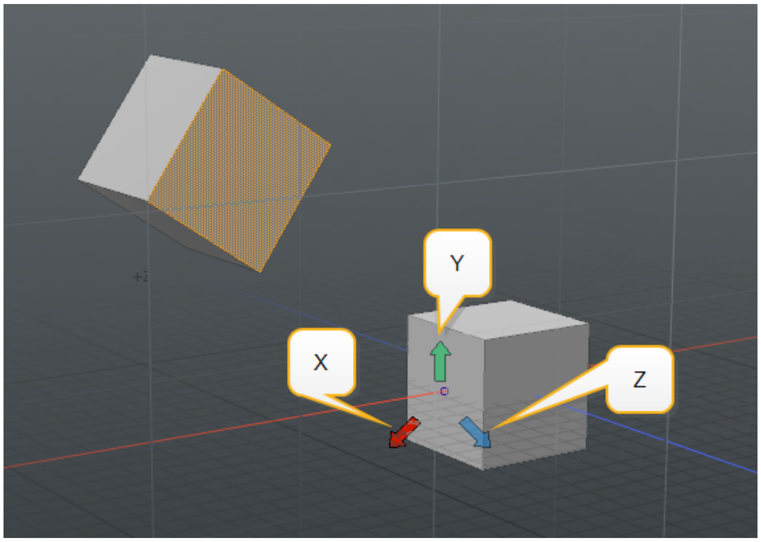
Select the component you want to align to something else. This can be a vertex, edge, or polygon.

1. In the **Modo** or **Model** layout, click the Model toolbox button in the toolbar on the left panel.
2. Go to the **Deform** sub-tab and under **Align Tools**, click **Quick Align**.



3. Click and hold the cursor on the component you want to align your geometry to.

   This displays three arrows corresponding to the three axes.



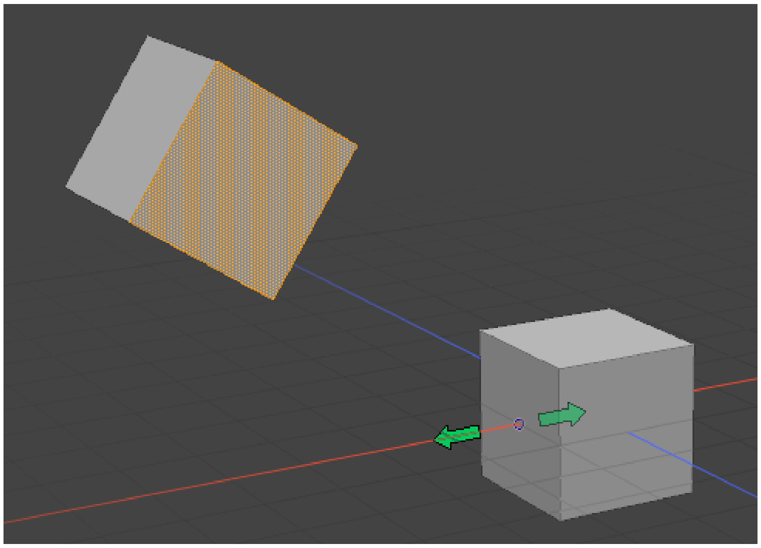
4. Still holding the cursor down, move it in the direction of the axis you want to align the geometry along. When hovering over an arrow, it becomes bigger. Release the cursor.

   Your selection is aligned to the selected geometry on the selected axis.

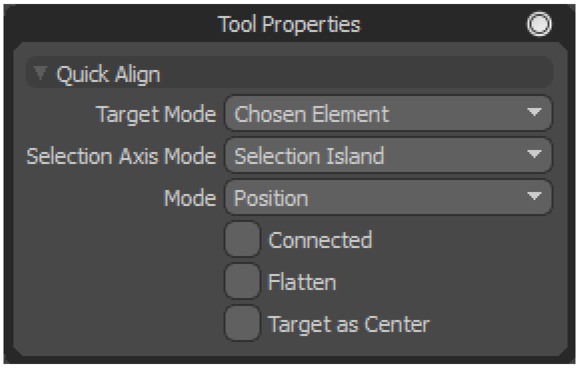
You can modify the tool's behavior using the following modifiers:

- To simultaneously flatten the geometry to the target axis, hold down the **Ctrl**/**Cmd** key, then click and hold to select the component and axis to align to. Alternatively, in the **Tool Properties**, enable **Flatten**. For more information on the properties, see Quick Align Properties.
- To move the selection and all connected geometry to the target position, hold down the **Shift** key, then click and hold to select the component and axis to align to. Alternatively, in the **Tool Properties**, enable **Connected**.
- To Align the selection to the target element's normal direction, hold down the right mouse button instead of the left. This displays two arrows corresponding to the normal directions. Move the cursor over one of the arrows and release.

  Alternatively, in the **Tool Properties**, set **Mode** to **Orientation** and left-click and hold the cursor over the target element.



Quick Align Properties



Tool Properties

Quick Align

Target Mode: Chosen Element

Selection Axis Mode: Selection Island

Mode: Position

☐ Connected

☐ Flatten

☐ Target as Center

Target Mode	Determines what your geometry is aligned to. You have the following options: • **Chosen Element** - The default behavior, aligns to an element you select in the 3D viewport. • **Local Axis** - Aligns to the center/rotation of each selection group, based on the set component selection mode. • **Origin** - Aligns to the workplane center or the world origin.
Selection Axis Mode	Specifies the selection axis. You have the following options: • **Selection Island** - Aligns each mesh selection to the selected component independently. • **Mesh Island** - Aligns the center of each mesh selection to the selected component independently.

	• **Total Selection** - Aligns the center of the entire selection to the selected component.
Mode	Sets the tool behavior when clicking in the viewport. The following options are available: • **Position** - Aligns the selection to the target position on the X, Y, or Z axis. • **Orientation** - Rotates the selection to match or oppose the target normal. You can also right-click and hold the cursor over the target element in the 3D viewport to do this.
Connected	When enabled, the tool affects any connected geometry. When disabled, only the selection is affected. You can also hold the **Shift** key while selecting the target element to move all connected geometry.
Flatten	When enabled, flattens the geometry to the target axis/normal. You can also hold the **Ctrl**/**Cmd** key while selecting the target element to flatten the geometry.
Target as Center	Use the target element as the rotation center when rotating the selected geometry.

Ground Align

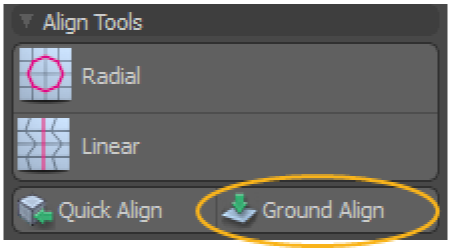
The Ground Align tool aligns any geometry in a selected layer to the Work Plane. You can drop items to the ground, or select a vertex, edge, or polygon, and rotate the item so that the selection is aligned to the ground. You can also drop items to the origin.

Note: Ground Align doesn't work with procedural geometry.

You can find the Ground Align tool in the **Deform** sub-tab of the Model toolbox, which is easily accessible from the left toolbar of the **Model** and **Modo** layouts.

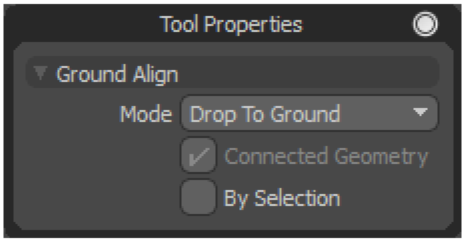
To use the tool:

1. In the 3D viewport or the **Items** list, select the geometry to be aligned.
2. Make sure that the selection mode is set to either **Vertices**, **Edges**, or **Polygons**, otherwise the Ground Align tool is inactive.
3. In the **Modo** or **Model** layout, click the Model toolbox button in the toolbar on the left panel.
4. Go to the **Deform** sub-tab and under **Align Tools**, click **Ground Align**.

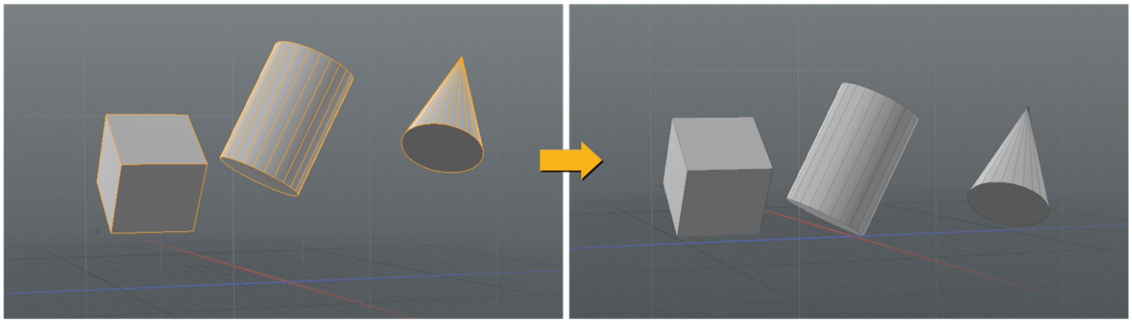

Align Tools
Radial
Linear
Quick Align
Ground Align

This activates the tool and opens the **Tool Properties**.

Note: If you are working in the **Modo** layout, this adds a **Tool Properties** button. Click it, or press the **K** key to open the properties.

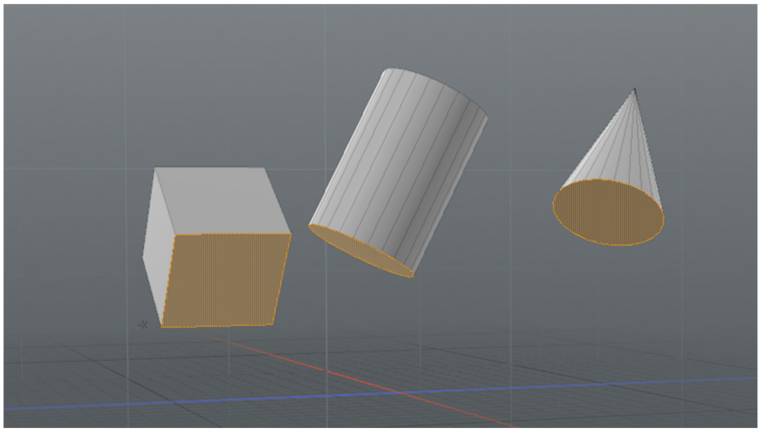

Tool Properties
Ground Align
Mode Drop To Ground
☑ Connected Geometry
☐ By Selection

5. **Mode** is set to **Drop To Ground** by default. Click in the viewport to align the geometry to the ground without rotating it.



To rotate the geometry before dropping:

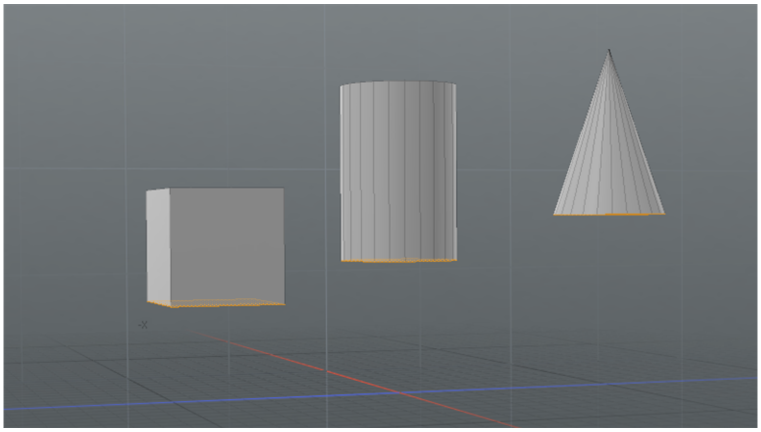
1. Select the element on the geometry you want to align to the ground. This can be a polygon, an edge, or a vertex.



2. Click **Ground Align** to activate the tool.
3. Right-click in the 3D viewport.

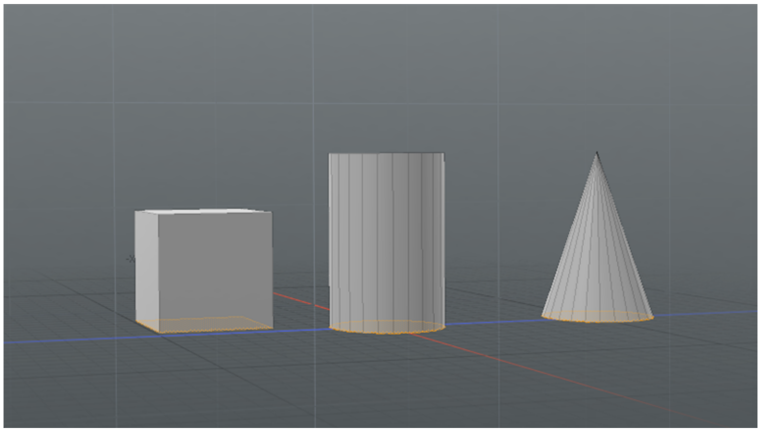
   Alternatively, in the Ground Align **Tool Properties**, set **Mode** to **Rotate to Ground**, then left-click in the viewport.

   This rotates the geometry so that the selected element faces the ground.



4. Click to drop the geometry to the ground.

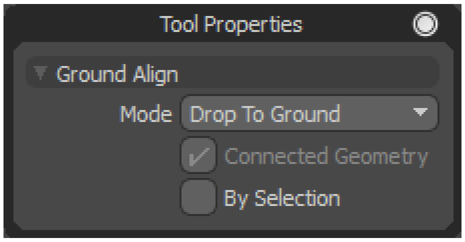
   Alternatively, in the **Tool Properties**, set **Mode** to **Drop To Ground**, then left-click in the viewport.



Hold **Shift** while clicking in the viewport to ignore any connected geometry and only drop or rotate the currently selected elements. Alternatively, in the **Tool Properties**, enable **By Selection**, and disable **Connected Geometry**.

To drop your selection to the origin, Hold **Ctrl**/**Cmd** and left-click in the 3D viewport. You can also set **Mode** to **Drop To Origin** in the **Tool Properties**, then left-click in the viewport.

Ground Align Properties



Mode: Drop To Grounds

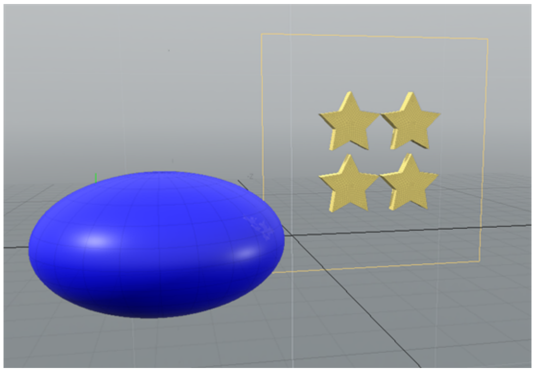
☑ Connected Geometry

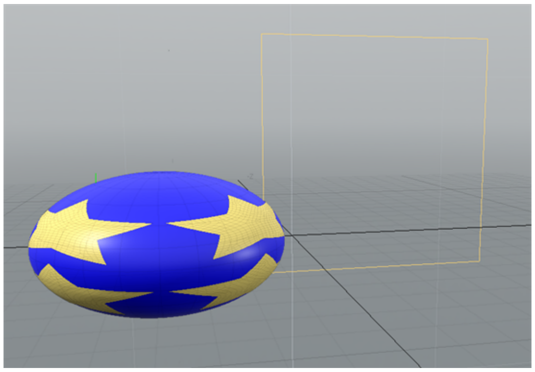
☐ By Selection

Mode	Sets the tool behavior when clicking in the viewport. The following options are available: • **Drop To Ground** - The default behavior. Drops the selected elements to 0 on the Y axis. • **Drop To Origin** - Drops the selected elements to the 0,0,0 position. • **Rotate To Ground** - Rotates the geometry so that the selected element faces the ground.
Connected Geometry	When enabled, all geometry connected to the selection is affected.
By Selection	When enabled, only the selected geometry is affected.

UV Transform

This is a procedural tool for transforming a source mesh's vertices to the world position of the surface of a target mesh, using a UV map on the target mesh. Essentially, this places the source mesh geometry on the surface of a target mesh, using the target UV map as a guide to locate vertex positions. By using this tool, you can quickly wrap a mesh over the surface of another mesh.





The tool operates in a similar way to UV mapping, but in this case the mapping region is located in 3D space, as is the source geometry. The tool also lets you use a UV map from the source mesh as a source of geometry, instead of geometry from the 3D space. This is done using the **Source UV Map** setting.

UV Transform Mapping Region

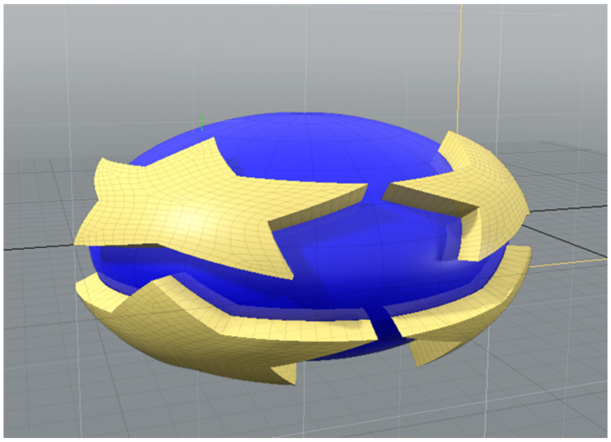
Modo uses a flat, square region in 3D space to determine the area from which to transform the source geometry. This area can be repositioned and resized as required, but is always aligned to one of the dominant planes (XY, XZ, YZ) using the **Axis** setting. You must ensure that the source geometry is placed over the region for the transform to take effect.

For example, if you want the mapping region to be on the XY plane (as shown above), set **Axis** to **Z**. Any target geometry that falls within the X, Y bounds of the mapping region will be transformed.

Translating the source geometry across the mapping region will move the transformed mesh over the surface of the target mesh, as it is effectively sliding over the surface of the target UV map.

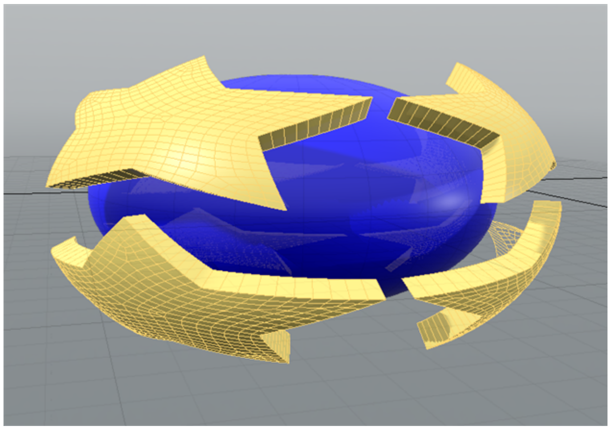
Wrap to Surface

By default, the source geometry is flattened onto the surface of the target mesh. However, by unchecking **Wrap to Surface**, you can preserve the depth of the source mesh when it is applied to the target mesh. This lets you wrap a 3D object around the surface of another mesh, as shown below.



Also, if **Wrap to Surface** is unchecked, the distance of the source mesh from the transform region plane creates a corresponding displacement of the transformed mesh from the target mesh surface. For example, if **Axis** is set to **Z**, the Z position of the source mesh determines the displacement distance.

The image below shows a transformed mesh displaced from the target mesh by increasing the **Position Z** of the source mesh. Observe also how the transformed mesh size increases as the distance increases.



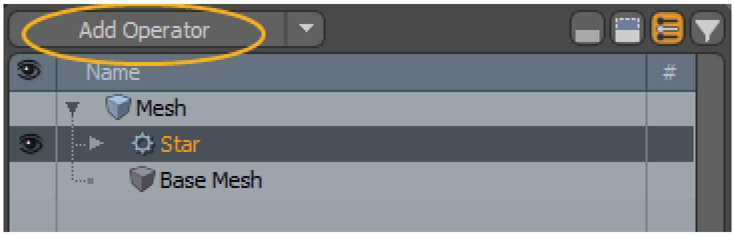
Note: The displacement from the surface does not occur when **Wrap to Surface** is unchecked. However, you can add a displacement to a flat transform mesh using the **Push** setting.

Using UV Transform

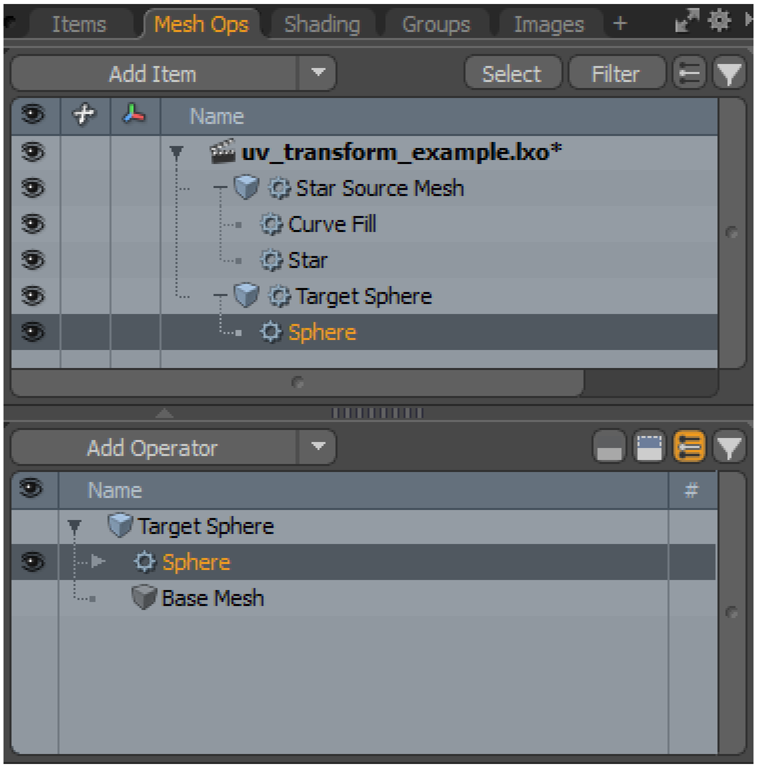
The steps below describe a typical use of the **UV Transform** tool using the **Mesh Operations**, starting with an empty scene. The example uses procedural meshes but this is not compulsory. If you already have two mesh layers containing a source mesh and a target mesh, where the target mesh has a UV map, you can skip to step 8.

Note: You can also right-click on the mesh in the 3D viewport and select **Item Operations** > **Add Mesh Operation**.

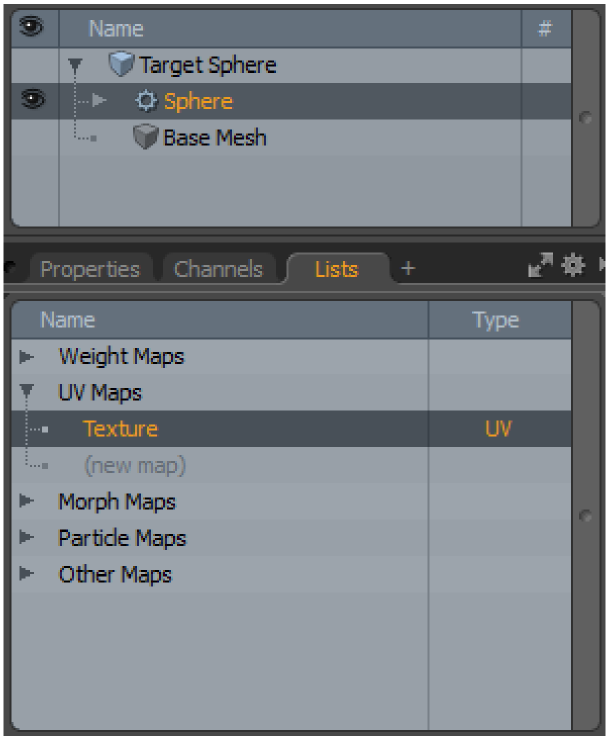
1. Add a source mesh to the scene. This will be transformed onto the target mesh. In this example, let's add a Star on the **XY** plane. In the **Mesh Ops** tab, click **Add Operator** and double-click **Mesh Operations** > **Create** > **Curves** > **Star**.



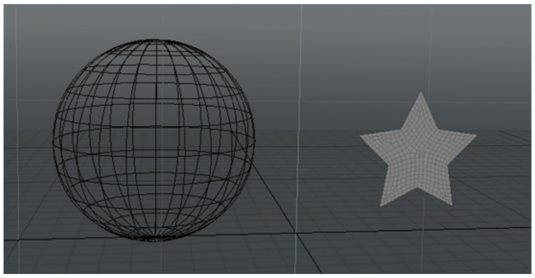
2. Let's name this mesh. In the **Mesh Ops** tab, click on the item **Mesh** and type **Star Source Mesh** in the **Properties** tab **Name** field.
3. Let's fill the star to make it solid. First select the star in your scene. Then click **Add Operator** and select **Mesh Operations** > **Curves** > **Curve Fill.**
4. Add a new mesh layer to the scene for the target mesh. To do this procedurally, click **Add Item** and select **Mesh**.
5. Let's name this mesh. In the **Properties** tab **Name** field below type the name **Target Sphere**.
6. Add a target mesh to the new layer. Let's add a procedural sphere. Click **Add Operator** and select **Mesh Operations** > **Create** > **Primitives** > **Sphere**.


Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
Name
uv_transform_example.lxo*
Star Source Mesh
Curve Fill
Star
Target Sphere
Sphere
Add Operator
Name
#
Target Sphere
Sphere
Base Mesh

7. Ensure the target mesh has a UV map (one should be created by default). Open the **Lists** tab and check the **UV Maps** list.

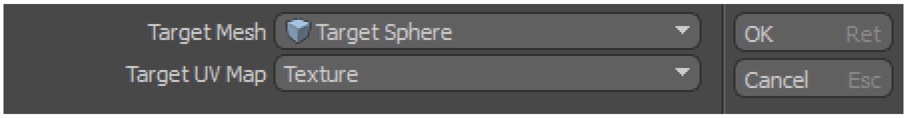

Name
#
Target Sphere
Sphere
Base Mesh
Properties
Channels
Lists
Name
Type
Weight Maps
UV Maps
Texture
UV
(new map)
Morph Maps
Particle Maps
Other Maps

8. Move the target mesh away from the origin so that you can see both meshes. You'll need to be in **Items** mode to move a procedural mesh (use the **W** key to open the move tool).



9. Select the **Star Source Mesh** item.
10. Add the **UV Transform** tool: Open the **Mesh Ops** tab, click **Add Operator** and select **Mesh Operations** > **Deform** > **UV Transform**.

    This opens the **Create UV Transform** dialog.



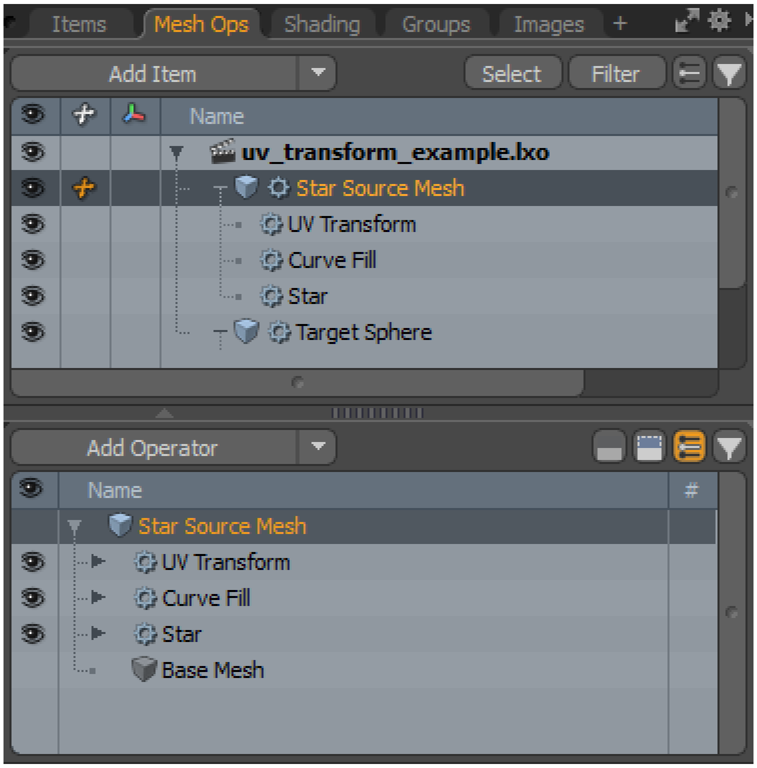
Target Mesh: Target Sphere

Target UV Map: Texture

OK Ret
Cancel Esc

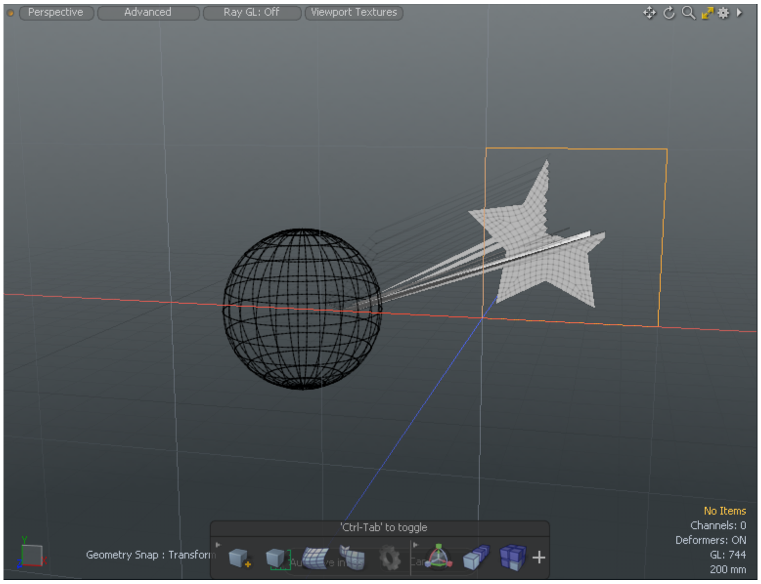
11. Set the **Target Mesh** to **Target Sphere**.
12. Set **Target UV Map** to **Texture** and click **OK**.

    The tool is added to the **Mesh Ops** list.



Items
Mesh Ops
Shading
Groups
Images
+
Add Item
Select
Filter
Name
uv_transform_example.lxo
Star Source Mesh
UV Transform
Curve Fill
Star
Target Sphere
Add Operator
Name
#
Star Source Mesh
UV Transform
Curve Fill
Star
Base Mesh

You'll see the square transform region in the 3D viewport.

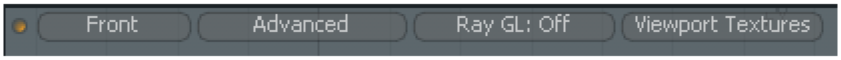

Perspective
Advanced
Ray GL: Off
Viewport Textures
'Ctrl-Tab' to toggle
Geometry Snap : Transform
No Items
Channels: 0
Deformers: ON
GL: 744
200 mm

Note: If the star overlaps the UV transform region (as shown above), only the part of the source mesh within the region is transformed. Consequently, the mesh is stretched between the original and transform locations.

13. Select the **UV Transform**, in the **Properties** tab, ensure the **Axis** is set to **Z**. In this case, this aligns the mapping region to the source mesh.
14. You can now transform the source mesh (the star) and/or the mapping region, so that the source mesh is enclosed by the mapping region.

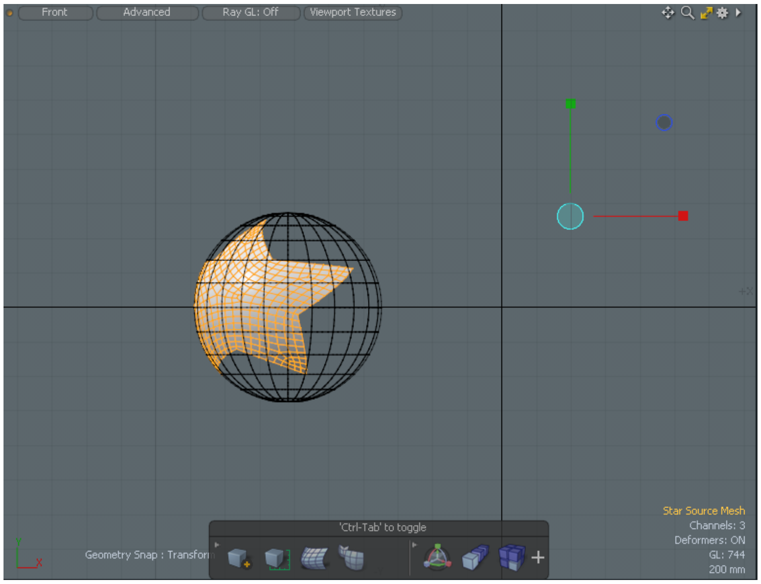
    a) In the **Items** list, select the **UV Transform** mesh operator, open the **Properties** tab on the lower right panel, and click the **Use Source Transform** checkbox.

    b) In the upper left corner of the viewport, change the **View Type** to **Front**.

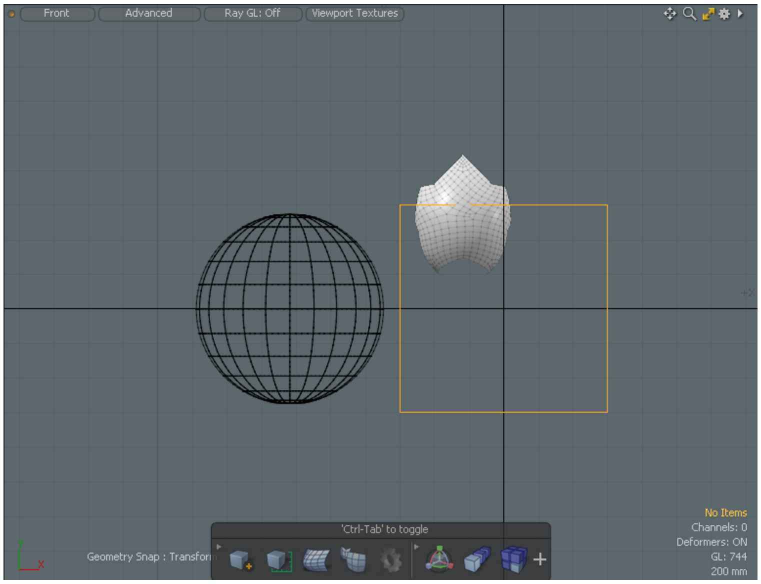

Front
Advanced
Ray GL: Off
Viewport Textures

    c) In the **Items** tab, select **Star Source Mesh**, press **W**, and use the handles to reposition the item.

    d) Press **R** to rescale the **Star Source Mesh** and use the handles to rescale the item.


Front
Advanced
Ray GL: Off
Viewport Textures
'Ctrl-Tab' to toggle
Geometry Snap : Transform
Star Source Mesh
Channels: 3
Deformers: ON
GL: 744
200 mm

Alternatively, disable the **Use Source Transform** option, adjust the **Offset** (mm) and **Scale** (%) properties respectively . For example, set **Offset U** and **V** to **0** and **Scale** to **120%**.

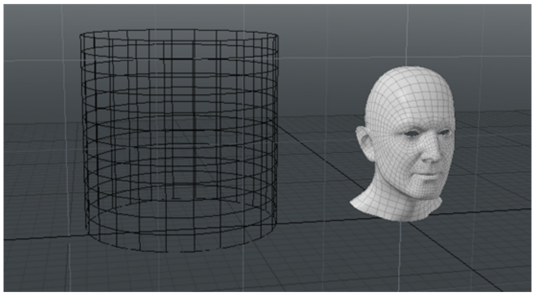

Front
Advanced
Ray GL: Off
Viewport Textures
'Ctrl-Tab' to toggle
Geometry Snap : Transform
No Items
Channels: 0
Deformers: ON
GL: 744
200 mm

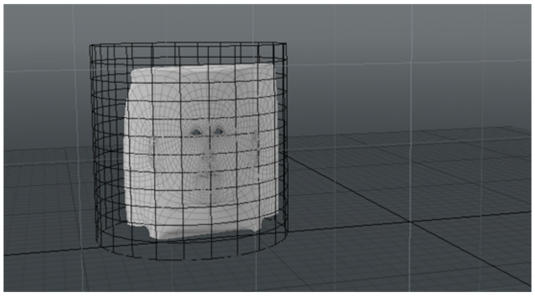
You can now see the source geometry transformed onto the surface of the target mesh. To see this more clearly, hide the target mesh to show only the transformed mesh.

Note: If you move the target mesh, the transformed mesh follows.

Using a Source UV Map

The tool provides two options for source geometry, you can either use a source mesh or a source UV map. To use a UV map, set the **Source UV Map** parameter in the tool properties. The image below shows a source mesh, and the result of transforming its UV map on to a cylinder.





UV Transform Properties

Edit the following parameters to customize the UV Transform tool. They are found in the **Properties** viewport on the **UV Transform** tab.

UV Transform	
Enable	Toggles the tool on and off.
Target Mesh	The source mesh is transformed to this mesh.
Target UV Map	Sets the UV map to use for the source transform. This is required to work out how to transform from the target to the source.
Source UV Map	Select a UV map to locate the vertices in the source mesh. By default this isn't set, and Modo uses the vertex position in the axis given by the **Axis** value to calculate the position to look up on the target UV map.
Use Source Transform	
Axis	Determines the axis that is used to calculate the target UV position. Choose one of the following: • **X** - Use YZ values. • **Y** - Use XZ values. • **Z** - Use XY values.

	Note: Axis is not used if **Source UV Map** is set.
Offset U/V	Sets the position of the center of the mapping region, relative to the origin.
Scale	Sets the scale of the mapping region. At **100%** the region is 1m x 1m.
Wrap to Surface	Check this to transform the mesh, preserving the depth of the source geometry. By default, this is off and the source geometry is flattened.
Push	The distance to offset the transformed source geometry from the surface of the target mesh.

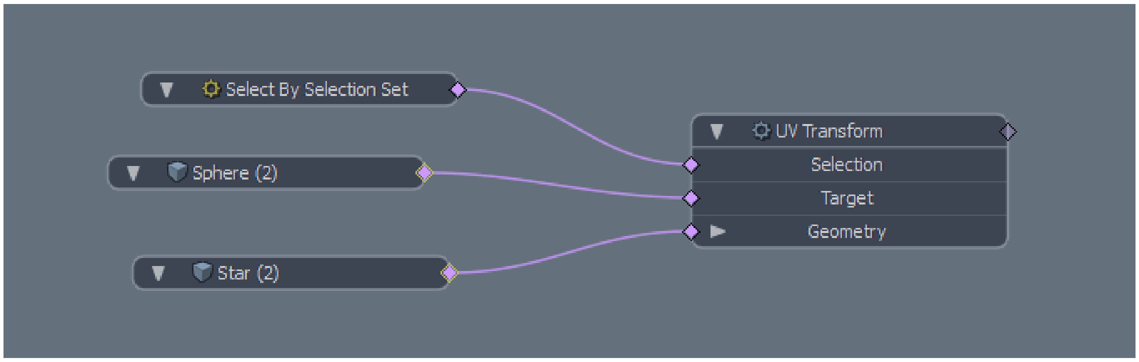
UV Transform Procedural

The UV transform has the following procedural sub-tools:

- **Selection** - Allows you to limit the transform to a selection of elements.
- **Target** - The target mesh.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

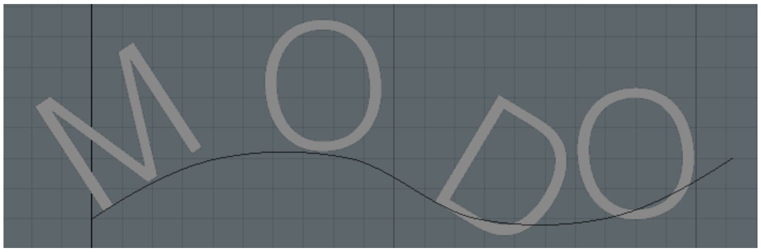
UV Transform Schematic

The UV Transform node has the same inputs as the procedural version.



Path Generator

The **Path Generator** sub-tool uses an existing curve to provide a path for other tools to walk and evaluate. This can be useful for things like extruding or transforming shapes along curves and controlling positions of elements along a curve. For example, you can use the Path Generator along with the Text tool, to place characters along a curve.



Using the Path Generator

Before using the Path Generator tool, you need an item in your scene to use it with. In this example, we use the Text primitive.

1. In the **Mesh Operations** list, click **Add Operator**.
2. Under **Procedural** > **Create** > **Primitives**, add the **Text** primitive.
3. In the **Properties** panel of the Text item, enter your text, and set the font.
4. In the list, expand the **Text** item to reveal the **Tool Pipe**, and click **(Add Tool Pipe)**.

   The procedural **Preset Browser** opens
5. Under **New**, double-click **Path Generator** to add it the Text item's tool pipe.
6. Add a new mesh layer in the **Item List** by clicking **Add Item**, and under **Locators**, select **Mesh**.
7. Draw a curve in the new mesh. The Curve tool can be found in the **Setup** layout under the **Setup** sub-tab, or in the **Model** layout's **Basic** sub-tab, by clicking and holding on the Pen tool.

> **Note:** For more information on the Curve tool, see the Adding Curves topic.

8. In the **Mesh Operations** list, expand the Path Generator item, and under **Curve**, click **(Add Curve)**.

   The procedural **Preset Browser** opens.
9. In the **Preset Browser**, under **Existing**, double-click the mesh layer that contains your Curve.

   Modo places your text on the curve.

You can adjust how the text is drawn along the curve in the Text item's **Properties** panel.

Note: For more information on text properties, see Text.

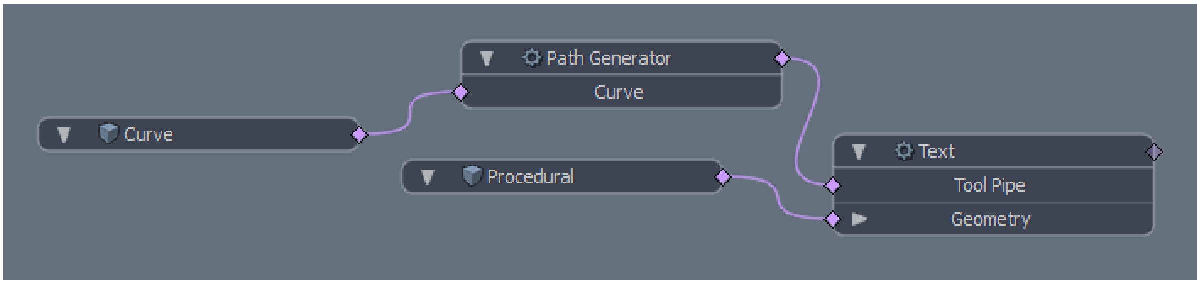
Path Generator Properties

The settings for the Path Generator are the following:

Path Generator	
Curve Index	Specifies the index of the curve to be used as path. If you have multiple curves in your scene, you can choose the one you need by entering its index.
Start	Specifies the beginning of the curve. Any section between the real start of the curve and this value is ignored.
End	Specifies the end of the curve. Any section between the real end of the curve and this value is ignored.
Curve Slide	Determines where the text should start on the curve. Use this setting to slide the text along the curve.

Path Generator Schematic

The Path Generator sub-tool has a single graph input, allowing any item that presents a curve group, such as a mesh layer, to be connected. The curve is evaluated and interpreted as a Path Generator for use as part of the procedural tool system.

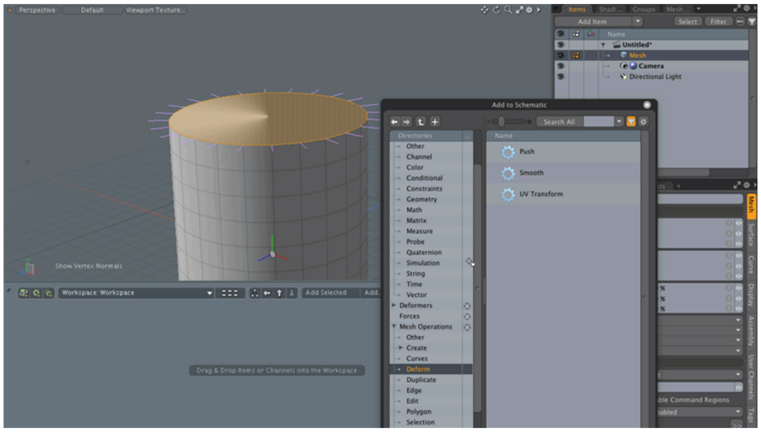


Scale Deform Tools

Learn about Modo's scale deform tools.

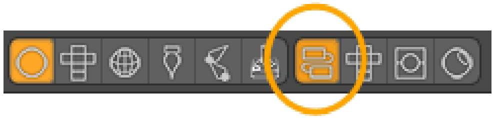
Push

The Push tool is a deformation tool that moves the selected geometry along the selected item's normals.


Perspective
Default
Viewport Texture...
Items
Shadi...
Groups
Mesh ...
Add Item
Select
Filter
Name
Untitled*
Mesh
Camera
Directional Light
Add to Schematic
Search All
Directories
Other
Channel
Color
Conditional
Constraints
Geometry
Math
Matrix
Measure
Probe
Quaternion
Simulation
String
Time
Vector
Deformers
Forces
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Edit
Polygon
Selection
Name
Push
Smooth
UV Transform
Show Vertex Normals
Workspace: Workspace
Add Selected
Add...
Drag & Drop Items or Channels into the Workspace
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

Accessing the Push Tool

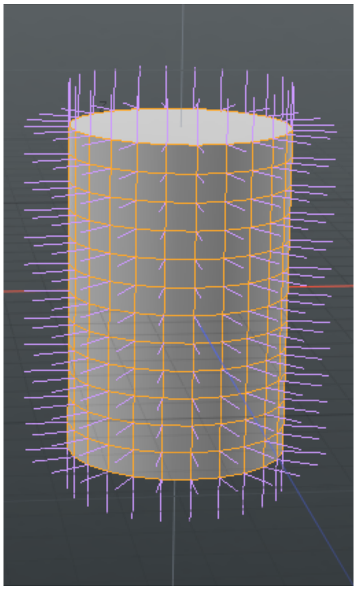
- On the left panel, open the **Deform** tab, and click **Push**.
- Alternatively, on the menu bar, click **Edit > Deform > Push**.
- Alternatively, on the right panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Deform** > **Push**.
- Alternatively, on the top left corner of the interface, click the **Schematic** palette icon , click **Add...**, and **Mesh Operations** > **Deform** > **Push**.



Applying the Procedural Push Tool

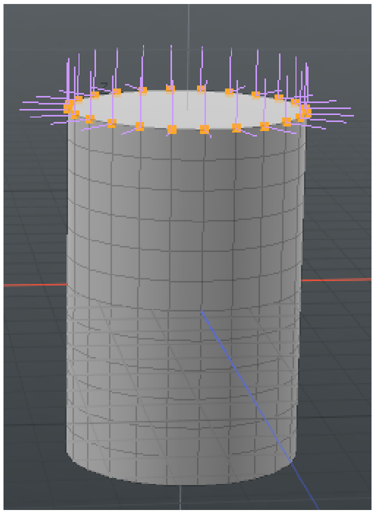
The following is an example of applying the procedural version of the **Push** tool using a vertices selection set. You can use any of the Vertices, Edges, or Polygons selection modes and apply the Push tool.

1. On the left panel, on the **Basic** tab, press **Ctrl**/**Cmd** + click on the cylinder icon.
2. On the menu bar click **Vertex Map** > **Create**, set **Vertex Map Type** to **Vertex Normal Map**, and click **OK**.
3. On the menu bar click **View** > **Show Vertex Normals**.

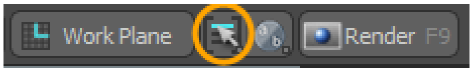


4. Under the layout menu bar, click **Edges**, and select the vertices on the top of the cylinder.

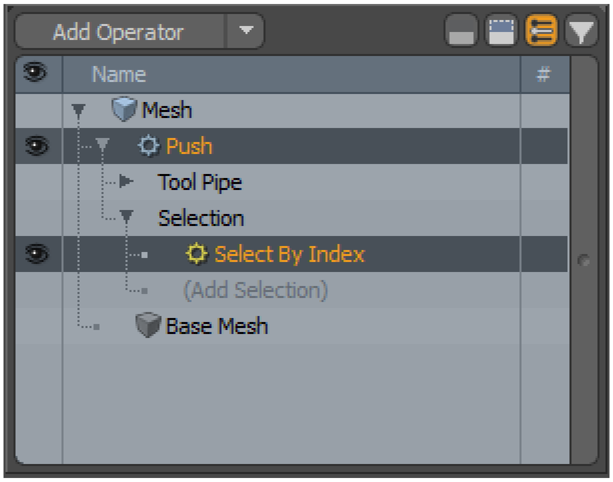
Tip: Shift + click on two vertices and press **L** to select the entire loop.



5. Under the layout menu bar, click the selection sets icon, and click **New set**.

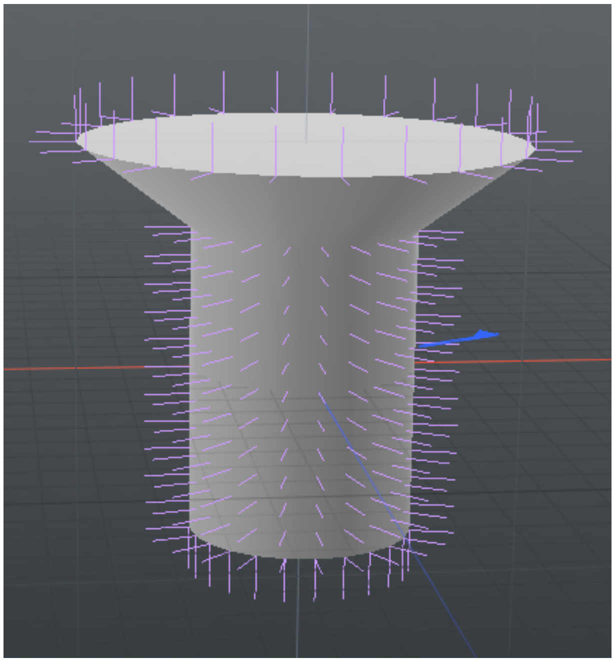

Work Plane
Render F9

6. In the **Create Selection Set** dialog, type **Top** in the **Name** field.
7. On the right panel, open the **Mesh Ops** tab, click **Add Operator** and double-click **Mesh Operations** > **Deform** > **Push.**
8. In the **Mesh Operations** list, expand **Push** > **Selection**, click **(Add Selection)** and double-click **Mesh Operations** > **Selection** > **Select By Selection Set**.

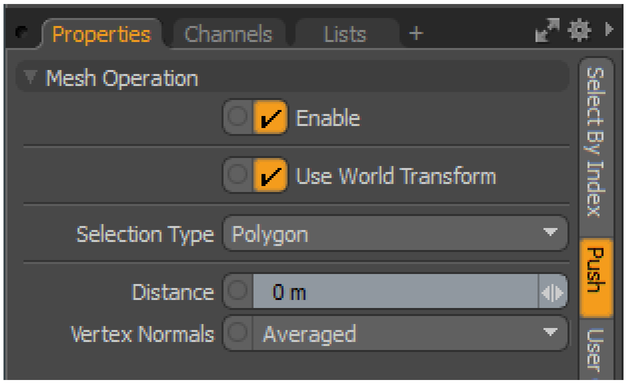

Add Operator
Name
#
Mesh
Push
Tool Pipe
Selection
Select By Index
(Add Selection)
Base Mesh

9. In the **Properties** tab, on the **Select By Selection Set** tab, and set **Name** to **Top**.
10. In the **Mesh Operations** list, select **Push** and click + drag the blue handle in the 3D viewport to apply the **Push** tool.

    The geometry moves along the selected item's normals



Push Tool Properties



☑ Enable

☑ Use World Transform

Selection Type Polygon

Distance 0 m

Vertex Normals Averaged

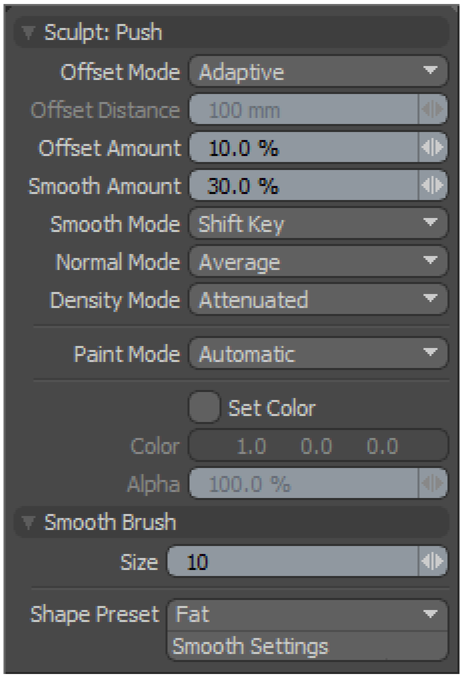
- **Enable** - Enables or disabled the Push tool.
- **Use World Transform** - Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
- **Distance** - Distance to push along vertex normal.
- **Selection Normals** - Specifies the method to compute vertex normals. The following options are available:
  - **Averaged** - Computes the vertex normal with all connecting polygons.
  - **Selected** - Computes the vertex normal from selected connecting polygons.
  - **Use Vertex Normal Map** - Uses the vectors from vertex normal map if the mesh has vertex normal map.

Sculpt

Before Modo had sculpting functions it had the Sculpt tool. This is a simple and direct way to shape polygons with a brush-like paradigm to smoothly deform a surface. When Modo introduced sculpting, the sculpting functionality made the older Sculpt tool unnecessary.

The Sculpt tool remains for your convenience. Using the superior sculpting functionality provides you with a more powerful way modify a mesh, but the current Sculpt tool still retains the intuitive simplicity of the older tool. To apply the tool, click the tool's icon and view the options in the tool's **Properties** panel. With the tool active, you can drag over the model's surface in the 3D viewport to deform the model outward along the polygons' normal. Pressing **Ctrl** pushes the surface inward along the normal. You may customize the effect of the brush on the surface by adjusting the options. For more on sculpting, see the Paint and Sculpting topic.

Tip: When a sculpt-based tool is activated, a circle is drawn under the mouse when you hover over a surface. This makes it easier to see where the sculpt is applied when you left-click and drag across the surface of your mesh. To change the size of the sculpt tip, right-click and drag your mouse.

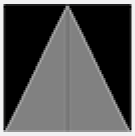
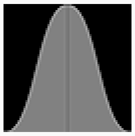
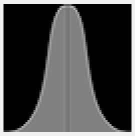
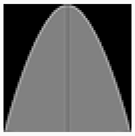


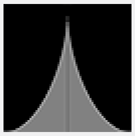
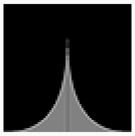
Sculpt: Push
Offset Mode: Adaptive
Offset Distance: 100 mm
Offset Amount: 10.0 %
Smooth Amount: 30.0 %
Smooth Mode: Shift Key
Normal Mode: Average
Density Mode: Attenuated
Paint Mode: Automatic
☐ Set Color
Color: 1.0 0.0 0.0
Alpha: 100.0 %
Smooth Brush
Size: 10
Shape Preset: Fat
Smooth Settings

Sculpt: Push	
Offset Mode	Provides two methods for Modo to calculate the way the tool moves the surface when brushing. • **Adaptive**: Adapts to the surface that is being brushed on based on the the **Offset Amount** value and the size of the brush. • **Absolute**: Uses the **Offset Distance** value for the brush regardless of the brush scale.
Offset Distance	Determines specifically how far to displace the surface. This is disabled unless **Offset Mode** is set to **Absolute**.
Offset Amount	Modulates the extent of the brushstroke. This is active only when **Offset Mode** is set to **Adaptive**. The total scale of the sculpt deformation is equal to the brush size times the offset amount percentage. **Note:** This is a very important value for controlling sculpted detail and, as such, it can be adjusted interactively by middle-clicking and dragging left or right.
Smooth Amount	Controls the amount of smoothing applied when pressing the **Shift** key with certain tools. The Smooth tool is available with almost every sculpting tool in Modo and can be accessed by pressing the **Shift** key while sculpting. This is a feature implemented specifically to enhance the sculpting workflow.
Smooth Mode	Provides alternate methods for applying smoothing while sculpting. • **Shift Key** - Enables Smooth mode when pressing the **Shift** key. • **Low Pressure** - Adjusts the smoothing based on the stylus' pressure when using a pressure-sensitive tablet. With this method, light sculpting strokes have a smoothing affect on the surface. Once you exceed the pressure threshold, the smoothing effect attenuates toward the full pressure effect (how the brush works normally as determined by the tool). This option provides a dual function to the tool. • **Uniform** - Applies smoothing at all times while sculpting and, therefore, makes softer strokes.
Normal Mode	Controls how Modo displaces the sculpted vertices. • **Brush Center** - Uses the geometry at the very center of the brush to calculate

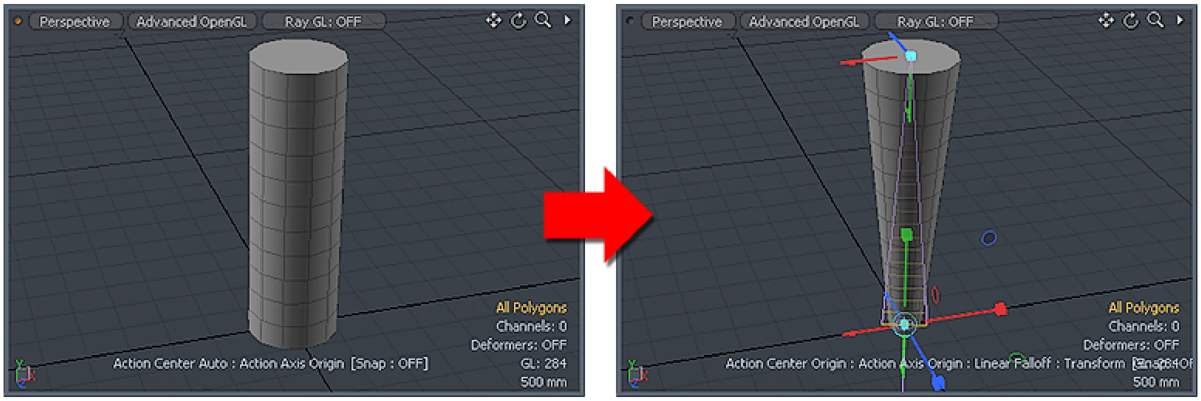
Sculpt: Push	
	the direction along which to displace the mesh. • **Vertex** - Displaces each vertex along its own normal. • **Average** - Calculates an average normal for all the vertices inside the brush. • **Screen** - Displaces the vertices directly towards the active viewport. • **Mouse Down** - Uses the normal of the mesh surface directly under the pointer when clicked. Modo then uses that vector for all displacements. • **X,Y,Z** - Specifies a world axis along which to displace the vertices.
Density Mode	Determines how the sculpt stroke affects areas of the mesh that it has already covered. For instance, brushing back and forth has the Sculpt tool re-apply itself. This option controls the sculpt during a single stroke. Once the stylus comes off the tablet or you release the mouse button, the density resets. • **Additive** - Continues to add to the displacement each time the brush crosses over its own path with no maximum cap to govern the sculpt. • **Attenuate** - Continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the **Offset Distance** or **Offset Amount** value. • **Max Stroke** - Enforces a maximum offset based on the largest displacement value of the current stroke. • **Max Map** - Caps the sculpt effect at the maximum amount in the current displacement image.
Paint Mode	Determines which method of sculpting Modo uses. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images and automatically switches to image-based sculpting when an image is available. There are times when you may want to force Modo to do mesh-based sculpting even if you have an image applied for displacement. This can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry. • **Automatic** - Switches between mesh- and image-based sculpting depending on availability of a displacement image map texture. • **Mesh** - Forces mesh-based sculpting only. • **Image Map** - Forces image-based sculpting only.
Set Color	When enabled with Multi-Res sculpting, you can simultaneously sculpt the mesh and paint into an image map layer with the same settings of the sculpting tool

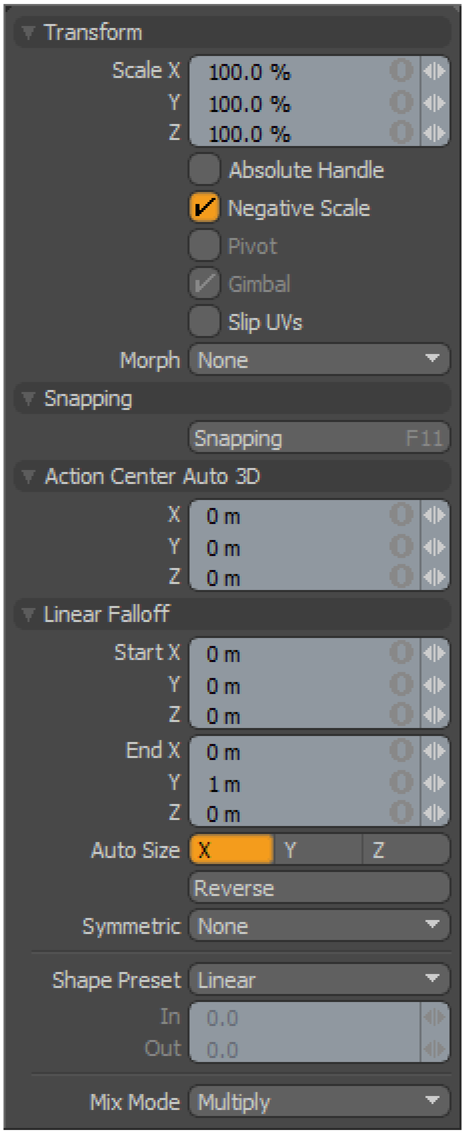
Sculpt: Push	
	itself.
Color	Determines the color applied to an image map when sculpting.
Alpha	Determines the opacity of a stroke applied to an image map.
Smooth Brush - The Smooth brush creates a feathered, airbrush effect over the surface.	
Size	Sets the radius of the brush in pixels. Zooming the view changes the result of the brush. With the brush size being fixed, zooming in on the model decreases the relative scale of the brush whereas zooming out increases it. You can interactively adjust the tip size by right-clicking and dragging in the viewport to adjust the brush's size.
Shape Preset	Includes several preset brush options for varying effects. A preset specifies the profile of the airbrush. You may also adjust the **In Slope** and **Out Slope** and the **In Weight** and **Out Weight** settings for the brush tip to further customize brushes.

	**Linear** - Creates a straight ramp from the center to the edge with a brush profile that looks like a cone.
	**Smooth** - Creates a brush stroke that is opaque in the center and falls off smoothly to the edge of the brush with an S-curve shape.
	**Bulge** - Changes the bias of the curve to start out slowly and then ramp more harshly towards the middle with a brush profile that looks similar to a dome.
	**Fat** - Starts out soft and ramps harshly at the outer edge to produce a soft U-shaped brush that varies the most under light pressure on a tablet.

	**Sharp** - Produces a fine stroke with the falloff very harsh and steep at the center.
	**Very Sharp** - Produces an even finer stroke than the Sharp preset with a more extreme falloff. This preset is nearly identical to the Sharp preset.

Taper


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 284
500 mm
Action Center Auto : Action Axis Origin [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
500 mm
Action Center Origin : Action Axis Origin : Linear Falloff : Transform



Transform

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

☐ Absolute Handle
☑ Negative Scale
☐ Pivot
☑ Gimbal
☐ Slip UVs

Morph None

Snapping

Snapping F11

Action Center Auto 3D

X 0 m
Y 0 m
Z 0 m

Linear Falloff

Start X 0 m
Y 0 m
Z 0 m
End X 0 m
Y 1 m
Z 0 m

Auto Size X Y Z

Reverse

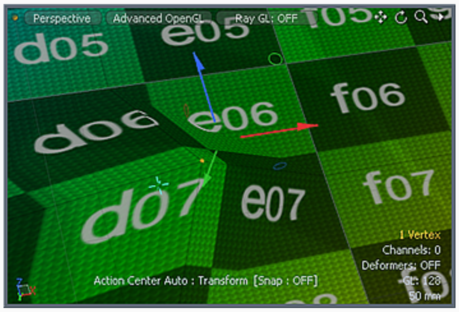
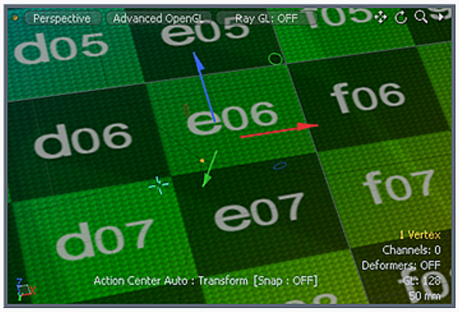
Symmetric None

Shape Preset Linear
In 0.0
Out 0.0

Mix Mode Multiply

Found in the **Deform** sub-tab of the modeling toolbox, the Taper tool combines the Transform tool with a Linear falloff. At first click, Modo sets the Linear falloff perpendicularly to the pointer and scales it automatically to match the currently selected geometry. Dragging now deforms the object with a tapering effect. To precisely place the falloff, you can either use the handles to manipulate the falloff after it has been autosized or right-drag away from the handles to reset the falloff to drag it out interactively.

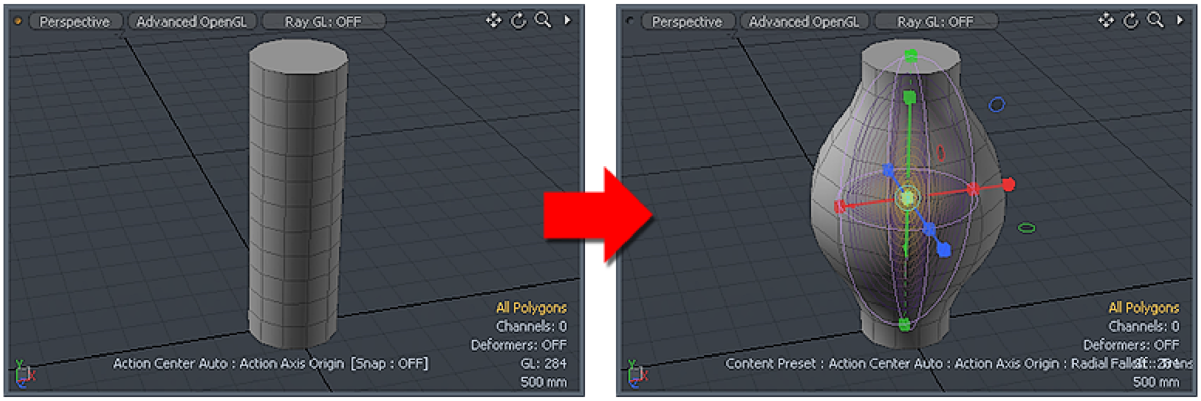
Taper	
Transform	
Scale X/Y/Z	Displays specific scaling values applied to the geometry. When adjusting an object interactively in the viewport, the **Scale** input values indicate the current scale factor applied. You may also specify the X, Y, and Z scaling values.

Taper	
Absolute Handle	When enabled, applies an additional function to the tool handles for you to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.
Negative Scale	When enabled, applies negative scale values, if appropriate, when you manipulate the tool handles in the viewport. When disabled, Modo does not automatically change the values if they are negative. You can always input negative values directly.
Pivot	Disabled for the Taper tool.
Gimbal	Disabled for the Taper tool.
Child Compensate	(Only available in **Items mode**) When enabled, ignores any transform values for child items. When disabled, passes along transform values to any child items of the current selection.
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping) >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or

Taper	
	Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Linear Falloff	
Start X/Y/Z	Defines the starting position of the falloff as a specific X,Y,Z coordinate value. This end of the line and everything beyond it receives the maximum amount of influence that attenuates toward the **End** position.
End X/Y/Z	Defines the end position of the falloff as a specific X,Y,Z coordinate value. This end of the line and everything beyond it receives no influence with the effect attenuating toward the **Start** position.
Auto Size X/Y/Z	Specifies the one axis to automatically scale the **Start** and **End** points of the falloff to match the bounding box size of the selected elements.
Reverse	Inverts the **Start** and **End** points to reverse the influence of the falloff.
Symmetric	Indicates how to automatically mirror the influence area of the falloff. • **None** - The Symmetric function is disabled. • **Start** - Mirrors the influence of the falloff symmetrically across the **Start** position. • **End** - Mirrors the influence of the falloff symmetrically across the **End**

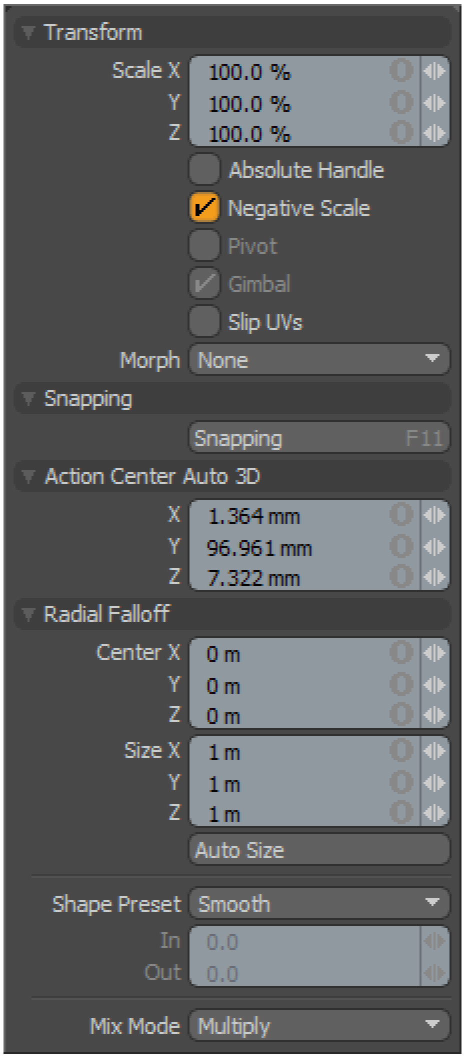
Taper	
	position.
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Bulge


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 284
500 mm
Action Center Auto : Action Axis Origin [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
500 mm
Content Preset : Action Center Auto : Action Axis Origin : Radial Falloff

The Bulge tool combines the Transform Scale tool with a Radial Falloff to produce a swelling effect on the target object or surface. You apply this tool similar to applying any normal scaling transform, but this tool smoothly fades the effects of the scaling across the object or surface.

Usage



Transform
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
☐ Absolute Handle
☑ Negative Scale
☐ Pivot
☑ Gimbal
☐ Slip UVs
Morph None

Snapping
Snapping F11

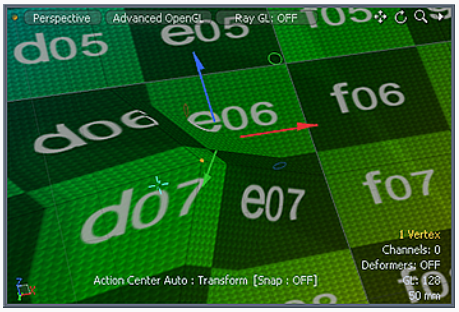
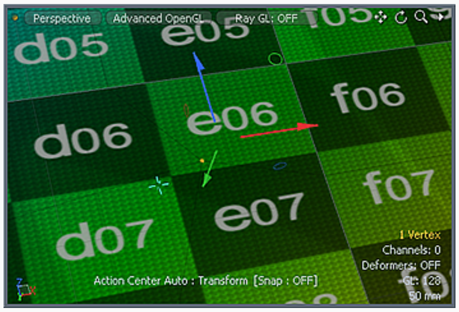
Action Center Auto 3D
X 1.364 mm
Y 96.961 mm
Z 7.322 mm

Radial Falloff
Center X 0 m
Y 0 m
Z 0 m
Size X 1 m
Y 1 m
Z 1 m
Auto Size
Shape Preset Smooth
In 0.0
Out 0.0
Mix Mode Multiply

The Bulge tool is in the **Deform** sub-tab of the modeling toolbox. (You can also choose **Edit > Deform > Bulge**.) Clicking the tool's button activates the tool. Next, click in the 3D viewport to enable the tool's interactive editing mode and for the Bulge tool to create an automatic falloff. Modo places the center of this falloff at the intersection of the pointer and the Work Plane, and Modo determines its size by the bounding box of the target element relative to the center. In some cases, this might not be the desired behavior; therefore, when the tool is active, you can manually position the falloff by right-clicking and dragging in the 3D viewport.

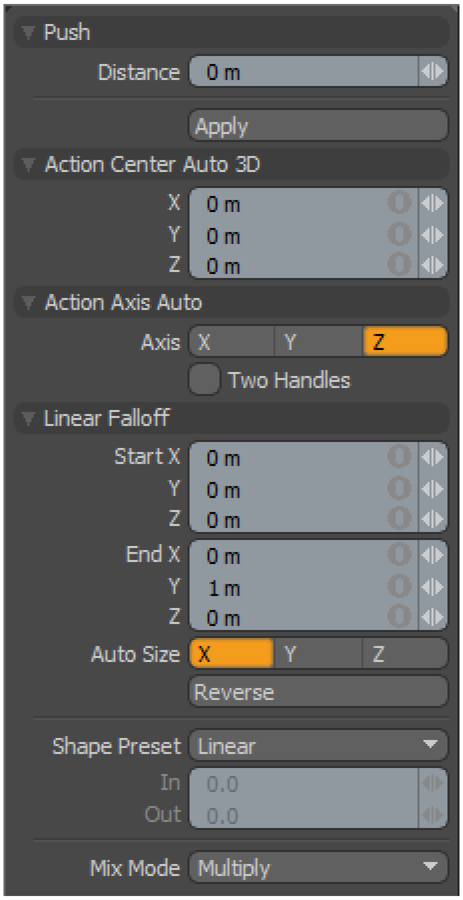
When you first activate the tool, Modo automatically fits the radial falloff to the current geometry and activates the Stretch tool at the center of the mesh. You use the Stretch handles to cause an apparent bulging of the object. To define a custom area of the mesh for applying the bulge, right-click and drag to set the Radial falloff region. Clicking applies the Stretch tool.

Bulge	
Transform	
Scale X/Y/Z	Displays specific scaling values applied to the geometry. You can enter specific values or adjust in the viewport. When adjusting an object interactively in the viewport, the **Scale** input values indicate the current scale factor applied.
Absolute Handle	When enabled, applies an additional function to the tool handles for you to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.
Negative Scale	When enabled, applies negative scale values, if appropriate, when you manipulate the tool handles in the viewport. When disabled, Modo does not automatically change the values if they are negative. You can always input negative values directly.
Pivot	Disabled for the Bulge tool.
Gimbal	Disabled for the Bulge tool.
Child Compensate	(Only available in **Items mode**) When enabled, ignores any transform values for child items. When disabled, passes along transform values to any child items of the current selection.
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. > >

Bulge	
	 Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping)  Perspective Advanced OpenGL Ray GL: OFF d05 e05 f05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Radial Falloff	
Center X/Y/Z	Defines the center of influence, which is where the strength of the falloff is the greatest (**100%**). The strength of the falloff attenuates toward the outer bounds of the spherical volume. The area outside the volume receives no tool influence (**0%**).
Size X/Y/Z	Defines the radius of a circle from the **Center** position and determines the outer

Bulge	
	area of the falloff where there is no effect.
Auto Size	Adjusts the falloff's **Center** and **Size** values automatically to match the bounding box of the current selection.
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Flare



Distance 0 m

X 0 m
Y 0 m
Z 0 m

Axis X Y Z

☐ Two Handles

Start X 0 m
Y 0 m
Z 0 m

End X 0 m
Y 1 m
Z 0 m

Auto Size X Y Z

Shape Preset Linear
In 0.0
Out 0.0

Mix Mode Multiply

The Flare tool combines the Push tool with a Linear falloff. With this tool, you can quickly flare out a specific portion of the mesh with the Linear falloff as the modulator.

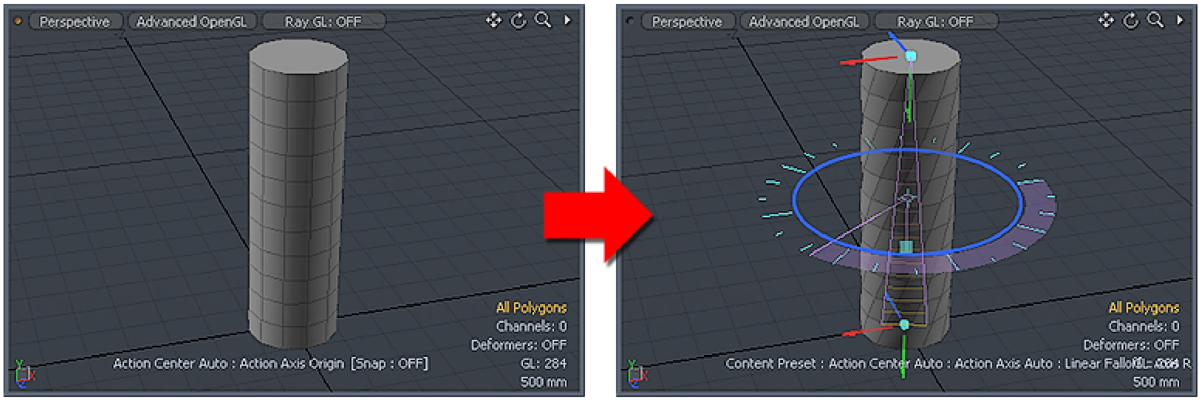
Flare	
Push	
Distance	Defines the actual amount Modo moves each vertex away from its original position. Modo moves the vertex directly along its normal. This value can be positive or negative.
Linear Falloff	
Start X/Y/Z	Sets a 3D position for the beginning of the Linear falloff. At this position Modo

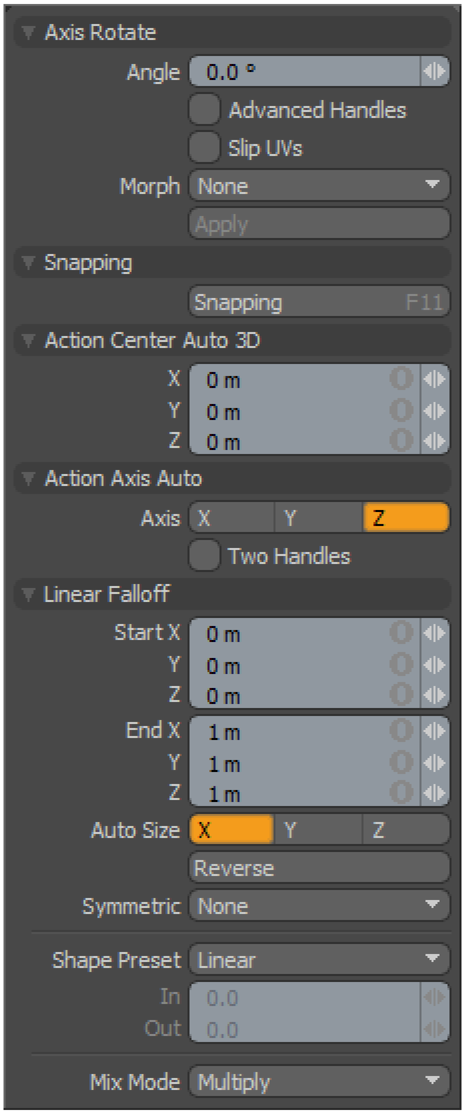
Flare	
	sets the falloff value to **0**. All vertices at this side of the falloff receive no effect from the tool.
End X/Y/Z	Sets a 3D position for the end of the Linear falloff. At this position, Modo sets the falloff value to **100**. All vertices at this side of the falloff are affected **100%** by the tool.
Auto Size	Snaps the Linear falloff to the specified axis and scales it to match the selected geometry.
Reverse	Swaps the **Start** and **End** values to flip the falloff.

Rotate Deform Tools

The following topics provide information on how to apply Modo's rotate deform tools, Twist, Bend, Vortex, and Swirl.

Twist


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 284
500 mm
Action Center Auto : Action Axis Origin [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
Content Preset : Action Center Auto : Action Axis Auto : Linear Falloff
500 mm



Axis Rotate

Angle 0.0 °

☐ Advanced Handles

☐ Slip UVs

Morph None

Apply

Snapping

Snapping F11

Action Center Auto 3D

X 0 m
Y 0 m
Z 0 m

Action Axis Auto

Axis X Y [✓ Z]

☐ Two Handles

Linear Falloff

Start X 0 m
Y 0 m
Z 0 m

End X 1 m
Y 1 m
Z 1 m

Auto Size [✓ X] Y Z

Reverse

Symmetric None

Shape Preset Linear

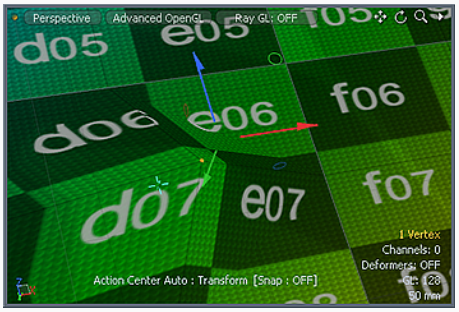
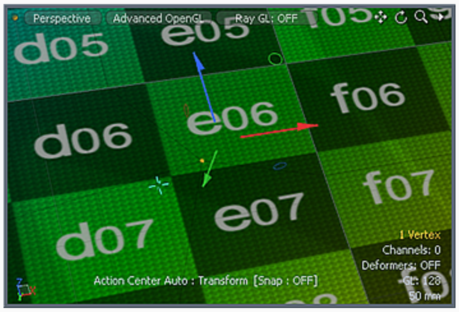
In 0.0
Out 0.0

Mix Mode Multiply

Found in the **Deform** sub-tab of the modeling toolbox, the Twist tool combines the Axis Rotate tool with a Linear falloff. At first click, Modo sets the Linear falloff perpendicular to the pointer and autofits it to the current geometry. Dragging now creates a twisting effect.

You can precisely set the effect of the tool by right-dragging out the Linear falloff to a specific position and then dragging to affect the rotation. This creates a twisting effect down the length of the falloff as long as the rotation axis is perpendicular to the falloff widget.

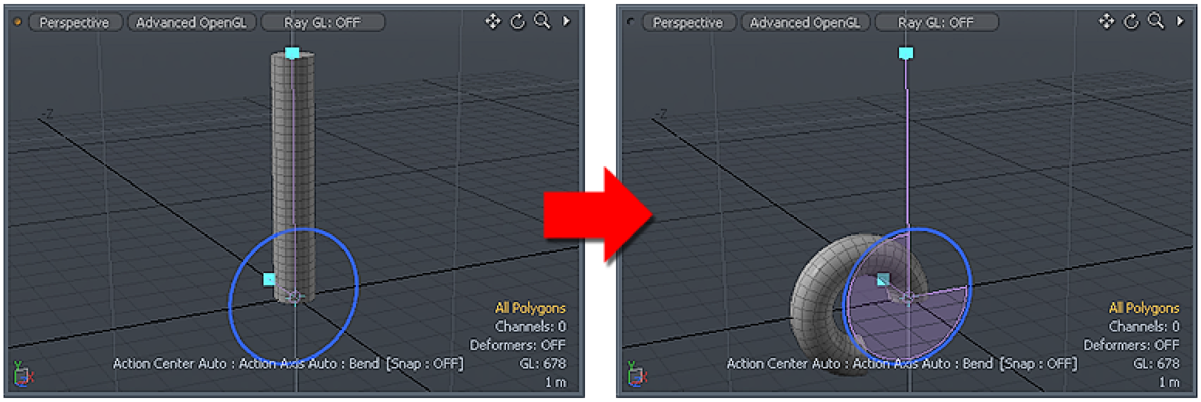
Twist	
Axis Rotate	
Angle	Displays a specific rotational value in degrees. You can enter a specific value or

Twist	
	adjust in the viewport. When adjusting an object interactively in the viewport, the **Angle** value indicates the current rotation.
Advanced Handles	When enabled, Modo displays additional information and controls in the viewport.
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  **Slip UVs** function disabled (notice the texture warping) >  **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions.

Twist	
	Note: In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Linear Falloff	
Start X/Y/Z	Defines the starting position of the falloff as a specific X,Y,Z coordinate value. This end of the line and everything beyond it receives the maximum amount of influence that attenuates toward the **End** position.
End X/Y/Z	Defines the end position of the falloff as a specific X,Y,Z coordinate value. This end of the line and everything beyond it receives no influence with the effect attenuating toward the **Start** position.
Auto Size X/Y/Z	Specifies the one axis to automatically scale the **Start** and **End** points of the falloff to match the bounding box size of the selected elements.
Reverse	Inverts the **Start** and **End** points to reverse the influence of the falloff.
Symmetric	Indicates how to automatically mirror the influence area of the falloff. • **None** - The Symmetric function is disabled. • **Start** - Mirrors the influence of the falloff symmetrically across the **Start** position. • **End** - Mirrors the influence of the falloff symmetrically across the **End** position.
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff.

Twist	
	- **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Bend

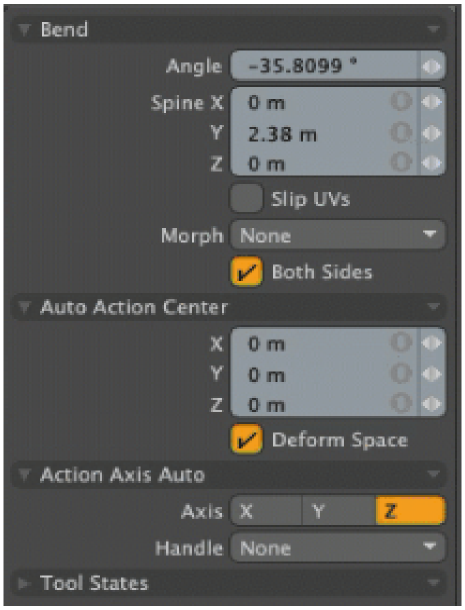

Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 678
1 m
Action Center Auto : Action Axis Auto : Bend [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 678
1 m
Action Center Auto : Action Axis Auto : Bend [Snap : OFF]

With the Bend tool you can quickly deform selected geometry in a curved manner. For example, take a straight cylinder (or a tube) and flex it into an arc to produce a section of a torus. This is similar to what the Bend tool does. The Bend tool consists of two handles: a bend spine and the bend angle disc. These control what gets bent and how much Modo deforms the geometry respectively.

Usage

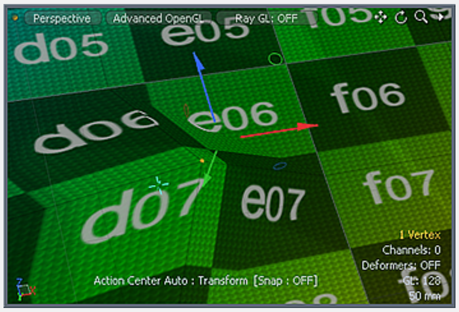
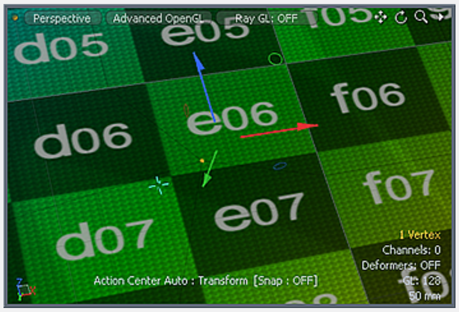
To use the Bend tool, click the tool's button in the **Deform** sub-tab of the default modeling toolbox. This activates the tool and shows the tool's attributes in the tool's **Properties** panel. Next, in the 3D viewport, click at the base location (where the bend should start from) to initiate the interactive editing mode and then draw the tool's handles. The center of influence for the Bend tool is at the intersection of where you click initially and the Work Plane. You may need to position the Work Plane appropriately prior to activating the tool because the Work Plane also determines the initial orientation of the tool.

Modo provides two handles to bend the target. The circular Angle handle, which is similar to the standard Rotate tool's handle, controls the amount of bending. The Spine handle determines the influence area for the operation. You can drag the Angle handle in a circular direction. The greater the distance is from where you clicked initially, the greater the bend amount is. Also, if you move the pointer (while dragging) further away from the handle, you have finer control for determining the angle. The spine determines which area of the target Modo bends. When you click and create the tool handle, its height is determined by the bounding box height of the target geometry. Because of this, Modo can deform the entire length of the target. You can drag the small blue square at the top of the handle to change its length to have a partial bend deformation. If you move the pointer over the blue square, Modo changes its color to yellow and dragging moves it. Dragging adjusts the position of the spine on the axes relative to the Work Plane. Pressing **Ctrl** while dragging constrains the move to a single axis. You can adjust the options in the **Properties** panel manually while the tool is active for a more precise bending deformation.



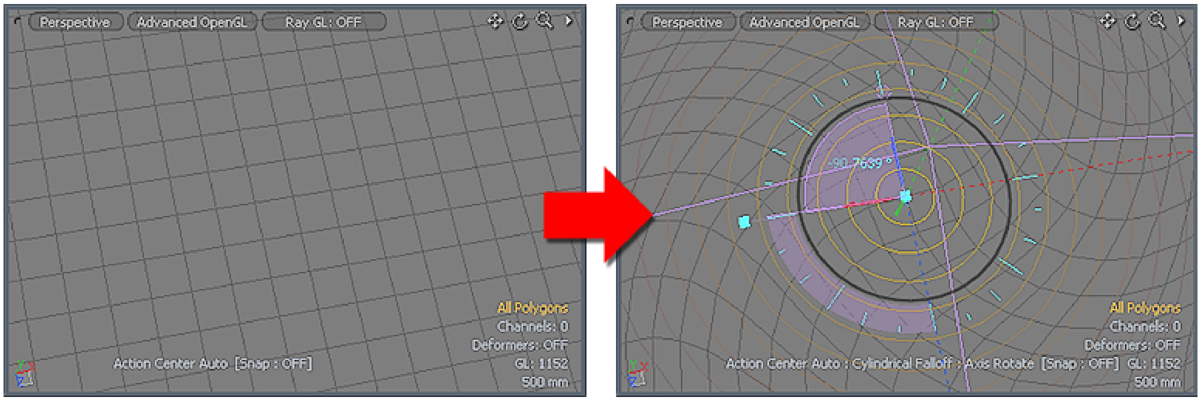
Bend
Angle -35.8099 °
Spine X 0 m
Y 2.38 m
Z 0 m
☐ Slip UVs
Morph None
☑ Both Sides
Auto Action Center
X 0 m
Y 0 m
Z 0 m
☑ Deform Space
Action Axis Auto
Axis X Y Z
Handle None
Tool States

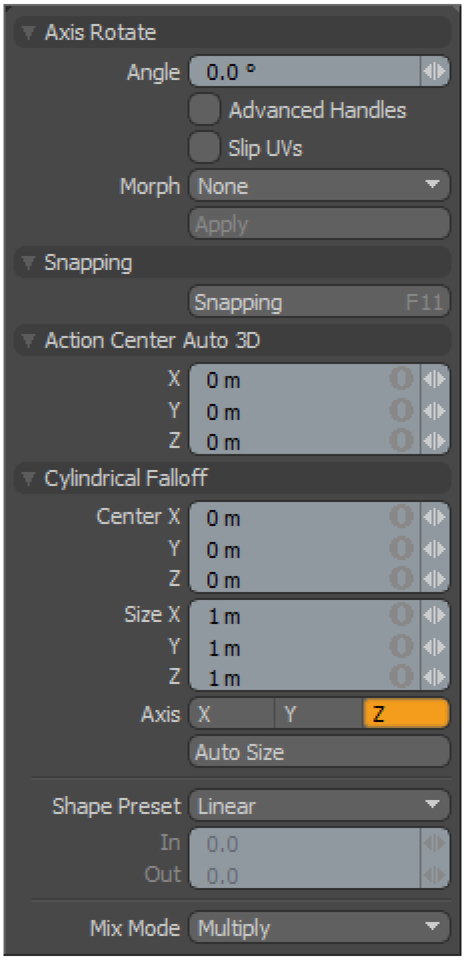
Bend	
Angle	Represents the amount of bend that is received by the geometry at the end of the spine. The Bend tool automatically fades this amount across the geometry selection down towards the base of the Bend tool to accomplish the intended bending effect.
Spine X/Y/Z	Sets the end of the spine handle. You can adjust the handle interactively in the 3D viewport by dragging or by typing positional values in the boxes.
Slip UVs	When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry.

Bend	
	>  Perspective Advanced OpenGL Ray GL: OFF Action Center Auto : Transform [Snap : OFF] 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm **Slip UVs** function disabled (notice the texture warping) >  Perspective Advanced OpenGL Ray GL: OFF Action Center Auto : Transform [Snap : OFF] 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm **Slip UVs** function enabled (texture remains even)
Morph	Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Both Sides	When **Both Sides** is enabled, the Bend tool deforms both the positive and negative sides of the spine vector.
Snapping	See the Applying Snapping topic for details about this feature.
Action Center Auto 3D	Positions the Angle adjustment handle relative to the position of the Work Plane when you first activate the tool in the 3D viewport. For more

Bend	
	information see the Specifying Action Centers and Falloffs topic.
Action Axis Auto	Specifies an alternate axis if the initial selection is incorrect for the desired bending direction.
Two Handles	When enabled, draws an additonal two handles in the 3D viewport for you to interactively adjust the orientation of the handle itself for off-axis bending.

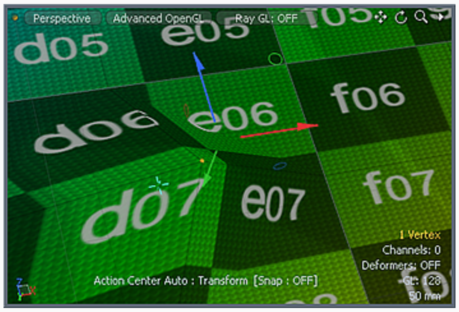
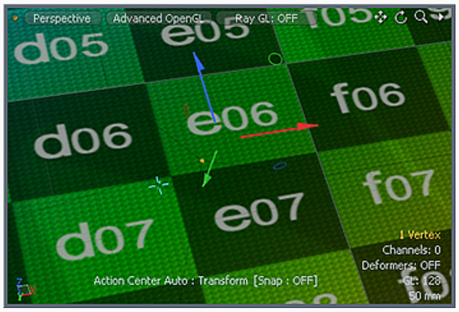
Vortex


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 1152
500 mm
Action Center Auto [Snap : OFF]
Perspective
Advanced OpenGL
Ray GL: OFF
-90.7639 °
All Polygons
Channels: 0
Deformers: OFF
Action Center Auto : Cylindrical Falloff : Axis Rotate [Snap : OFF] GL: 1152
500 mm


Axis Rotate
Angle 0.0 °
☐ Advanced Handles
☐ Slip UVs
Morph None
Apply
Snapping
Snapping F11
Action Center Auto 3D
X 0 m
Y 0 m
Z 0 m
Cylindrical Falloff
Center X 0 m
Y 0 m
Z 0 m
Size X 1 m
Y 1 m
Z 1 m
Axis X Y Z
Auto Size
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply

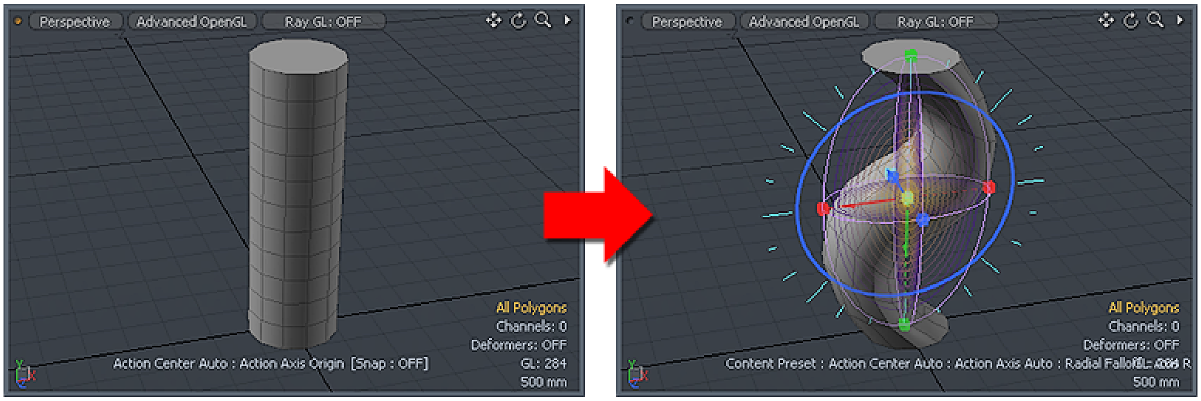
Found in the **Deform** sub-tab of the modeling toolbox, the Vortex tool combines the Axis Rotate tool with the Cylindrical falloff. After you click the first time, Modo creates the falloff region in a 2D plane centered on the pointer's position and automatically scaled to match the current geometry. It centers the Rotate widget also on the pointer's position and, if you drag now, Modo creates a vortex effect on the mesh. For more precise control of the affected region, right-drag out the falloff area first and the drag to affect the rotation.

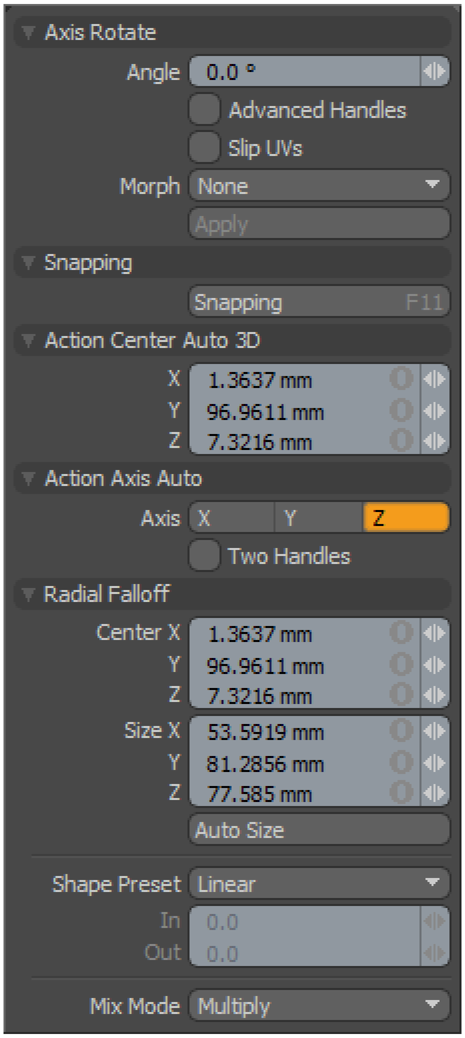
Vortex	
Axis Rotate	
Angle	Displays a specific rotational value in degrees. You can enter a specific value or adjust in the viewport. When adjusting an object interactively in the viewport, the **Angle** value indicates the current rotation.
Advanced Handles	When enabled, Modo displays additional information and controls in the viewport.

Vortex	
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  **Slip UVs** function disabled (notice the texture warping) >  **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo deals with the morph map vertex data when applying any transforms. **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. **Transform** - Transforms morph data along with the base mesh. **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions.

Vortex	
	Note: In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Cylindrical Falloff	
Center X/Y/Z	Defines the center of influence, which is where the strength of the falloff is the greatest (**100%**). The strength of the falloff attenuates toward the outer bounds of the cylinder. The falloff volume extends upward and downward toward infinity.
Size X/Y/Z	Defines the radius of a circle from the **Center** position and determines the outer area of the falloff where there is no effect.
Axis	Defines the projection axis for the cylinder and determines the direction for the falloff.
Auto Size	Adjusts the falloff's **Center** and **Size** values automatically to match the bounding box of the current selection.
Shape Preset	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Swirl


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 284
500 mm
Action Center Auto : Action Axis Origin [Snap : OFF]
Content Preset : Action Center Auto : Action Axis Auto : Radial Falloff



Axis Rotate

Angle 0.0 °

☐ Advanced Handles

☐ Slip UVs

Morph None

Apply

Snapping

Snapping F11

Action Center Auto 3D

X 1.3637 mm
Y 96.9611 mm
Z 7.3216 mm

Action Axis Auto

Axis X Y Z

☐ Two Handles

Radial Falloff

Center X 1.3637 mm
Y 96.9611 mm
Z 7.3216 mm

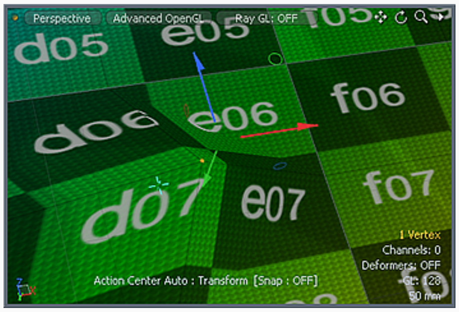
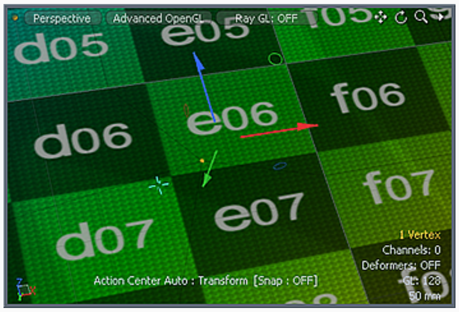
Size X 53.5919 mm
Y 81.2856 mm
Z 77.585 mm

Auto Size

Shape Preset Linear
In 0.0
Out 0.0

Mix Mode Multiply

Found in the **Deform** sub-tab of the modeling toolbox, the Swirl tool combines the Axis Rotate tool with a Radial falloff. When you first click in the viewport, Modo automatically fits the Radial falloff to the current geometry. Dragging in the 3D view applies a rotation modulated by this falloff. For more control over the region of influence, use the right mouse button to quickly define the falloff region, then use the left mouse button to apply the rotation. To precisely place the falloff, either use the handles to manipulate the falloff after it has been auto-sized or right-click and drag away from the handles to reset the falloff and drag it out interactively.

Swirl	
Axis Rotate	
Angle	Displays specific rotational values applied to the geometry (calculated in degrees). When adjusting an object interactively in the viewport, the **Angle** input values indicate the current rotational amount applied.
Advanced Handles	When enabled, displays additional information and controls in the viewport.
Slip UVs	(Only available in Component modes) When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function disabled (notice the texture warping) >  Perspective Advanced OpenGL Ray GL: OFF d05 e05 d06 e06 f06 d07 e07 f07 1 Vertex Channels: 0 Deformers: OFF GL: 128 50 mm Action Center Auto : Transform [Snap : OFF] **Slip UVs** function enabled (texture remains even)
Morph	(Only available in Component modes) Determines how Modo treats stored morphing information when applying transforms (such as the Move, Rotate, or Scale transforms) to geometry. There are three options for controlling how Modo

Swirl	
	deals with the morph map vertex data when applying any transforms. • **None** - Transforms selected (visible) morphs independent of their source, but does not affect unselected morph data. • **Transform** - Transforms morph data along with the base mesh. • **Keep Positions** - Converts morph data into an absolute morph map. All vertices retain their pre-transformed positions. **Note:** In previous versions of Modo, to transform a morph along with its base, you needed to select it in the **Vertex Map** list. If you didn't, when Modo recalled the relative morph map data, it would produce distorted, undesirable results; therefore, it was easy to accidentally spoil a model. To remedy this problem, current versions of Modo have three options to deal with the morph map vertex data.
Snapping	See the Applying Snapping topic for details about this feature.
Radial Falloff	
Center X/Y/Z	Defines the center of influence, which is where the strength of the falloff is the greatest (**100%**). The strength of the falloff attenuates toward the outer bounds of the spherical volume. The area outside the volume receives no tool influence (**0%**).
Size X/Y/Z	Defines the radius of a perfect circle from the **Center** position and determines the outer area of the falloff where there is no effect.
Auto Size	Adjusts the falloff's **Center** and **Size** values automatically to match the bounding box of the current selection.
Shape	Controls the strength of the falloff's influence along the extent by using a shape preset. • **Linear** - Attenuates the falloff evenly across its range. • **Ease-In** - Strengthens the falloff toward the start position. • **Ease-Out** - Strengthens the falloff toward the end position. • **Smooth** - Strengthens the falloff toward the center of the falloff. • **Custom** - Fine tunes the strength of the falloff based on the **In** and **Out** values.

Swirl	
In/Out	Determines the strength of the falloff nearer to the start or end position.
Mix Mode	Defines how each falloff interacts with the other(s) in instances where there are multiple falloffs applied to a transform (by using **Add** in the **Falloff** menu).

Editing Items

Modo offers a variety of tools to divide geometry in different ways. To reshape existing surfaces, you can use the various slicing functions to add more geometry to any surface area lacking in polygonal density. The Boolean and Drill tools shape forms by adding and subtracting simple shapes to create more complex ones. The subdivide commands provide a simple way to increase overall density across a mesh (perhaps in preparation for applying sculpting tools). For information on using a specific tool, click a link below to see that tool's topic.

Slice		
	Slice	Sets a slicing plane through the currently-selected geometry.
	Axis Slice	Creates one or more evenly spaced slices aligned on the X, Y, or Z axes.
	Curve Slice	Cuts the mesh with a user-defined curve.
	Julienne	Creates multiple evenly-spaced slices that you can position interactively in the 3D viewport with resulting geometry that contains inner faces.
	Edge Slice	Creates a slice between edges by clicking from edge to edge.
	Topology Pen	Re-topologizes high-resolution geometry.
	Pen Slice	Similar to the pen tool, creates a slice plane.

Slice		
	Contour	Generates contours from user-defined slices across an existing polygon surface.
	Loop Slice	Lets you slice additional edge loops into your geometry.
	Shatter	Breaks a model into a specified number of individual pieces.
Edit		
	Booleans	Combines mesh layers in various ways (Union, Intersect, Subtract, and Add).
	Solid Drill	Performs Boolean operations onto flat 2D polygons by using overlapping 3D geometry.
	Axis Drill	Performs Boolean-like operations with 2D polygons projected down a specified axis.
	Subdivide	Recursively splits the selected polygons into smaller polygons.
Other		
	Edit Commands - Cut/Copy/Paste	Cut, Copy, and Paste commands.
	Edit Commands - Delete/Remove	Discards selected elements from a model/scene.
	Mesh Cleanup	Repairs problematic geometry.
	Slice Effector	Cuts geometry on a mesh along the particle positions provided by a tool generator connecting to **Tool Pipe**. This is used in combination with the Pen Slice and Curve Slice tools.
	Auto Retopo	Generates a new mesh item from the input surface. For more information, see Automatic Retopology Tool.

Slice		
🔧	Retopo Edge Lock	Refines your geometry results based on curves drawn on sharp edges. For more information, see Refining Your Geometry Results.
🔧	Retopo Edge Flow	Refines your geometry results based on curves drawn for directional flow. For more information, see Refining Your Geometry Results.

Slice Tools

The following topics describe the various ways you can slice your geometry in Modo.

Slice

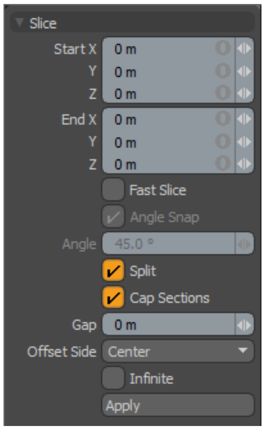
Found in the **Mesh Edit** sub-tab of the **Modeling** toolbox, the **Slice** tool sets a slicing plane through the currently-selected geometry. The slice plane is interactively set by dragging out the tool across a 3D viewport. The first click in the 3D view sets the **Start** position for the slice. Clicking and dragging extends the slicing plane out to the **End** position, which can also be dragged interactively. The orientation of the plane is relative to the **Work Plane**, drawing perpendicular to its current position. The **Start** and **End** positions are simply X,Y, and Z positions that are set with the mouse, or numerically on the **Tool Properties** panel.

As long as the tool is active, you can drag the cutting plane by clicking on either of the blue end handles, or clicking and dragging anywhere along the pink line that connects the start and end points. To create a new slice using the same basic length and axis, but set a new center position, you can middle-click at the new location. The slice plane jumps up to that location and create a new slice. To begin a completely new slice, use the **Shift**+click and drag shortcut. This leaves the previous slice in place and begins a reset slice action at the mouse down position.

Tip: To improve your modeling workflow, use the **Select by Previous Operation** mesh operation. The following **Name** options are available for the **Slice** tool: **Intersecting Edges** and **Cap Sections**. For more information, see Select by Previous Operation.

Slice Properties

The following **Slice** options are available:



Slice

Start X 0 m

Y 0 m

Z 0 m

End X 0 m

Y 0 m

Z 0 m

☐ Fast Slice

☑ Angle Snap

Angle 45.0 °

☑ Split

☑ Cap Sections

Gap 0 m

Offset Side Center

☐ Infinite

Slice	
Start X/Y/Z	Determines the beginning position that defines the slicing plane. You can edit the position interactively in the viewport, or accurately define values with the XYZ coordinate inputs.
End X/Y/Z	Determines the ending position that defines the slicing plane. You can edit the position interactively in the viewport, or accurately define values with the XYZ coordinate inputs.
Fast Slice	When this option is enabled, the slicing action won't occur, as the plane is edited until the mouse button is released. This can help to increase performance when slicing dense meshes.
Angle Snap	When snapping is enabled (press the **X** key to temporarily toggle snapping states), the slicing plane snaps (jumps) to the nearest specified angle.
Angle	Determines the snapping angle for **Angle Snap** in degrees of rotation.
Split	Slices the surface into distinct separated surfaces on either side of the slice line.
Cap Sections	With the **Split** option enabled, the **Cap** option adds a polygon cap to the separated objects, where there are enclosed volumes.
Gap	With the **Split** option enabled, the **Gap** option increases the thickness of the cutting plane, eliminating a larger portion of the target surface.
Offset Side	Adjusts the placement of the gap to be above, below or centered on the cutting plane.

Slice	
Infinite	With the **Infinite** option enabled, the defined line section is extended indefinitely beyond the **Start** and **End** values. **Tip:** To slice everything you want, enable the **Infinite** option, and then make sure that you are inside the mesh layer you want to slice with either everything selected or nothing selected.

Procedural Slice

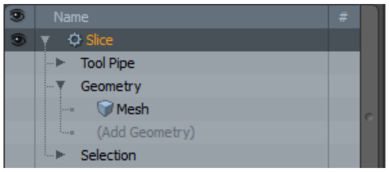
Slice is also available as a procedural mesh operation. You can add the tool in the **Mesh Ops** tab. In the **Model** layout, you can find the list on the right side of the interface, next to the **Item List**.

To add the Slice mesh operation:

1. In the **Mesh Ops** tab, click the **Add Operator** button, and navigate to **Mesh Operations** > **Edit** > **Slice**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

   Slice is added to the list, and a slicing plane is set through the polygon mesh.

You can expand the Slice item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses:



Slice can use the following inputs:

- **Tool Pipe** - Allows you to add sub-tools to the operation.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.
- **Selection** - Allows you to select the elements to which you want to apply the operation.

Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

The Slice mesh operation has the same properties as the traditional Slice tool. See Slice.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove.

You can also use the **Select by Previous Operation** mesh operation. The following **Name** options are available for the **Slice** tool: **Intersecting Edges** and **Cap Sections.** For more information, see Procedural Selection

Curve Slice

The Curve Slice tool cuts the mesh with a user-defined curve. There are two methods possible:

1. Create a curve and place it in the background.

Note: Make it visible in the **Item List**, but do not select it.

When you activate the tool, the background curve becomes the cutting source.

2. If there is no visible background curve when you activate the tool, draw a slicing curve interactively in the 3D viewport and then edit it as you would with a normal curve. (See the Curve tool topic for information about editing curves.) The Path Steps Generator section of this tool's **Properties** panel has settings to control the number of steps along the curve, which affects the smoothness of the newly created edges.

Procedural Curve Slice

Curve Slice is also available as a procedural mesh operation. It is a combination of the Slice Effector, Path Segment Generator, and Path Generator sub-tools. You can add the tool from the **Mesh Ops** tab, on the right-side of the **Model** layout.

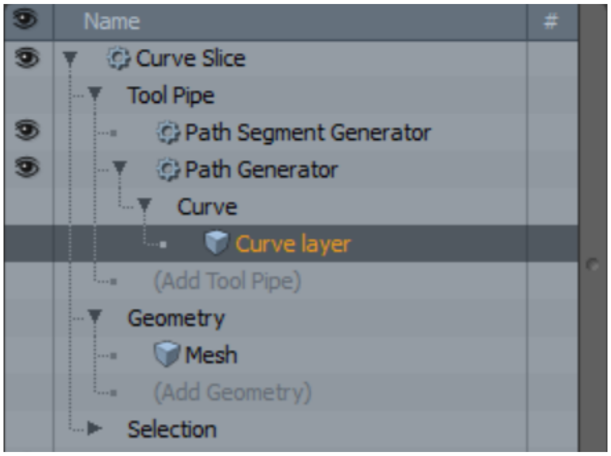
To use the tool, you need a geometry to cut and a slicing curve to cut with.

1. Add the geometry you want to slice using the **Add Operator** button in the **Mesh Ops** tab.

2. Add a new mesh layer to place the curve in: Click **Add Item** in the upper half of the **Mesh Ops** tab and select **Item** > **Mesh**.
3. With the new mesh layer selected, draw a curve in it, using the traditional curve tools in **Curve** tab of the modeling toolbox. For more information, see Adding Curves.
4. Select your original mesh layer, the one that contains your geometry, and in the **Mesh Ops** tab, click **Add Operator**, then select **Mesh Operations** > **Edit** > **Curve Slice**.

   The **Create Slice Clone** dialog opens.
5. Select the mesh that contains your curve as **Path**, then click **OK**.

   Curve Slice is added to the **Mesh Operations** list. When you expand it by clicking the small arrow in front of it, you can see its inputs.



- Curve Slice
  - Tool Pipe
    - Path Segment Generator
    - Path Generator
      - Curve
        - Curve layer
    - (Add Tool Pipe)
  - Geometry
    - Mesh
    - (Add Geometry)
  - Selection

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove.

You can also use the **Select by Previous Operation** mesh operation. The following **Name** options is available for the **Curve Slice** tool: **Intersecting Edges**. For more information, see Select by Previous Operation.

Curve Slice Properties

The following options are available:

Option	Description

Enable	Activates/Deactivates tool. By default, this property is enabled.
Use World Transform:	Centers a geometry selection at the World Origin position. By default this property is enabled.
Axis	Defines the axis across which to project the drill function.
Vector X, Y, Z	Sets the axis vector position for X, Y, and Z.
Select Sliced Edges	Sets edge selections for sliced edges to the selection set. The selection is automatically created and displayed in the **Mesh Ops** tab under **Lists**.
Edge Selection Set	Defines the name of the edge selection set for the **Select Sliced Edges** option. This name can be entered in the **Add Operator** > **Mesh Operations** > **Selection** > **Select By Selection Set** which can be used with Edge mesh operations.

Pen Slice

The Pen Slice tool uses the same underpinnings as the Pen tool. Therefore, you can leverage the same structured or organic flow for creating a slice plane as you would when creating a polygon with the standard Pen tool.

In the **Type** list, select either **Open** or **Closed**. By default, in **Closed** mode, Modo connects the first and last point in the slice profile to create a closed loop.

Self-snapping options include **World Axis**, **Straight Line**, and **Right Angles**.

Procedural Pen Slice

Pen Slice is also available as a procedural mesh operation. It is a combination of the Slice Effector and the Pen Generator sub-tools. You can add the tool from the **Mesh Ops** tab, on the right-side of the **Model** layout.

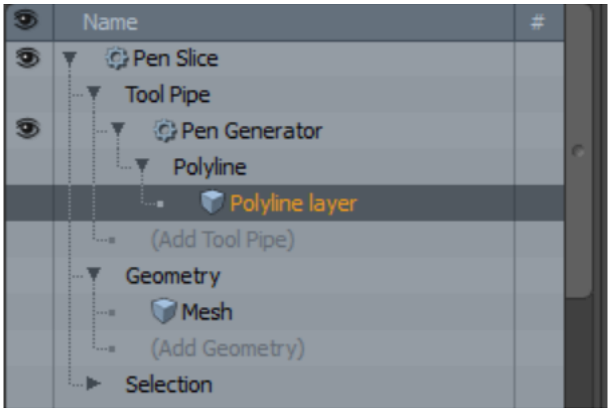
To use the tool, you need a geometry to cut and a polyline to cut with.

1. Add the geometry you want to slice using the **Add Operator** button in the **Mesh Ops** tab.

2. Add a new mesh layer to place the curve in: Click **Add Item** in the upper half of the **Mesh Ops** tab and select **Item** > **Mesh**.
3. With the new mesh layer selected, draw a polyline in it, using the Pen tool in the the modeling toolbox. For more information, see Pen.
4. Select your original mesh layer, the one that contains your geometry, and in the **Mesh Ops** tab, click **Add Operator**, then select **Mesh Operations** > **Edit** > **Pen Slice**.

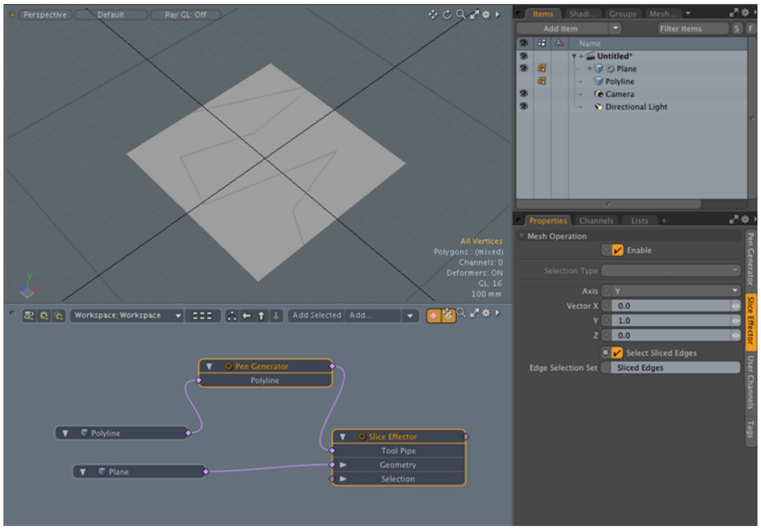
   The **Create Pen Slice** dialog opens.
5. Select the mesh that contains your polyline as **Polyline**, then click **OK**.

   Pen Slice is added to the **Mesh Operations** list. When you expand it by clicking the small arrow in front of it, you can see its inputs.



Slice Effector

The **Slice Effector** tool is used to cut geometry on a mesh along the particle positions provided by a tool generator connecting to **Tool Pipe**. This is used in combination with the Pen Slice and Curve Slice tools. The Pen Slice tool slices the geometry along the polyline path with the pen generator tool. The Curve Slice tool slices the geometry along the curve path and path steps generator tools.


Perspective
Default
Ray GL: Off
Items
Shadi...
Groups
Mesh ...
Add Item
Filter Items
Name
Untitled*
Plane
Polyline
Camera
Directional Light
All Vertices
Polygons : (mixed)
Channels: 0
Deformers: ON
GL: 16
100 mm
Workspace: Workspace
Add Selected
Add...
Pen Generator
Polyline
Polyline
Slice Effector
Tool Pipe
Plane
Geometry
Selection
Properties
Channels
Lists
Mesh Operation
Selection Type
Pen Generator
Slice Effector
User Channels
Tags

☑ Enable

Axis: Y

Vector X: 0.0

Y: 1.0

Z: 0.0

☑ Select Sliced Edges

Edge Selection Set: Sliced Edges

Usage

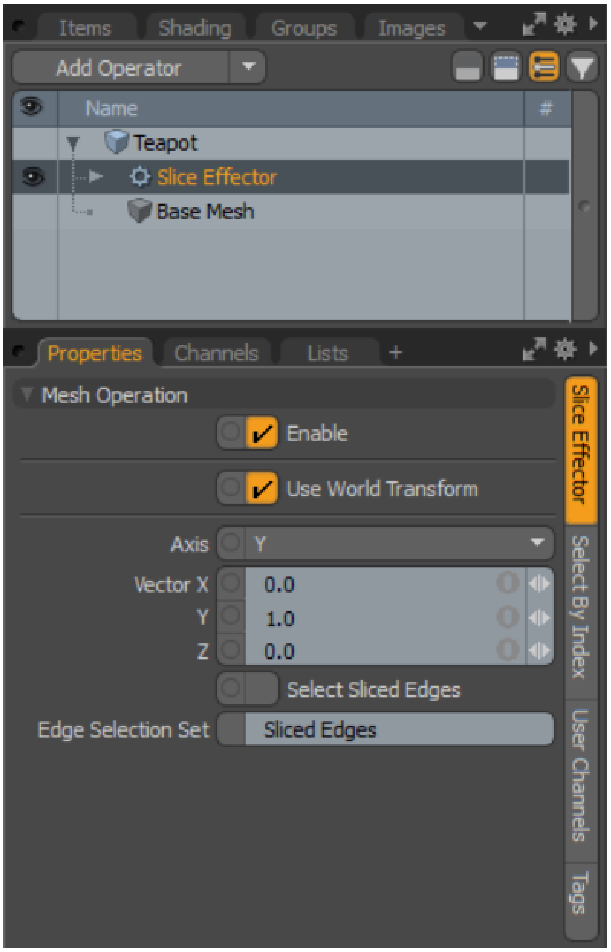
To use the Slice Effector tool, you need a geometry to cut and a slicing curve to cut with.

1. Add the geometry you want to slice using the **Add Operator** button in the **Mesh Ops** tab.
2. Add a new mesh layer to place the curve in: Click **Add Item** in the upper half of the **Mesh Ops** tab and select **Item** > **Mesh**.
3. With the new mesh layer selected, draw a curve in it, using the traditional curve tools in **Curve** tab of the modeling toolbox. For more information, see Adding Curves.
4. Select your original mesh layer, the one that contains your geometry, and in the **Mesh Ops** tab, click **Add Operator**, then select **Mesh Operations** > **Edit** > **Curve Effector**.

   The **Create Slice Clone** dialog opens.
5. Select the mesh that contains your curve as **Path**, then click **OK**.

   Curve Slice is added to the **Mesh Operations** list. When you expand it by clicking the small arrow in front of it, you can see its inputs.

Slice Effector Properties


Items
Shading
Groups
Images
Add Operator
Name
#
Teapot
Slice Effector
Base Mesh
Properties
Channels
Lists
+
Mesh Operation
Slice Effector
Select By Index
User Channels
Tags

☑ Enable

☑ Use World Transform

Axis: Y

Vector X: 0.0
Y: 1.0
Z: 0.0

☐ Select Sliced Edges

Edge Selection Set: Sliced Edges

The following options are available:

Option	Description
Enable	Activates/Deactivates tool. By default, this property is enabled.
Use World Transform:	Centers a geometry selection at the World Origin position. By default this property is enabled.
Axis	Sets the axis direction to slice geometry.
Vector X, Y, Z	Sets the axis vector position for X, Y, and Z.
Select Sliced Edges	Sets edge selections for sliced edges to the selection set. The

	selection is automatically created and displayed in the **Mesh Ops** tab under **Lists**.
Edge Selection Set	Defines the name of the edge selection set for the **Select Sliced Edges** option. This name can be entered in the **Add Operator** > **Mesh Operations** > **Selection** > **Select By Selection Set** which can be used with Edge mesh operations.

Edge Slice

With the **Edge Slice** tool you can click from edge to edge, creating a slice between the edges as you go. After clicking down on an edge, you can drag along the edge's length to set a precise cut location. You can also edit previously-made edge cuts by clicking directly on the handle point that represents the desired cut point and dragging it to a new location.

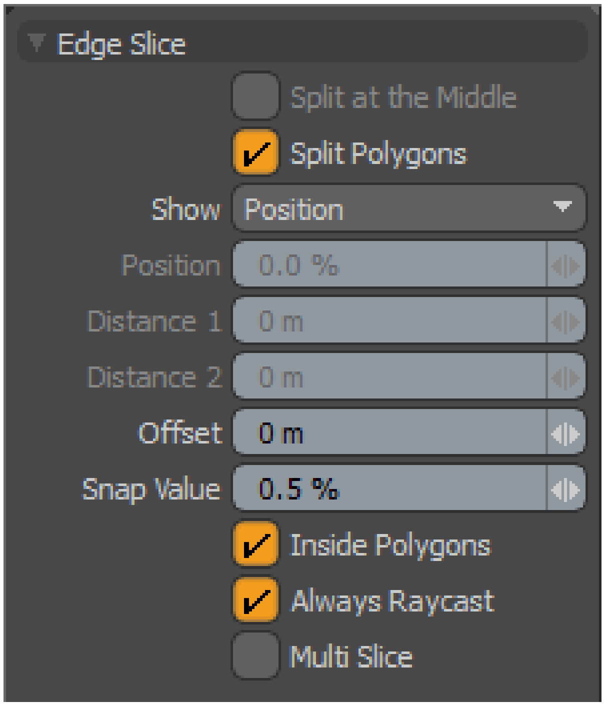
Often you may wish to use the **Edge Slice** to create a closed cut where the last point and the first point are joined by an edge so that you have created a new "edge loop". To do this you can create a final cut on the edge that received the first cut. As you drag the new cut point towards the first cut point on the edge, the color of both points changes when they overlap, so you can see that they have been merged. At this point, you can deactivate the tool as you have completed the edge loop.

Activating the Edge Slice Tool

- In the **Model** layout **Toolbox**, under the **Mesh Edit** sub-tab, click **Edge Slice**.
- In the **Topology** layout **Tools** sub-tab, click **Edge Slice**.

From the menu bar, click **Geometry** > **Slice** > **Edge**.

Edge Slice Properties



Edge Slice

☐ Split at the Middle

☑ Split Polygons

Show: Position

Position: 0.0 %

Distance 1: 0 m

Distance 2: 0 m

Offset: 0 m

Snap Value: 0.5 %

☑ Inside Polygons

☑ Always Raycast

☐ Multi Slice

Split at Middle	When active, this option forces all cuts to be centered on each edge as it is cut. This option can be activated temporarily by holding down the **Ctrl** key while cutting an edge.
Split Polygons	When enabled, the **Split Polygons** function creates a new edge between the cut points of the previous edges. With this disabled, new points are added to the polygons, but no edges connect them.
Show (None/Position/Distance)	Provides a small heads up display of data related to the current cut point as a position along the edge as a percentage, or a two distance values indicating the distance to either point in the cut edge. There is also the **None** option that disabled the display.
Offset	Defines the distance from an edge cut to a cut position within the polygon. The length value is updated automatically when a cut is performed and while the tool is interactive, the value can be adjusted measuring the length of the resulting cut line from the prior edge to the interior cut.

Snap Value	Constrains the newly created vertices to specified increments. To adjust the slice vertices without constraint, simply set this value to 0%.
Inside Polygons	With this toggle setting active, you can click anywhere inside a polygon to create a new edge that goes into the interior of the polygon. This option makes it very convenient to carve in additional details to existing polygons.
Always Raycast	Helps when cutting elements while working in wireframe mode, culling out any occluded polygons, meaning that, when enabled only, front-most facing polygons can be cut by the tool.
Multi Slice	When enabled, a dashed line is drawn in the viewport, showing the link between the location of the last cut to the current cursor position.

Loop Slice

Loop Slice is a powerful tool that lets you slice additional edge loops into your geometry. **Loop Slice** is found in the **Model** layout **Toolbox**, under the **Edit** sub-tab. It is also found in the **Topology** layout, under the **Tools** sub-tab. In addition, you can also activate it by choosing **Geometry** > **Slice** > **Loop**.

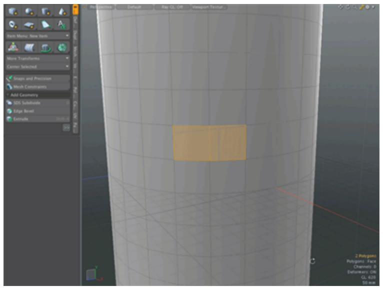
Direct Loop Slice

To use **Loop Slice** in direct modeling:

1. To define the loop, select two adjoining polygons or an edge shared by the polygons on your mesh.
2. Go to the **Edit**sub-tab and click the **Loop Slice** button to activate the tool.

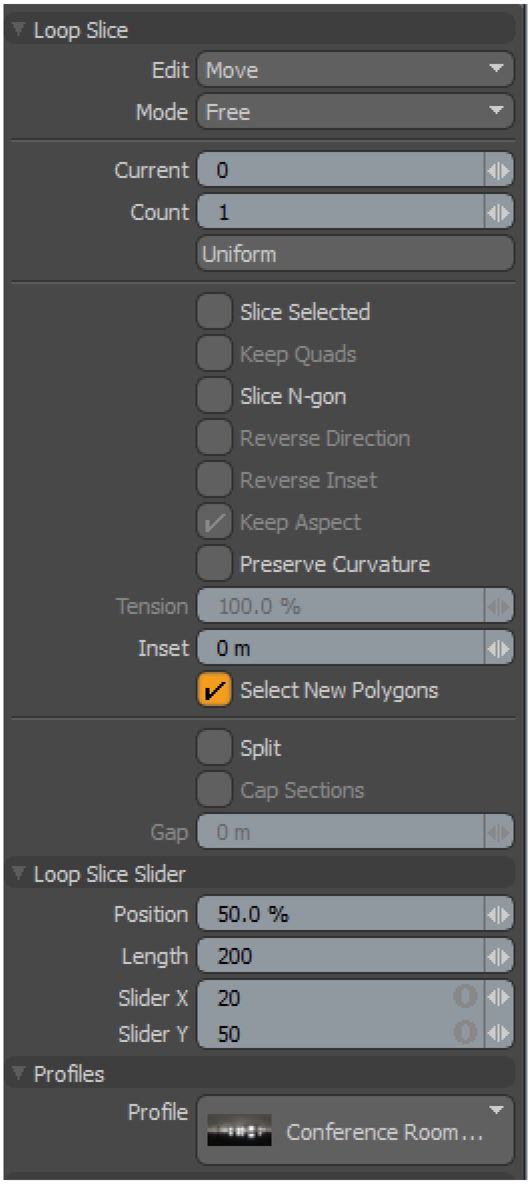
   A purple bar appears at the top of the viewport, which you can manipulate to modify the slicing.
3. Drag the small yellow triangle to adjust the currently-active slice position. The far end dots can be dragged to change the size of the input bar. In addition, you can enable the Select New Polygon option for better visibility of the polygons, which you are manipulating.

Click the image below to view the animation.



Direct Loop Slice Properties

Click the **Tool Properties** button to open the following options.



Loop Slice

Edit: Move

Mode: Free

Current: 0

Count: 1

Uniform

☐ Slice Selected

☐ Keep Quads

☐ Slice N-gon

☐ Reverse Direction

☐ Reverse Inset

☑ Keep Aspect

☐ Preserve Curvature

Tension: 100.0 %

Inset: 0 m

☑ Select New Polygons

☐ Split

☐ Cap Sections

Gap: 0 m

Loop Slice Slider

Position: 50.0 %

Length: 200

Slider X: 20

Slider Y: 50

Profiles

Profile: Conference Room…

Edit	Defines what function the mouse plays: **Move** repositions the slice within the loop by hauling in the viewport or dragging on the slice indicator, **Add** creates additional slices when clicking on the indicator bar, and **Remove** deletes them.
Mode	Defines how Modo creates and positions slices: **Free** allows any position along the indicator bar, **Uniform** evenly spaces the slices within the loop, and **Symmetry** mirrors the loop slices over the 50% position.
Current	Denotes which slice is the active slice for editing. Slices are numbered sequentially left to right in the input bar.
Count	The total number of slices added to the loop.
Uniform	Clicking this button automatically repositions **Freely** spaced spans evenly across the width of the loop.
Position	Slice position expressed as a spacing percentage in relation to the loop width. 50% is the center.
Length	Controls the length in pixels of the purple slice input bar in the viewport.
Slider X/Y	Controls the position of the input bar in the viewport.
Slice Selected	Restricts the slice to only the selected portion of the mesh, rather than using the selection to simply determine the loop to cut. This setting respects corners as well.
Keep Quads	Forces all new polys to be quadrangular inside the sliced area.
Slice N-gon	When this option is enabled, Modo attempts to continue a loop slice across polygons with more than four sides. For better control over how **Loop Slice** slices across the N-gon, select the loop as you wish it to be sliced and enable the **Slice Selected** option.
Reverse Direction	If this option is enabled, the 1D profile is evaluated from top to bottom.
Reverse Inset	If this option is enabled, **Inset** is reversed.
Keep Aspect	Automatically sets the **Inset** value based on the aspect ratio of the profile.
Preserve Curvature	The new position is computed along a curve through the edges to maintain the overall curvature or surface continuity. Once enabled, the **Tension** setting becomes available, controlling the tightness of the curvature preservation.

Tension	Determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length, placing the new vertex appropriately, where a value of 0% would be like disabling the **Preserve Curvature** option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.
Inset	Specifies the offset amount of **X** direction of 1D profile.
Select New Polygon	When enabled, provides better visibility to see the selected polygons. A guide color is displayed for the slicing edges of the selected polygons.
Split	When the **Split** option is enabled the resulting 'loop' split the surface, producing two distinct overlapping edges, rather than a single connected edge.
Cap Sections	When using the **Split** option, the **Cap Sections** option can be added to create a single polygon "cap" connected to the two resulting edge loops, in effect closing off the two new edges.
Gap	When splitting surfaces, the **Gap** option defines a width around the split line, in effect thickening the width of the resulting cut by the defined value.
Profiles	
Profile	A mini **Preset Browser** for viewing various profiles. Works the same as the standard Preset Browser. You can simply click on a profile to load that particular shape as a **Loop Slice** cutter. Has the effect of what a router does when cutting in to and engraving a wood panel.

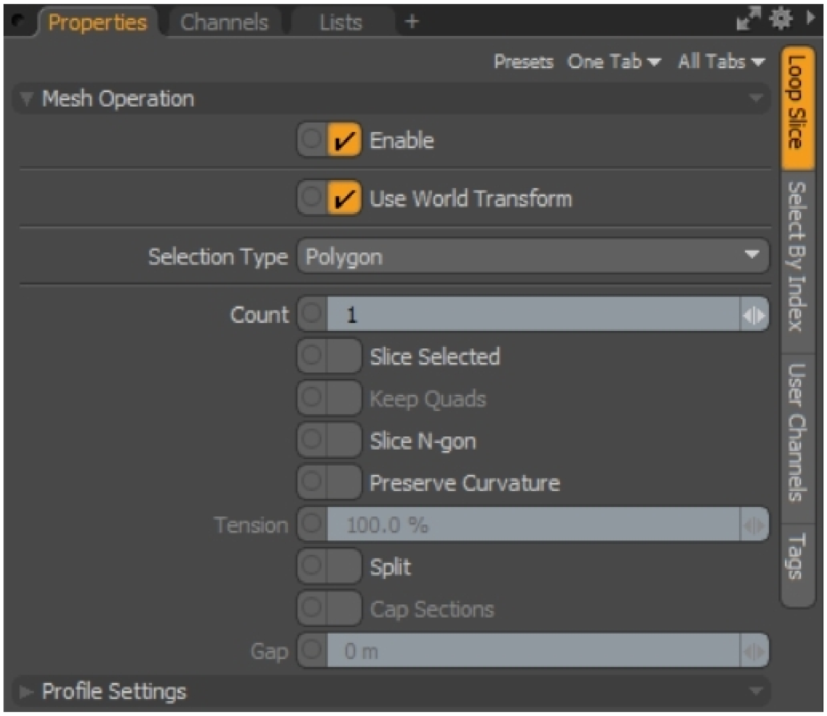
Procedural Loop Slice

The procedural version of **Loop Slice** evenly slices polygon loops by specific **Count**, much like the **Uniform** mode of the direct version. Instead of setting the slice position interactively, the procedural version uses an **Array** modifier in its input channel for the **Slice Position**.

To use **Loop Slice** in procedural modeling:

1. To define the loop, select two adjoining polygons or an edge shared by the polygons on your mesh.
2. In the **Mesh Ops** tab, click **Add Operator**> **Mesh Operations** > **Edit** > **Loop Slice**.
3. Go to the **Properties** tab to modify the **Loop Slice** operation.

Procedural Loop Slice Properties



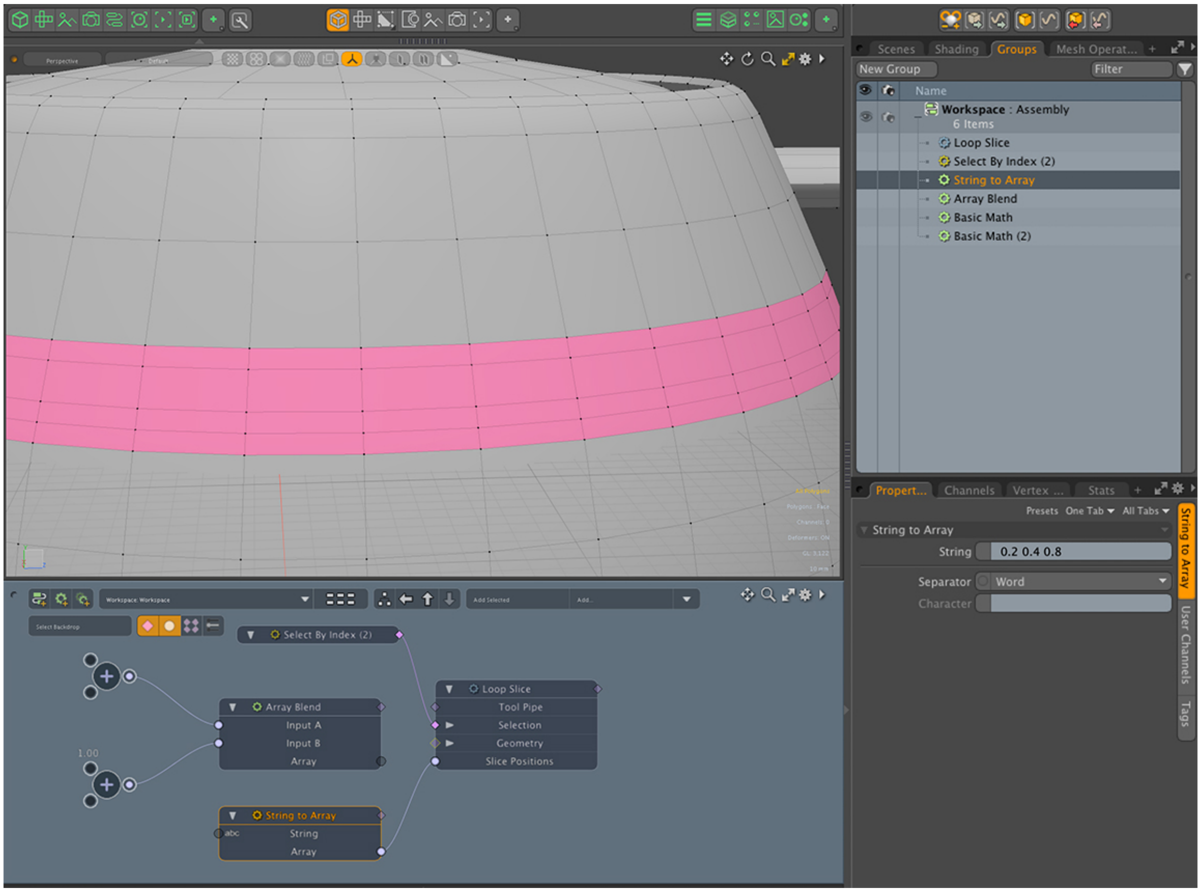
☑ Enable
☑ Use World Transform
Selection Type Polygon
Count 1
☐ Slice Selected
☐ Keep Quads
☐ Slice N-gon
☐ Preserve Curvature
Tension 100.0 %
☐ Split
☐ Cap Sections
Gap 0 m

Selection Type	Seed mesh element (Polygon or Edge).
Count	The total number of slices added to the loop.
Slice Selected	Restricts the slice to only the selected portion of the mesh, rather than using the selection to simply determine the loop to cut. This setting respects corners as well.
Keep Quads	Forces all new polys to be quadrangular inside the sliced area.
Slice N-gon	When this option is enabled, Modo attempts to continue a loop slice across polygons with more than four sides. For better control over how **Loop Slice** slices across the N-gon, select the loop as you wish it to be sliced and enable the **Slice Selected** option.
Preserve Curvature	The new position is computed along a curve through the edges to maintain the overall curvature or surface continuity. Once enabled, the **Tension** setting becomes available, controlling the tightness of the curvature preservation.
Tension	Determines the strength of the curvature preservation. The default value of 100% calculates the curvature similar to a spline generated along the edge's length,

	placing the new vertex appropriately, where a value of 0% would be like disabling the **Preserve Curvature** option. Negative values inset the curvature and higher values further offset the vertex out from the curve's position.
Split	When the **Split** option is enabled the resulting 'loop' splits the surface, producing two distinct overlapping edges, rather than a single connected edge.
Cap Sections	When using the **Split** option, the **Cap Sections** option can be added to create a single polygon "cap" connected to the two resulting edge loops, in effect closing off the two new edges.
Gap	When splitting surfaces, the **Gap** option defines a width around the split line, in effect thickening the width of the resulting cut by the defined value.
Profile Settings	
Inset	Specifies the amount of offset on the **X** axis for a 1D profile.
Reverse Direction	If this option is enabled, the 1D profile is evaluated from top to bottom.
Reverse Inset	If this option is enabled, Inset is reversed.
Keep Aspect	Automatically sets the Inset value based on the aspect ratio of the profile.

Working with the Array Channel Modifier

String to Array is a simple way to set multiple slice positions by text. The loop slice decodes the numeric words to slice positions. When an **Array** channel is connected to the **Slice Positions** input on a Loop Slice node, the **Count** channel of the **Loop Slice** node is overwritten with the number set in the array size.


Scenes
Shading
Groups
Mesh Operat...
New Group
Filter
Name
Workspace : Assembly
6 Items
Loop Slice
Select By Index (2)
String to Array
Array Blend
Basic Math
Basic Math (2)
Propert...
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
String to Array
User Channels
Tags
Select By Index (2)
Array Blend
Input A
Input B
Array
1.00
Loop Slice
Tool Pipe
Selection
Geometry
Slice Positions
String to Array
abc
String
Array

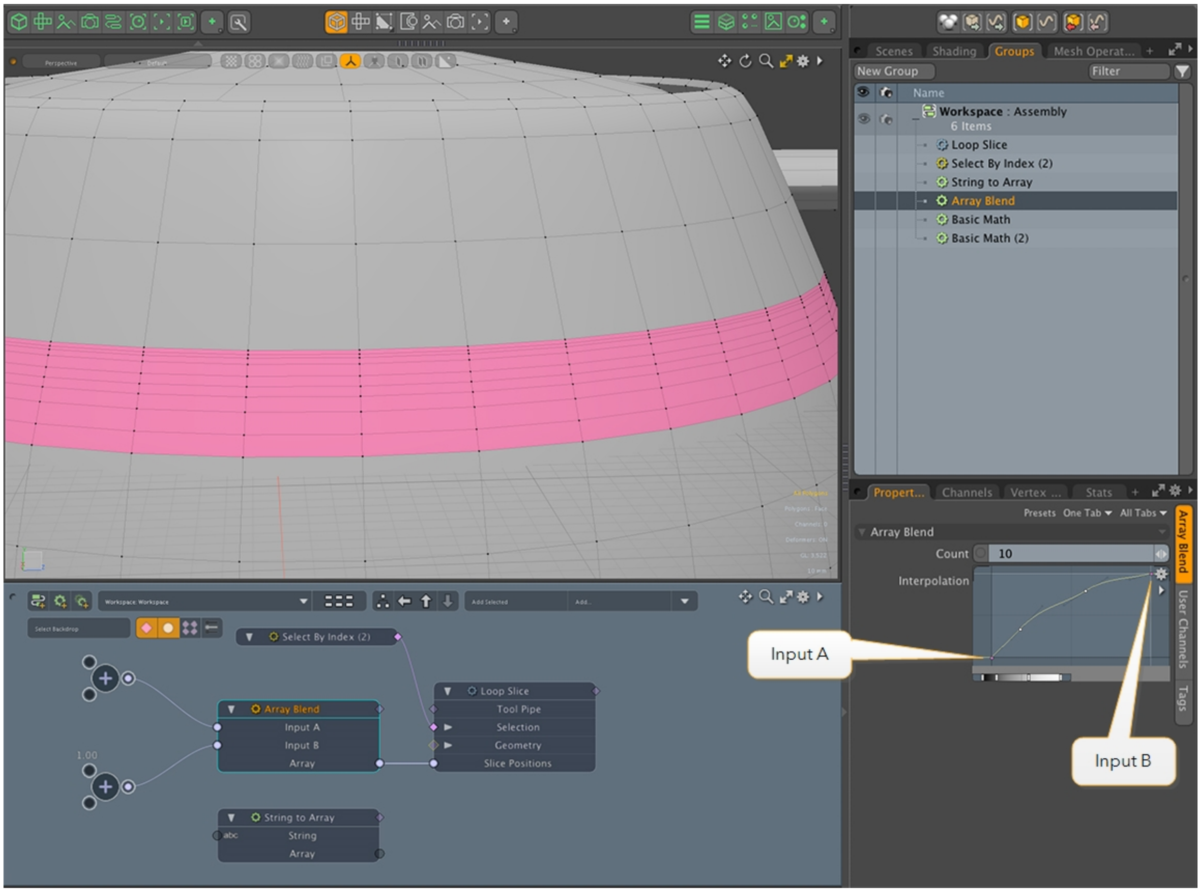
String to Array

String: 0.2 0.4 0.8

Separator: Word

Character:

The **Array Blend** operator sets slice positions by **Count** between **InputA** and **InputB**. The slice positions are interpolated by the gradient curve of the **Array Blend** operator.

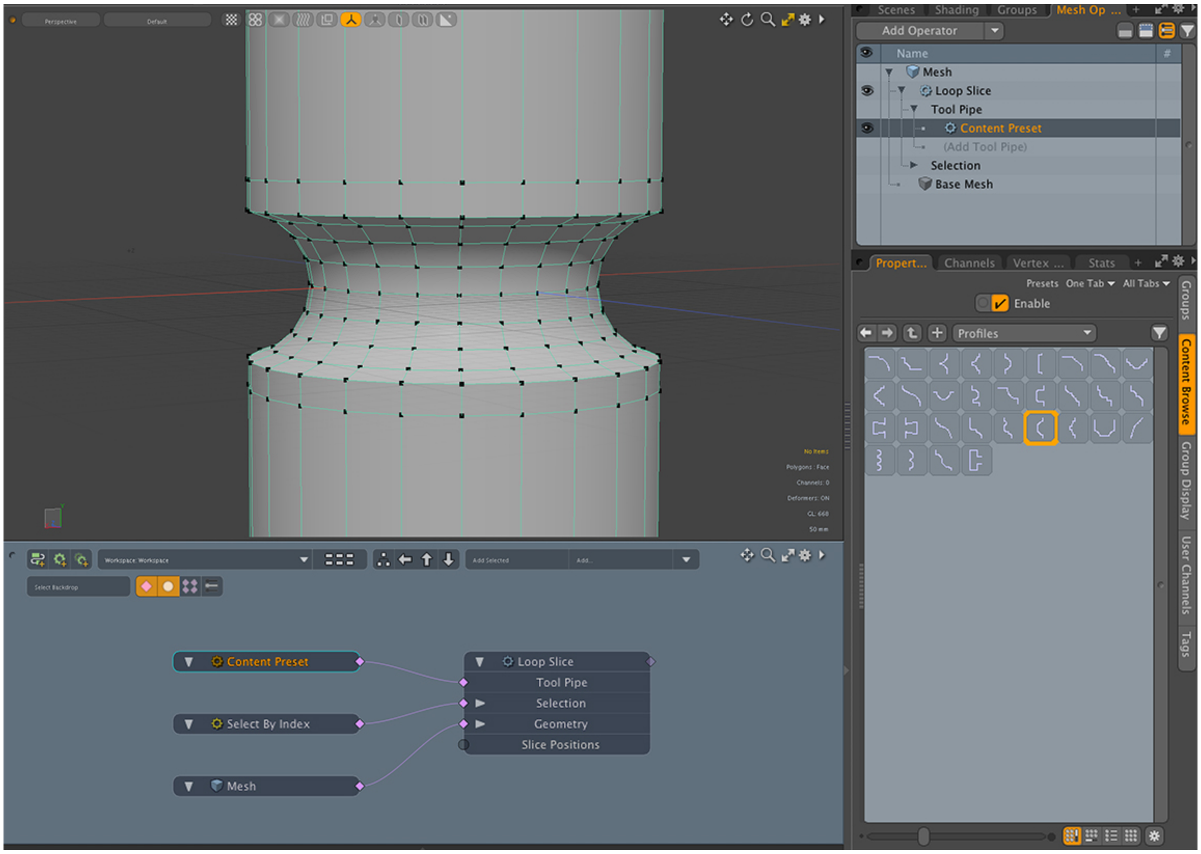


Scenes
Shading
Groups
Mesh Operat...
New Group
Filter
Name
Workspace : Assembly
6 Items
Loop Slice
Select By Index (2)
String to Array
Array Blend
Basic Math
Basic Math (2)
Propert...
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Array Blend
Interpolation
User Channels
Tags
Select By Index (2)
Loop Slice
Tool Pipe
Selection
Geometry
Slice Positions
Array Blend
Input A
Input B
Array
1.00
String to Array
abc
String
Array
Input A
Input B

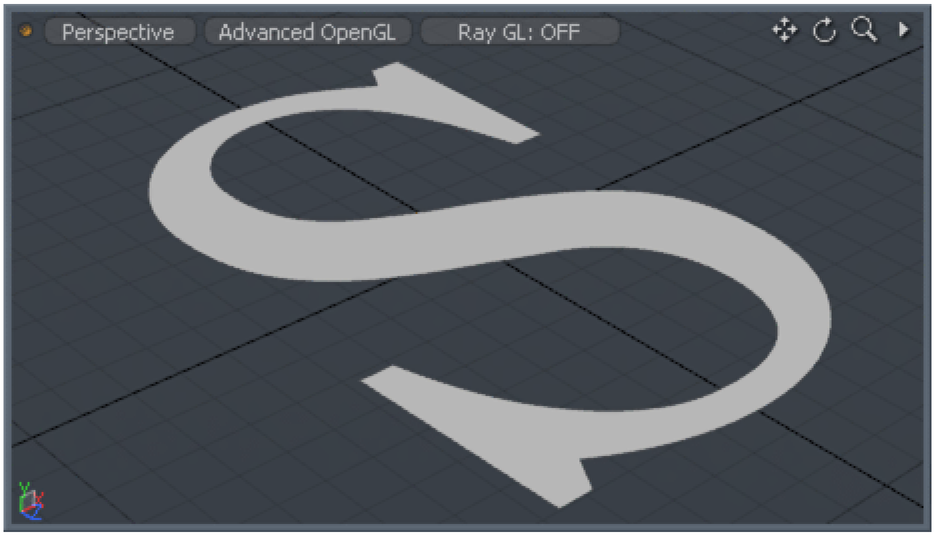
Count 10

Working with 1D Profile Presets

The loop slice supports 1D Profiles from **Content Preset** operator. When the **Content Preset** operator is set to the tool pipe of the **Loop Slice** operator, the Loop Slice tool applies the shape of the selected 1D profile. The example below shows a simple vertical curve applied. Both slice position and depth are set along the 1D profile curve.


Scenes
Shading
Groups
Mesh Op ...
Add Operator
Name
Mesh
Loop Slice
Tool Pipe
Content Preset
(Add Tool Pipe)
Selection
Base Mesh
Propert...
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Enable
Profiles
Groups
Content Browse
Group Display
User Channels
Tags
Content Preset
Select By Index
Mesh
Loop Slice
Tool Pipe
Selection
Geometry
Slice Positions

Axis Slice


Perspective
Advanced OpenGL
Ray GL: OFF

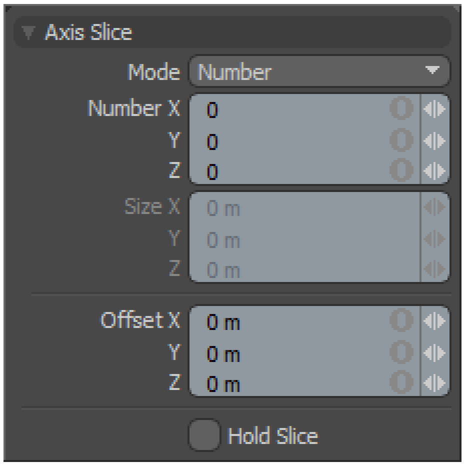
The Axis Slice tool creates one or more evenly spaced slices aligned on the X, Y, or Z axes. The **Number** and **Size** values determine the number of slices and their positions; the **Offset** value determines their positions across the surface. Modo bases the slice spacing on the overall bounding box size of the selected geometry. You can edit the **Offset** value interactively by dragging in the 3D viewport. This tool is useful for quickly dicing up a mesh with uniform slices. This can be valuable when you want to deform a large multi-edged polygon (n-gon). By adding additional geometry to the polygon, Modo can render the deformation as expected.

Usage

The Axis Slice tool adds edges across polygons. Therefore, make sure you are in **Polygons** selection mode and that you select the target polygons to be sliced before activating the tool. You activate the tool by clicking the **Axis Slice** button in the **Mesh Edit** sub-tab of the modeling toolbox. You can also choose **Geometry** > **Slice** > **Axis Slice**. Once you activate the tool, you can adjust the **Number** attribute for each axis in the **Properties** panel to control the number of slices across the selected area. You can adjust the positions of the slices interactively in the 3D viewport by dragging horizontally or vertically.

Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove. You can also use the **Select by Previous Operation** mesh operation. For more information, see Procedural Selection

Axis Slice Properties


Axis Slice
Mode: Number
Number X: 0
Y: 0
Z: 0
Size X: 0 m
Y: 0 m
Z: 0 m
Offset X: 0 m
Y: 0 m
Z: 0 m
☐ Hold Slice

Slice Mode	Sets the style of slicing. Depending on which option you select (**Number** or **Size**),

	Modo activates the relevant options. • **Number** - Spaces the slices evenly within the overall bounding box of the target selection. • **Size** - Determines the placement of the slices based on a defined measurement.
Number X/Y/Z	Sets a specific number of slices along each axis within the bounding box of the selected geometry. This is available only in **Number** mode.
Size X/Y/Z	Sets a specific distance between each slice on the three axes. This is available only in **Size** mode.
Offset X/Y/Z	Sets the center position of the slices. By default, the values are **(0,0,0)**. Changing these values moves the slices on their respective axes. You can adjust these values interactively by dragging in any 3D viewport.
Hold Slice	When enabled, disables the slicing operation temporarily when for you to adjust the values interactively. Enabling this option is useful when you want to adjust the values without a calculation lag when dicing a dense mesh.

Axis Slice in the Procedural

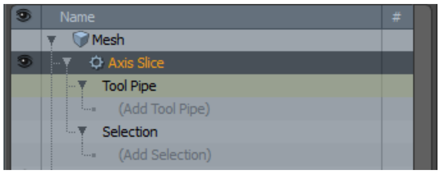
Axis Slice is also available as a procedural mesh operation. You can find the tool in the **Mesh Ops** tab. In the **Model** layout, this tab is on the right side of the interface, next to the **Item List**.

To add the Axis Slice mesh operation:

1. Click the **Add Operator** button, and navigate to **Mesh Operations** > **Edit** > **Axis Slice**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

   Axis Slice is added to the list, and you can adjust its properties in the tool's **Properties** panel in the lower-right of the screen.

You can expand the Axis Slice item in the list by clicking the small arrow in front of it. This reveals the inputs the operation uses:



Axis Slice can use the following inputs:

- **Tool Pipe** - Allows you to add sub-tools to the operation.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.
- **Selection** - Allows you to select the elements to which you want to apply the operation.

Note: The **Geometry** input is only visible when the list is not in Solo mode. For more information, see Solo Mode.

The Axis Slice mesh operation has the same properties as the traditional Axis Slice tool. See Axis Slice Properties.

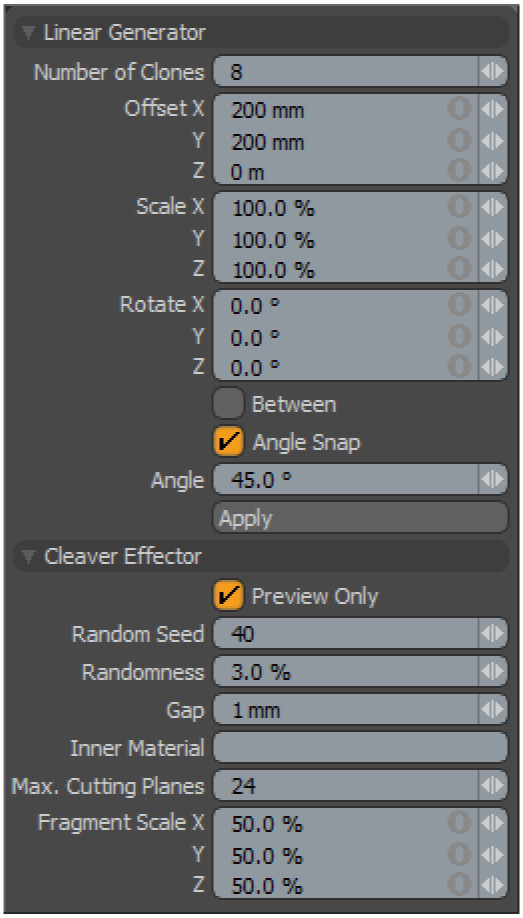
Julienne

The Julienne tool creates multiple evenly-spaced slices that you can position interactively in the 3D viewport with resulting geometry that contains inner faces. This tool is useful for quickly dicing up a mesh into uniform slices. You can specify the number of slices and their positions by providing **Number** and **Offset** values. You can edit the **Offset** value interactively by dragging in the 3D viewport.

Using the Julienne Tool

The Julienne tool is in the modeling toolbox of the **Model** and **Model Quad** interface tabs in the **Mesh Edit** sub-tab. Mainly, you use the Julienne tool interactively. Once you click the tool's icon, drag in the 3D viewport. When you click in the viewport initially, you set the origin for the tool, which is relative to the Work Plane, and dragging determines the offset. By off-handle dragging in the viewport, you interactively adjust the final angle and positions of the slices.

Julienne Tool Properties



Linear Generator
Number of Clones 8
Offset X 200 mm
Y 200 mm
Z 0 m
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Rotate X 0.0 °
Y 0.0 °
Z 0.0 °
☐ Between
☑ Angle Snap
Angle 45.0 °

Cleaver Effector
☑ Preview Only
Random Seed 40
Randomness 3.0 %
Gap 1 mm
Inner Material
Max. Cutting Planes 24
Fragment Scale X 50.0 %
Y 50.0 %
Z 50.0 %

Number of Clones	Determines the total number of possible slices. Depending on the origin and **Offset** positions, not all slices intersect with the target mesh.
Offset X/Y/Z	Determines the distance from the initial origin as an offset. These values either determine the second slice in the arrangement of slices or the final slice when you enable **Between**.
Size X/Y/Z	Determines the scaling of the cutting planes.
Rotate X/Y/Z	Determines the rotations of the cutting planes.
Between	When enabled, spaces the slicing planes evenly between the origin and the **Offset** position. When disabled, the cutting planes extend beyond the **Offset**.
Angle Snap	When enabled, snaps the slices gently when the angle is close to the value

	specified.
Angle	Specifies the angle for which snapping occurs. This only applies when you enable **Angle Snap**.
Cleaver Effector - The Cleaver effector cuts apart geometry by using a series of recursive Boolean operations. When used with an associated generator in the Tool Pipe (such as the Linear generator), Modo ignores the randomness options.	
Preview Only	When enabled, offers a simplified GL preview of the final cuts to speed up the interactivity of the tool. (Modo doesn't actually cut the geometry.) To actually create the cuts, disable this option before dropping the tool.
Random Seed	Determines the initial value for the random number generator. Each seed produces its own unique results.
Randomness	Determines the strength of jitter values applied to the cutting planes when not used with a generator.
Gap	Determines the thickness of the cutting planes. This is the amount of space between surfaces after the cut is complete.
Inner Material	Defines a name that Modo applies as a material tag to the interior cut areas of the mesh.
Max. Cutting Planes	Specifies the maximum number of total cutting planes.
Fragment Scale X/Y/Z	Controls the relative separation of the planes in the various dimensions. When not used with a generator, Modo makes chunks that are long and thin or flat instead of roughly cubic based on these values.

Topology Pen

When creating high-resolution assets with functions such as multi-resolution sculpting, you may create geometry that would be difficult to rig and animate. You could have geometry with too many vertices or with little attention paid to surface topology. These and similar things can lead to odd, unrealistic deformations. The solution for getting a free-form, high-resolution model into a format where it can be easily animated is to retopologize the model. To re-structure geometry, you take a high-resolution mesh and use that as a base to generate a much lower resolution mesh. The lower resolution mesh can have more thoughtfully placed geometry that is easier to weight and rig. From there, you can bake a displacement map to create in a final asset that is visually the same as the free-form geometry, but a lot lighter to provide better deformations and smoother interactivity when animating.

To facilitate retopologizing models, you use the Topology Pen tool to create that lower-resolution mesh. This combined with the Pen tool and Sketch tool set to a **Background Point Constraint** should be all that you need to completely retopologize any high-resolution geometry.

The **Topology Pen** tool is found in the **Model** layout Toolbox, under the **Polygon** sub-tab. It is also found in the **Topology** layout **Tools** sub-tab. The Topology layout is dedicated to the functions of retopologizing models and provides all the related tools and settings in a single convenient viewport.

Note: Previously the Topology Pen tool was known as the Drag Weld tool. When features were added to increase its usability, it was renamed, but the functionality remains.

Using the Tool

Retopology works with a high resolution mesh as a background element. (It is visible in the **Item List**, but it is unselected.) Next, make sure Modo is in Polygons component mode (not **Items mode**). Create a new empty mesh layer by pressing **N**. Select the new mesh layer in the **Item List**. (Rename it if you wish.) Place some initial geometry with either the Pen Tool (possibly using **Quads** mode to lay down strips of quadrangle polygons) or the Topology Sketch Tool (which also lays down a series of quadrangle polygons along a drawn curve that conforms to the background surface). Once you map out the basic contours, you can activate the Topology Pen tool and begin to extend edges and adjust the geometry into the final surface. The Topology Pen tool, depending on settings, offers a broad range of functions. Some functions even duplicate those of other tools to provide a more convenient and fluid workflow. It also enables you to smooth and even out the spacing between vertices on a surface to clean up topology.

By default, the Topology Pen tool also activates a Background Constraint type of **Point** to ensure that all created vertices lie directly on the surface of the background mesh. To use, click the **Topology Pen** icon in the **Topo** interface tab's toolbox. In its **Properties** panel select the proper mode and options, and then manipulate the geometry in the 3D viewport. You can see the geometric element to edit by the pre-highlighting (drawing a ghosted color over the element as the you move the pointer). Dragging applies the defined action, represented by a small yellow box. When you move the box, Modo displays the distance as the **Offset X/Y/Z** values.

Modo provides some shortcuts to use when the tool is in Move mode.

- Click = Move

- Right-click = Move Edge Loop
- Middle-click = Split
- **Shift**+click = Duplicate
- **Shift**+right-click = Duplicate Loop
- **Shift**+middle-click = Add Loop
- **Ctrl**+click = Slide
- **Ctrl**+middle-click = Remove
- **Shift**+ **Ctrl** + left click = Smoothing
- **Shift**+ **Ctrl** + right click = Smoothing + Edge Loop

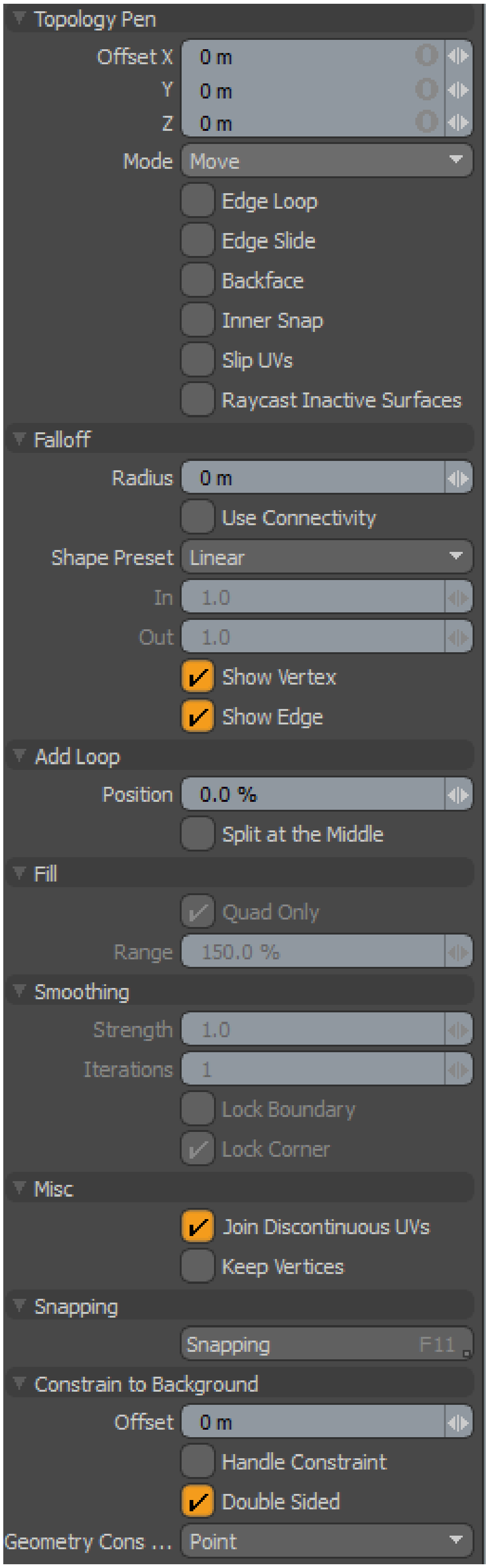
Smoothing

The Topology Pen tool has a smoothing feature which allows you even out the spacing between vertices on a surface to clean up topology. You can focus the smooth operation on a section of your mesh. As you drag, the point/edges around the current location are highlighted to indicate the radius of the falloff and the selected polygons relax into an even distribution.

Smart Snapping

The Topology Pen tool has a custom smart-snapping feature that automatically snaps edges together to weld them into a single vertex. Modo determines the snapping based on the proximity of vertices to each other as you drag an element. Modo indicates snapped edges by a purple color. The welding is not permanent until you release the mouse button; therefore, you can continue to drag and adjust the position of the element (and modifying the snapping) should the wrong vertex snap. You can snap individual vertices together by dragging one vertex onto another.

Topology Pen Properties



Topology Pen

Offset X: 0 m
Y: 0 m
Z: 0 m
Mode: Move
☐ Edge Loop
☐ Edge Slide
☐ Backface
☐ Inner Snap
☐ Slip UVs
☐ Raycast Inactive Surfaces

Falloff

Radius: 0 m
☐ Use Connectivity
Shape Preset: Linear
In: 1.0
Out: 1.0
☑ Show Vertex
☑ Show Edge

Add Loop

Position: 0.0 %
☐ Split at the Middle

Fill

☑ Quad Only
Range: 150.0 %

Smoothing

Strength: 1.0
Iterations: 1
☐ Lock Boundary
☑ Lock Corner

Misc

☑ Join Discontinuous UVs
☐ Keep Vertices

Snapping

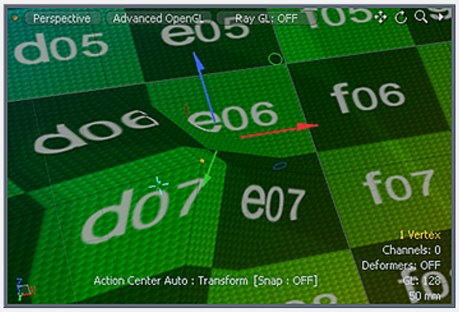
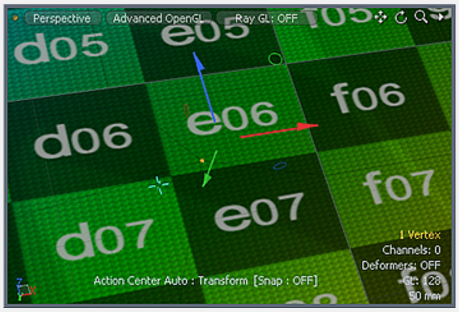
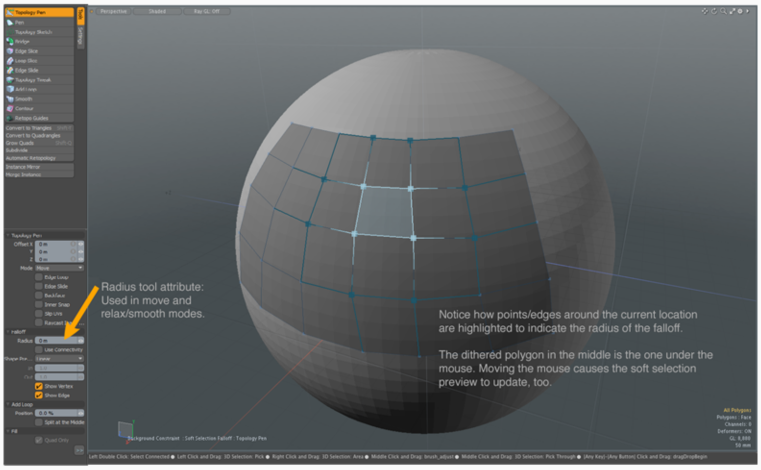
Snapping F11

Constrain to Background

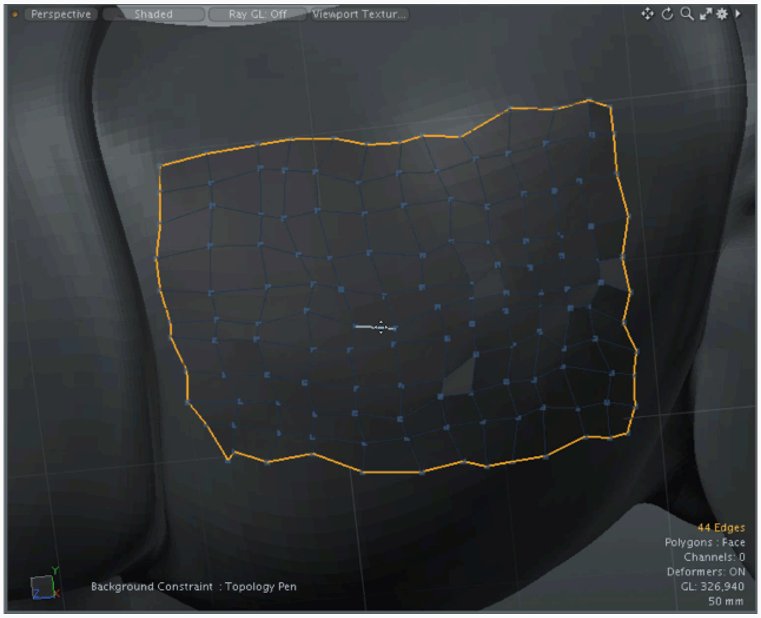
Offset: 0 m
☐ Handle Constraint
☑ Double Sided
Geometry Cons ...: Point

Topology Pen	
Offset X/Y/Z	Specifies the distance the point of reference (a small yellow box) travels as you drag it in the 3D viewport. Once you set the location by clicking the mouse, you may also manually adjust these values directly to adjust the target component.
Mode	Provides several modes that directly relate to the process of retopologizing geometry. • **Move** - Moves an element as you drag it, but it remains fixed against the background surface as it slides around. This is useful in editing vertices, edges, and polygons. • **Duplicate** - Duplicates an edge as you drag it. With **Edge Loop** enabled or by dragging with the right mouse button, you can duplicate an entire edge loop simultaneously. This mode also duplicates individual vertices to make triangle polygons. • **Remove** - Removes the element completely from the geometry. This mode does not modify individual outer edges, but removes interior edges, vertices, and entire polygons. Using the right mouse button removes an entire loop. • **Split** - Creates an edge, which can then be dragged over and snapped to an adjoining vertex, when dragging a vertex. This splits the geometry with a new edge. • **Add Loop** - Create a new edge across the entire span or loop of geometry until another edge or a triangle is encountered. Dragging interactively adjusts the position of the loop — unless you have **Split at the Middle** enabled. • **Point** - Draws a one point polygon when you click the mouse and the pointer is in an empty area. If you click any geometry component in this mode, then the tool works as in the **Move** mode. • **Fill** - Looks for nearby edges and points (vertices) and tries to automatically generate a quad or triangle polygon from them. You can use this mode to fill large areas with a grid of polygons after using the Point mode used to place the vertices. After placing vertices, switch to Fill mode and click in the center of where each quad (or triangle) polygon should appear to create the geometry. Modo pre-highlights the polygon to indicate what geometry it creates when you click. • **Smoothing** - Relaxes vertices by clicking and dragging elements in a falloff

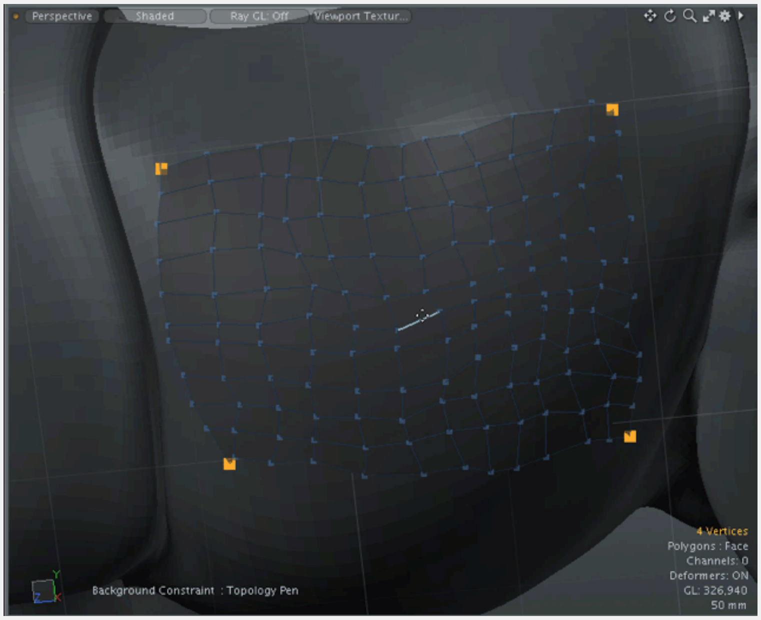
Topology Pen	
	range. You can use this mode to clean up the topology of a mesh to even out the spacing between vertices on a surface, while constraining to the background. When Edge Loop is enabled, the vertices on the edge loop are relaxed along the curve. If the Radius of the Falloff is zero, all vertex positions are relaxed. Strength controls the amount of smoothing applied to the model. Iterations determines the number of times that Modo applies the smoothing. A combination of both Strength and Iterations controls the overall smoothing of the model.
Edge Loop	When enabled, modifies an entire continuous loop of geometry until an opposing edge or triangle is encountered (instead of a single geometric element) when making any adjustments.
Edge Slide	When enabled, restricts the moving of edges and vertices to the positions of adjoining edges. This slides edges and vertices along the edges of neighboring geometry. Pressing **Ctrl** while in Move mode provides the same functionality.
Backface	When enabled, snaps geometric components against the back side of a surface. Normally, when disabled, Modo ignores backward-facing polygons for snapping.
Innersnap	When enabled, snaps any vertex to any other vertex regardless of its location (in the same layer). Normally, when disabled, Modo does not snap interior vertices for a surface (those with no open edges) to other interior faces because snapping only occurs between perimeter vertices.
Slip UVs	When enabled, edits applied to the geometry do not change the existing UV map. UV values are generally fixed to specific vertices; therefore, further edits to the geometry may warp, deform, or otherwise distort the UV values in undesirable ways. When this happens, you may need to adjust the map or to redo it altogether. To avoid this undesirable result, you enable **Slip UVs** so as to not disturb any existing UV mapping applied to the geometry. > >

Topology Pen	
	 **Slip UVs** function disabled (notice the texture warping)  **Slip UVs** function enabled (texture remains even)
Raycast Inactive Surfaces	When enabled, hides certain foreground geometry components behind the background geometry where you cannot select them. This is useful when retopologizing layered elements. For example, when working from a side view with a character that has fingers, several fingers could overlap in the viewport to make it difficult for the Topology Pen tool to discern which edge to indicate for duplicating. With this option enabled, Modo ignores the foreground retopology geometry that is behind the nearest surface and occluded by the background object (the other fingers). In this case, only the retopology surface that is the closest to the viewer is active. This does slow the tool's interactivity, which is why this option is disabled by default.
Falloff - The Falloff distance is very similar to the Selection Falloff tool.	
Radius	Determines the range outside of the selected edge to attenuate the falloff.  Radius tool attribute: Used in move and relax/smooth modes. Notice how points/edges around the current location are highlighted to indicate the radius of the falloff. The dithered polygon in the middle is the one under the mouse. Moving the mouse causes the soft selection preview to update, too.

Topology Pen	
Use Connectivity	When enabled, only affects single surface/connected elements. Unconnected elements within the range are ignored.
Shape Preset	The strength of the falloff's influence can be controlled along the extent using a shape preset. The following options are available: • **Linear** - Attenuation of the falloff occurs evenly across its range. • **Ease-In** - Strength of the falloff is greater toward the start position. • **Ease-Out** - Strength of falloff is greater toward the end position. • **Smooth** - Strength of falloff is greater toward the center of the falloff. • **Custom** - Use the **In/Out** options to fine-tune the strength of the falloff.
In	When **Custom** is set for **Shape Preset**, use this option to set the incoming value to fine-tune the strength of the falloff. The **In** value determines the strength of the falloff nearer to the start position.
Out	When **Custom** is set for **Shape Preset**, use this option to set the outgoing value to fine-tune the strength of the falloff. The **Out** value determines its strength nearer the end of the falloff.
Show Vertex	Used to display the angular points of the polygons on a model. This option is enabled by default.
Show Edge	Used to display the edges of the polygons on a model. This option is enabled by default.
Add Loop	
Position	When enabled, determines the position along an edge where Modo adds the new line of polygon edges. Modo computes the position as a percentage along each edge encountered. After adding the loop by clicking in the 3D viewport, you can directly modify this value by adjusting the resulting position.
Split at the Middle	When enabled, creates new edge loops fixed at the 50% (center) position.
Fill - The Fill options are only available when the Topology Pen tool is in the **Fill** mode.	
Quad Only	When enabled, tries to force only four-sided (quad) polygons.
Range	Determines the search area around the pointer when looking for vertices to convert to polygons.

Topology Pen	
Smoothing	
Strength	Controls the amount of smoothing applied to the mesh.
Iterations	Determines the number of times that Modo applies the smoothing.
Lock Boundary	When enabled, vertex positions on boundary edges are locked at the original positions. This option is disabled by default. Click the image below to view an animation.  Perspective Shaded Ray GL: Off Viewport Textur... 44 Edges Polygons : Face Channels: 0 Deformers: ON GL: 326,940 50 mm Background Constraint : Topology Pen
Lock Corner	When enabled, vertices that are shared with one surface polygon on boundary edges are locked. This option is enabled by default. Click the image below to view an animation.

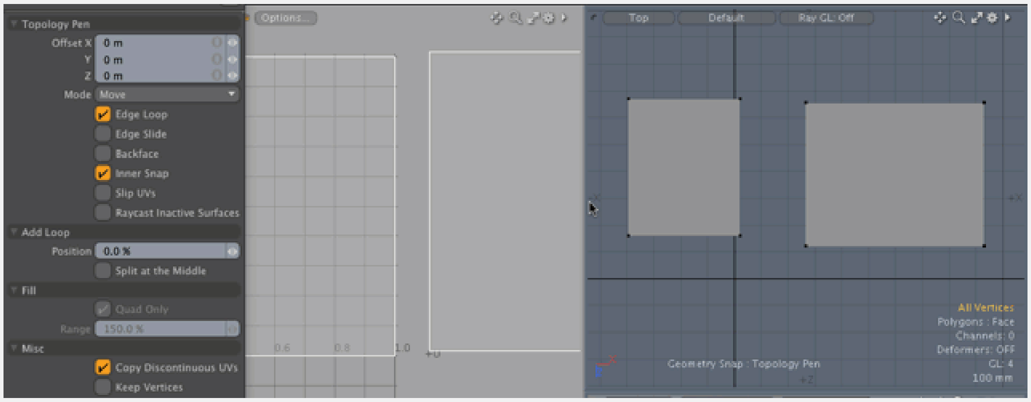
Topology Pen


Perspective
Shaded
Ray GL: Off
Viewport Textur...
4 Vertices
Polygons : Face
Channels: 0
Deformers: ON
GL: 326,940
50 mm
Background Constraint : Topology Pen

Misc

Copy Discontinuous UVs

Specifies how to set discontinuous UVs after welding a 3D vertex pair. When enabled, the discontinuous UV from the source vertex of polygon is used for the new destination vertex, otherwise the UV of the destination vertex is used for the source vertex. This option is enabled by default.

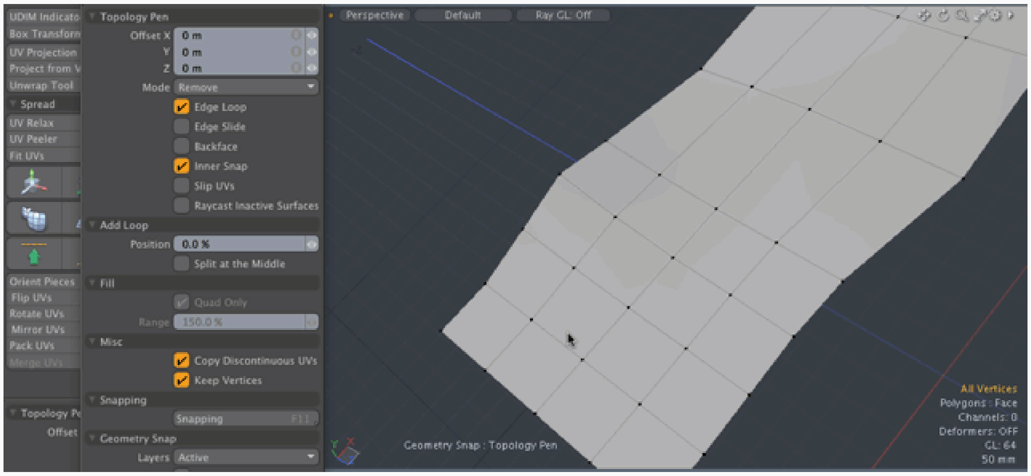

Topology Pen
Offset X 0 m
Y 0 m
Z 0 m
Mode Move
☑ Edge Loop
☐ Edge Slide
☐ Backface
☑ Inner Snap
☐ Slip UVs
☐ Raycast Inactive Surfaces
Add Loop
Position 0.0 %
☐ Split at the Middle
Fill
☑ Quad Only
Range 150.0 %
Misc
☑ Copy Discontinuous UVs
☐ Keep Vertices

Options...
0.6
0.8
1.0
Top
Default
Ray GL: Off
All Vertices
Polygons : Face
Channels: 0
Deformers: OFF
GL: 4
100 mm
Geometry Snap : Topology Pen

Keep Vertices

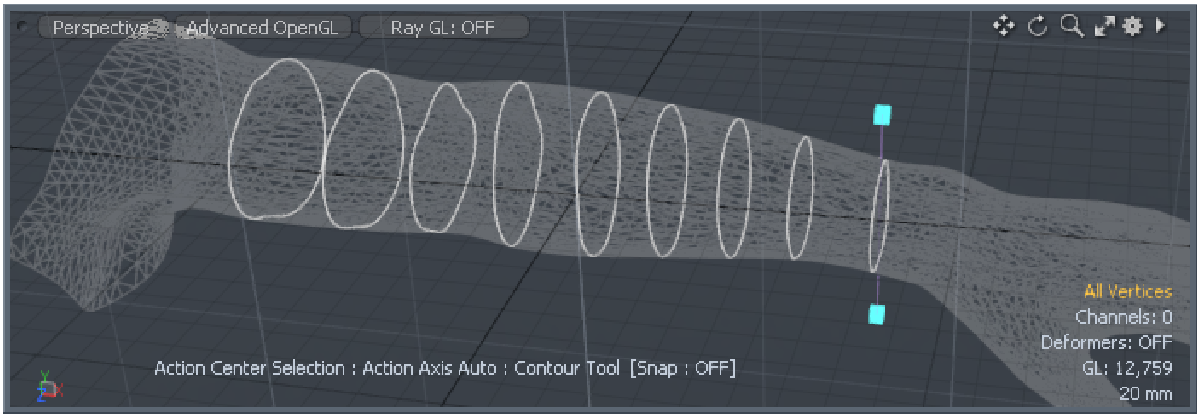
When disabled, the end vertices of the edges are removed by the **Remove** mode operation with the **Edge Loop** option enabled. This option is disabled by default.

Topology Pen


Topology Pen
Offset X 0 m
Y 0 m
Z 0 m
Mode Remove
☑ Edge Loop
☐ Edge Slide
☐ Backface
☑ Inner Snap
☐ Slip UVs
☐ Raycast Inactive Surfaces
Add Loop
Position 0.0 %
☐ Split at the Middle
Fill
☑ Quad Only
Range 150.0 %
Misc
☑ Copy Discontinuous UVs
☑ Keep Vertices
Snapping
Snapping F11
Geometry Snap
Layers Active

UDIM Indicato
Box Transforn
UV Projection
Project from V
Unwrap Tool
Spread
UV Relax
UV Peeler
Fit UVs
Orient Pieces
Flip UVs
Rotate UVs
Mirror UVs
Pack UVs
Merge UVs
Topology Pe
Offset
Perspective
Default
Ray GL: Off
Geometry Snap : Topology Pen
All Vertices
Polygons : Face
Channels: 0
Deformers: OFF
GL: 64
50 mm

Contour


Perspective
Advanced OpenGL
Ray GL: OFF
Action Center Selection : Action Axis Auto : Contour Tool [Snap : OFF]
All Vertices
Channels: 0
Deformers: OFF
GL: 12,759
20 mm

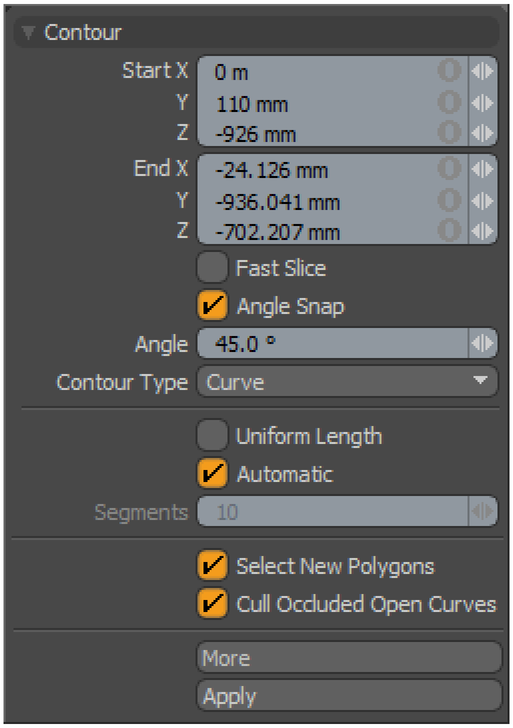
The Contour tool generates contours from user-defined slices across an existing polygon surface. The resulting contours can be curves, polygons, or lines. (In this case, lines are simple strings of two-point polygons.) The contours, themselves, represent the intersection of the slice across the target surface. You can use these contours combined with the Bridge tool for retopologizing a surface.

Usage

The **Contour** tool is found in the **Model** layout **Toolbox** under the **Mesh Edit** sub-tab and in the **Topology** layout **Toolbox** under the **Tools** sub-tab. You can also choose **Geometry** > **Slice** > **Contour**. Prior to activating the tool, you need to have the target surface used to capture the contours from in the background (visible but unselected in the **Item List**). The target mesh for the contours needs to be a selected foreground layer. Clicking the **Contour Tool** icon (or choosing the command) activates the tool. With the tool active, dragging across an existing surface in the viewport generates the contour. The tool's handle is a lavender line with two cyan squares at each end. If you move the pointer over either square, it changes to yellow, and you can drag it to adjust the contour's position. (You can only edit the currently active contour slice.) Pressing **Shift** as you drag creates a new contour slice.

Note: Contour slices project through the target based on the current view angle and orientation. Therefore, when slicing in a perspective view, the resulting contours have perspective as well. Switching to an appropriate orthogonal viewport keeps the contour slices parallel.

Contour Tool Properties



Contour

Start X: 0 m
Y: 110 mm
Z: -926 mm
End X: -24.126 mm
Y: -936.041 mm
Z: -702.207 mm
☐ Fast Slice
☑ Angle Snap
Angle: 45.0 °
Contour Type: Curve
☐ Uniform Length
☑ Automatic
Segments: 10
☑ Select New Polygons
☑ Cull Occluded Open Curves

Start X/Y/Z	Specifies the beginning of the slicing handle that creates the contour. Generally, these display the current start position when you interactively draw in the 3D viewport, but you can also enter specific values.
End X/Y/Z	Specifies the end of the slicing handle that creates the contour. Generally, these

	display the current end position when you interactively draw in the 3D viewport, but you can also enter specific values.
Fast Slice	When enabled, creates the contour after you release the mouse button to increase the tool's performance. By default, Modo creates contours interactively as you draw the slice in the 3D viewport. On dense meshes this may slow the tool down.
Angle Snap	When enabled, snaps the contour slice handle to a user-defined increment whenever the drawn line is close to the defined **Angle** value.
Angle	Determines the snapping angle when you enable **Angle Snap**.
Contour Type	Determines the type of contour generated. • **Lines** - Generates a chain of two-point polygons (lines). You can select these lines in either Edges or Polygons mode. • **Curve** - Generates a standard curve that closely represents the target surface. • **Face** - Produces a closed polygon face with edges that represent the target surface.
Uniform Length	When enabled, distributes the resulting vertex positions of the generated contour evenly along its length.
Automatic	When enabled, determines vertex positions by the contour slices' intersection with a polygon edge along the target surface. When disabled, Modo determines the vertex positions by the **Segments** value.
Segments	Determines the number of vertices along the contour for the selected contour type when you disable **Automatic**. By providing a value, you specify how to distribute the vertices.
Select New Polygons	When enabled, selects automatically any new geometry that Modo generates. When you deactivate the Contour tool, Modo retains the selection as you activate another tool (such as the Bridge) to connect the resulting curves.
Cull Occluded Open Curves	When enabled, eliminates automatically any open or incomplete contours for areas of geometry that face away from you (from the current viewport's perspective). For example, if Modo generated a contour for a cylinder but the contour didn't cross the entire shape, Modo generates a short curve to represent the front of the shape and another curve on the reverse side. Enabling **Cull Occluded Open Curve** eliminates the curve generated on the opposing face.
More	Generates automatically a new contour with an offset based on the two previous

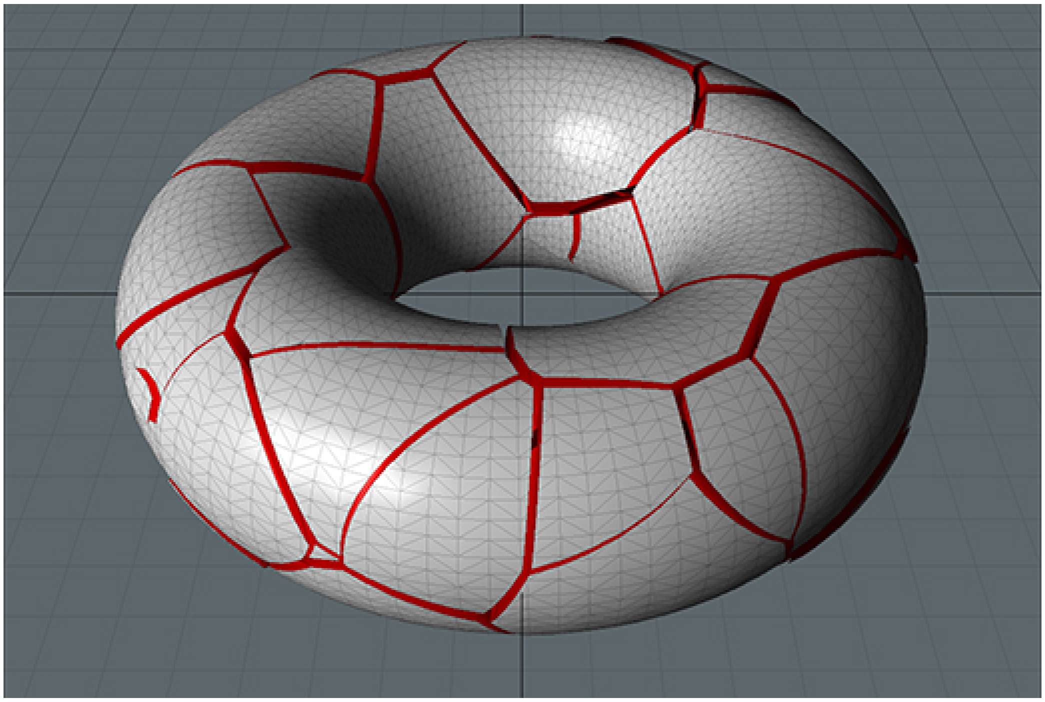
	contours. By clicking **More**, you can produce a set of evenly-spaced contours after drawing the first two.

Shatter

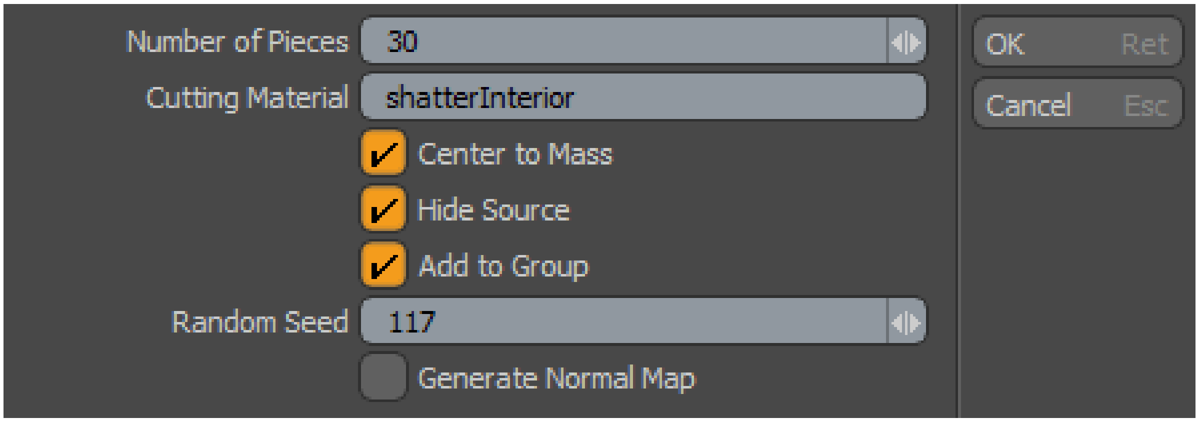
The Shatter command breaks a model into a specified number of individual pieces, which can then be used with Modo dynamics to produce animated mayhem. The Shatter command performs a Veronoi-type of fragmentation within the bounding box of the target selection to generate cutting shards that are necessary for the Boolean operations that form the actual pieces. You can use any arbitrary shape for the target geometry, but it must be completely water-tight (with no open faces, floating vertices, or floating curves). The geometry generated by the Shatter command is not specifically SubD-compatible geometry, so these meshes should be frozen to an appropriate level prior to running the command. For information on freezing meshes, see Freezing Geometry.

Each individual shard becomes its own item layer (with an option for grouping pieces) with an independent center point. Once generated, you can apply dynamics at once to all selected elements. You can also glue them together to create groupings that must reach a certain force threshold before breaking apart. There are several different types of shattering that you can apply. Each has its own options that influence the look of the resulting shards. The Shatter commands are in the **Geometry > Mesh Shatter** submenu. You must select the target geometry to break apart prior to invoking the command.

Uniform



By choosing **Geometry** > **Mesh Shatter** > **Uniform**, you have Modo generate pieces that are roughly uniform in size for the Veronoi shards that make up the Boolean operation. The resulting Boolean shards may not be strictly uniform depending on how the pieces intersect with the target geometry during the Boolean stage.



Number of Pieces: 30

Cutting Material: shatterInterior

☑ Center to Mass

☑ Hide Source

☑ Add to Group

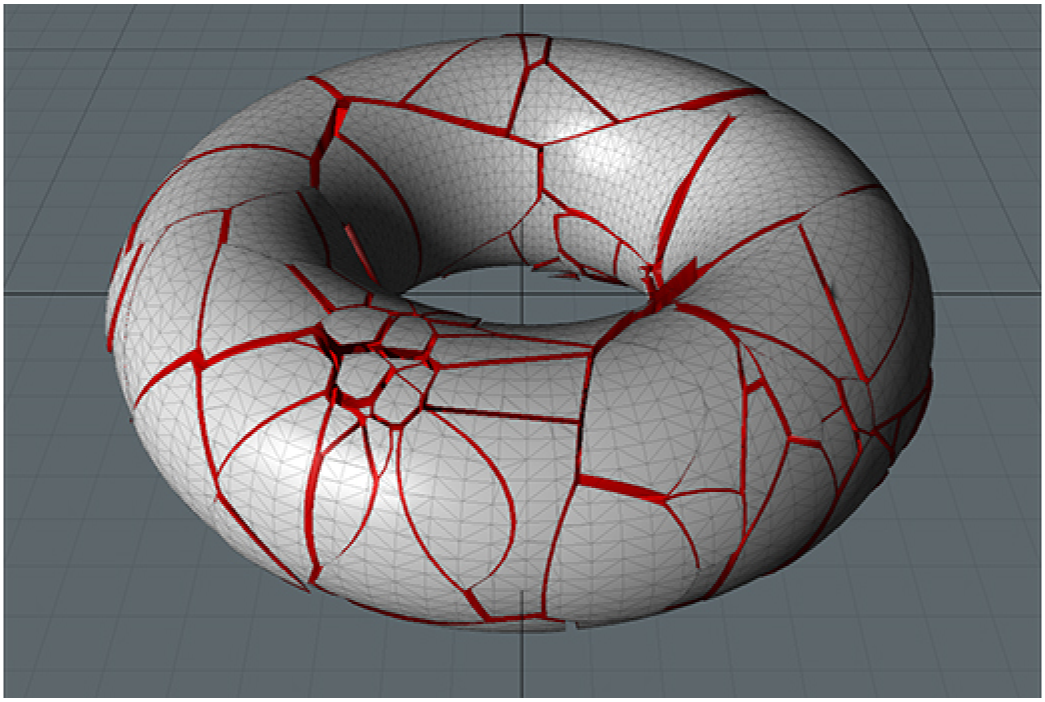
Random Seed: 117

☐ Generate Normal Map

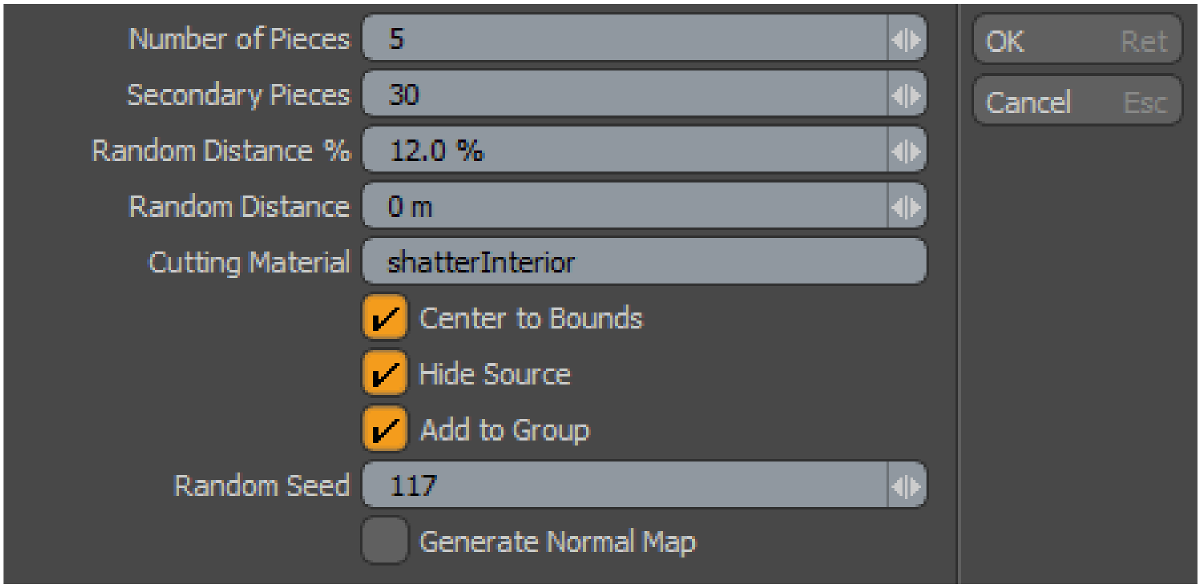
The following Shatter Uniform options are available:

Shatter Uniform	
Number of Pieces	Determines the number of Veronoi shards generated for the Boolean cutting operations. Depending on the shape of the target geometry, this value roughly determines the final number of separate items generated.
Cutting Material	Defines an interior surface for the resulting shards (colored red by default). If you leave this blank, Modo tags the interior surfaces with the **Default** material tag.
Center to Mass	When enabled, places the center position of each resulting shard item on its overall mass. This is a requirement for using the shards with dynamics and produces the most realistic results for its movements.
Hide Source	When enabled, disables the original Mesh Item layer in the **Items** list to make it invisible (unless its **Render Control** option is **On**). Modo always preserves the original Mesh Item from which the shards originate.
Add to Group	When enabled, parents all the resulting shard items under a single group locator (folder) item. When disabled, the items are added as new Mesh Item layers to the **Items** list.
Random Seed	Specifies the initial value Modo uses when generating the random Veronoi values. Each seed produces different, random results.
Generate Normal Map	When enabled, creates a vertex normal map prior to cutting the shards to freeze their shading normals. When disabled, shading normals are influenced by the shard edges and the material's **Smoothing Angle**.

Cluster



The **Cluster** command is similar to the **Uniform** command. By choosing **Geometry > Mesh Shatter > Cluster**, you have Modo generate additional, smaller shards around the initial pieces to create a more randomized look to the final shards.

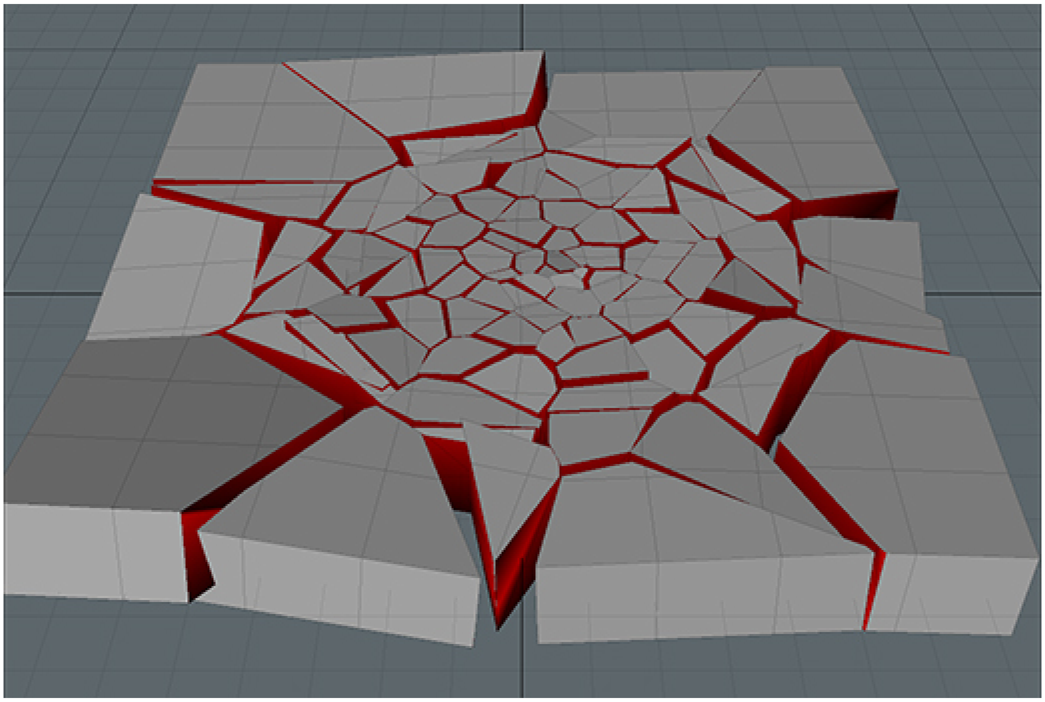


Number of Pieces: 5
Secondary Pieces: 30
Random Distance %: 12.0 %
Random Distance: 0 m
Cutting Material: shatterInterior
☑ Center to Bounds
☑ Hide Source
☑ Add to Group
Random Seed: 117
☐ Generate Normal Map

The following Shatter Cluster options are available:

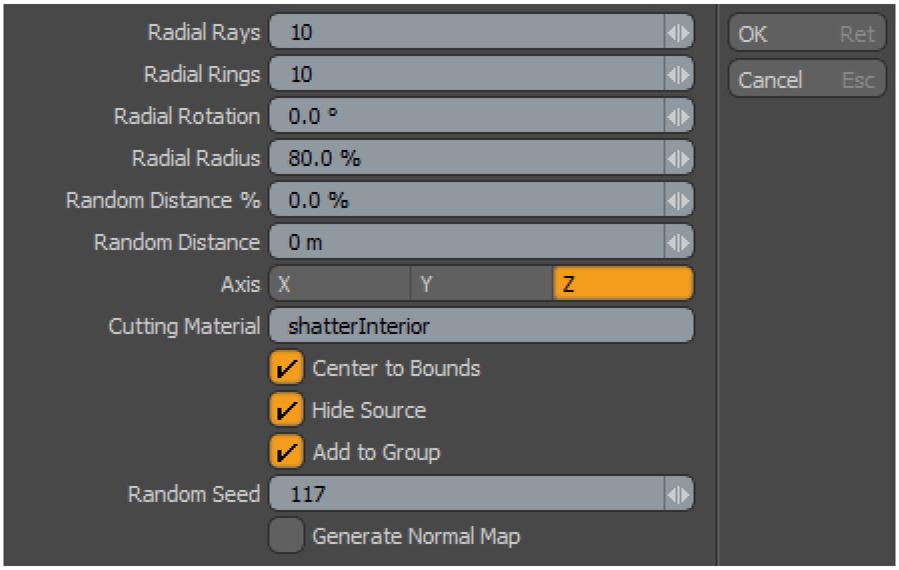
Shatter Uniform	
Number of Pieces	Determines the number of Veronoi shards generated. This is similar to the option with the Uniform command.
Secondary Pieces	Determines the number of secondary shards generated around the initial shards.
Random Distance %	Specifies the distance offset that Modo uses to create the secondary shards from the initial shards. This value is a percentage relative to the size of the original piece.
Random Distance	Specifies the distance form the initial pieces for the secondary pieces as an absolute World value.
Cutting Material	Defines an interior surface for the resulting shards (colored red by default). If you leave this blank, Modo tags the interior surfaces with the **Default** material tag.
Center to Bounds	When enabled, places the center position of each resulting shard item on the bounding box volume's center. You must enable this to use the shards with dynamics.
Hide Source	When enabled, disables the original Mesh Item layer in the **Items** list to make it invisible (unless its **Render Control** option is **On**). Modo always preserves the original Mesh Item from which the shards originate.
Add to Group	When enabled, parents all the resulting shard items under a single group locator (folder) item. When disabled, the items are added as new Mesh Item layers to the **Items** list.
Random Seed	Specifies the initial value Modo uses when generating the random Veronoi values. Each seed produces different, random results.
Generate Normal Map	When enabled, creates a vertex normal map prior to cutting the shards to freeze their shading normals. When disabled, shading normals are influenced by the shard edges and the material's **Smoothing Angle**.

Radial



Choose **Geometry** > **Mesh Shatter** > **Radial** to have Modo generate a radial graph-type of look to the resulting shards similar to a spider's web. Randomized shard values look like broken glass from a central strike point with smaller pieces in the center and the size increasing as the pieces move outward.

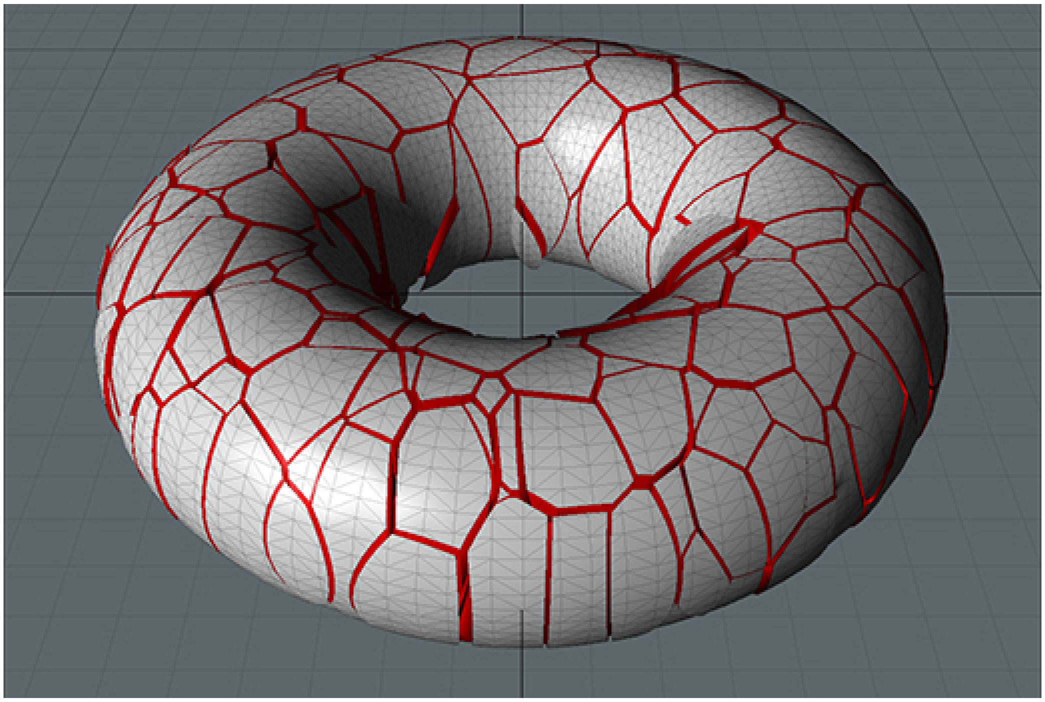
The following Shatter Radial options are available:



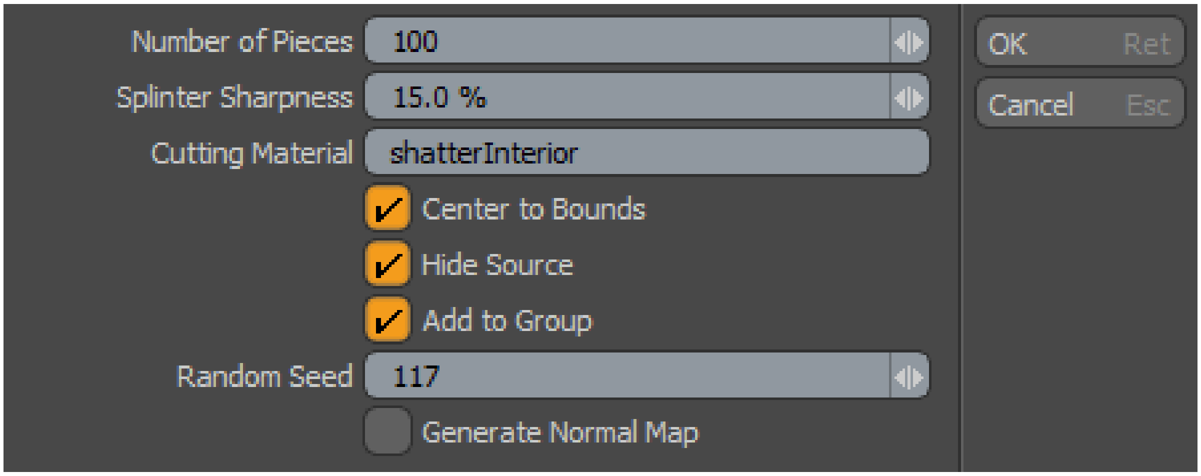
Radial Rays 10
Radial Rings 10
Radial Rotation 0.0 °
Radial Radius 80.0 %
Random Distance % 0.0 %
Random Distance 0 m
Axis X Y Z
Cutting Material shatterInterior
☑ Center to Bounds
☑ Hide Source
☑ Add to Group
Random Seed 117
☐ Generate Normal Map

Shatter Radial	
Radial Rays	Determines the number of radial divisions. Modo calculates the total ray length value by the selected form's overall bounding box size.
Radial Rings	Determines the number of divisions along the rays length – similar to rings on a tree.
Radial Rotation	Spins the radial rays to control where they land on the target geometry.
Radial Radius	Determines size of radial rings along the length of the radial rays. For example, with a setting of **50%**, Modo positions the final radial ring at 50% of the total radial ray length to concentrate the shards toward the center of the form.
Random Distance %	Specifies an offset of the initial cut location based on a percentage of the final shard's size.
Random Distance	Specifies an amount of random offset from the initial shard position as an absolute World value.
Axis	Specifies the axis plane that the radial cuts align along.
Cutting Material	Defines an interior surface for the resulting shards (colored red by default). If you leave this blank, Modo tags the interior surfaces with the **Default** material tag.
Center to Bounds	When enabled, places the center position of each resulting shard item on the bounding box volume's center. You must enable this to use the shards with dynamics.
Hide Source	When enabled, disables the original Mesh Item layer in the **Items** list to make it invisible (unless its **Render Control** option is **On**). Modo always preserves the original Mesh Item from which the shards originate.
Add to Group	When enabled, parents all the resulting shard items under a single group locator (folder) item. When disabled, the items are added as new Mesh Item layers to the **Items** list.
Random Seed	Specifies the initial value Modo uses when generating the random Veronoi values. Each seed produces different, random results.
Generate Normal Map	When enabled, creates a vertex normal map prior to cutting the shards to freeze their shading normals. When disabled, shading normals are influenced by the shard edges and the material's **Smoothing Angle**.

Splinter



Choose **Geometry** > **Mesh Shatter** > **Splinter** to have Modo generate long shards that have jagged interiors, which is similar to how wood breaks. The Splinter command is similar to the Uniform command, but it generates longer shards.

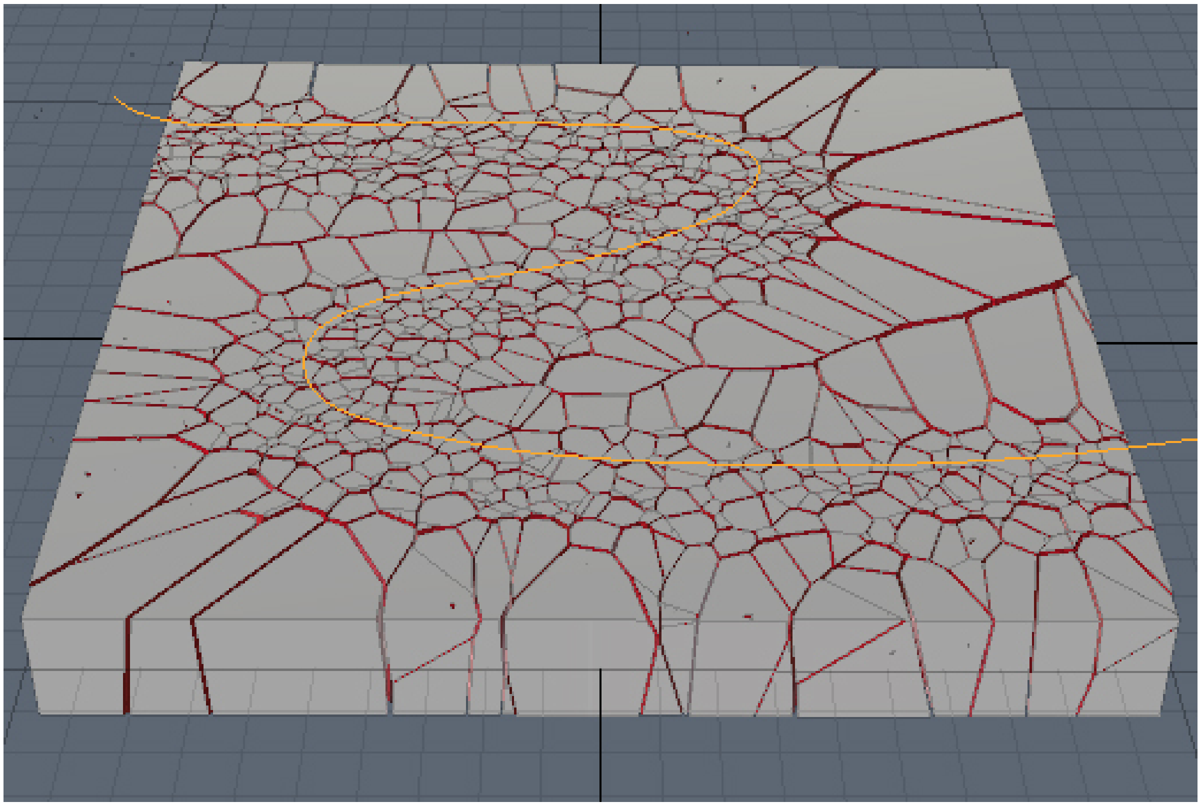


Number of Pieces: 100
Splinter Sharpness: 15.0 %
Cutting Material: shatterInterior
☑ Center to Bounds
☑ Hide Source
☑ Add to Group
Random Seed: 117
☐ Generate Normal Map

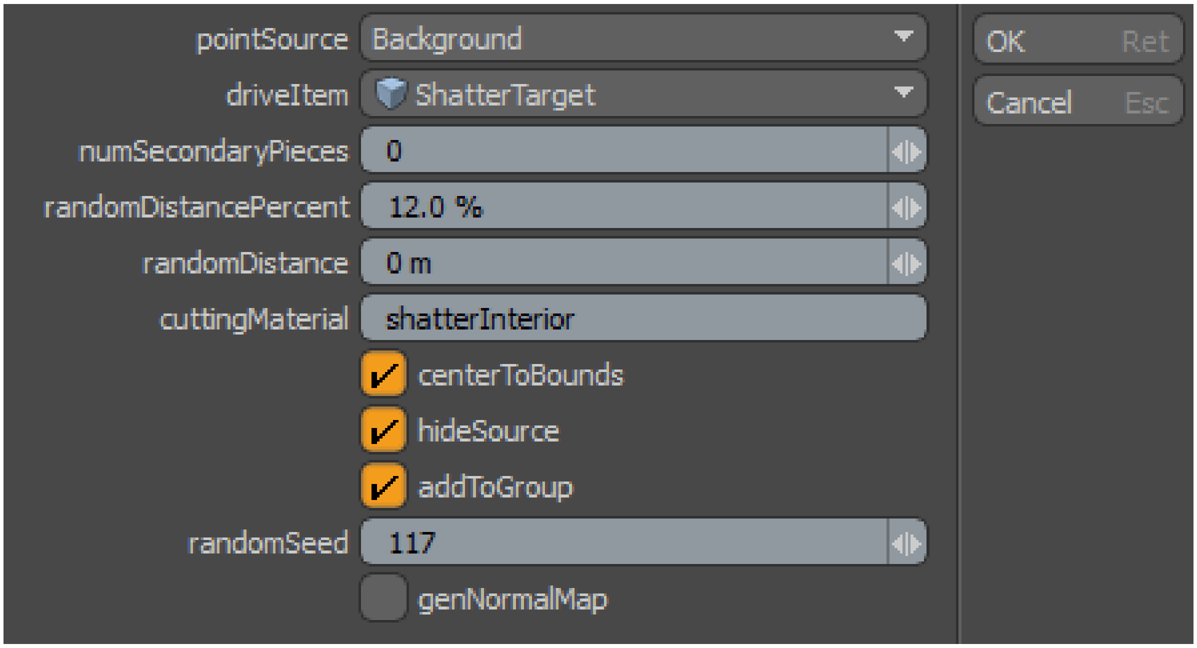
The following Splinter Uniform options are available:

Splinter Uniform	
Number of Pieces	Determines the number of Veronoi shards generated for the Boolean cutting operations. Depending on the shape of the target geometry, this value roughly determine the final number of separate items that Modo generates.
Splinter Sharpness	Determines the amount of jaggedness at the center cut line.
Cutting Material	Defines an interior surface for the resulting shards (colored red by default). If you leave this blank, Modo tags the interior surfaces with the **Default** material tag.
Center to Bounds	When enabled, places the center position of each resulting shard item on the bounding box volume's center. You must enable this to use the shards with dynamics.
Hide Source	When enabled, disables the original Mesh Item layer in the **Items** list to make it invisible (unless its **Render Control** option is **On**). Modo always preserves the original Mesh Item from which the shards originate.
Add to Group	When enabled, parents all the resulting shard items under a single group locator (folder) item. When disabled, the items are added as new Mesh Item layers to the **Items** list.
Random Seed	Specifies the initial value Modo uses when generating the random Veronoi values. Each seed produces different, random results.
Generate Normal Map	When enabled, creates a vertex normal map prior to cutting the shards to freeze their shading normals. When disabled, shading normals are influenced by the shard edges and the material's **Smoothing Angle**.

Source



The **Source** shatter option allows you to use Mesh Items to drive the cutting of the target mesh. Curves or point sources can be used as well, with the positions of the resulting shards placed relative to the vertices of the driving geometry. To add it, choose **Geometry** > **Mesh Shatter** > **Source**.



pointSource: Background

driveItem: ShatterTarget

numSecondaryPieces: 0

randomDistancePercent: 12.0 %

randomDistance: 0 m

cuttingMaterial: shatterInterior

☑ centerToBounds

☑ hideSource

☑ addToGroup

randomSeed: 117

☐ genNormalMap

The following Source Shatter options are available:

Source Shatter	
Point Source	The **Point Source** option defines what particular mesh is used as the point source; where the vertex positions within the layer defines the relative locations of the generated shards. **Background** uses any visible, but inactive mesh layers as the driving item(s), **Last Selected** uses any geometry that was actively selected in the current layer when the command was invoked as the driving item, and **Specific Item** uses the mesh layer defined in the **Drive Item** input field.
Drive Item	When the **Specific Item** option is selected in the **Point Source** field, this input defines the item that is used as the driver for the resulting shards.
Secondary Pieces	This value determines the number of secondary shards generated around the initial shards.
Random Distance %	This value is an offset of the initial cut location based on a percentage of the final shards size.
Random Distance	This value is an absolute World value that specifies an amount of random offset from the initial shard position.
Cutting Material	Defines an interior surface for the resulting shards (colored red by default). If you leave this blank, Modo tags the interior surfaces with the **Default** material tag.

Source Shatter	
Center to Bounds	When enabled, places the center position of each resulting shard item on the bounding box volume's center. You must enable this to use the shards with dynamics.
Hide Source	When enabled, disables the original Mesh Item layer in the **Items** list to make it invisible (unless its **Render Control** option is **On**). Modo always preserves the original Mesh Item from which the shards originate.
Add to Group	When enabled, parents all the resulting shard items under a single group locator (folder) item. When disabled, the items are added as new Mesh Item layers to the **Items** list.
Random Seed	Specifies the initial value Modo uses when generating the random Veronoi values. Each seed produces different, random results.
Generate Normal Map	When enabled, creates a vertex normal map prior to cutting the shards to freeze their shading normals. When disabled, shading normals are influenced by the shard edges and the material's **Smoothing Angle**.

Edit Tools

These topics provide information on Modo's geometry edit tools.

Edit Commands - Cut/Copy/Paste

The Cut, Copy, and Paste commands work together to move geometry from one layer or scene to another or to duplicate geometry within the same layer. These commands work by holding the cut or copied geometry in an internal Modo-specific buffer called the Clipboard. The Cut and Copy commands place data into the Clipboard, and the Paste command pulls it out and places it in the active layer.

Usage

The Cut and Copy commands take the current selection (determined by the current selection mode) and move it to the Clipboard until you exit Modo or you perform another Cut or Copy operation and overwrite the contents of the Clipboard. (There is only one Clipboard that Modo uses.) The Paste operation places the contents of the Clipboard into the current mesh layer regardless of the selection mode. For example, if you cut selected vertices to the Clipboard, Modo doesn't magically convert them to polygons if you paste them into a layer while in Polygons mode.

Note: Although if you select edges, cut or copy them, and then paste them, Modo pastes the edges as a string of polylines (two-point polygons) and not true edges. (Edges cannot exist by themselves without a polygon.) You can select these polylines in both Edges and Polygons modes.

To move geometry from multiple layers into a single layer:

1. In the **Item List**, press **Ctrl** as you select multiple Mesh Item layers by clicking each layer's name.
2. In the 3D viewport, paint around or marquee select the target geometry.
3. Choose **Edit** > **Cut** or press **Ctrl**+**X** (or **Cmd**+**X**) on the keyboard to cut the selection.
   The selected geometry moves from the scene into the Clipboard.
4. To place this data into another layer or scene, select the target Mesh Item layer in the appropriate scene and choose **Edit** > **Paste** or press **Ctrl**+**V** (or **Cmd**+**V**).
   If you retain the original selection of item layers, Modo inserts all geometry into the topmost selected layer.

To duplicate geometry in the same layer:

1. In the 3D viewport, select the component elements to duplicate with either the paint or marquee selection tool.
2. Choose **Edit** > **Copy** or press press **Ctrl**+**C** (or **Cmd**+**C**) on the keyboard to copy the selection into the Clipboard.
3. Activate one of the transform tools (the Move, Rotate, or Scale tool) and transform the original selection as needed. (You can also press the **W**, **E**, or **R** key respectively on the keyboard.)
4. Choose **Edit** > **Paste** or press **Ctrl**+**V** (or **Cmd**+**V**) to paste the selection back into its initial position or orientation.
5. Repeat steps 2-4 to make subsequent duplicates.

Edit Commands - Delete/Remove

Modo has two functions for discarding selected elements from a model or scene. The Delete command will delete a selected item. The Remove command removes the selected polygons or edges and leaves the vertices behind. This is useful when you want to use the vertices as a profile that you may want to extrude.

Delete Command

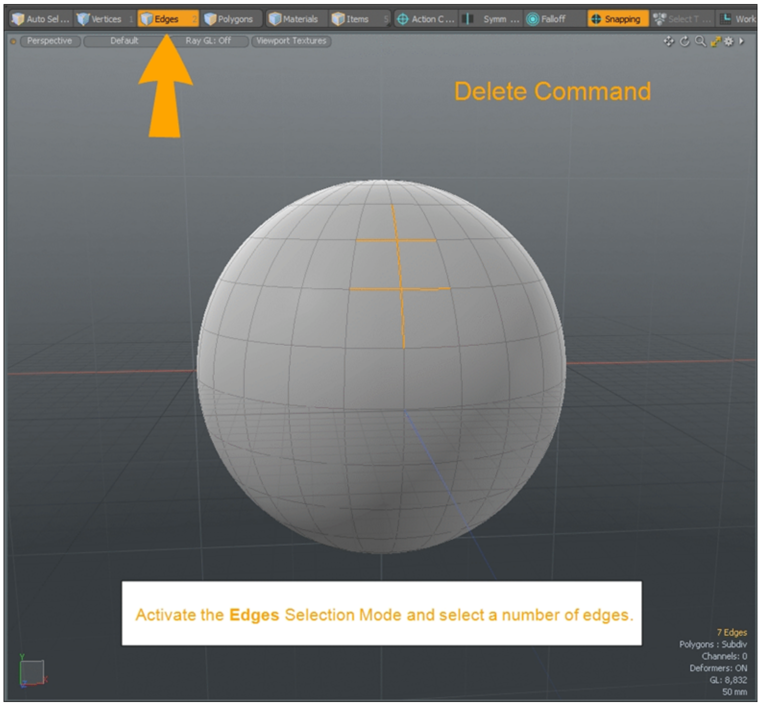
You can choose **Edit** > **Delete** or press the **Delete** or the **Backspace** key on the keyboard to delete selected polygons. This command universally discards any selected item — whether the selected item is in the **Shader Tree**, **Item List**, **Graph Editor**, or in the **3D Viewport**. When deleting a selection, Modo eliminates the entire selection. For example, with a selected polygon, Modo also deletes the associated edges and vertices, but it doesn't delete the components of surrounding unselected geometry. Deleting an edge also deletes the associated polygons that relied on the edge.

Remove Command

You can choose **Geometry** > **Remove**, or press **Shift**+**Backspace** on the keyboard to remove the selected polygons or edges. Invoking this command discards component selections only. When applied, the Remove command acts similarly to Delete, with some slight differences in the case of polygons and edges.

The Remove command only deletes the selected polygons, keeping vertices available. Removing an edge leaves the geometry intact and only removes the edge itself.

Click the image below to view an animation, which demonstrates a workflow.

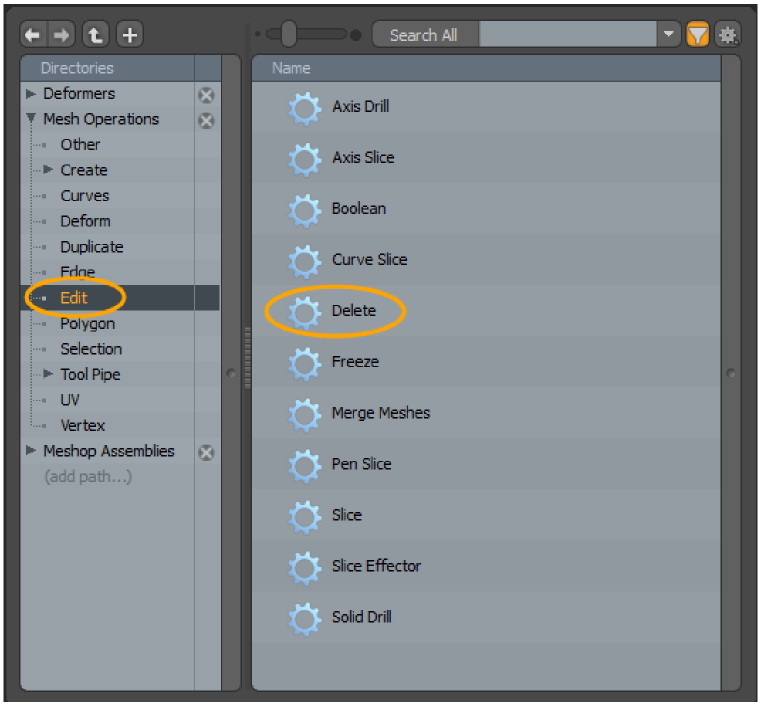

Auto Sel ...
Vertices
1
Edges
2
Polygons
Materials
Items
5
Action C ...
Symm ...
Falloff
Snapping
Select T ...
Work
Perspective
Default
Ray GL: Off
Viewport Textures
7 Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
50 mm

Delete Command

Activate the **Edges** Selection Mode and select a number of edges.

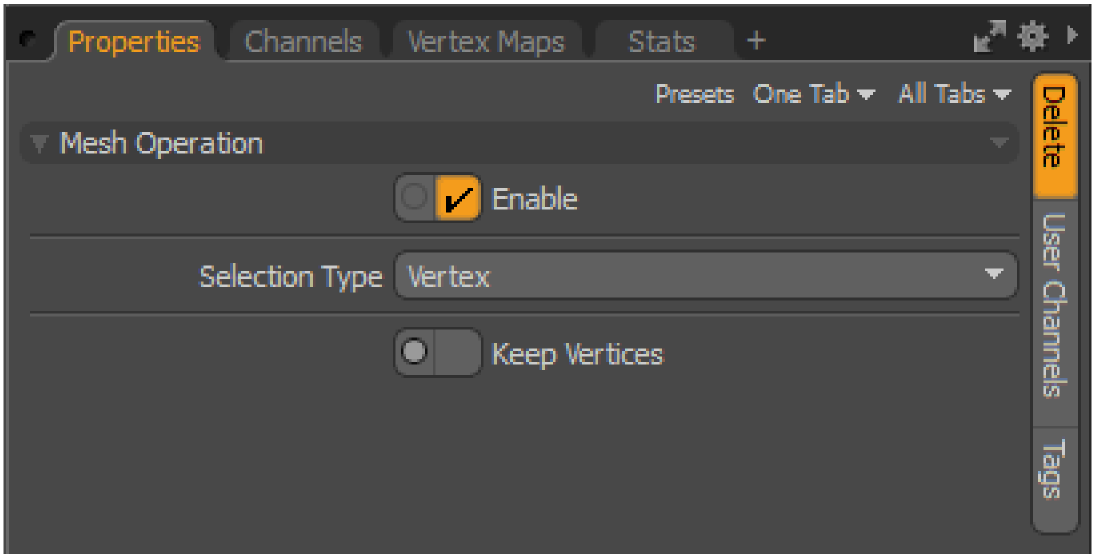
Delete in Procedural Modeling

When working with Modo's procedural modeling tools, you can use the procedural **Delete** operation to delete vertices, polygons, or edges. Select the item in the **Mesh Ops** lists that you want to delete from your scene, click **Add Operator** > **Edit** > **Delete** and specify the properties. As a result, only the selected vertices, edges, or polygons are displayed in your scene.


Search All
Directories
Deformers
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Edit
Polygon
Selection
Tool Pipe
UV
Vertex
Meshop Assemblies
(add path...)
Name
Axis Drill
Axis Slice
Boolean
Curve Slice
Delete
Freeze
Merge Meshes
Pen Slice
Slice
Slice Effector
Solid Drill

To view your original item, turn off the **Delete** Enable option.

In the **Delete Properties** tab, on the bottom of the right panel, the following options are available:

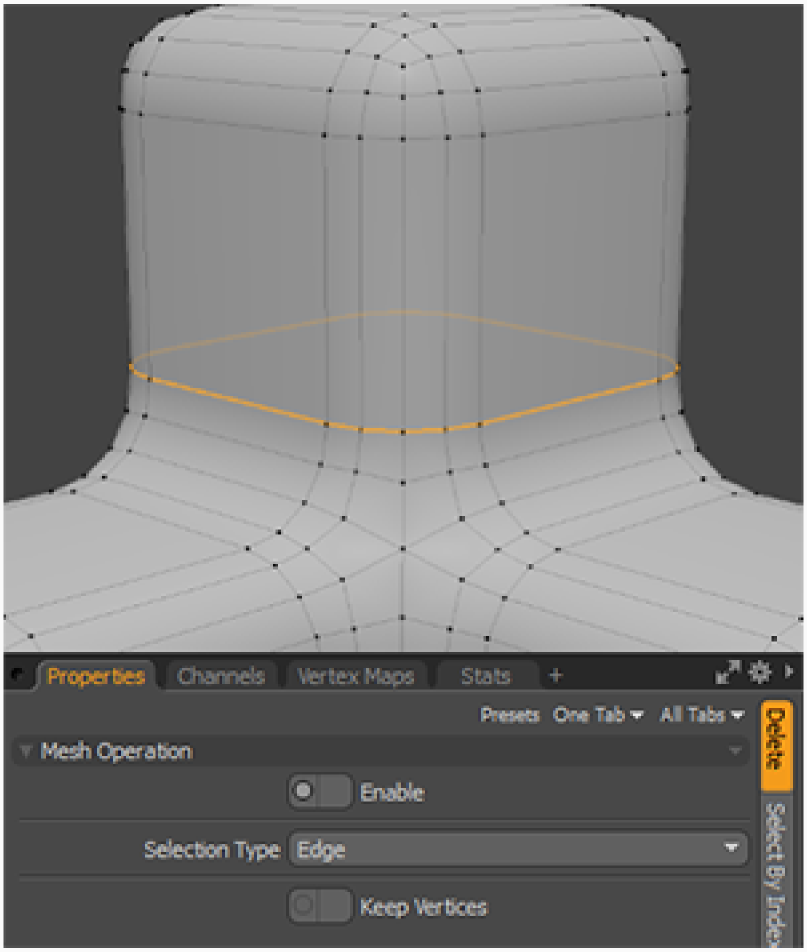

Properties
Channels
Vertex Maps
Stats
+
Presets
One Tab
All Tabs
Mesh Operation
Delete
User Channels
Tags

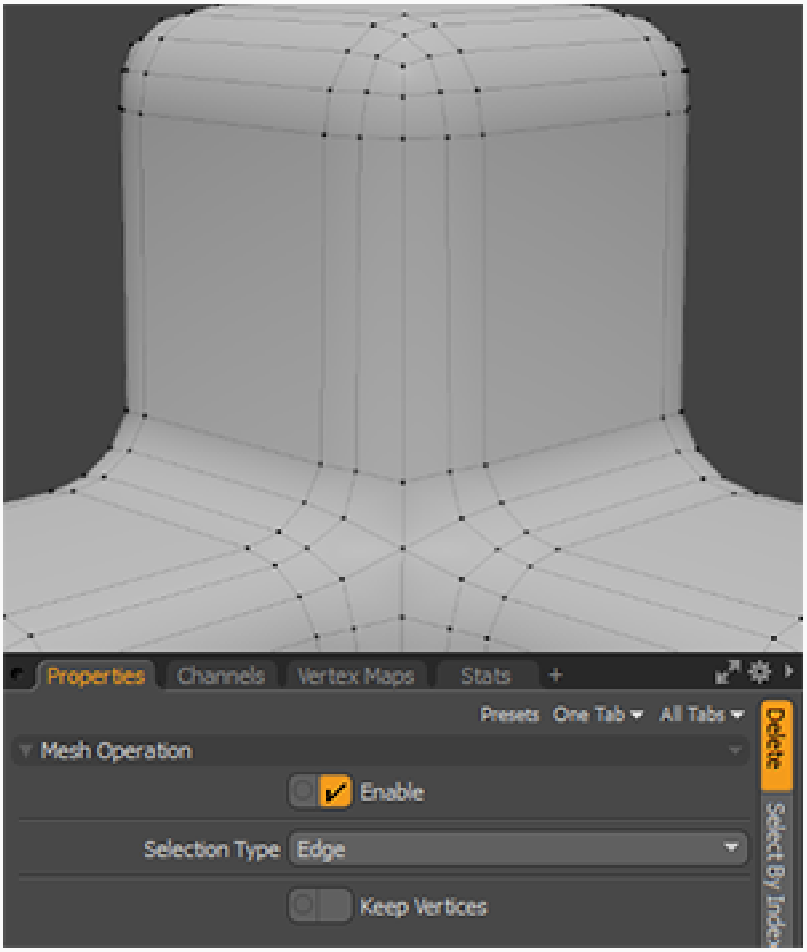
☑ Enable

Selection Type: Vertex

○ Keep Vertices

Option	Description
Enable	Toggles the tool on and off.
Selection Type	Allows you to specify the selection type to delete.

	• **Vertex** - Deletes the selected vertices and leaves the unselected vertices in your scene. • **Edge** - Deletes the selected edges and leaves the unselected edges in your scene. • **Polygon** - Deletes the selected polygon and leaves the unselected polygons in your scene.
Keep Vertices	This option allows you to keep vertices when using the **Edge** or **Polygon Selection Types**.  Properties Channels Vertex Maps Stats + Presets One Tab All Tabs Mesh Operation Delete Select By Index Enable Selection Type Edge Keep Vertices Selected edge If disabled, the **Delete** operator removes the selected edge or polygon and its adjoining vertices.



Properties
Channels
Vertex Maps
Stats
+
Presets
One Tab
All Tabs
Mesh Operation
Delete
Select By Index

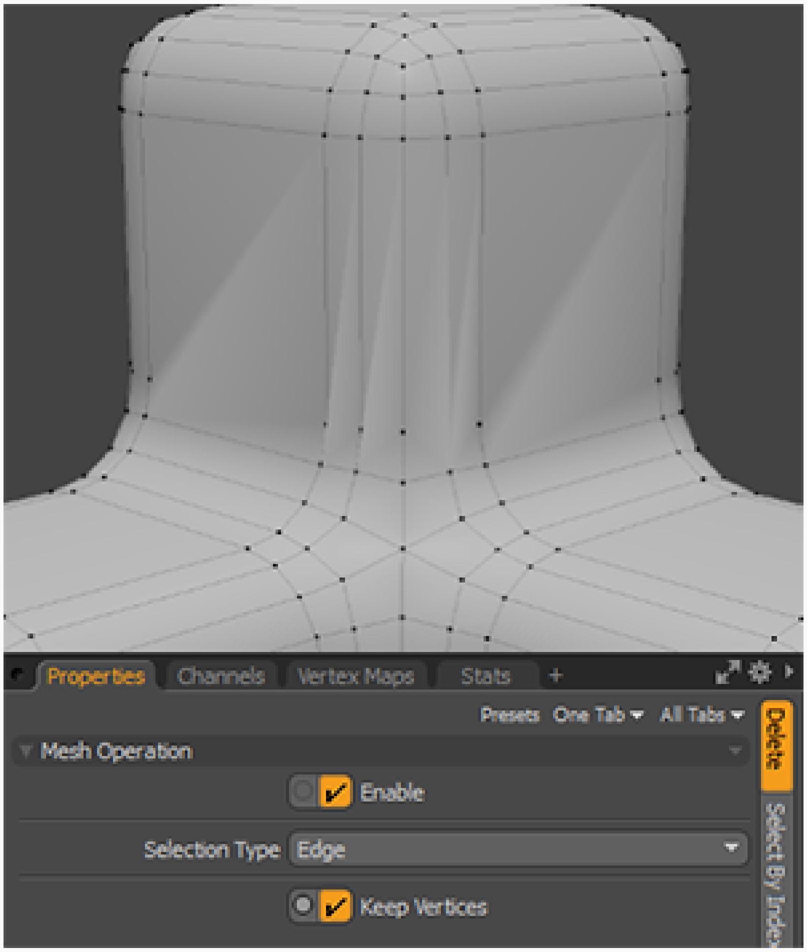
☑ Enable

Selection Type: Edge

☐ Keep Vertices

Keep Vertices disabled so the selected edge and adjoining vertices are deleted

If enabled, the **Delete** operator removes the selected edge or polygon only.


Properties
Channels
Vertex Maps
Stats
Presets
One Tab
All Tabs
Delete
Select By Index
Mesh Operation

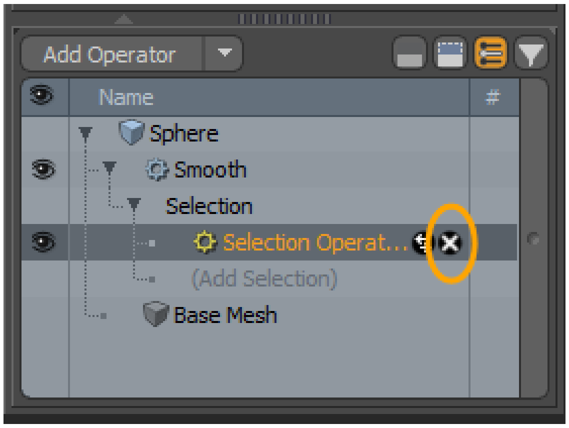
☑ Enable

Selection Type: Edge

☑ Keep Vertices

Keep Vertices enabled so only the selected edge is deleted

Alternatively, you can delete a selected item in the **Mesh Ops** list by clicking the **X** button.


Add Operator
Name
#
Sphere
Smooth
Selection
Selection Operat...
(Add Selection)
Base Mesh

You are presented a **Delete Item** dialog. If the selected item is referenced by other items in your scene you are presented with a number of delete options.

- Clicking **Yes** deletes only the selected item.
- Clicking **Yes to All** deletes the selected item and all of the referenced items.

Tip: You can also right-click on an item and select **Delete** from the contextual menu.

Automatic Retopology Tool

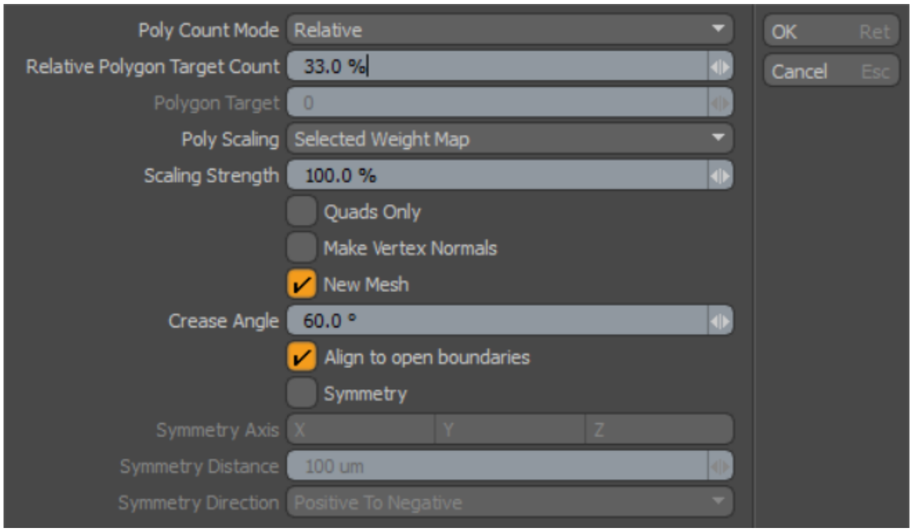
The Automatic Retopology tool generates a new mesh item from the input surface. The new mesh is produced according to an automatic retopology algorithm, which outputs a clean mesh with good topology that follows the curvature of the input at a density you specify. The density can be controlled by weight maps and the mesh flow can be controlled by sketching guide curves on the surface.

The **Automatic Retopology** tool converts a mesh into a mesh made of all 3- and 4-point polygons of a uniform size. To determine this polygon size, the tool looks at the number of polygons you requested in the output and divides the surface of the area by that number. As a result, the target polygon number, either as a number or a percent, is approximated. You can also apply a subdivision to the mesh to produce pure quads. For more information about subdivisions, see Surface Tab.

The retopology algorithm detects sharp edges in the input mesh and preserves them by looking for faces that meet at an angle greater than the crease angle you specify. Although the process is automatic, you can guide the results using standard Modo mesh attributes, such as the Retopo density **Weight Map**, or the **Retopo Edge Lock** and **Retopo Edge Flow** edge selection sets. For more information about applying edge lock and flow curves, see Refining Your Geometry Results.

The **Automatic Retopology** tool is available from the following UI:

- Open the **Topology** layout, in the Toolbox on the left pane open the **Tools** sub-tab, and click **Automatic Retopology**.
- Open the **Model** layout, in the Toolbox open the **Mesh Edit** sub-tab, and click **Auto Retopo**.
- On the **Geometry** application menu bar, click **Automatic Retopology...**.



Poly Count Mode: Relative
Relative Polygon Target Count: 33.0 %
Polygon Target: 0
Poly Scaling: Selected Weight Map
Scaling Strength: 100.0 %
☐ Quads Only
☐ Make Vertex Normals
☑ New Mesh
Crease Angle: 60.0 °
☑ Align to open boundaries
☐ Symmetry
Symmetry Axis: X Y Z
Symmetry Distance: 100 um
Symmetry Direction: Positive To Negative

Controlling Results

You can guide the retopology results in three ways:

- by constraining the positions of points,
- by constraining the direction of edges, or
- by changing the relative size of the polygons.

Constraining the Positions of Points by Edge Locking

To constrain positions for points, thus edges in the output mesh, create an edge selection set in the input mesh called **Retopo Edge Lock**. The retopo algorithm places the points along the edges. You can also apply a position lock by adding a curve to the mesh with the selection set tag **Retopo Edge Lock**. The curve must also have a vertex normal map with normals that match the surface. You can draw this curve using the **Retopo Sketch** tool. For more information, see Applying Retopo Guides and Refining Your Geometry Results.

Constraining the Direction of Edges

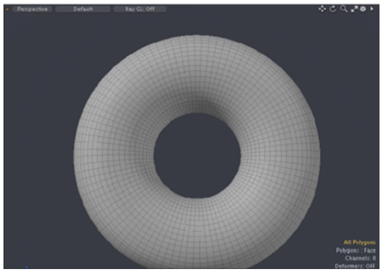
To constrain directions for edge flows in the output mesh, create an edge selection set in the input mesh called **Retopo Edge Flow**. The retopo algorithm then aligns the poly flow to the edges. You can also add a flow guide by adding a curve to the scene with the selection set tag **Retopo Edge Flow**. The curve must also have a vertex normal map with normals that match the surface. You can draw this curve using the **Retopo Sketch** tool. For more information about vertex maps, see Vertex Normal Maps. For more information about applying curves used to constrain the direction of the edges, see Refining Your Geometry Results.

Changing the Relative Size of the Polygons

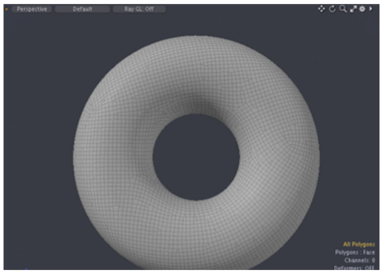
A metric is a way of measuring distance on a surface. By defining a different scale at different parts of the mesh, you can warp the uniform polygons to be bigger in some places and smaller in others. This is particularly useful if you want to capture small details in a model without needing a large number of small polygons in the large flat places as well. The smallest polygons are determined by the **Relative Polygon Target Count**. To get very small detailed polygons, when using a custom metric, you may need to set more than 100% of the polygons in the model. You can get additional control over the metric using the **Scale Strength** option.

Examples

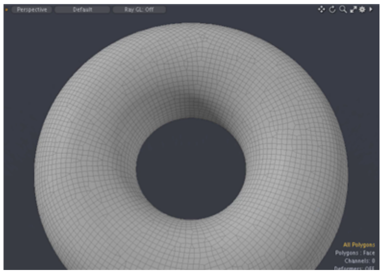
Click on an image below to enlarge.



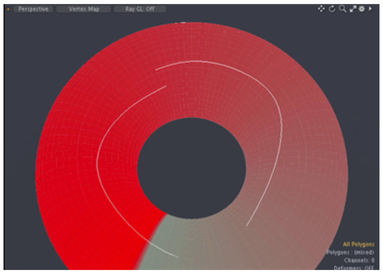
Default Torus



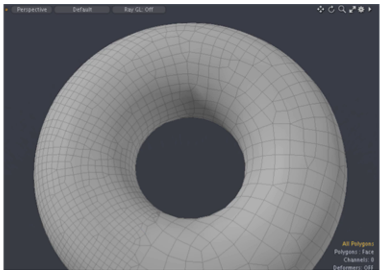
Relative Polygon Target Count 100%



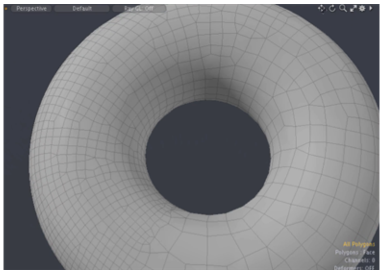
Apply a curve selection for a position lock using **Retopo Guides**.
Relative Polygon Target Count 100%



View Vertex Map - UMap applied.



Weight Map - UMap applied.
Relative Polygon Target Count 100%
Scaling Strength 200%



Weight Map - UMap applied.
Relative Polygon Target Count 100%
Scaling Strength 50%

Automatic Retopology Properties

The following options are available:

Automatic Retopology	
Poly Count Mode	Determines the number of polygons as a percentage of the original mesh, with an absolute number or by adaptively or scaling based on input mesh. The following options are available: • **Relative** - Scales the existing polygon count set in the Relative Polygon Target Count option to determine the minimum scale of polygons. • **Absolute** -Determines the minimum scale of polygons. • **Adaptive** - Attempts to automate the settings for Relative Polygon Target Count and Scaling Strength. It tries to find a high enough mesh density to capture the detail from the smallest edges on the mesh, then computes an appropriate scaling strength for the **Shortest Edge** metric to scale the lowest density to match the lowest density of the source mesh. **Note:** When the Adaptive mode is used, the metric argument,Poly Scaling, is disabled but the Relative Polygon Target Count and the Scaling Strength settings can be used to increase or decrease the base density, or to modify the impact of the metric, respectively.
Relative Polygon Target Count	Number of polygons expressed as a percentage of the current polygon count. This number determines the average poly area, which guides the retopology process. **Note:** This calculation is not exact.
Polygon Target	Number of polygons. If this is not zero, it overrides the **Relative Polygon Target Count**value. This number determines the average polygon area, which guides the retopology process. **Note:** This calculation is not exact. **Tip:** Watch out for large polygons next to small polygons. Bump up the **Polygon Target** count to capture small polygon details.
Poly Scaling	This option sets the Custom Metric that changes the relative polygon sizes based on

Automatic Retopology	
	a particular mesh attribute. **Tip:** Use larger **Poly Scaling** values to create larger polygons in areas where detail is not needed. The following options are available: • **Selected Weight Map** - Use the currently selected weight map, or if none is selected, a weight map named **RetopoDensity** is used to control the polygon size. Values from 0 to 1.0 drive the scale of polygons from largest to smallest. For information about weight maps, see Weighting Tools. • **Poly Area** - This metric scales the output polygon size based on the input polygon size. This is useful on models with small details. • **Shortest Edge** - This metric is very similar to the Poly Area metric but instead of using the area of the polygons to scale the density, it uses the length of the minimum edge on the polygon. This can make a difference when a mesh has very large and narrow triangles, as is commonly seen in models imported from CAD data. • **Curvature** - This metric computes the surface curvature and makes polygons larger in flatter areas. It uses the surface curvature at each vertex to scale the density of quads in the retopology result mesh, making the mesh more dense where the mesh is curved and less dense where it is flat. This is intended to capture the shape of the surface more accurately without wasting a lot of polygons in the flat areas.
Scaling Strength	A multiplier to weight the influence the polygon scaling. Amplify the strength of the custom metric by increasing this over 100%.
Quads Only	This option causes the retopology result mesh to be subdivided once to eliminate any 3-point polygons.
Make Vertex Normals	Enable to generate a vertex map with vertex normals.
New Mesh	Enable to create a new mesh. When the **New Mesh** option is disabled, the Auto-Retopo result replaces the source mesh. In this case, the polygon type of the source mesh is preserved. This works best if the source doesn't mix regular and subdiv polygons.

Automatic Retopology	
Crease Angle	The minimum angle that is treated as a sharp crease. Polygons that meet at a sharper angle than this value are assumed to have a sharp edge.
Align to open boundaries	This option preserves any open boundaries on the mesh and guides the edge flow.
Symmetry	Turn on this option to perform automatic retopology on only one side of the mesh, then mirror and merge the result. This creates a new mesh that is symmetrical along the selected **Symmetry Axis**. It also produces results faster.
Symmetry Axis	Choose the axis of symmetry.
Symmetry Distance	This option provides a margin of error for the input mesh, in case some of its middle points are over the line. It also sets the merge distance when mirroring the result mesh and merging seam vertices. If your result has gaps along the seam, you may need to increase this number.
Symmetry Direction	Specifies the direction of the operation. The following options are available: • **Positive To Negative**: Delete polygons on the negative side and mirror the remaining polygons on the positive side. • **Negative To Positive**: Delete polygons on the positive side and mirror the remaining polygons on the negative side.

Applying Auto Retopology

The following is a simple example of applying the **Automatic Retopology** tool using the **Model** layout.

1. Open the **Model** layout, open the **Basics** sub-tab in the Toolbox, and then add a **Torus Primitive** to your scene.
2. Open the **Mesh Edit** sub-tab in the Toolbox and click **Auto Retopo**.

   The Auto Retopo dialog is displayed.
3. Change the Automatic Retopology Properties settings and then click **OK**.

Retopo Guides

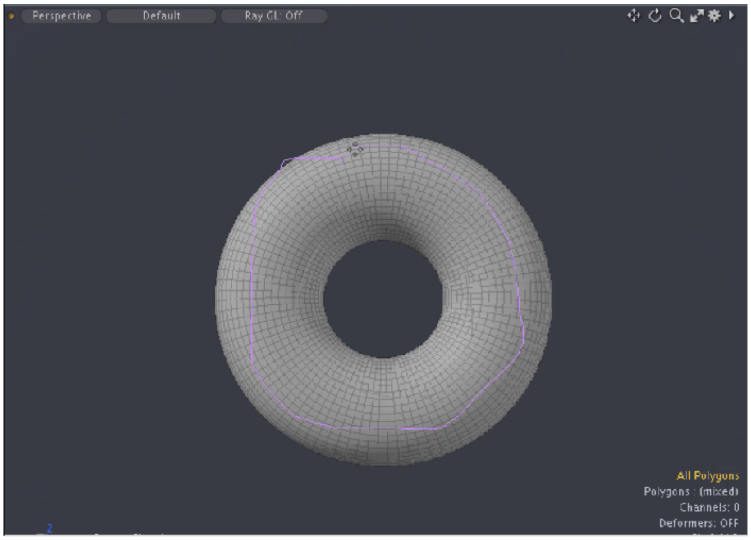
The **Retopo Guides** tool allows you to sketch curves (polylines) on a surface, which are set up as position and edge flow guides for **Automatic Retopology**. These curves are tagged with a special selection set name that the **autoretopo** command looks for and they have a vertex normal map with normals taken from the selected surface. As a result of placing **Retopo Guides** on your mesh, the curve polygon on the surface creates a polygon selection set, which improves the alignment flow to the curves and to the natural direction of the selected mesh.

Tip: The **Retopo Guides** tool can also be launched with the command **tool.set retopo.sketch on**.

Applying Retopo Guides

The following is an example applying curves to constrain directions for edge flows in the output mesh.

1. Open the **Topology** layout and select a mesh in your scene.
2. In the Toolbox on the left pane, open the **Tools** sub-tab, click **Automatic Retopology** and apply the default setting.
3. In the Toolbox on the left pane, under the **Tools** sub-tab, click **Retopo Guides**.
4. Click and drag on the surface to defines curves.



5. In the Toolbox on the left pane, under the **Tools** sub-tab, click **Automatic Retopology**.
6. Click **OK** to apply the default settings.

   The alignment flow is updated.

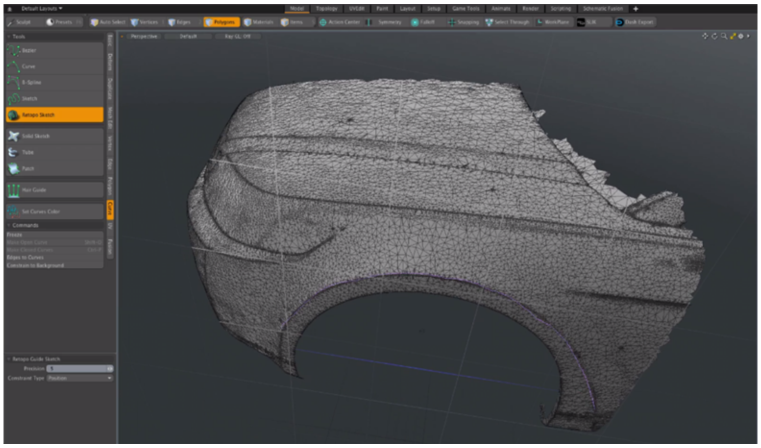
Refining Your Geometry Results

To refine your geometry results, to create better tessellation of the polygons on your model, you can set the Automatic Retopology tool to lock to specified sharp edges on your mesh and to curves used to specify the direction of the flow of polygons. The following procedure walks you through the process of using both **Retopo Edge Lock** and **Retopo Edge Flow** to apply the edge selection sets to the selected edges.

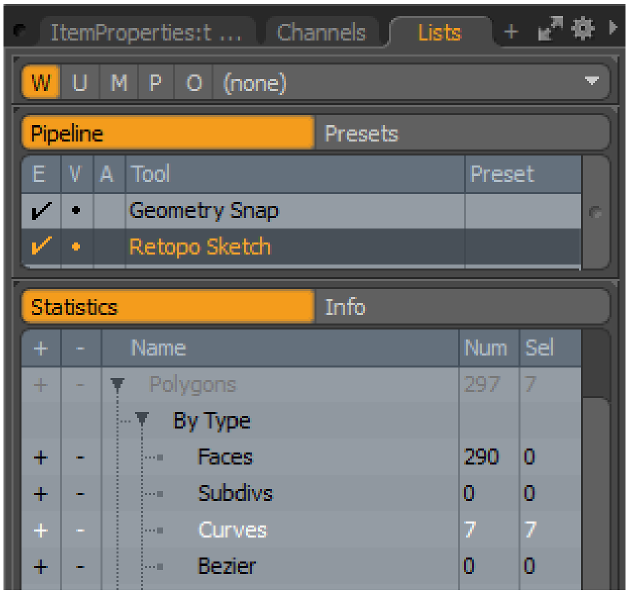
To refine your geometry results:

1. In the **Model** layout, on the Toolbox on the left pane, click the **Curve** sub-tab.
2. Click the **Retopo Sketch** tool and sketch curves onto the geometry of your mesh.

   In this example, curves are drawn on sharp edges.



3. Select all of the curves you have created. To quickly do this, open the **Mesh Operations** List, click the **Lists** sub-tab in the lower pane. Under the **Statistics** sub-tab, expand **Polygons** > **By Type** and click the **+** (plus) button beside **Curves**. All of the curves are now selected.



E	V	A	Tool	Preset
✓	•		Geometry Snap	
✓	•		Retopo Sketch	

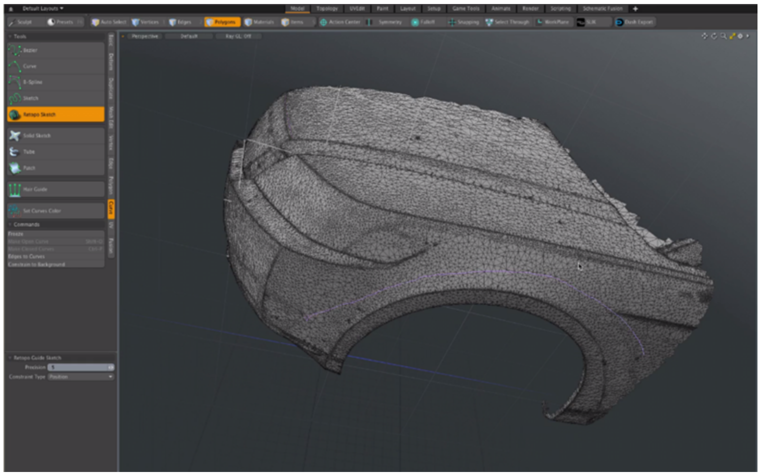
+	-	Name	Num	Sel
+	-	Polygons	297	7
		By Type		
+	-	Faces	290	0
+	-	Subdivs	0	0
+	-	Curves	7	7
+	-	Bezier	0	0

4. Set the **Auto Retopology** tool to lock to edge curves. With the selection of curves activated, click the **Mesh Edit** sub-tab in the Toolbox, and then click **Retopo Edge Lock**.

   When using this option, you are using the curves drawn on sharp edges to tell the Auto Retopology tool to calculate a more uniform mesh based on the selected curves.

5. Specify the flow direction of the geometry between the locked edges by drawing flow curves onto your mesh. Click the **Curves** sub-tab in the Toolbox, click **Retopo Sketch**, and draw your flow curves.

   In this example, directional flow curves are drawn around the wheel fender and the front of the hood.



6. In the Toolbox, click the **Mesh Edit** sub-tab, and then click **Retopo Edge Flow**.

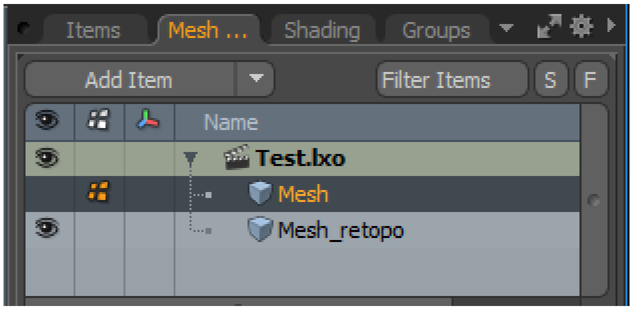
   When using this option, you are using the curves drawn to specify the flow direction of the geometry and to tell the Auto Retopology tool to calculate a more uniform mesh based on the selection of curves.

Tip: A selection set is automatically created. To view this selection set, in the **Mesh Operations** List, under the **Statistics** sub-tab, expand **Polygons** > **By Selection Set**. Depending on the method you have applied, **RetopoEdgeLock** or **RetopoFlow** is listed with their respective curve counts.

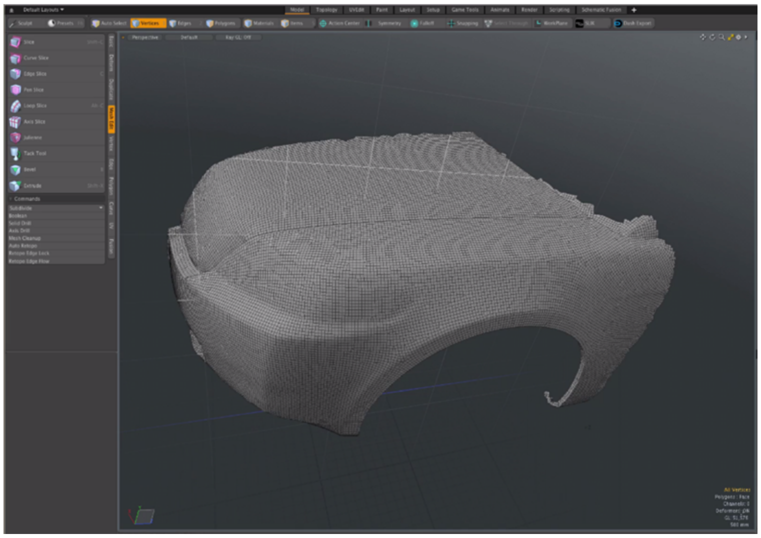
7. Run the **Auto Retopology** tool. In the **Mesh Edit** sub-tab, click **Auto Retopo** and set the **Relative Polygon Target Count** percentage or **Polygon Target** value, and then click **OK**. In this example, I have set **Relative Polygon Target Count** to 200%.

   For more information about these options, see Automatic Retopology Properties.

8. To view the results, in the **Mesh Operations** list, open the **Items** sub-tab, and click the 👁 visibility button for the parent mesh.


Items
Mesh ...
Shading
Groups
Add Item
Filter Items
S
F
Name
Test.lxo
Mesh
Mesh_retopo

The **Mesh_retopo** item is displayed in the viewport. Notice the polygon count has been reduced and the tessellation of the geometry has been improved.



Merge Meshes

The **Merge Meshes** tool merges multiple mesh layers into a single procedural mesh. You can also apply Merge Meshes to direct modeling mesh items to create a single mesh.

Use the Merge Meshes tool to:

- Copy any item into a procedural layer.

  For example, the source geometry can be a MeshFusion item, the procedural mesh, VDB voxels, or one of the other procedural items. Once merged, you can modify or deform the procedural result like any other procedural mesh.

Note: The geometry type (triangles or quads) is dependent on the geometry provided by the procedural surface. No post-processing is done to clean up or modify the geometry in any way.

- Use the Merge Meshes tool as a mesh operator to control which properties are merged.

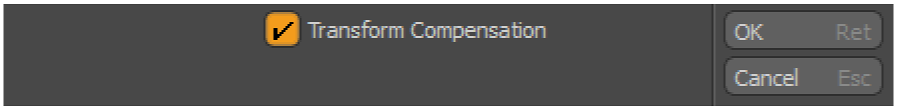
  For example, you can read replicator items as geometry inputs allowing replicated geometry to be manipulated using procedural modeling tools. These replicator items can be exported as a single mesh item to use in other projects. For more information, see Using the Merge Meshes Operator
- Apply the Merge Meshes operation to a group item. For more information, see Applying Merge Meshes to a Group.
- Use Merge Meshes to convert a particle simulation into vertices in a mesh layer. You can then use deformers to modify the simulation.

Note: If you modify the source mesh, your changes are also applied to the target mesh. However, the changes you make to the mesh in the target layer are not carried over to the source mesh. Deleting the original mesh also removes the mesh from the target layer.

Merging Meshes

1. On the right panel, open the **Items** list and select the item layers to merge.
2. Right-click and select **Merge Meshes**,

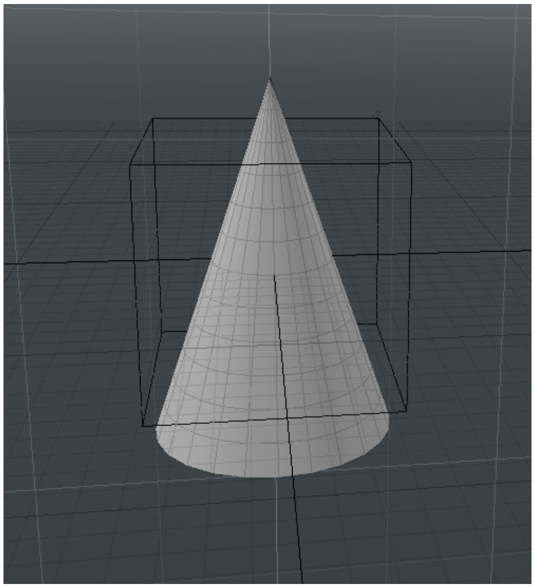
   The Merge Meshes dialog displays with the **Transform Compensation** option enabled. This option transforms the normal map values to correspond to the new vertex positions in the merged mesh.



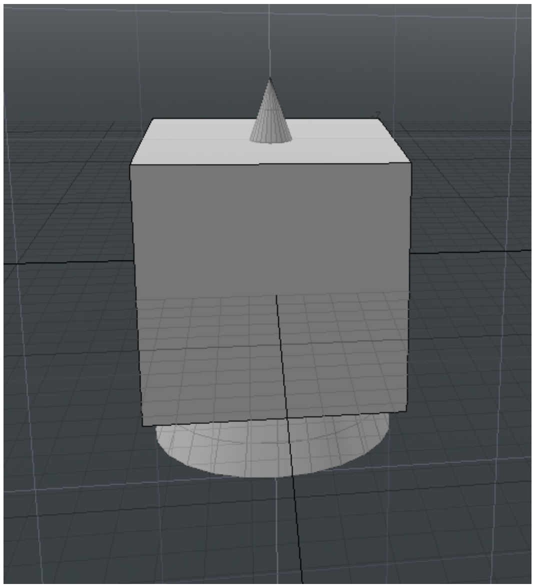
☑ Transform Compensation

3. Click **OK**.

   All mesh layers in your scene are merged into the target parent mesh. No vertices are merged.



The target mesh in the 3D viewport before the Merge Meshes operation.



The target mesh in the 3D viewport after the Merge Meshes operation.

Applying UnMerge Meshes

1. On the right panel, open the **Items** list.
2. Right-click on a parent mesh item and select **UnMerge Meshes**.

   All the separate mesh items within a single parent mesh item are created as individual mesh items.

Applying Merge Meshes - Selected

1. On the right panel, open the **Items** list.
2. Right-click on a mesh item and select **Merge Meshes - Selected**.
3. Name the new mesh item.

   The new mesh item links the items selected in the items list.

Using the Merge Meshes Operator

Use the Merge Meshes mesh operation to specify which Merge Meshes Properties are merged.

You can apply Merge Meshes to:

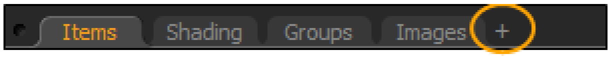
- selected items
- a parent item

or

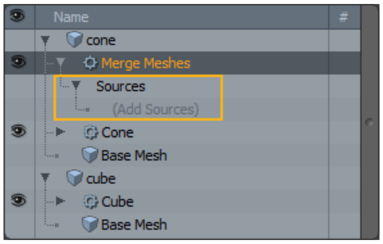
- group item

1. Open the **Mesh Ops** tab and click **Add Operator**.

Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **New Tab** > **Data Lists** > **Mesh Ops**.



2. Navigate to **Mesh Operations** > **Edit** and double-click **Merge Meshes**.
3. Expand the **Merge Meshes** operator and click **(Add Sources)**.
4. Navigate to **Modifier Mesh Operations** > **Existing** and select the parent source mesh item.

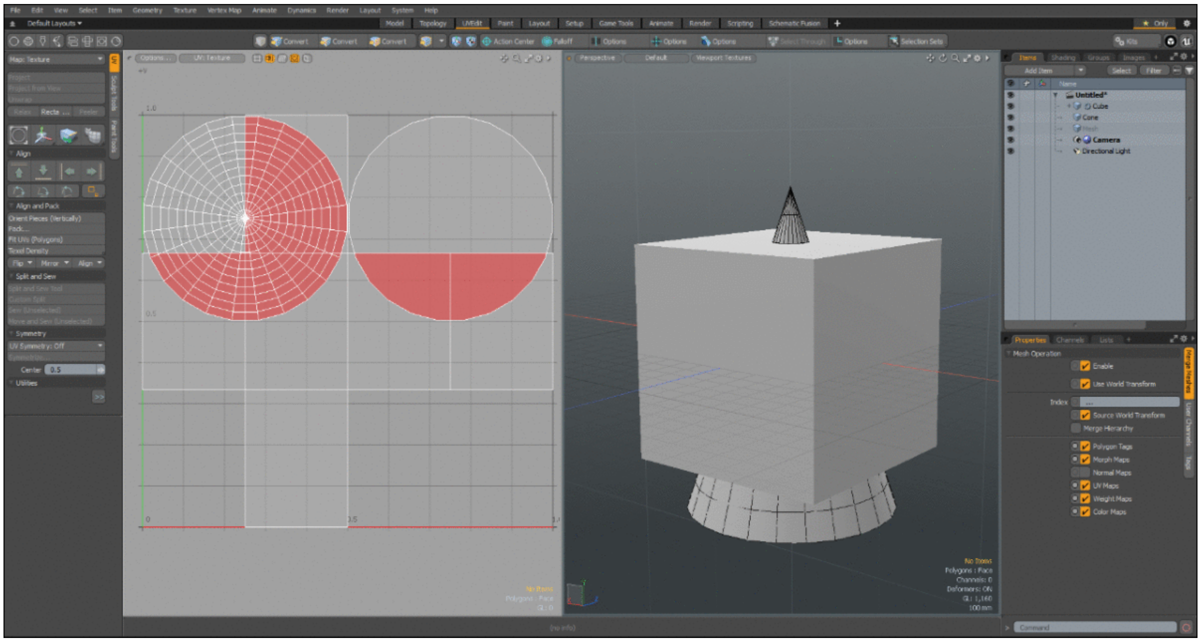


Tip: When specifying the **(Add Sources)** value, you can also select **Modifier Mesh Operations** > **New** > **Particles** > **Replicator**. The replicator items are converted to geometry items and you can continue applying various procedural operations to modify each layer. Once done, right-click **Merge Meshes** and select **Freeze** and then export your selection (**File > Export As...**). For more information, see Replicators.

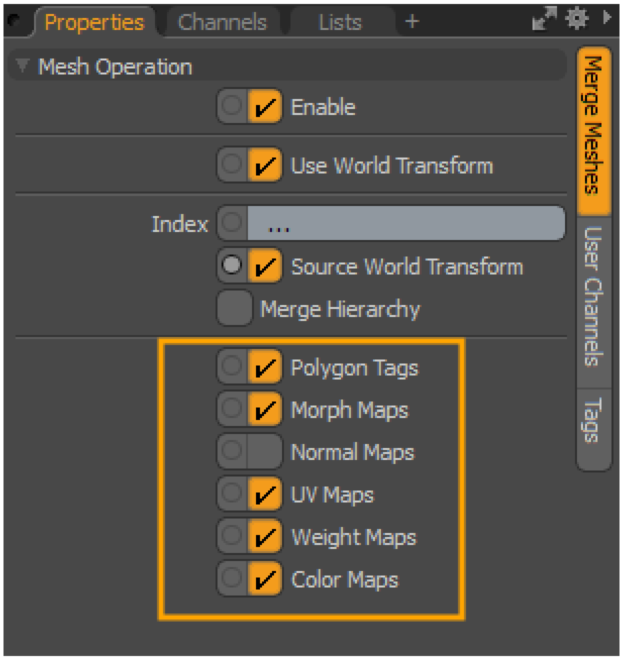
5. In the **Properties** panel, on the bottom of the **Mesh Ops** tab, enable Merge Hierarchy.

Partial Merges

Merge Meshes can perform partial merges, allowing certain mesh elements to be skipped. This is useful when merging surfaces that contain normal maps, as the presence of the normal map is often undesirable for further edits.



In the Merge Meshes Properties tab, on the right panel, enable or disable the following channels:


Properties
Channels
Lists
+
Merge Meshes
User Channels
Tags

Mesh Operation

☑ Enable

☑ Use World Transform

Index ...

☑ Source World Transform

☐ Merge Hierarchy

☑ Polygon Tags

☑ Morph Maps

☐ Normal Maps

☑ UV Maps

☑ Weight Maps

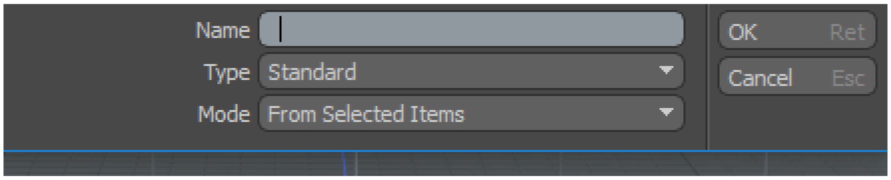
☑ Color Maps

Applying Merge Meshes to a Group

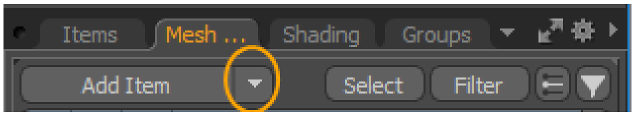
Once the Merge Meshes tool is applied to a group, you can later add other items to that group. The Merge Meshes tool controls all items added to a group.

Tip: To add an item to an existing group, select it in the 3D viewport and open the **Groups** tab on the right panel. Right-click on the group item and select **Add Item** from the contextual menu. For more information, see Groups Viewport.

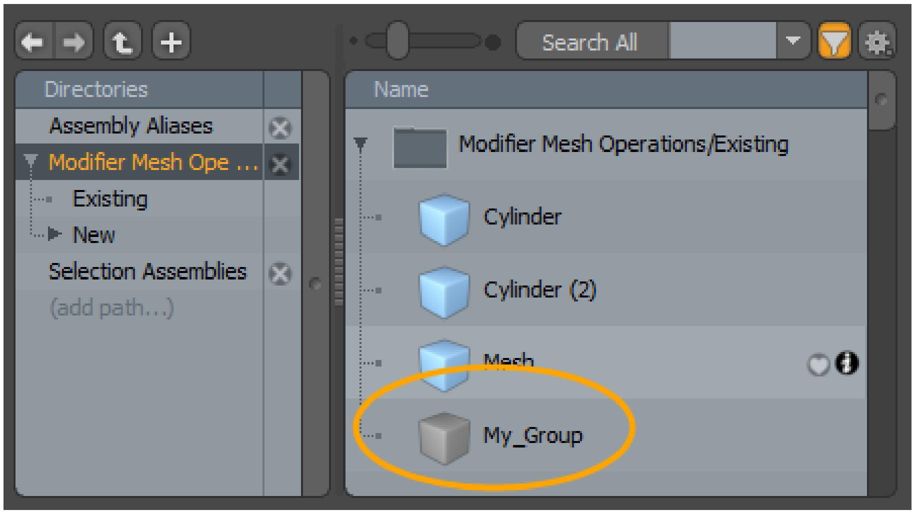
1. In the **Model** layout, open the **Items** tab, and select the mesh items to group.
2. On the right panel, open the **Groups** tab, and click **New Group**.


Name
Type Standard
Mode From Selected Items
OK Ret
Cancel Esc

3. Enter a meaningful **Name** and click **OK**.
4. Open the **Mesh Ops** tab, click the down arrow beside **Add Item** and select **Mesh**.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter

5. Click and drag this mesh layer and move it to the top of the list.
6. Click **Add Operator**, expand **Mesh Operations** > **Edit**, and double-click **Merge Meshes**.
7. Expand **Merge Meshes** > **Sources**, and click **(Add Sources)**.
8. In the Add Operator dialog, expand **Modifier Mesh Operations** > **Existing** and double-click **My_ Group**.


Search All
Directories
Assembly Aliases
Modifier Mesh Ope ...
Existing
New
Selection Assemblies
(add path...)
Name
Modifier Mesh Operations/Existing
Cylinder
Cylinder (2)
Mesh
My_Group

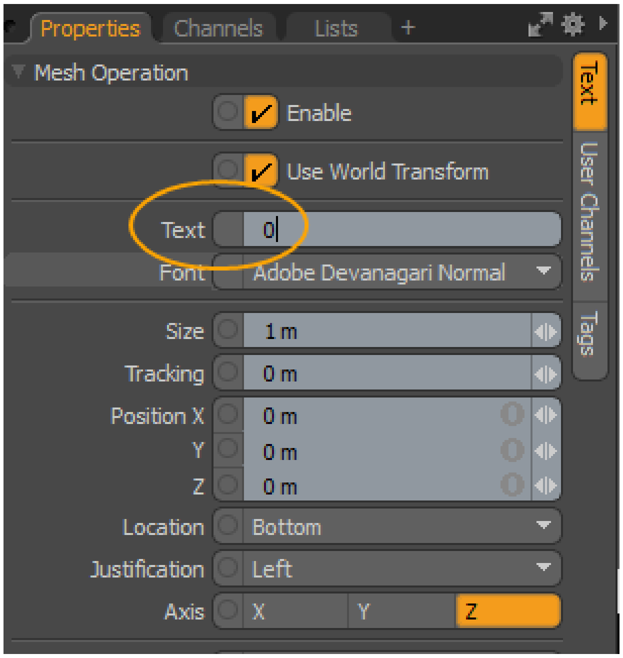
Applying Merge Meshes to Text

You can use the **Index** option to specify which Text mesh items to include in a Merge Meshes operation.

Index ...

Note: For more information about the Text primitive, see Text.

1. In the **Model** layout, open the **Mesh Ops** tab, click **Add Item**, and double-click **Locators** > **Mesh**.
2. On the right panel, click **Add Operator**, expand **Mesh Operations** > **Create** > **Primitives**, and double-click **Text**.
3. In the **Properties** tab, on the bottom right panel, enter in a number into the **Text** field.



Enable ☑

Use World Transform ☑

Text 0

Font Adobe Devanagari Normal

Size 1 m

Tracking 0 m

Position X 0 m

Y 0 m

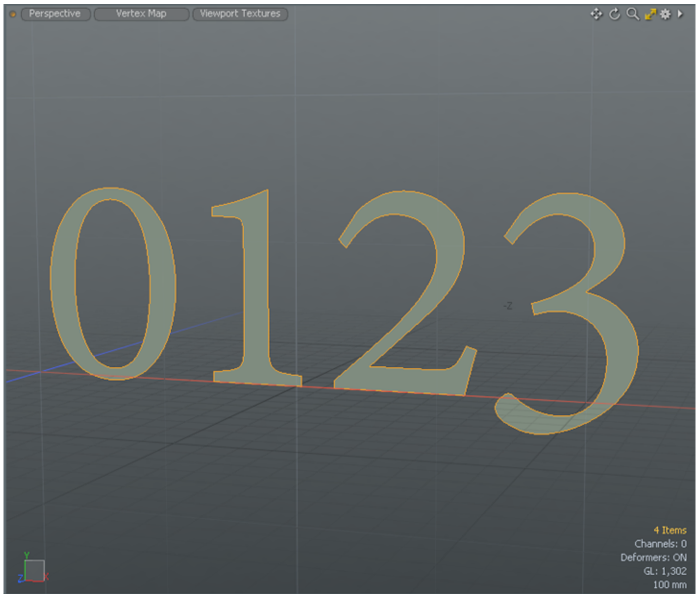
Z 0 m

Location Bottom

Justification Left

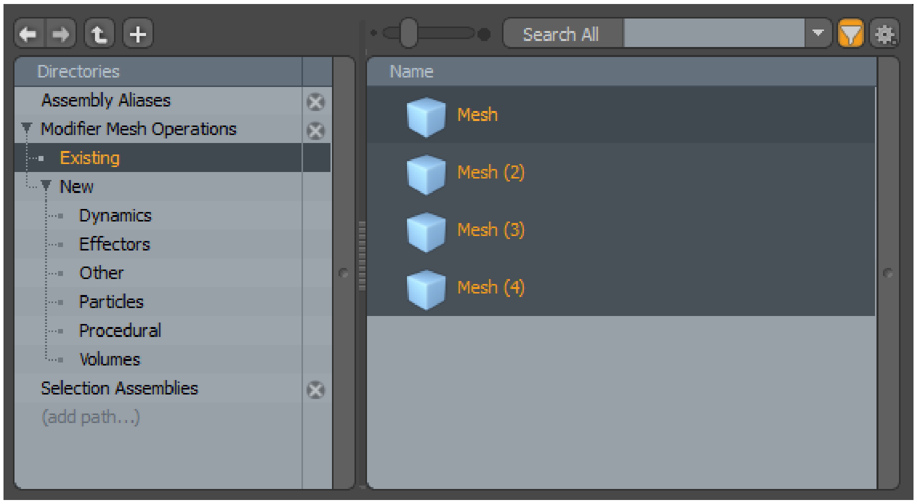
Axis X Y Z

4. Select the Text primitive, press **W**, and drag the text field into an new position along the **X** axis.
5. Repeat steps 1 - 4 to add the four separate text meshes to your scene, such as 0, 1, 2, and 3.


Perspective
Vertex Map
Viewport Textures
0123
4 Items
Channels: 0
Deformers: ON
GL: 1,302
100 mm

Tip: To keep organized, double-click on each mesh item in the **Items** list and type a new name.

6. Click **Add Operator**, expand **Mesh Operations** > **Edit**, and double-click **Merge Meshes**.
7. Expand **Merge Meshes** > **Sources**, and click **(Add Sources)**.
8. In the **Add Operator** dialog, expand **Modifier Mesh Operations** > **Existing** and select all of the mesh items you previously created.

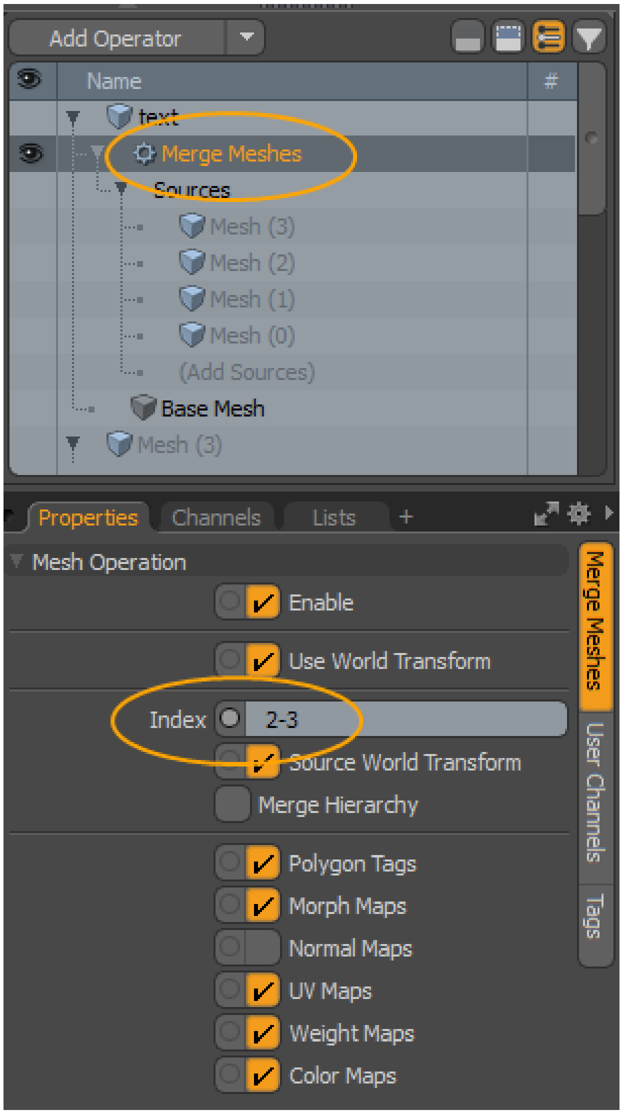

Search All
Directories
Assembly Aliases
Modifier Mesh Operations
Existing
New
Dynamics
Effectors
Other
Particles
Procedural
Volumes
Selection Assemblies
(add path...)
Name
Mesh
Mesh (2)
Mesh (3)
Mesh (4)

Warning: If you're trying to select your active mesh, Modo displays an error message.

9. Drag-and-drop your selection onto the Merge Meshes **Sources** field.
10. In the **Mesh Ops** tab, click the eye icon to hide the original meshes.

The merged mesh items remain displayed in the 3D viewport.

11. Select the **Merge Meshes** item, and type the **0-2** value in the **Index** field and press **Enter**.


Add Operator
Name
#
text
Merge Meshes
Sources
Mesh (3)
Mesh (2)
Mesh (1)
Mesh (0)
(Add Sources)
Base Mesh
Mesh (3)
Properties
Channels
Lists
+
Merge Meshes
User Channels
Tags

Mesh Operation

☑ Enable

☑ Use World Transform

Index 2-3

☑ Source World Transform

☐ Merge Hierarchy

☑ Polygon Tags

☑ Morph Maps

☐ Normal Maps

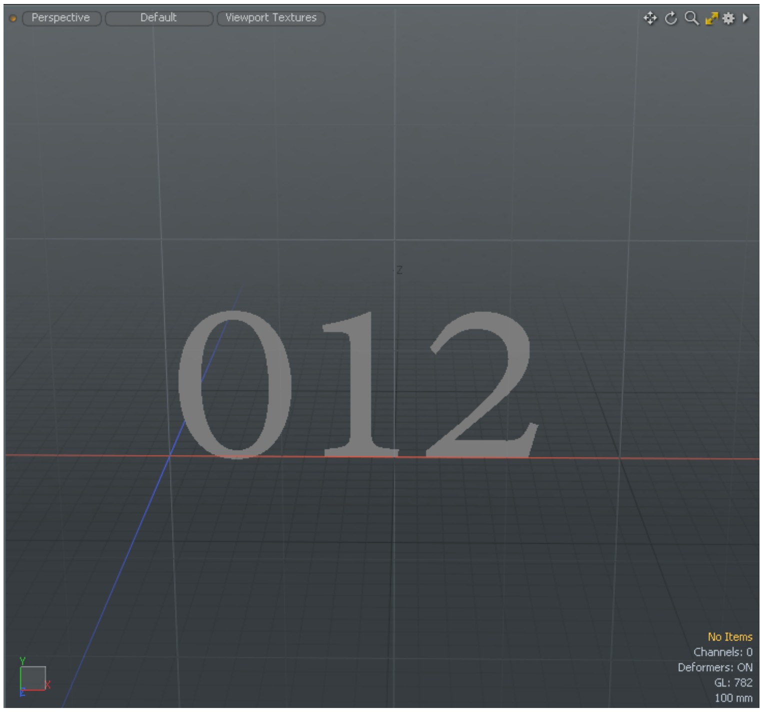
☑ UV Maps

☑ Weight Maps

☑ Color Maps

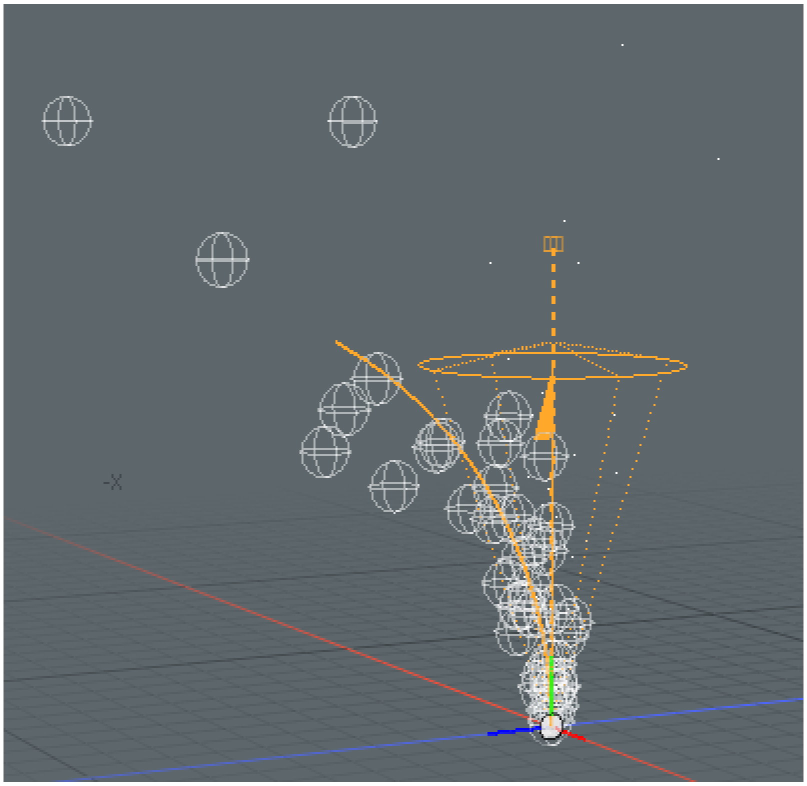
Note: The mesh items are numbered based on the order they were created, starting at 0 for the first item created.

Only the first three mesh items are displayed in the viewport.


Perspective
Default
Viewport Textures
No Items
Channels: 0
Deformers: ON
GL: 782
100 mm

Applying Merge Meshes to Particle Simulations

The Merge Meshes operation also works with particles. You can specify any kind of Particle Simulation as the **Source** of the operation, and it converts particle sources to vertices in a mesh layer. You can then apply deformers to the simulation.



To merge your particle simulation into a mesh layer:

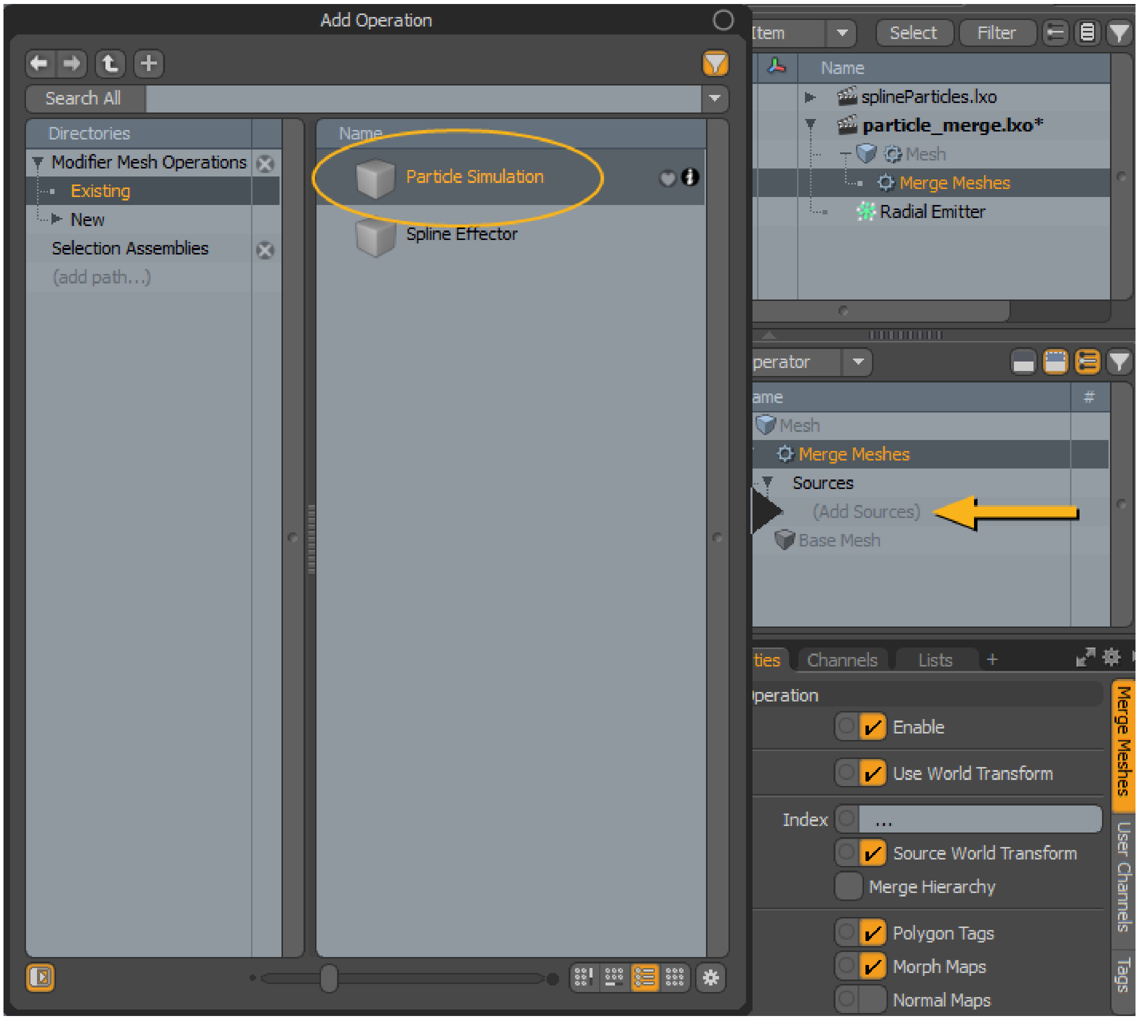
1. Have a scene with a particle simulation and a mesh layer set up.

Tip: To learn how to set up a particle simulation, see Particles and Simulations.

2. In the **Mesh Ops** tab go to your mesh layer and click **Add Operator**.

   This opens the **Add Operator** dialog.
3. Click **Mesh Operations** > **Edit**, and double-click **Merge Meshes**.

   The Merge Meshes operation is added to the Mesh Operations list.
4. Expand **Merge Meshes** in the Mesh Operations list to reveal its **Sources** input.
5. Click **Add Sources**, and under **Existing**, double-click your particle simulation.


Add Operation
Search All
Directories
Modifier Mesh Operations
Existing
New
Selection Assemblies
(add path...)
Name
Particle Simulation
Spline Effector
Item
Select
Filter
Name
splineParticles.lxo
particle_merge.lxo*
Mesh
Merge Meshes
Radial Emitter
perator
ame
#
Mesh
Merge Meshes
Sources
(Add Sources)
Base Mesh
ties
Channels
Lists
+
peration
Merge Meshes
User Channels
Tags

☑ Enable
☑ Use World Transform
Index ...
☑ Source World Transform
☐ Merge Hierarchy
☑ Polygon Tags
☑ Morph Maps
☐ Normal Maps

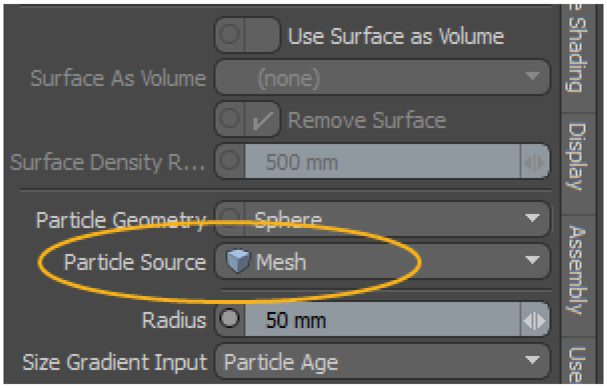
This merges the particle simulation to the mesh layer. The particles are converted to vertices that you can deform.

To deform the simulation:

1. In the Mesh Operations list, add a deformer, for instance, a Bend Effector, and set it up in your scene.

Note: For details on the various types of effectors and how to use them, see Applying Deformers.

2. If you're using a Volume item, select the it to open its **Properties** on the right.



☐ Use Surface as Volume

Surface As Volume (none)

☑ Remove Surface

Surface Density R... 500 mm

Particle Geometry Sphere

Particle Source Mesh

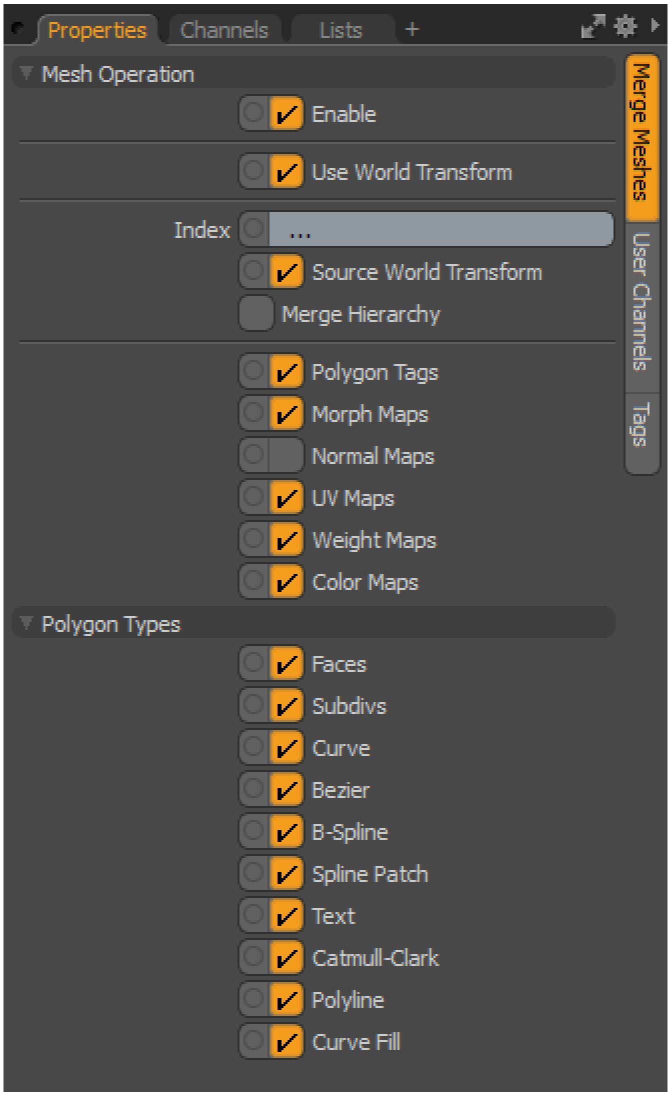
Radius 50 mm

Size Gradient Input Particle Age

3. In the **Volume Particles** sub-tab, under **Geometry**, set **Particle Source** to the mesh layer to which your simulation was merged.

   When you play the simulation now, the particles are deformed by the deformer you've added.

Merge Meshes Properties


Properties
Channels
Lists
+
Merge Meshes
User Channels
Tags
Mesh Operation

☑ Enable

☑ Use World Transform

Index ...

☑ Source World Transform

☐ Merge Hierarchy

☑ Polygon Tags

☑ Morph Maps

☐ Normal Maps

☑ UV Maps

☑ Weight Maps

☑ Color Maps

Polygon Types

☑ Faces

☑ Subdivs

☑ Curve

☑ Bezier

☑ B-Spline

☑ Spline Patch

☑ Text

☑ Catmull-Clark

☑ Polyline

☑ Curve Fill

Enable	Toggles merged meshes on and off.
Use World Transform	When enabled, the mesh operation maintains the world space on the input layer.
Index	Specify the geometry to merge.

	Note: The mesh items are numbered based on the order they were created, starting at 0 for the first item created. For example: **...** - All mesh items are merged. **1** - Only the second Merge Meshes Source mesh item is merged. **1...** - The second and remaining Merge Meshes Source mesh items are merged. **1-3** - The second, third, and forth Merge Meshes Source mesh items are merged.
Source World Transform	Input items are merged using their world space.
Merge Hierarchy	Input items and all child items are merged.
Polygon Tags	When enabled, merges polygon tags. For more information, see Polygon Set Material/Part/Smoothing.
Morph Maps	When enabled, merges morph maps. For more information, see Morph Influence.
Normal Maps	When enabled, merges normal maps. For more information, see Vertex Normal Maps.
UV Maps	When enabled, merges UV Maps
Weight Maps	When enabled, merges Weight Maps.
Color Maps	When enabled, merges Color Maps.

Merge Mesh - Polygon Types Properties

Use the following options to specify what type of polygons you want to extract from a mesh.

Faces	Copy face polygons from the source mesh to the procedural mesh.

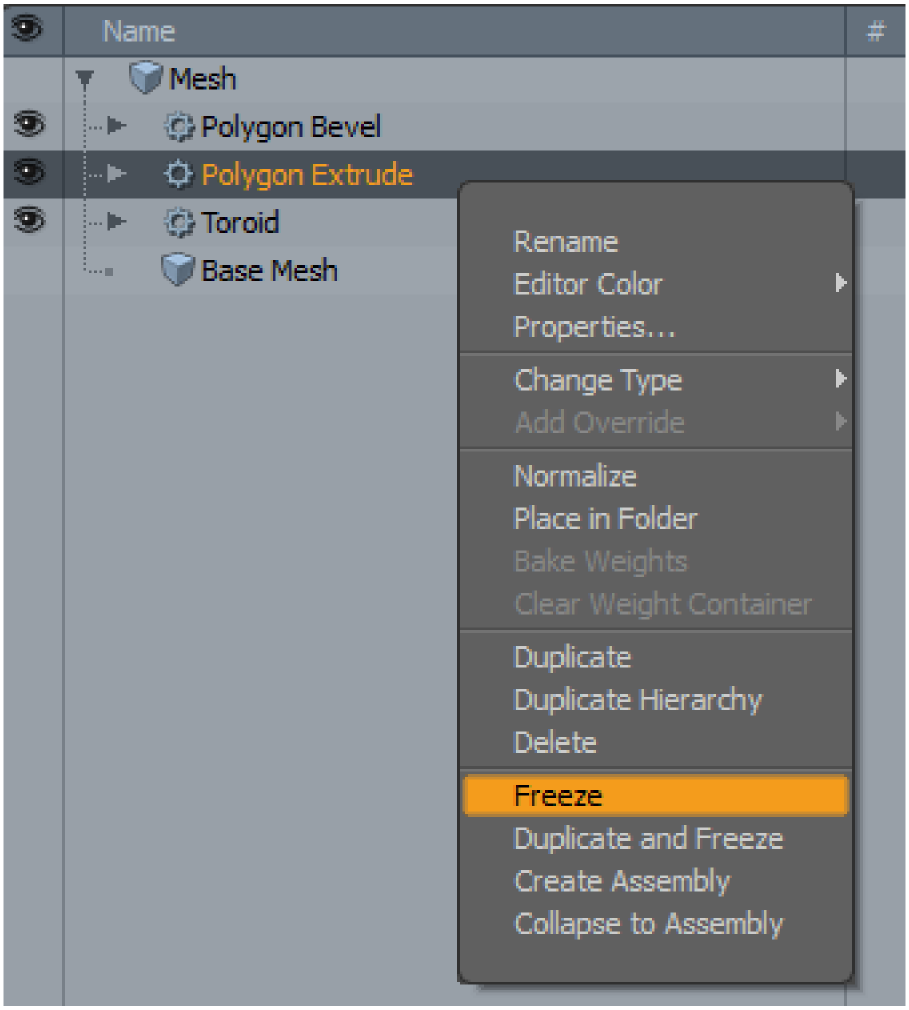
	A face polygon creates a faceted surface. An individual polygon is commonly called a face, which is bounded by three or more vertices and their associated edges. When many faces are connected together to create a network of faces, it's called a polygon mesh.
Subdivs	Copy subdivision patches from the source mesh to the procedural mesh. Subdivs are subdivision surface smoothed polygons. Subdividing polygons recursively splits the selected polygons into smaller polygons. For more information, see Subdivide.
Curve	Copy curve polygons from the source mesh to the procedural mesh. For more information, see Adding Curves.
Bezier	Copy Bezier polygons from the source mesh to the procedural mesh. For more information, see Bezier.
B-Spline	Copy B-Spline polygons from the source mesh to the procedural mesh. B-Spline polygons are spline curves defined by at least four vertices and a Weight Map. For more information, see B-Spline.
Spline Patch	Copy spline patches from the source mesh to the procedural mesh. Spline patches produce a spline patch curve for further patching. This requires 3 or 4 vertices. For more information, see Patch Curves.
Text	Copy text polygons from the source mesh to the procedural mesh. Text polygons contain metadata in the form of polygon tags that store all the information about the font used, size, kerning amount, and so on. For more information, see Text.
Catmull-Clark	Copy Catmull-Clark patches from the source mesh to the procedural mesh. Catmull-Clark patches are Pixar Catmull-Clark smoothed polygons. For more information, see Set Polygon Type.
Polyline	Copy line polygons from the source mesh to the procedural mesh. A polyline is a continuous line composed of one or more line segments. For more information, see Line.

Curve Fill	Copy curve fill polygons from the source mesh to the procedural mesh. Curve fill polygons are closed, non-intersecting curves with quadrangles. For more information, see Curve Fill.

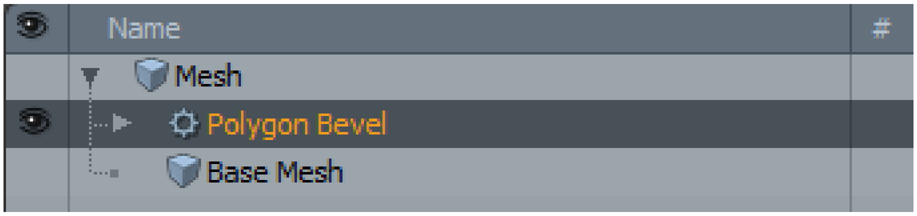
Freezing Mesh Operations

When you have a long list of mesh operation, it may be useful to simplify your stack. Modo allows you to consolidate your mesh up to a particular operation, using the Freeze command. This way, Modo only evaluates the stack until the selected operation.

You can freeze mesh operations in the **Mesh Operations** list by right-clicking the last operation you'd like to include in the evaluation, then clicking **Freeze**.


Name
#
Mesh
Polygon Bevel
Polygon Extrude
Toroid
Base Mesh
Rename
Editor Color
Properties...
Change Type
Add Override
Normalize
Place in Folder
Bake Weights
Clear Weight Container
Duplicate
Duplicate Hierarchy
Delete
Freeze
Duplicate and Freeze
Create Assembly
Collapse to Assembly

Modo freezes the mesh of the selected operation, which means that all previous operations are pushed back into the base mesh.

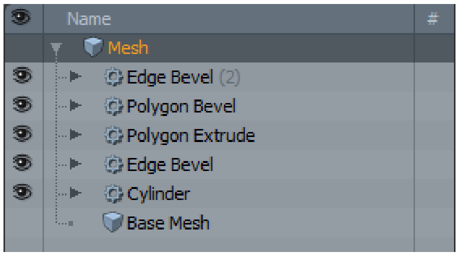

Name
#
Mesh
Polygon Bevel
Base Mesh

Duplicate and Freeze

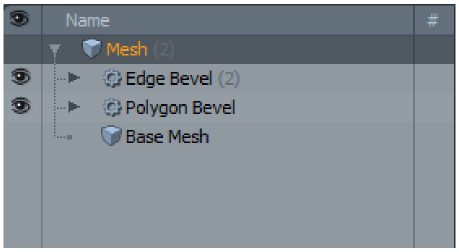
The **Duplicate and Freeze** command is an alternative to **Freeze**. This freezes mesh operations the same way as **Freeze**, but it also creates a copy of the original mesh. This way you can still preserve your fully-editable mesh.

Modo creates a new mesh layer with the collapsed **Mesh Operations** list, while preserving the original mesh stack.

The image below shows the frozen and the duplicated mesh layers in the **Mesh Operations** list.


Name
#
Mesh
Edge Bevel (2)
Polygon Bevel
Polygon Extrude
Edge Bevel
Cylinder
Base Mesh

The **Mesh Operations** list before Freeze


Name
#
Mesh (2)
Edge Bevel (2)
Polygon Bevel
Base Mesh

The **Mesh Operations** list with the Freeze command applied at the Polygon Bevel level

Booleans

With the Boolean command you can combine mesh layers in various ways (**Union**, **Intersect**, **Subtract**, and **Add**). By doing so, you can create more complex shapes that would otherwise be difficult to create manually.

Usage

Any mesh that you want to use for a Boolean operation needs to fulfill specific criteria to work properly:

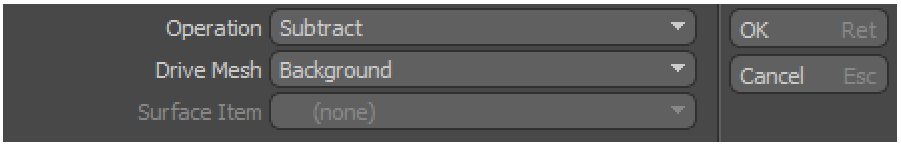
- Meshes must not have coincidental overlapping (non-manifold) geometry. This means no vertices, edges, or faces that share the same position in space.
- The mesh should not have any floating vertices.
- Both the driver and driven mesh must have some volume and must be water-tight (no open edges), which means that if the geometry was a filled solid, nothing could leak from it.

Note: It might be useful to run the Mesh Cleanup script before attempting a Boolean operation to ensure a clean, error-free mesh.

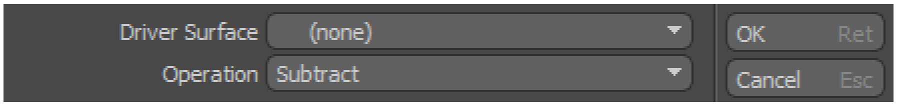
Once you make sure that the geometry is correct for the Boolean operation, place the driven mesh to the foreground and the driver mesh to the background (unless specified differently). You can do this through the **Item List**. Foreground layers are selected and visible. Background layers are visible but not selected. For information on working with layers, see the Items (Scenes) List topic.

Tip: It may be helpful and speed up the procedure to disable any unnecessary background layers. To disable a background layer, drag over the eye column of the **Item List**.

Once you have set the geometry and assigned the foreground and background layers, you can invoke the Boolean tool. It is in the **Mesh Edit** sub-tab of the default **Model** layout toolbox. Clicking the button opens the **Boolean CSG** dialog where you can select the desired **Operation** and then click **OK** for Modo to perform the Boolean operation. For information on the direct modeling Boolean tool options, see Boolean Properties - Direct Modeling Tool.


Operation Subtract
Drive Mesh Background
Surface Item (none)

The Boolean tool is also available in the **Mesh Operations** list. Open the **Mesh Ops** tab, click **Add Operator**, and select **Mesh Operations** > **Edit** > **Boolean**. Once done, the **Create Boolean Mesh Operator** dialog box is displayed. From the **Drive Surface** drop-down menu, select the mesh item in your scene, from the Operation drop-down menu select and click **OK**. For information on the procedural Boolean tool options, see Boolean Properties - Procedural Modeling Tool.


Driver Surface (none)
Operation Subtract

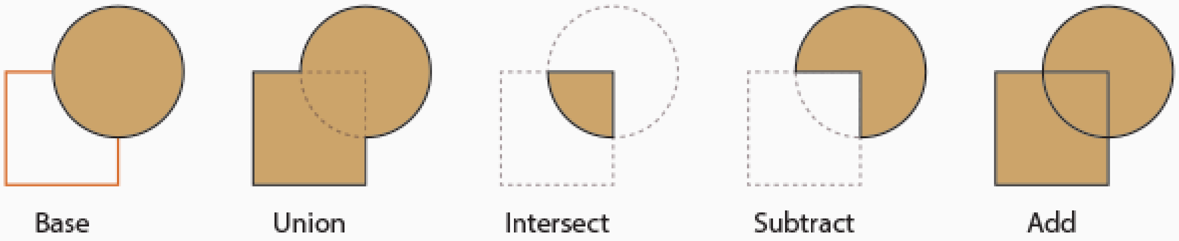
Tip: To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove.

You can also use the **Select by Previous Operation** mesh operation. The following **Name** options are available for the **Boolean** tool: **Intersecting Edges**, **Polygons from Geometry**, and **Polygons from Driver Surface**. For more information, see Select by Previous OperationProcedural Selection

Tip: The results produced by Boolean operations are not generally compatible with subdivision surfaces. Depending on the shapes and how they overlap, the result of the Boolean operation may produce oddly shaped polygons with more than four edges that tend pinch and pucker. You may have to do some manual cleanup to generate results to use a subdivision surfaces model. To produce a rounded edge at an intersection, you might try applying the Edge Bevel tool to the new edges from the Boolean operation (if the original geometry was simple enough). You could get better visual results by applying the **Rounded Edges** function of the material item. Consider also the Axis Drill and Solid Drill commands, which perform similar operations without merging the geometry.

Boolean Properties - Direct Modeling Tool

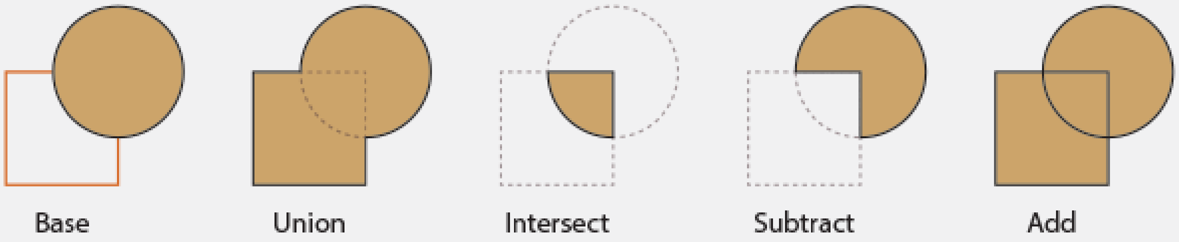
The following properties are available:

Option	Description
Operation	Determines how Modo treats the different layers.  • The **Base** example shows the square shape as the background element with the circle in the foreground. The other examples show the results of the different operations. • **Union** - Combines the two layers together as a seamless mesh and removes any overlapping geometry. • **Intersect** - Results in a mesh created only in areas where the two layers overlap. Modo

	removes all sections of the two layers that do not overlap. • **Subtract** - Removes the background layer from the foreground layer to cut away any section of the foreground that overlaps with the background layer. Modo creates new faces to close the area where the background layer cuts through the foreground. • **Add** - Combines the geometry from both layers into a single mesh. Modo retains all geometry and creates extra vertices to ensure the meshes are stitched together.
Drive Mesh	Determines which mesh has an effect over the Boolean operation. • **Background** - Controls the driven mesh (as specified by the item's background or foreground designation in the **Item List**). This is the default. • **Last Selected** - Uses the mesh layer you most recently selected before invoking the Boolean command. • **Specific Surface Item** - Designates a specific mesh layer (which can include groups of multiple items defined in the Groups palette) to act upon the current active layer.
Surface Item	Specifies which Mesh Item layer Modo uses to drive the Boolean operation. This is available if you have selected **Specific Surface Item** as the **Drive Mesh** option. To use a group, you must specify it here.

Boolean Properties - Procedural Modeling Tool

The following properties are available:

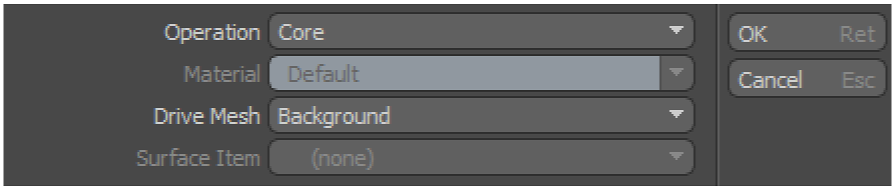
Option	**Description**
Driver Surface	Determines which mesh has an effect over the Boolean operation.
Operation	Determines how Modo treats the different layers.  • **Union** - Combines the two layers together as a seamless mesh and removes any

	overlapping geometry. • **Intersect** - Results in a mesh created only in areas where the two layers overlap. Modo removes all sections of the two layers that do not overlap. • **Subtract** - Removes the background layer from the foreground layer to cut away any section of the foreground that overlaps with the background layer. Modo creates new faces to close the area where the background layer cuts through the foreground. • **Add** - Combines the geometry from both layers into a single mesh. Modo retains all geometry and creates extra vertices to ensure the meshes are stitched together.

Solid Drill

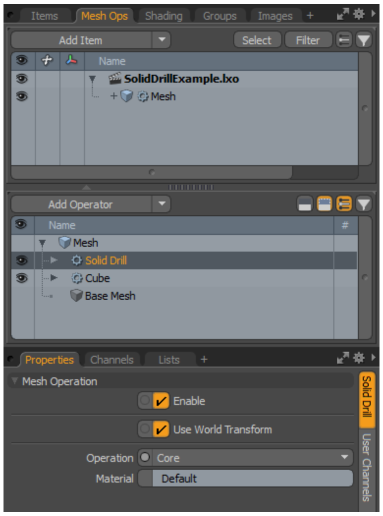
The Solid Drill command performs Boolean operations onto flat 2D polygons by using overlapping 3D geometry. Modo uses the background layer geometry as the Boolean drill operator on the foreground geometry.

For direct modeling, this tool is available in the modeling Toolbox. Open the **Mesh Edit** tab and click **Solid Drill**.



Operation: Core
Material: Default
Drive Mesh: Background
Surface Item: (none)

The **Solid Drill** tool is also available in the **Mesh Operations** list. Open the **Mesh Ops** tab, click **Add Operator** and select **Mesh Operations** > **Mesh Operations/Edit** > **Solid Drill**. Once done, you are presented with the Create Solid Drill Mesh Operator dialog box. Set the **Drive Surface** to any mesh item you have in your scene and click **OK**.


Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
Name
SolidDrillExample.lxo
Mesh
Add Operator
Name
#
Mesh
Solid Drill
Cube
Base Mesh
Properties
Channels
Lists
Mesh Operation
Solid Drill
User Channels

☑ Enable

☑ Use World Transform

Operation: Core

Material: Default

> **Tip:** To improve your modeling workflow, use the procedural modeling **Delete** operation. For more information, see Edit Commands - Delete/Remove.
>
> You can also use the **Select by Previous Operation** mesh operation. The following **Name** options are available for the **Solid Drill** tool: **Intersecting Edges**, **Stencil Polygons**, and **Polygons from Geometry**. For more information, see Select by Previous Operation.

In addition, the **Solid Drill** tool is also available in the Schematic Viewport. Click **Add** and select **Procedural Modeling** > **Edit** > **Solid Drill**.

Solid Drill Properties

The following options vary, depending on the type modeling tool selected (Traditional Toolbox modeling tool vs Procedural Modeling Mesh Operations tool):

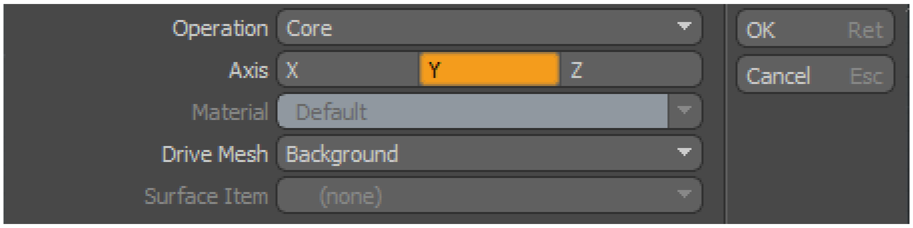
Option	**Description**
Enable	Enables or disables the Solid Drill tool.
Use World Transform	Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Operation	Specifies the type of **Solid Drill** operation to apply to geometry. • **Core**: Cuts away everything outside the 2D outline projected onto the

	polygons. - **Stencil**: Projects the 2D outline onto the polygons without removing either the inside or outside. Applies a default material to the new geometry within the outline. - **Tunnel**: Cuts a hole using the 2D outline and removes the inside. - **Slice**: Projects the 2D outline onto the polygons without removing either the inside or outside. This is the same as **Stencil**, but it does not affect the surface material.
Material	Specifies the material name used with Stencil mode. This is applied to the new polygons.
Drive Mesh	Determines which mesh has an effect over the Axis Drill operation. - **Background**: Controls the driven mesh (as specified by the item's background or foreground designation in the **Items** list). This is the default. - **Last Selected**: Uses whatever geometry you most recently selected before invoking the Boolean operation. With this, you can apply Boolean operations within a single Mesh Item layer. - **Specific Surface Item**: Designates a specific mesh layer (which can include groups of multiple items defined in the Groups palette) to act upon the current active layer.
Surface Item	Specifies which Mesh Item layer Modo uses to drive the Axis Drill operation. This is available if you have selected **Specific Surface** Item as the **Drive Mesh** option. To use a group, you must specify it here.

Axis Drill

With the Axis Drill tool you can perform Boolean-like operations with 2D polygons projected down a specified axis. Modo uses the background layer geometry as the Boolean operator onto the foreground geometry.

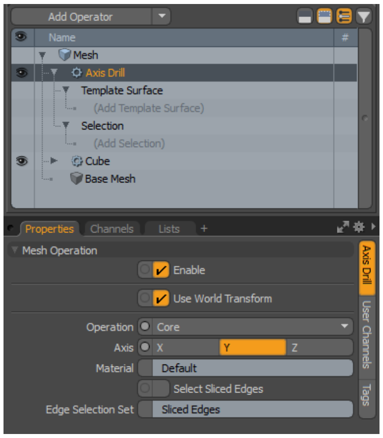
For direct modeling, this tool is available in the modeling Toolbox. Open the **Mesh Edit** tab and click **Axis Drill**.


OK Ret
Cancel Esc

Operation: Core
Axis: X Y Z
Material: Default
Drive Mesh: Background
Surface Item: (none)

The **Axis Drill** tool is also available in the **Mesh Operations** list. Open the **Mesh Ops** tab, click **Add Operator**, and select **Mesh Operations** > **Edit** > **Axis Drill**. The **Create Axis Drill Mesh Operator** dialog is displayed. From the **Template Surface** dropdown list, select a mesh in your scene and click **OK**.

Tip: To improve your modeling workflow, use the **Select by Previous Operation** mesh operation. The following **Name** options are available for the **Axis Drill** tool: **Intersecting Edges** and **Stencil Polygons**. For more information, see Select by Previous Operation.


Add Operator
Name
Mesh
Axis Drill
Template Surface
(Add Template Surface)
Selection
(Add Selection)
Cube
Base Mesh
Properties
Channels
Lists
+
Mesh Operation
Axis Drill
User Channels
Tags

☑ Enable
☑ Use World Transform
Operation: Core
Axis: X Y Z
Material: Default
☐ Select Sliced Edges
Edge Selection Set: Sliced Edges

Axis Drill Properties

The following options vary, depending on the type of modeling tool selected (Traditional Toolbox modeling tool vs Procedural Modeling Mesh Operations tool):

Enable: Activates/Deactivates tool. By default, this property is enabled.

Use World Transform: Center a geometry selection at the World Origin position. By default this property is enabled.

Operation: Select the type of Axis Drill operation to apply to the geometry.

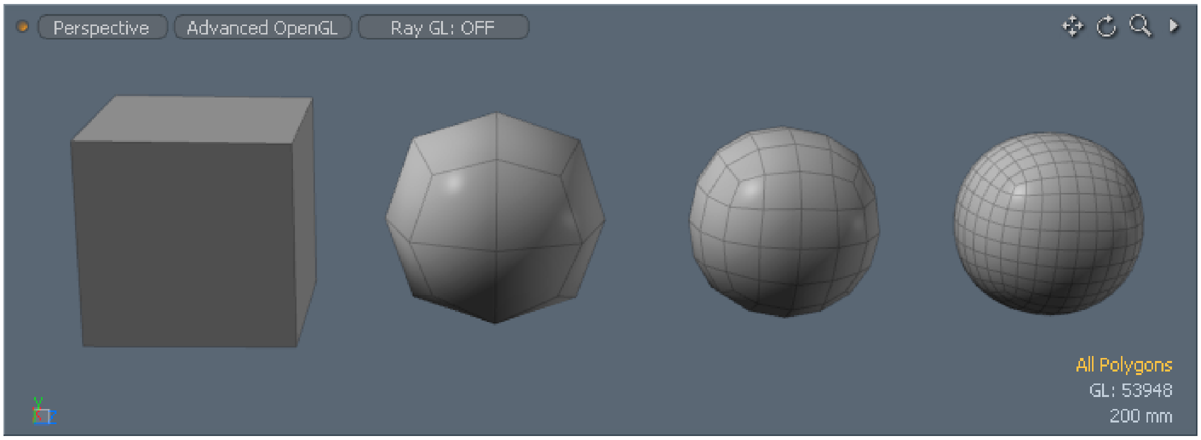
- **Core**: Cuts away everything outside the 2D outline projected onto the polygons.
- **Stencil**: Projects the 2D outline onto the polygons without removing either the inside or outside, and applies a default material to the new geometry within the outline.
- **Tunnel**: Cuts a hole using the 2D outline and removes the inside.

  - **Slice**: Projects the 2D outline onto the polygons. This is the same as **Stencil**, but **Slice** does not affect the surface material.
- **Axis**: Defines the axis across which to project the drill function.
- **Material**: Specify the material name used with **Stencil** mode. This is applied to the new polygons.
- **Select Sliced Edges**: Sets edge selections for sliced edges to the selection set. The selection is automatically created and displayed in the **Mesh Ops** tab under **Lists**.
- **Edge Selection Set**: Define the name of the edge selection set for the **Select Sliced Edges** option. This name can be entered in the **Add Operator** > **Mesh Operations** > **Selection** > **Select By Selection Set** which can be used with Edge mesh operations.
- **Drive Mesh**: Determines which mesh has an effect over the Axis Drill operation.
  - **Background**: Controls the driven mesh (as specified by the item's background or foreground designation in the **Items** list). This is the default setting.
  - **Last Selected**: Uses whatever geometry you most recently selected before invoking the Boolean operation. With this, you can apply Boolean operations within a single Mesh Item layer.
  - **Specific Surface Item**: Designates a specific mesh layer (which can include groups of multiple items defined in the Groups palette) to act upon the current active layer.

- **Surface Item**: Specifies which Mesh Item layer Modo uses to drive the Axis Drill operation. This is available if you have selected **Specific Surface Item** as the **Drive Mesh** option. To use a group, you must specify it here.

Note: The **Drive Mesh** and **Surface Item** properties are only available from the Direct Modeling Toolbox.

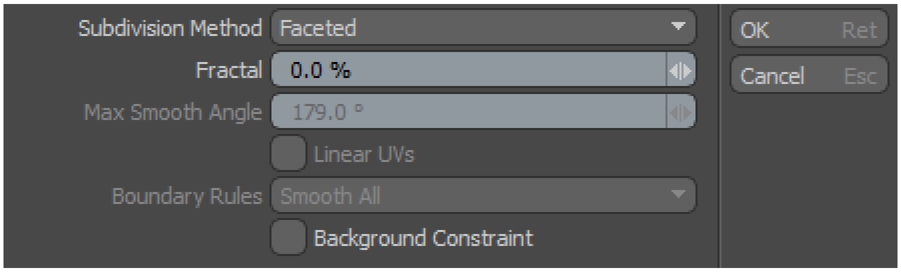
Subdivide



Subdividing polygons recursively splits the selected polygons into smaller polygons. A single subdivision increases the polygon count by a factor of four; Modo divides each quad polygon into four smaller polygons.

To subdivide:

- select **Subdivide** from the **Mesh Edit** subtab, or
- press **D** while in **Polygons** component mode, or
- press **Shift**+**D** to open the **Subdivide Polygons** dialog, or
- choose **Geometry** > **Polygon** > **Subdivide**.



Subdivision Method: Faceted

Fractal: 0.0 %

Max Smooth Angle: 179.0 °

☐ Linear UVs

Boundary Rules: Smooth All

☐ Background Constraint

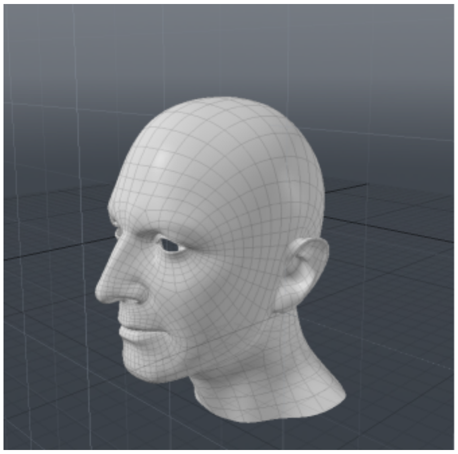
Subdivide Polygons Properties

- **Subdivision Method**: The type of subdivision to apply to the geometry.
  - **Faceted** - Creates new polygons with vertices that are perfectly aligned with the plane of the original polygon. Visually, there is no significant change to the shape of the model (unless these polygons are Subdivision Surface polygons) other than added vertices (increased resolution) for further deformation.
  - **Smooth** - Uses the **Max Smoothing Angle** value to determine where there are edges in the model that are beyond the sharpness threshold and should not be smoothed. Modo breaks all other polygons that meet with edges below the **Max Smooth Angle** not only into 4 additional polygons, but Modo moves the resulting vertices slightly to round out (or smooth) the surface of the mesh. To see what this option does, create a sphere with few sides and segments, and then apply **Subdivide Polygons** with the **Smooth** option.
  - **SDS Subdivide** - Moves all points in the mesh, including the original points, towards the limit surface that Modo derives by using the original mesh as the control points. This is similar to the **Smooth** option, and it gives a similar result to activating Subdivision Surfaces with a subdivision level of 1 and then freezing the geometry. This adds uniform density to an entire mesh when working with subdivision surfaces.
  - **Catmull-Clark** - Provides smoothing of the selected limit surface, although this algorithm has slightly different results than Modo's **SDS Subdivide** option. Developed by Pixar's Ed Catmull and Jim Clark, this subdivision type provides some benefits in that it distorts the UV map much less and you can apply **Edge Creasing**.

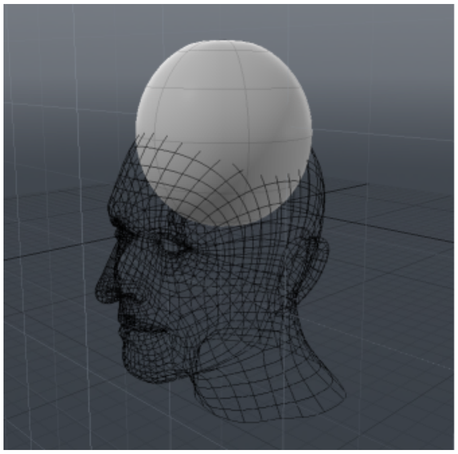
- **Fractal**: Adds some randomization to the position of points after Modo subdivides them. This can be useful when creating organic objects such as rough terrains or rocks.
- **Max Smoothing Angle**: Determines when the **Smooth** option should move the newly created vertices. If the angle between the original vertices is greater than the **Max Smoothing Angle**, the new vertices fall in the same plane as the original vertices. For example, with a cube that has right angles, if the **Max Smoothing Angle** is less than 90 degrees, the new vertices are in the same plane without any rounding effect.
- **Linear UVs**: When enabled, interpolates new values by retaining the initial UV positions without smoothing. Use with baked maps (such as a normal map or a diffuse color map for a game character) that require fixed polygon positions. When disabled, the additional, newly created vertices from subdividing polygons require UV values (to not break the existing UVs maps), which are smoothly interpolated based on existing neighboring values. This provides the best, most pleasing results (except for baked maps).
- **Boundary Rules**: Determines how to move the points on the boundaries of a mesh. Creasing produces a sharper transition between surfaces; smoothing produces smooth surface transitions.
    - **Smooth All**- Moves all positions on the boundary based on the smooth rule of Pixar Subdivision Surfaces.
    - **Crease All**- Moves the points that have two incident edges (valence = 2) based on the crease rule. Modo smooths other positions on the boundary by the two boundary edges.
    - **Crease Edges**- Moves the points that have two incident edges (valence = 2) based on the smooth rule. Modo smooths other positions on the boundary by two boundary edges.
- **Background Constraint**: Select this to use a background mesh to constrain the subdivision. When subdivided, the mesh will attempt to mold itself to the contours of the background mesh. This is useful if you already have a mesh and you want another mesh to fit it.

Note: The effect is similar to molding over a background mesh by constraining to the background when transforming. For more information, see Constrain to Background.

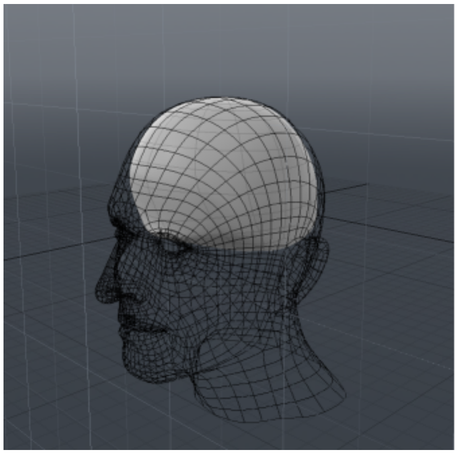
The images below show an example.



A head mesh.



A sphere is added to the layer above the head.



The sphere is subdivided with **Background Constraint** checked,

Subdivide in the Procedural

Modo has a procedural Subdivide mesh operation available in the **Mesh Operations** list. In the **Model** layout, you can find the stack in the lower half of the **Mesh Ops** tab on the right side of the interface.

Note: For information on how to use the **Mesh Ops** tab, see Using the Mesh Ops Tab.

You can add the tool by clicking the **Add Item** button above the **Mesh Operations** list. Under **Mesh Operations** > **Polygon**, double-click **Subdivide**.

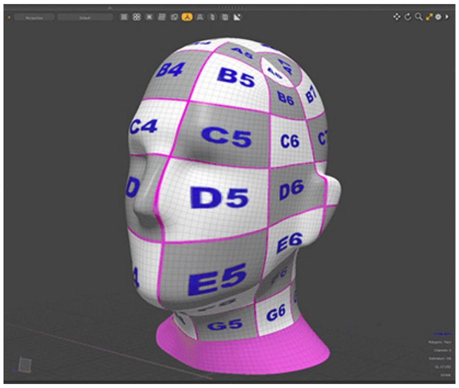
You can expand the Subdivide item in the stack by clicking the small arrow in front of it. This reveals the inputs the operation uses. Subdivide requires the following inputs:

- **Selection** - Allows you to select the polygons you want to subdivide.
- **Tool Pipe** - Allows you to add falloffs and sub-tools to the operation.

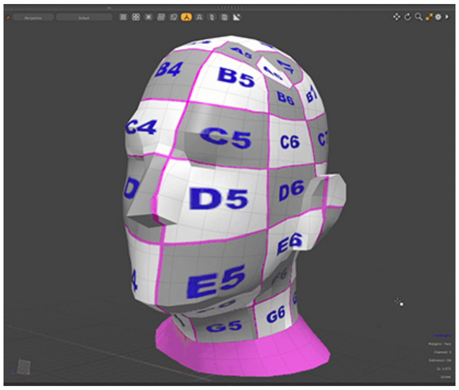
The procedural Subdivide mesh operation has the same properties as the traditional tool. You can find the properties in the **Properties** panel below the **Mesh Operations** list when the item is selected.

UnSubdivide

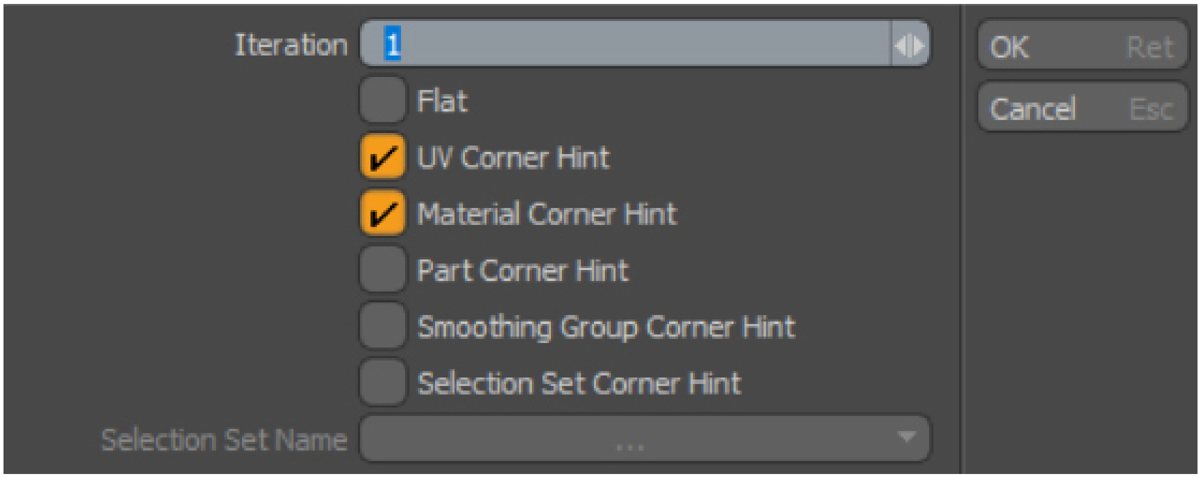
The **UnSubdivide** command reconstructs subdivided polygons using the Catmull-Clark algorithm in reverse. This operation locates four quadrangle pairs from the source subdivided mesh, then merges them and repositions the corner points to their original positions before subdividing. **UnSubdivide** can be applied both in direct modeling and procedurally as a Polygon Mesh Operation.



Subdivided.



UnSubdivided using 2 Iterations.



Iteration 1

☐ Flat

☑ UV Corner Hint

☑ Material Corner Hint

☐ Part Corner Hint

☐ Smoothing Group Corner Hint

☐ Selection Set Corner Hint

Selection Set Name ...

Iteration sets the number of times the operation is repeated.

To recreate the original cage mesh, **UnSubdivide** expands quad pairs of polygons starting from initial pairs. The **UnSubdivide Polygon** menu offers additional options to find the first 4 quadrangle pairs.

Flat - Disable to move vertex positions.

UV Corner Hint - Use UV corner to locate 4 quadrangle pairs at root level.

Material Corner Hint - Use Material corner to locate 4 quadrangle pairs at root level.

Part Corner Hint - Use Part corner to locate root quadrangles.

Smoothing Group Corner Hint - Use Smoothing Group Corner to locate 4 quadrangle pairs at root level.

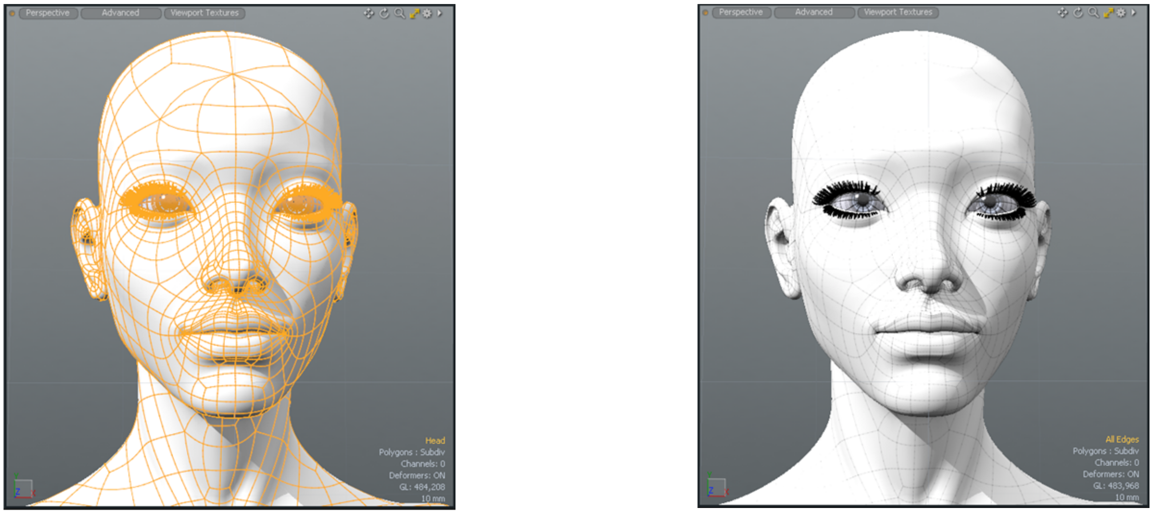
Selection Set Corner Hint - Use Polygon Selection Set corner to locate 4 quadrangle pairs at root level.

Selection Set Name - Select a named polygon Selection Set.

Note: Gaps may appear in cases where some polygons are not merged in quad pairs.

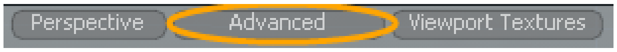
Fix Black Dots

Use the **Fix Black Dots** tool to correct topology, removing pinched areas and holes from your mesh item.


Perspective
Advanced
Viewport Textures
Head
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 484,208
10 mm
All Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 483,968
10 mm

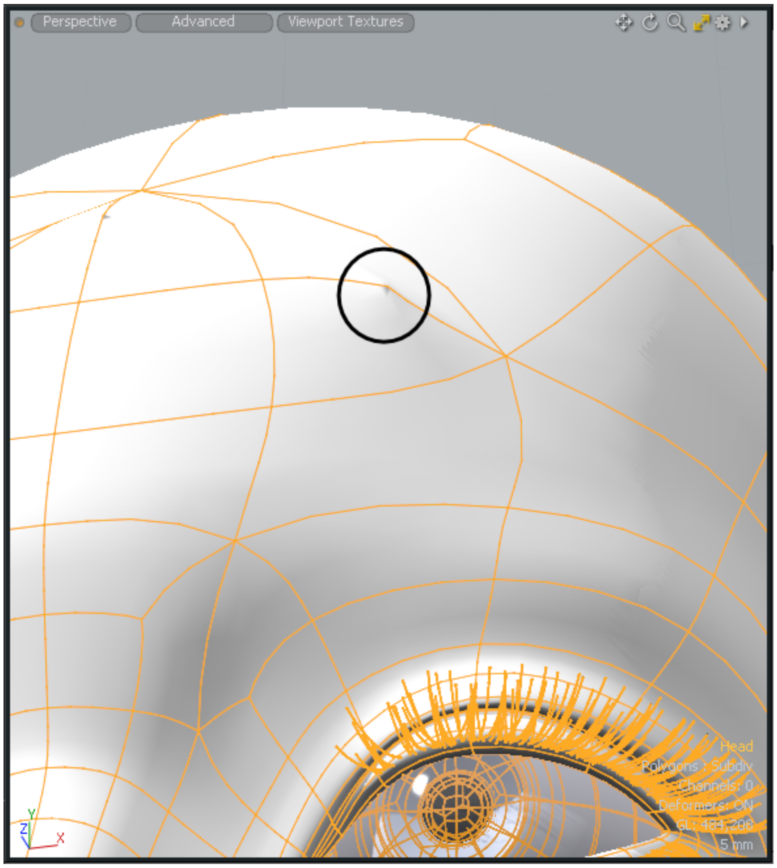
Before After

1. Download our example scenes and open **fix_black_dots.lxo**.
2. In the **Model** layout, set the viewport style to **Advanced**.


Perspective
Advanced
Viewport Textures

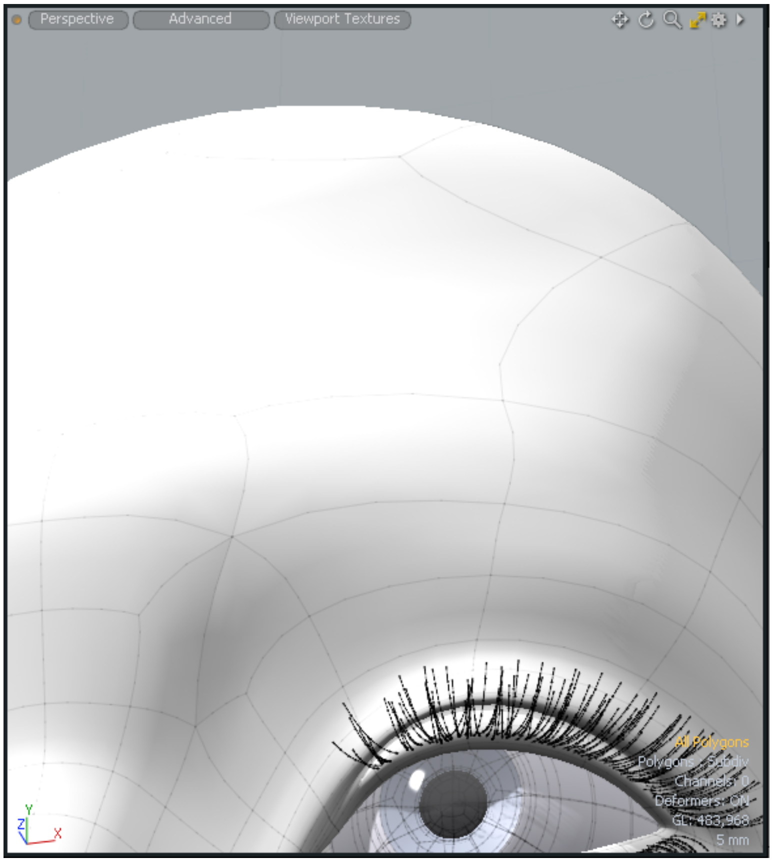
3. Zoom into the forehead.

   Pinches are displayed on either side of the forehead.


Perspective
Advanced
Viewport Textures
Y
Z
X
Head
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 484,208
5 mm

4. On the left panel, open the **Mesh Edit** tab, and click **Fix Black Dots**.

   The model is retopologized and the dents are not longer displayed.

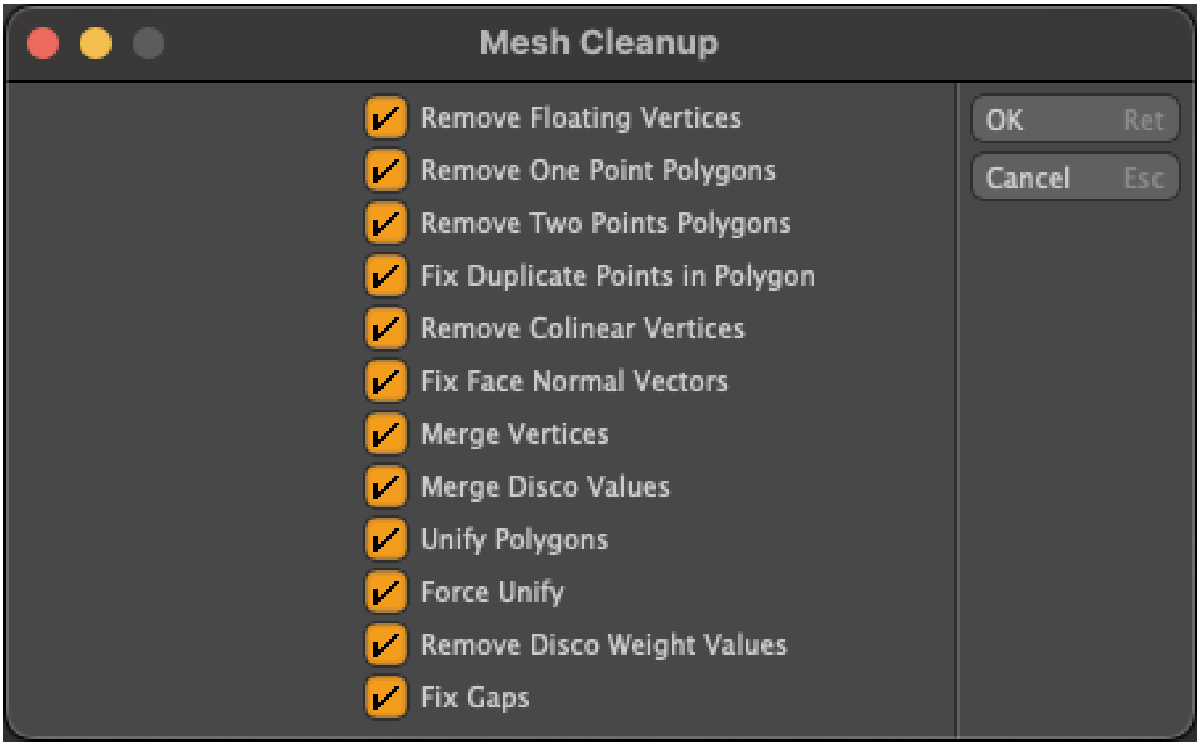

Perspective
Advanced
Viewport Textures
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 483,968
5 mm
Y
Z
X

Other Geometry Editing Tools

The following topics describe how you can cut, copy, paste, delete and remove geometry in Modo, and how you can do a mesh cleanup.

Mesh Cleanup

The **Mesh Cleanup** function wraps a bunch of useful functionalities into a single command, aiding you in repairing problem geometry. Inevitably, even the most conscientious modelers eventually creates some geometry that contains collocated polygons, random floating vertices, or other errors that can build up and may eventually cause rendering errors among other problems. The **Mesh Cleanup** command is a one-stop-shop for error elimination. You can simply invoke the command found in the menu bar under **Geometry** > **Mesh Cleanup** to open the **Mesh Cleanup** dialog. You can then toggle the different options on or off, if applicable, and press **OK** to apply the command to the currently-selected geometry.



☑ Remove Floating Vertices
☑ Remove One Point Polygons
☑ Remove Two Points Polygons
☑ Fix Duplicate Points in Polygon
☑ Remove Colinear Vertices
☑ Fix Face Normal Vectors
☑ Merge Vertices
☑ Merge Disco Values
☑ Unify Polygons
☑ Force Unify
☑ Remove Disco Weight Values
☑ Fix Gaps

The different options include:

- **Remove Floating Vertices** - removes single vertices that aren't connected to any polygons.
- **Remove One Point Polygons** - removes vertices that may be left behind when deleting polygon faces.
- **Remove Two Point Polygons** - removes edges of polygons where there is a visible line connecting vertices, but no faces.
- **Fix Duplicate Points in Polygons** - eliminates invalid duplicated vertices within a polygon, merging them into a single vertex.
- **Remove Colinear Vertices** - removes unnecessary extra vertices along an edge.
- **Fix Face Normal Vectors** - shifts the vertex order of face polygons in order to compute the correct normal vector. This resolves issues when first, second, and last vertices are arranged in a straight line.
- **Merge Vertices** - combines vertices that have identical coordinates (collocated).
- **Merge Disco UVs** - each vertex can store multiple UV values, one set for each connected face. In cases where identical values are stored as multiple discontinuous values to a single vertex, this option eliminates them, merging the duplicates and optimizing the UV map.
- **Unify Polygons** - removes duplicated (collocated) polygons that share identical Vertex Map values and tags.
- **Force Unify** - works in conjunction with **Unify Polygons** but removes all duplicated (collocated) polygons, ignoring Vertex Map values and tags.
- **Remove Disco Weight Values** - removes unnecessary discontinuous weight values from all Weight Maps in the active meshes. Discontinuous values are automatically generated for various purposes, but can occasionally cause problems in standard workflows.

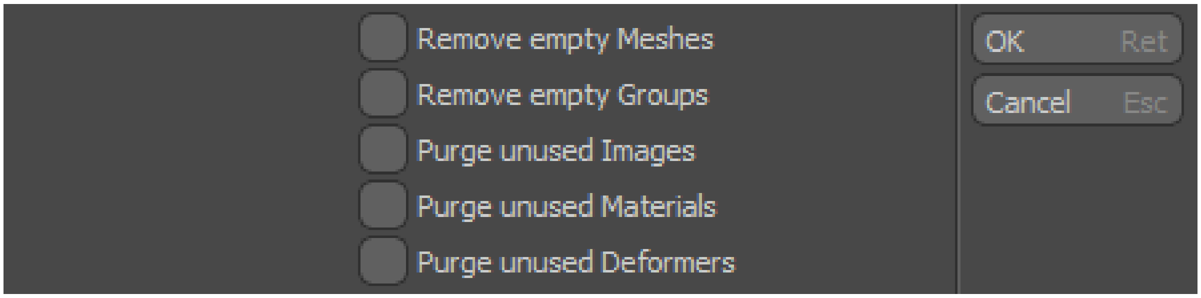
- **Fix Gaps** - repairs gaps when co-linear vertices only belong to one side of shared polygons. When this type of gap is detected on an edge, Fix Gaps adds co-linear vertices to the vertex list of the connecting polygon.

Tip: Mesh Cleanup can also be added to Mesh Operations to work with procedural meshes.

Scene Cleanup

The **Scene Cleanup** function is useful for removing empty mesh items, empty groups, and unused items your current scene. The function is available in the **Item** menu by clicking **Scene Cleanup**. This function can also be run in a command line.

The following options are available:



☐ Remove empty Meshes
☐ Remove empty Groups
☐ Purge unused Images
☐ Purge unused Materials
☐ Purge unused Deformers

Option	Description
Remove empty Meshes	Removes meshes that have no geometry.
Remove empty Groups	Removes groups that have no children.
Purge unused Images	Removes clips that are not used by any other item.
Purge unused Materials	Removes materials that are not used (similar to **Texture** > **Purge Unused Materials**)
Purge unused Deformers	Removes deformers that are not used by any other scene item.

Running this Function in a Command-line

You can also run **scene.cleanup** in a command-line to apply this function.

The following arguments are available:

- **emptyMesh:true/false**
- **emptyGroup:true/false**
- **unusedNodes:true/false**
- **unusedImages:true/false**
- **unusedMaterials:true/false**
- **unusedDeformers:true/false**

To run the command:

- With your scene open in Modo, on the right panel, type the following into the command line:

scene.cleanup <emptyMesh_value> <emptyGroup_value> <unusedImages_value> <unusedMaterials_value> <unusedDeformers_value>

For example, type **scene.cleanup true true true true true**

Alternatively, type **scene.cleanup** to display the **Scene Cleanup** dialog and select the options you wish to apply.

Specifying Action Centers and Falloffs

Learn about how to define action centers and falloffs.

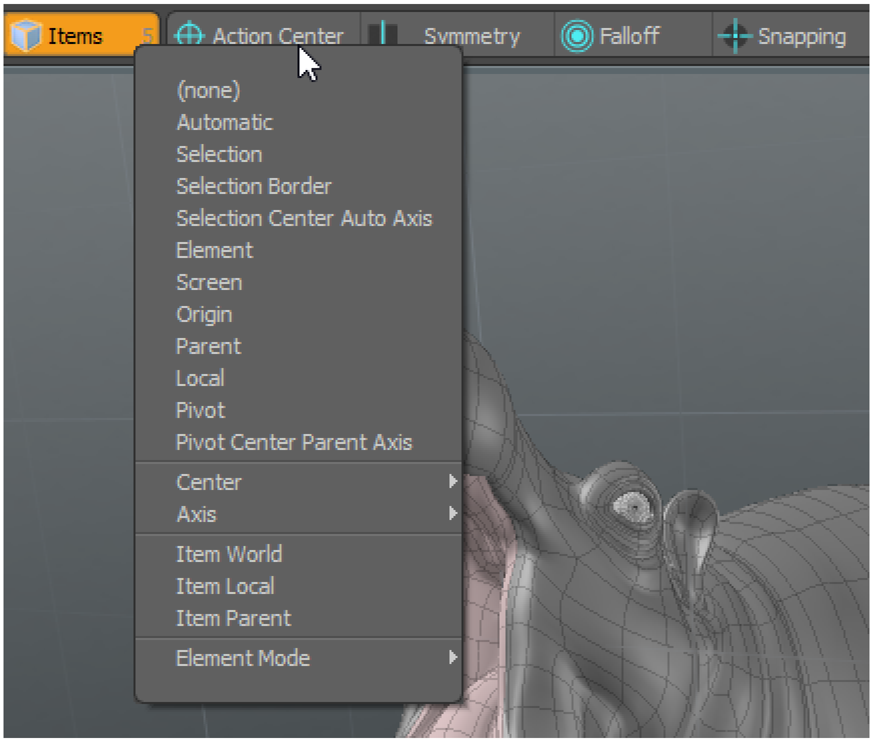
Activating Action Centers

Action centers are a definable origin position for modeling and transform actions in Modo. While other 3D software programs use the fixed mesh geometry pivot point as the center for most actions, Modo separates the Action Center into an independent dynamically definable component. This provides a means to easily change the positions from where the action of any tool originates. You can activate action centers in the **Action Center** dropdown menu, or in the menu bar, under **Edit** > **Action Center**. You can also break an action center further into its center and axis components for even greater control.

Generally, you can't see anything happen in the 3D viewport when selecting an action center until a tool is made active, as action centers themselves are represented by the positions of an active tool's handle. When you select a new action center, the position and orientation of the handle updates to reflect the new axis and center. To move your object in the 3rd dimension (forward and backward) in the **3D Viewport** press **Shift** + **Ctrl/Cmd** + **right-mouse-button** and drag.

Action Center Types

The following table lists the available action center types.


Items
Action Center
Symmetry
Falloff
Snapping
(none)
Automatic
Selection
Selection Border
Selection Center Auto Axis
Element
Screen
Origin
Parent
Local
Pivot
Pivot Center Parent Axis
Center
Axis
Item World
Item Local
Item Parent
Element Mode

Type	Description
None	When no action center is selected, Modo defaults to **None**, which is similar to **Automatic**, except there is no ability to explicitly define the axis.
Automatic	When a tool is activated, the **Automatic** action center positions the tool's handle at the center of the current selection (or on all active geometry in the case of no selection) with the axis aligned to the World or Work Plane axes. You can explicitly select the axis in the properties panel and interactively modify it by adjusting the orientation handle, the

Type	Description
	thin blue line with the yellow dot. The **Automatic** action center is not fixed, like others action centers are. You can define a new center by clicking away from the tool handle, setting the positions at the intersection of where the pointer clicked in the viewport and the Work Plane.
Selection	Bases the action center on the average vertex position of the selection in the **3D Model View**, effectively centering the action center to whatever is selected. Setting the action center to **Selection** changes the tool's handles to have the center and axis aligned to the current selection.
Selection Border	Combines the selection border center with the selection-based axis. This combination centers all tool handles around the border of the selected elements with the orientation of the tool derived from the average normals of all selected elements. The border of selection is defined by the open edge area where the selection stops. This is extremely useful for posing character type operations. For example, you can select a loop of polygons on an elbow, then select the rest of the arm using the **Select** > **Close loop** command, and activate **Selection Border**, which positions a rotate transform handle aligned to the initial loop selection making it easy to bend the arm.
Selection Center Auto Axis	Keeps the center of the tool handles centered on the geometry, but re-aligns the tool handles to the major axes. It provides an axis field in the **Tool Properties Action Axis Auto** section, so you can easily change from one axis to another by clicking the appropriate button.
Element	Allows you to click on any element in the 3D model to set the action center. If you click on a polygon, Modo realigns the tool's handles to its normal. If you click on an edge, or a vertex, Modo aligns the tool handle to the normal of those elements. For example, if you want to stretch an entire model along the normal of any particular polygon, you simply activate the stretch tool, click on the polygon and now you can scale along that axis. **Element** action center does not require the element that is selected to be the element that is used for the action center. Once you have a polygon selected, you can choose **Element** in the **Action Center** field, then click on a different polygon. That polygon is the reference action

Type	Description
	center for transforming the original selected polygon.
Screen	Allows the action center and axis to be based on the frame of the viewport, or screen space. When **Screen Center** and **Axis** is chosen for any transform, the tool handles realign themselves to the screen and the transform becomes in relation to the picture plane rather than to any world or local coordinate system.
Origin	Uses the world coordinates 0,0,0 for the center of any transform, regardless of item or element position. While less useful with the Move transform, this is handy for Rotate or Stretch if you are building something off the origin. You can easily rotate around the center of the 3D view using this setting. It also makes it easy to accurately set functions like Mirror, which happens across the origin's planar axes.
Parent	Uses the pivot position of the parented item, as opposed to the center position. This is useful for transforming items in a hierarchy when you wish to keep items in proper relationship to their parents.
Local	Uses the center of individual element clusters for the operation center, like having a separate axis and center for each selection group. This lets you select multiple elements and have them each rotate around their own local axis. When using the Move tool, the handles are axis-aligned to the last selected polygon. This is particularly useful when working with edges, as you can use the stretch tool to scale several edges simultaneously along each one's individual local axis.
Pivot	Uses the pivot position of the item as the center of any transform. This is especially useful for animated items that might not be near the origin center.
Pivot Center Parent Axis	Uses the current items pivot position as the center, but adjusts the axis to match the parent item.

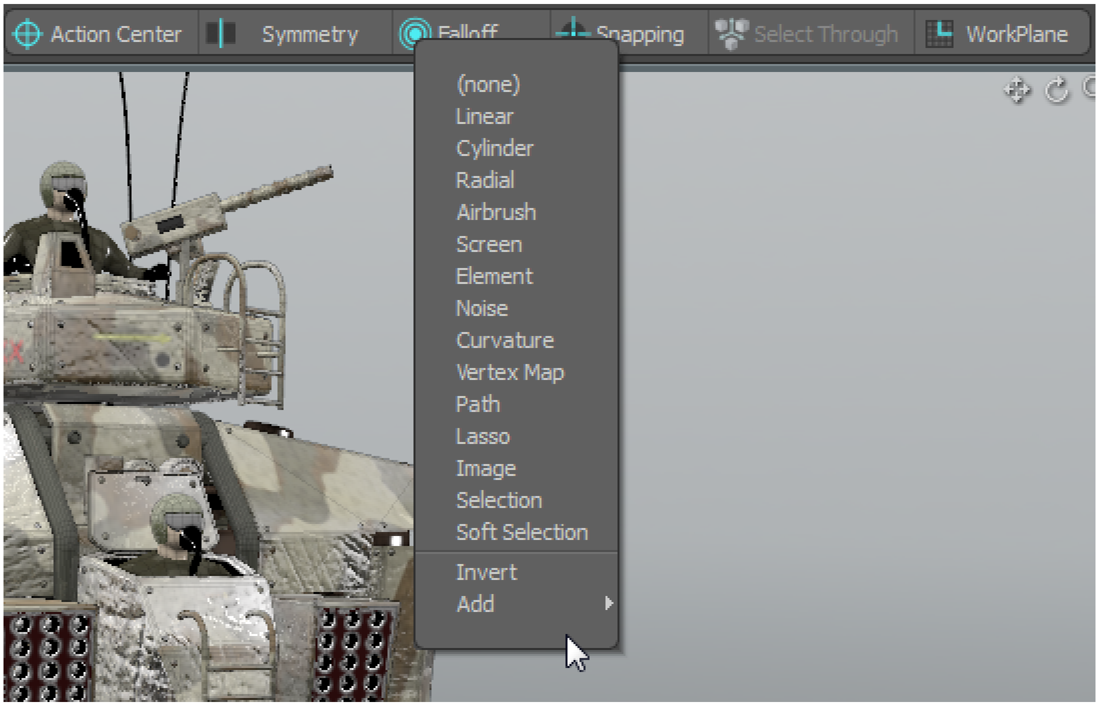
Using Falloffs

In Modo, the effect any tool has upon a selected region of geometry is rigid, meaning applications of a tool happen at a constant rate across the selected range. Modo provides a way to attenuate the effect a tool has over a selection as a falloff, a gradual decrease in intensity across a defined range. Falloffs are universal in application, once activated, they are persistent until disabled, and work with all the various tools. In fact, many tools are simply pre-made combinations of a specific falloff and a transform. For example, if you add the Rotate tool to a Linear Falloff, you get a Twist tool. This is made possible using the power of the Tool Pipe.

You can define a falloff in multiple ways.

1. Select one in the **Falloff** dropdown menu above the 3D viewport, in the **Falloff** selector menu, or in the menu bar under **Edit** > **Falloff**.

   The selected falloff has a checkmark next to its name to indicate its active state.



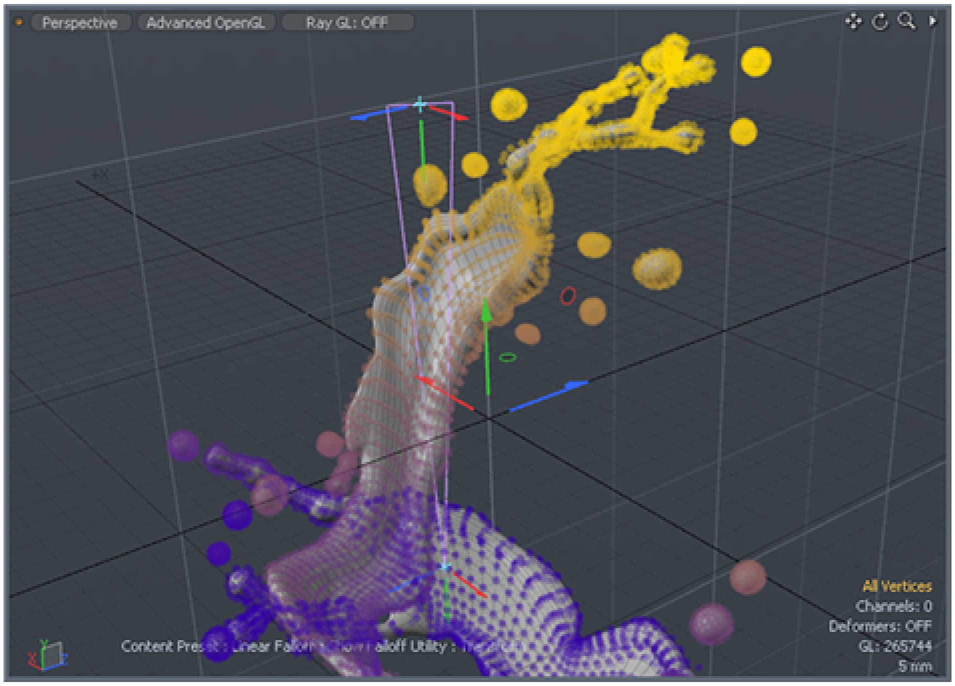
2. Once you've selected a falloff, click and drag in the viewport to define the range.

   Taking its position cue from the Work Plane, the first click defines the origin or center of the falloff, and the drag action defines the size or range.
3. Release the mouse button to accept the falloff.

   Any subsequent tool applications utilize the falloff until it is disabled by selecting **(none)** or pressing **Esc**.

When a falloff is active, its properties are visible in the tool's **Properties** panel as an adjunct to the tool's properties. Each has its own specific settings. Anytime a tool is in interactive mode, you can modify the input values and receive real-time feedback of the results. Additionally, you can adjust the tool directly in the viewport by clicking and dragging the widgets visible on the **Falloff** display.

If a tool is active when setting a falloff, such as the Move or Rotate tools, the act of simply selecting the falloff type in the menu automatically scales the falloff to the bounding box size of the active selection (if nothing is selected, all geometry is affected). This functionality makes it easy to accurately apply falloffs when modeling. To better visualize the effect a falloff has over a selection, enable the **Show Falloffs** option in the menu bar under **View** > **Show Falloffs**. This colorizes the vertices depending on the amount of influence a falloff has over them, dark purple for no influence, transitioning toward yellow for full influence as demonstrated by the image below (with a **Linear** falloff on the Y axis).



> **Note:** Commands such as **Copy**, **Paste**, or **Delete** are typically all or nothing type events and are therefore unaffected by falloffs.

Airbrush Falloff

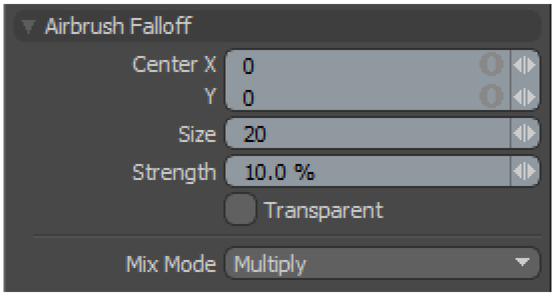
The Airbrush falloff allows a tool's influence to be painted onto a mesh. Unlike the other falloff tools, airbrush continually adds to the tool influence while you drag the brush around the mesh. This tool is similar to the Airbrush tool found in 2D paint programs, except rather than smoothly painting pixels into an image, the brush paints the tool influence into the mesh. This can be combined with many tools in Modo for a wide variety of effects. You can right-click and drag to interactively define the size of the brush applied.

To use the Airbrush falloff:

1. Under the interface layout tabs, click **Falloffs** > **Airbrush** and apply the effect (such as Move).

   At first you can't see any results.
2. Click and drag over the surface of the model to apply the tool's action to the geometry.

Airbrush Properties



Center X 0

Y 0

Size 20

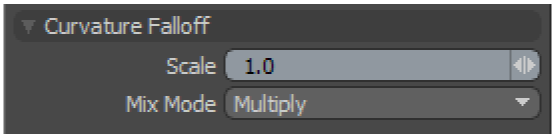
Strength 10.0 %

☐ Transparent

Mix Mode Multiply

Center X/Y	Determines the current center position in the viewport of the brush tip. You can interactively click and drag over a model's surface to actually apply the falloff.
Size	Sets the pixel size of the Airbrush tip's influence, calculated as a radius from the **Center** position.
Strength	Defines the intensity of the falloff's application when brushing over a surface.
Transparent	When disabled, the falloff only affects polygons facing toward the viewport. When enabled, the falloff works through surface also affecting those facing away from the viewports.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.

Curvature Falloff



Scale 1.0

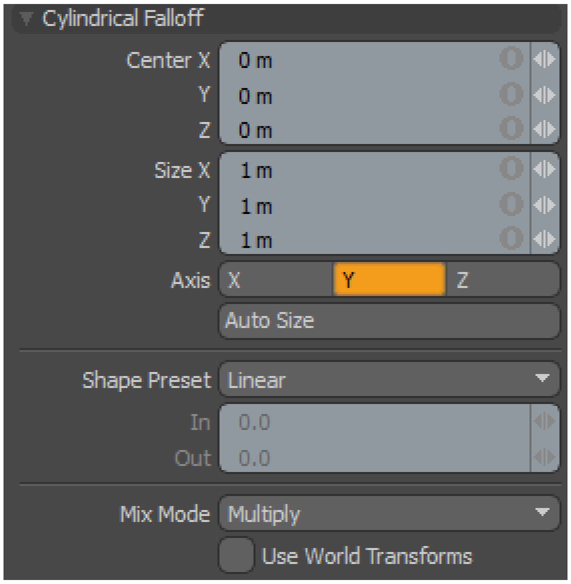
Mix Mode Multiply

The Curvature falloff is defined by surface variation. Computing the same type of effect as the RT Curvature, the Curvature falloff is used to generate a falloff that attenuates across changes in surface topology, especially noticeable in areas of abrupt change or protrusion (producing an effect sometimes called an accumulation or dirt shader). Adjust the **Scale** value to apply the effect to concave or convex areas of the surface.

> **Note:** The curvature is sampled based on continuous surfaces, intersecting surfaces are calculated independently. Also, surfaces with little topological variation (smooth surfaces) likely appear to produce little to no results.

Scale	This value determines the attenuation amount from one area to the next across topological changes. Reducing the scale tends to bias the effect toward concave areas, increasing it biases it outward from them. Setting the scale to negative values inverts the effect where the convex areas get masked by the falloff and the concave areas are affected.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.

Cylinder Falloff


Cylindrical Falloff

Center X 0 m
Y 0 m
Z 0 m

Size X 1 m
Y 1 m
Z 1 m

Axis: X, Y (selected), Z

Shape Preset: Linear

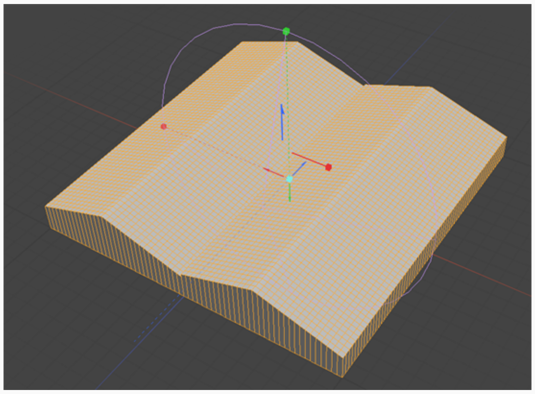
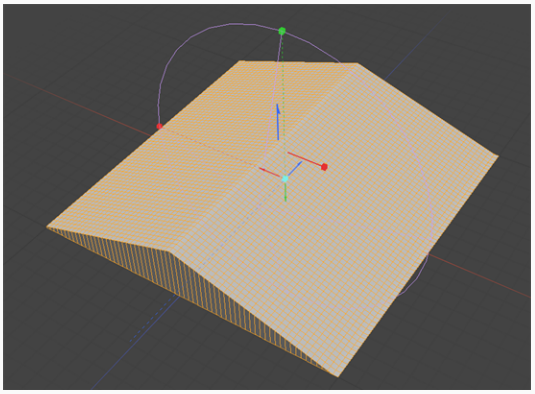
In 0.0
Out 0.0

Mix Mode: Multiply

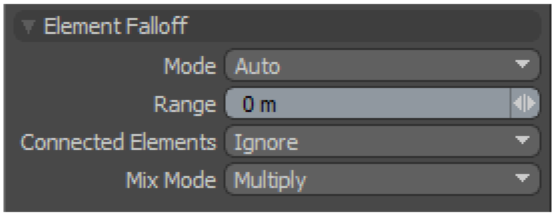
☐ Use World Transforms

The Cylinder Falloff creates a circular falloff that extends across the length of a virtual cylinder. All vertices at the center of the cylinder receive 100% of the tool's influence, attenuated toward the outer edge and beyond the cylinder, where vertices receive none of the tool's influence.

Center X/Y/Z	Defines the center of influence, where the strength of the falloff is greatest (100%). The strength of the falloff attenuates toward the outer bounds of the cylinder. The falloff volume extends upward and downward toward infinity.
Size X/Y/Z	Defines the radius of a perfect circle from the center and determines the outer area of the falloff where there is no effect.
Axis	Defines the projection axis for the cylinder, determining the direction for the falloff.
Auto Size	You can select this option to automatically size the falloff's **Center** and **Size** values to match the bounding box of the current selection.
Shape Preset	The strength of the falloff's influence can be controlled along the extent using a shape preset. • **Linear** - Attenuation of falloff occurs evenly across its range. • **Ease-In** - Strength of falloff is greater toward the **Start** position. • **Ease-Out** - Strength of falloff is greater toward the **End** position. • **Smooth** - Strength of falloff is greater toward the center of the falloff. • **Custom** - You can use the **In**/**Out** options to fine-tune strength of falloff.
In/Out	The **In** value determines the strength of the falloff nearer to the **Start** position, and the **Out** value determines the strength on nearer the **End** side of the falloff.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.
Use World Transforms	By default, falloffs calculate their influence over transforming vertices according to their relative position, using object space vertex positions. When this option is enabled, all selected mesh surfaces are considered one surface and they deform as one.

	 Use World Transform disabled  Use World Transform enabled

Element Falloff



Element Falloff

Mode Auto

Range 0 m

Connected Elements Ignore

Mix Mode Multiply

Element falloff limits the tool influence to the element (vertex, edge or polygon) in the mesh that is clicked on when using the current tool. You can hover the mouse pointer over a component (vertex, edge or polygon) and see the target element pre-highlighted. Clicking on the element applies the tool's action to the element, attenuating outward based on the **Range** value.

Mode	There are several modes available for selecting the element type. • **Auto** - You can choose from any component, based on which element is clicked. • **Auto Center** - Same as **Auto**, but the **Range** is calculated from the element's center point. • **Vertex** - You are limited to only selecting vertices. • **Edge** - You are limited to only selecting edges. • **Edge Center** - Same as **Edge**, but the **Range** is calculated from the element's

	center point. • **Polygon** - You are limited to only selecting polygons. • **Polygon Center** - Same as **Polygon**, but the **Range** is calculated from the elements center point.
Range	This value determines the extent of the falloff range, measured outward from the tool handle. When the **Center** options are selected, the **Range** is calculated from the element's center point. At the default 0 m value, only the clicked element is affected.
Connected Elements	Defines what connected geometry does when an element is transformed. • **Ignore** - Ignores all connected elements and only moves the element specified. • **Use Connectivity** - Only affects single surface and connected elements. Unconnected elements within range are ignored. • **Rigid Connections** - Moves all connected elements equally the specified distance. • **Edge Loops** - Moves the connected loop, defined by all the connected quad polygons/edges/vertices in a single row.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.

Image Falloff

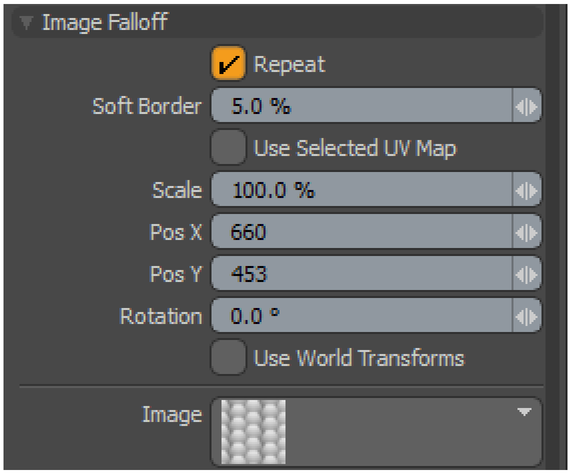


Image Falloff

☑ Repeat

Soft Border: 5.0 %

☐ Use Selected UV Map

Scale: 100.0 %

Pos X: 660

Pos Y: 453

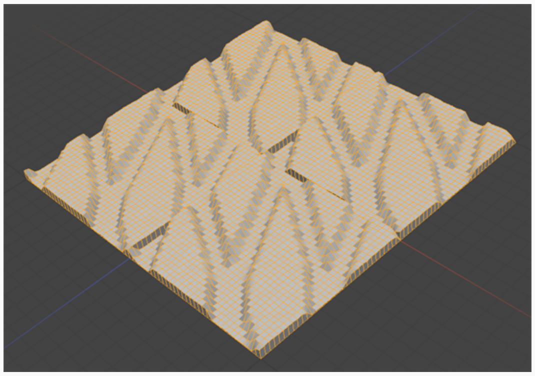
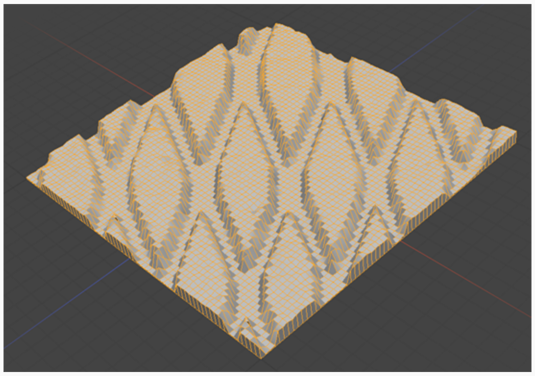
Rotation: 0.0 °

☐ Use World Transforms

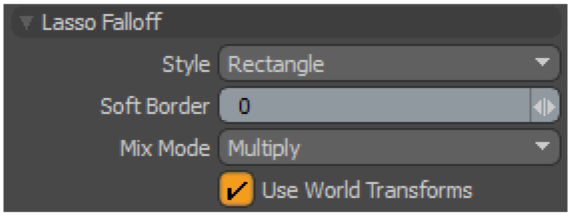
Similarly to the Image Ink option in the painting toolbox, you may apply an image, in screen space, as a falloff, where the brightness (luminosity) of the image affects the selected geometry. Additionally, by selecting the **Use Current UV** option of the falloff, you can apply the current image as a falloff in 3D object space opening up a whole variety of options.

To use, you need to place the masking image into the images folder of the presets, or add the path to the **Preset Browser**. Once the image is selected, clicking in the viewport activates the image widget, where you can use the controls to position the image within the 3D viewport using the various handles.

Repeat	When this option is enabled, the image is repeated outside of the images borders. This has no effect when **Use Selected UV Map** is enabled.
Soft Border	Attenuates the strength of the image's effect over the geometry. At 100% the strength of the falloff is strongest at the center falling off toward the edges of the image. At 0%, the white areas of the entire image affect the geometry.
Use Selected UV Map	When this option is enabled, Modo bypasses the viewport image and applies the image's effect to the surface with the currently selected UV map. You can select UV maps within the **List** viewport under the UV Map section.
Scale	This value determines the size of the image in the 3D viewport, and when **Repeat** is disabled, the size of the falloff's area of influence as well. Click the right side handle to interactively rotate the image.

Pos X/Y	Positions the center point of the Image within the 3D viewport. You can also click the center handle on the widget to interactively position the image.
Rotation	Rotates the image within the 3D viewport. Click the top handle to interactively rotate the image.
Image	Use this button to browse through the preset bitmap textures, selecting the image to use for the falloff.
Use World Transforms	By default, falloffs calculate their influence over transforming vertices according to their relative position, using object space vertex positions. When this option is enabled, all selected mesh surfaces are considered one surface and they deform as one.  Use World Transform disabled  Use World Transform enabled

Lasso Falloff



Lasso Falloff

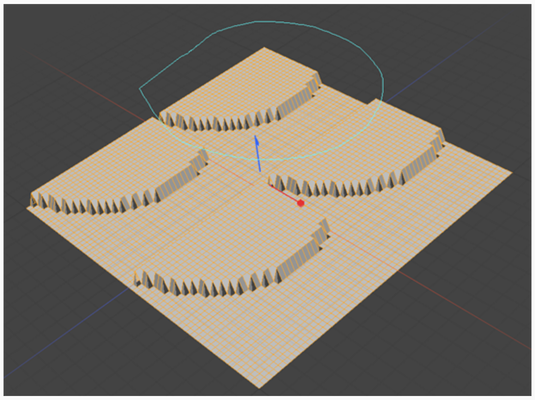
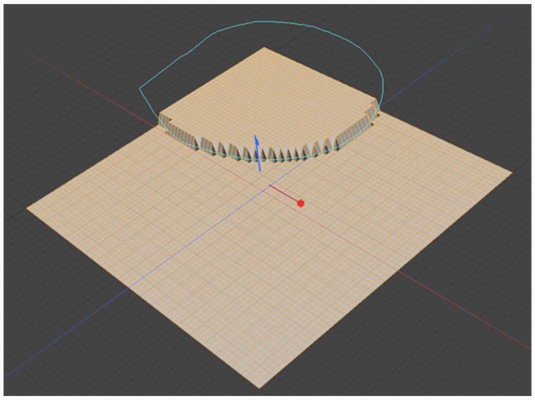
Style: Rectangle

Soft Border: 0

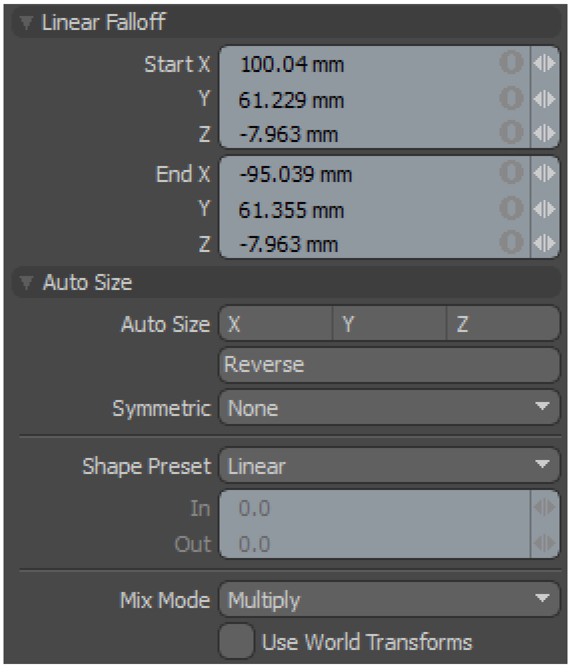
Mix Mode: Multiply

☑ Use World Transforms

Several styles of Lasso selections drawn in screen space projecting outward toward infinity, producing very specific falloffs. Settings allow you to then to define areas of influence around the lasso shape with the 'Soft Border' option. All areas within the Lasso receive 100% of the tools effect; the attenuation of the falloff occurs across the 'Soft Border' distance only outside the Lasso area.

Style	You can choose from the various Lasso styles. • **Lasso** - Allows you to draw freeform shapes onto the screen defining the falloff. • **Rectangle** - You can drag out rectangular shapes to define the falloff area. • **Circle** - You can drag out perfectly circular shapes to define the falloff area. The shape originates from the center of the circle. • **Ellipse** - You can drag out elliptical shapes to define the falloff area. The shape originates from the upper or lower corner of the ellipse.
Soft Border	This option defines the actual falloff radius around the Lasso area that attenuates across the distance defined. Calculated in screen-space pixels from the lasso border.
Mix Mode	In instances where there are multiple falloffs applied to a transform (using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.
Use World Transforms	By default, falloffs calculate their influence over transforming vertices according to their relative position, using object space vertex positions. When this option is enabled, all selected mesh surfaces are considered one surface and they deform as one.  Use World Transform disabled  Use World Transform enabled

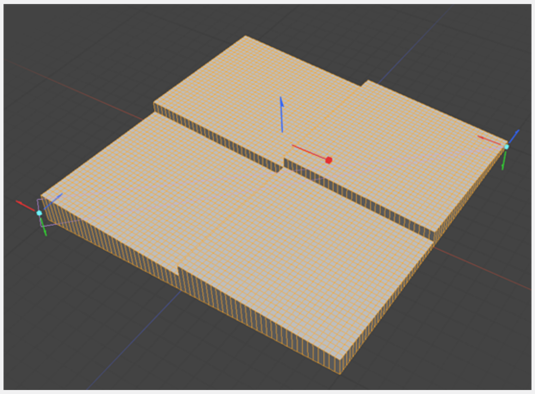
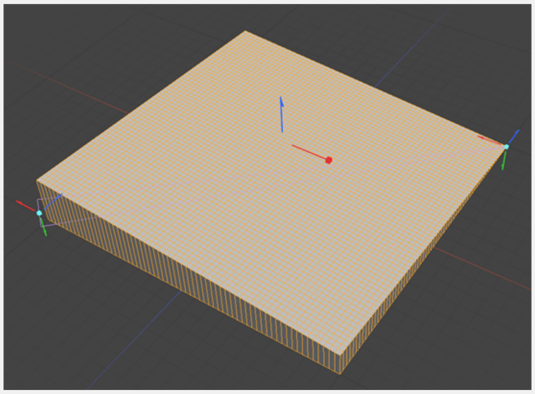
Linear Falloff



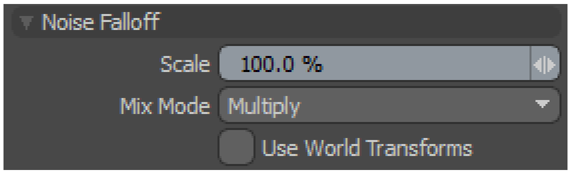
Start X 100.04 mm
Y 61.229 mm
Z -7.963 mm
End X -95.039 mm
Y 61.355 mm
Z -7.963 mm
Symmetric None
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply
☐ Use World Transforms

The Linear falloff uses a straight line to define the falloff. The line itself determines the direction of the falloff's influence with the strength of the effect itself, reaching out toward infinity. All the vertices at one end of the line receive 100% of the tool's influence. The vertices at the other side receive 0% of the tool's influence. The points between the two lines receive some partial value based on the interpolation between the two end points.

Start X/Y/Z	Defines the starting position of the falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives the maximum amount of influence attenuating toward the **End** position.
End X/Y/Z	Defines the end position of the falloff as a specific XYZ coordinate value. This end of the line and everything beyond it receives no influence attenuating toward the **Start** position.
Auto Size X/Y/Z	You can select either of these three options to automatically scale the **Start** and **End** points of the falloff to match the bounding box size of the selected elements along one of the three axes.
Reverse	You can select this option to invert the **Start** and **End** points reversing the influence of

	the falloff.
Symmetric	You can choose from several options to automatically mirror the influence area of the falloff: • **None** - The **Symmetric** function is disabled. • **Start** - Influence of falloff is symmetrically mirrored across the **Start** position. • **End** - Influence of falloff is symmetrically mirrored across the **End** position.
Shape Preset	The strength of the falloff's influence can be controlled along the extent using a shape preset. • **Linear** - Attenuation of falloff occurs evenly across its range. • **Ease-In** - Strength of falloff is greater toward the **Start** position. • **Ease-Out** - Strength of falloff is greater toward the **End** position. • **Smooth** - Strength of falloff is greater toward the center of the falloff. • **Custom** - You can use the **In/Out** options to fine-tune the strength of the falloff.
In/Out	The **In** value determines the strength of the falloff nearer to the **Start** position and the **Out** value determines the strength on nearer the **End** side of the falloff.
Mix Mode	In instances where there are multiple falloffs applied to a transform (using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.
Use World Transforms	By default, falloffs calculate their influence over transforming vertices according to their relative position, using object space vertex positions. When this option is enabled, all selected mesh surfaces are considered one surface and they deform as one.  Use World Transform disabled  Use World Transform enabled

Noise Falloff



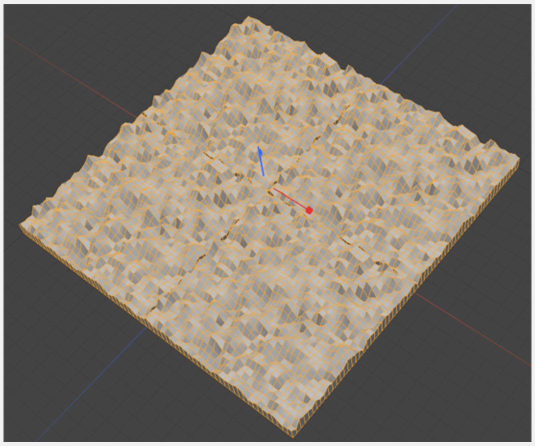
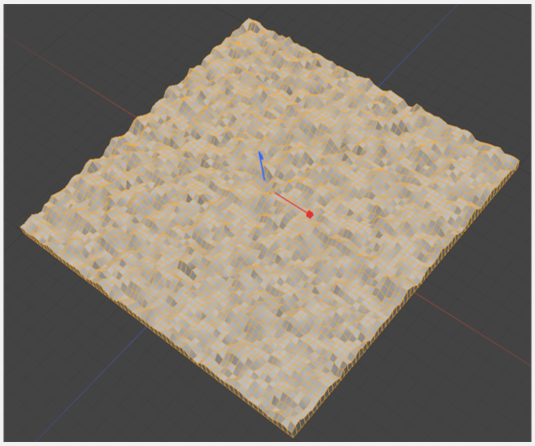
Noise Falloff

Scale 100.0 %

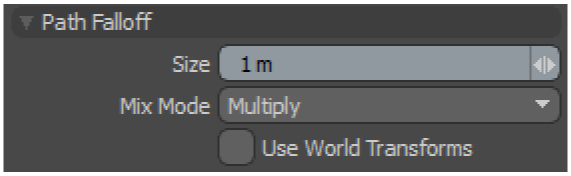
Mix Mode Multiply

☐ Use World Transforms

The Noise falloff uses a three-dimensional noise pattern to modulate the influence of the tool. You can adjust the size of the noise pattern with the scale parameter. Combined with the Move or Stretch tool, this falloff can be useful for roughing up a surface.

Scale	This value determines the size of the noise in relation to the geometry it is affecting.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.
Use World Transforms	By default, falloffs calculate their influence over transforming vertices according to their relative position, using object space vertex positions. When this option is enabled, all selected mesh surfaces are considered one surface and they deform as one.  Use World Transform disabled  Use World Transform enabled

Path Falloff

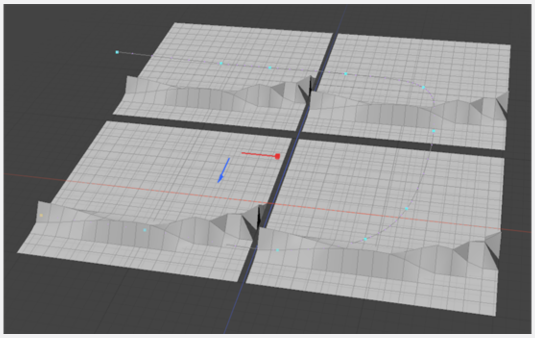
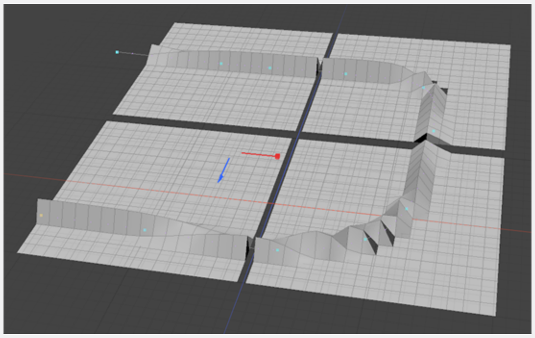

Path Falloff

Size 1 m

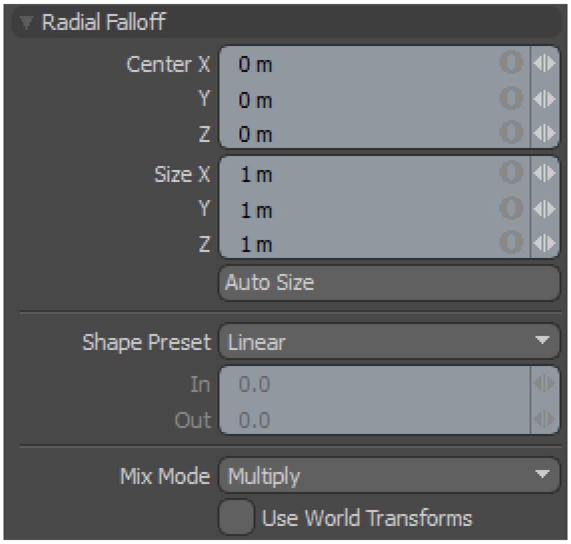
Mix Mode Multiply

☐ Use World Transforms

The Path falloff combines the curve generator with the Path falloff. The curve generator creates a curve and the falloff reads the curve points to determine where the tool should be active. If the user has the Path falloff in the **Tool Pipe** at the same time as a mesh edit or Transform Tool, they may need to select the Curve Path (in **Polygons** mode) prior to activating the falloff. This allows you to define the curve to be used. Once the basic curve is created, you can select the tool and begin to edit. You can also click on the curve control points to edit them while the tool/falloff combination is active.

Size	This value determines the area of attenuation around the path itself, calculated as a tube around the curve's position.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.
Use World Transforms	By default, falloffs calculate their influence over transforming vertices according to their relative position, using object space vertex positions. When this option is enabled, all selected mesh surfaces are considered one surface and they deform as one.  Use World Transform disabled  Use World Transform enabled

Radial Falloff



Radial Falloff

Center X 0 m

Y 0 m

Z 0 m

Size X 1 m

Y 1 m

Z 1 m

Shape Preset Linear

In 0.0

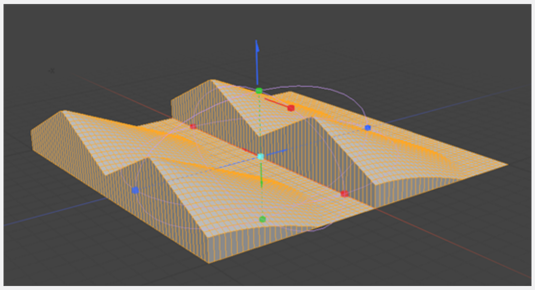
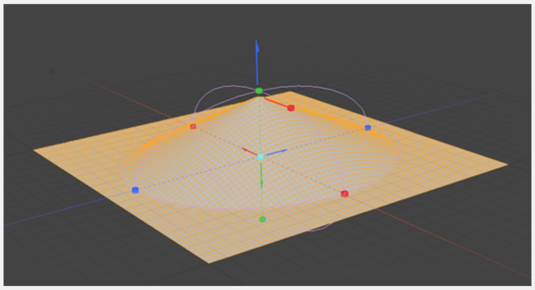
Out 0.0

Mix Mode Multiply

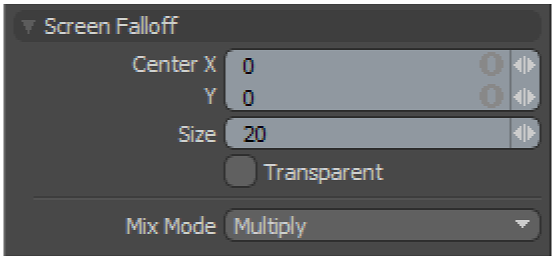
☐ Use World Transforms

Radial falloff uses a sphere where all vertices at the very center of the sphere receive 100% tool influence and any vertices outside the sphere receive 0% of the tool's influence. There is a smooth gradient of tool influence between the center and edge boundary of the sphere that utilizes the falloff shape preset to determine the interpolation style.

Center X/Y/Z	Defines the center of influence, where the strength of the falloff is the greatest (100%). The strength of the falloff attenuates toward the outer bounds of the spherical volume, the area outside the volume receives no tool influence.
Size X/Y/Z	Defines the radius of a perfect circle from the center and determines the outer area of the falloff where there is no effect.
Auto Size	You can select this option to automatically size the falloff's **Center** and **Size** values to match the bounding box of the current selection.
Shape Preset	The strength of the falloff's influence can be controlled along the extent using a shape preset. • **Linear** - Attenuation of falloff occurs evenly across its range. • **Ease-In** - Strength of falloff is greater toward the **Start** position. • **Ease-Out** - Strength of falloff is greater toward the **End** position. • **Smooth** - Strength of falloff is greater toward the center of the Falloff.

	• **Custom** - You can use the **In/Out** options to fine-tune strength of falloff.
In/Out	The **In** value determines the strength of the falloff nearer to the **Start** position, where the **Out** value determines the strength on nearer the **End** side of the falloff.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.
Use World Transforms	By default, falloffs calculate their influence over transforming vertices according to their relative position, using object space vertex positions. When this option is enabled, all selected mesh surfaces are considered one surface and they deform as one.  Use World Transform disabled  Use World Transform enabled

Screen Falloff



Screen Falloff

Center X 0

Y 0

Size 20

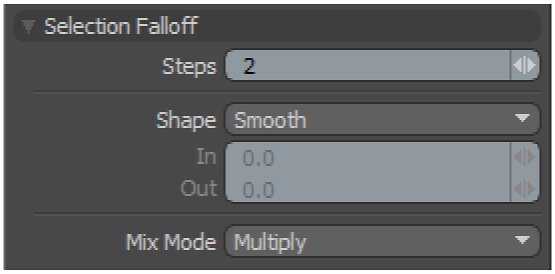
☐ Transparent

Mix Mode Multiply

The Screen falloff is similar to the Cylinder falloff, except rather than using one of the major axes to determine the angle of the projection, the Screen falloff projects a cylinder that is perpendicular to the viewport window (the computer's screen). The falloff is defined by a disc drawn in screen space extending infinitely into the distance. This disc is used to project a virtual cylinder of influence.

Center X/Y	Defines the center of influence in screen space, where the strength of the falloff is greatest (100%). The strength of the falloff attenuates toward the outer bounds of the spherical volume, the area outside the volume receives no tool influence.
Size	Sets the pixel size of the circular area of influence, calculated as a radius from the **Center** position.
Transparent	When disabled, the falloff only affects polygons facing toward the viewport. When enabled the falloff works through surface also affecting those facing away from the viewports.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each Falloff interacts with the other.

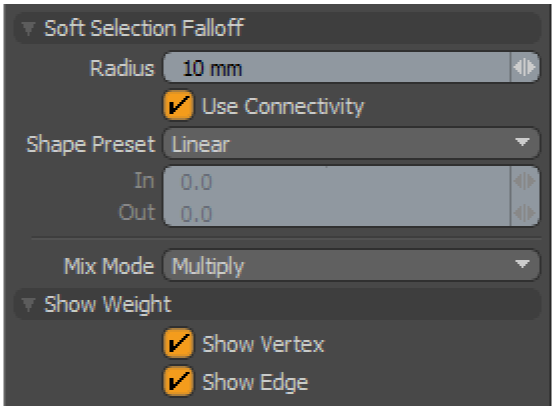
Selection Falloff


Selection Falloff
Steps 2
Shape Smooth
In 0.0
Out 0.0
Mix Mode Multiply

The Selection falloff allows you to modulate a geometry edit based on a selection area. The area of influence is at 0% on the selection border and ramps to 100% based on the **Steps** value. If the **Steps** are set to 1, there is only a slight falloff range, resulting in a harder edit. Increasing the number of **Steps** softens the influence at the selection border and the number of steps can be controlled by right-clicking or by editing the text field in the tool's **Properties**. You might think of **Steps** as rows or loops of selected elements: for each **Step** the falloff reduces the selection area and uses the range between the original selection and the reduced selection as the soft border for the falloff area.

Steps	This value determines the extent of the falloff range, where each **Step** represents a loop of polygons or a ring of edge sections. The greater the **Steps** value, the further the falloff attenuates.
Shape	The strength of the falloff's influence can be controlled along the extent using a shape preset. • **Linear** - Attenuation of the falloff occurs evenly across its range. • **Ease-In** - Strength of the falloff is greater toward the **Start** position. • **Ease-Out** - Strength of falloff is greater toward the **End** position. • **Smooth** - Strength of falloff is greater toward the center of the falloff. • **Custom** - You can use the **In/Out** options to fine-tune the strength of the falloff.
In/Out	The **In** value determines the strength of the falloff nearer to the **Start** position, and the **Out** value determines its strength nearer the **End** side of the falloff.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.

Soft Selection Falloff


Soft Selection Falloff

Radius: 10 mm

☑ Use Connectivity

Shape Preset: Linear

In: 0.0

Out: 0.0

Mix Mode: Multiply

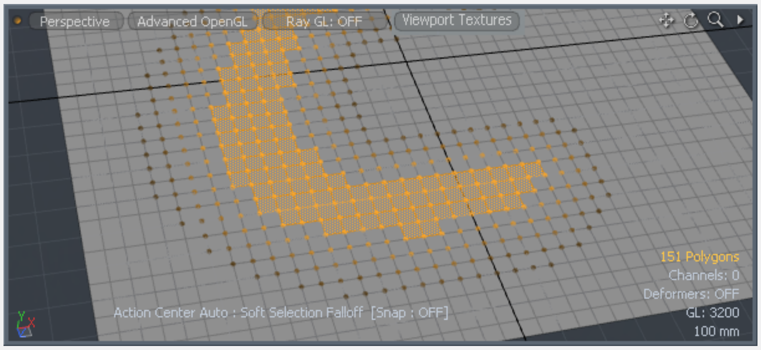
Show Weight

☑ Show Vertex

☑ Show Edge

The Soft Selection Falloff is unique in that it is the only falloff that modulates the geometry outside of the selection.

Radius	Determines the range outside of the selected edge to attenuate the falloff.

Use Connectivity	When enabled, only affects single surface/connected elements. Unconnected elements within range are ignored.
Shape	The strength of the falloff's influence can be controlled along the extent using a shape preset. • **Linear** - Attenuation of the falloff occurs evenly across its range. • **Ease-In** - Strength of the falloff is greater toward the **Start** position. • **Ease-Out** - Strength of falloff is greater toward the **End** position. • **Smooth** - Strength of falloff is greater toward the center of the falloff. • **Custom** - You can use the **In**/**Out** options to fine-tune the strength of the falloff.
In/Out	The **In** value determines the strength of the falloff nearer to the **Start** position, and the **Out** value determines its strength nearer the **End** side of the falloff.
Mix Mode	In instances where there are multiple falloffs applied to a transform (by using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.
Show Weights	
Show Vertex/Show Edge	You can use these toggles to enable/disable the display of geometry highlights for both vertices and edges, giving a visual reference to the application of the Soft Selection falloff (demonstration of the vertices display mode below).  Perspective Advanced OpenGL Ray GL: OFF Viewport Textures Action Center Auto : Soft Selection Falloff [Snap : OFF] 151 Polygons Channels: 0 Deformers: OFF GL: 3200 100 mm

Coplanar Falloff

The Coplanar falloff affects faces that are located on the same plane as the selection. It works similarly to the Soft Selection falloff, but it considers the angle between normals instead of the distance.

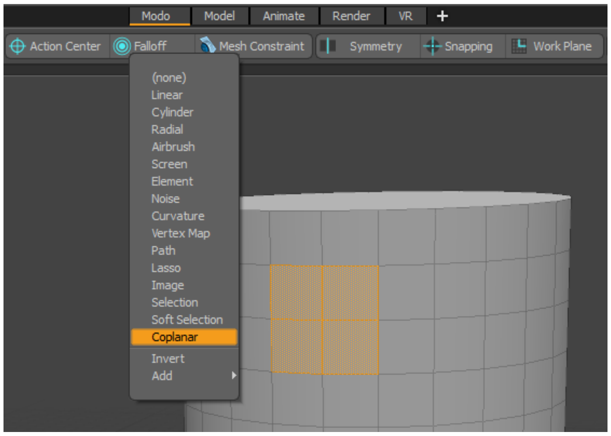
Tip: For more information on using falloffs in general, see Using Falloffs.

Using the Coplanar Falloff

1. Select the elements you want to modify.

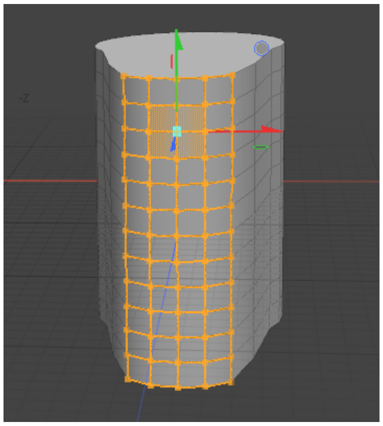
Note: If nothing is selected, all geometry is affected.

2. At the top of the 3D viewport, click **Falloff**, and select **Coplanar**.


Modo
Model
Animate
Render
VR
+
Action Center
Falloff
Mesh Constraint
Symmetry
Snapping
Work Plane
(none)
Linear
Cylinder
Radial
Airbrush
Screen
Element
Noise
Curvature
Vertex Map
Path
Lasso
Image
Selection
Soft Selection
Coplanar
Invert
Add

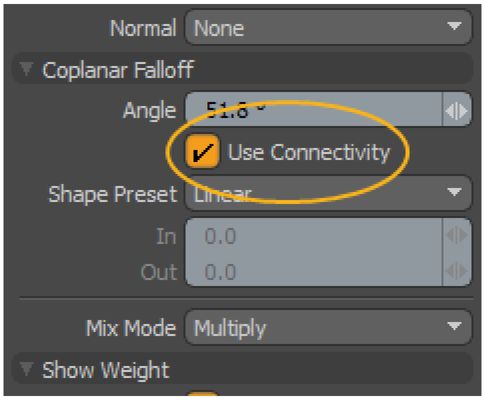
3. Activate the tool you want to use, for example, the Move tool.
4. To adjust the range of the falloff, right-click and drag in the viewport.
5. Click and drag on the tool handles in the 3D viewport, or edit the tool's properties.

   The tool's effect is attenuated by the falloff.



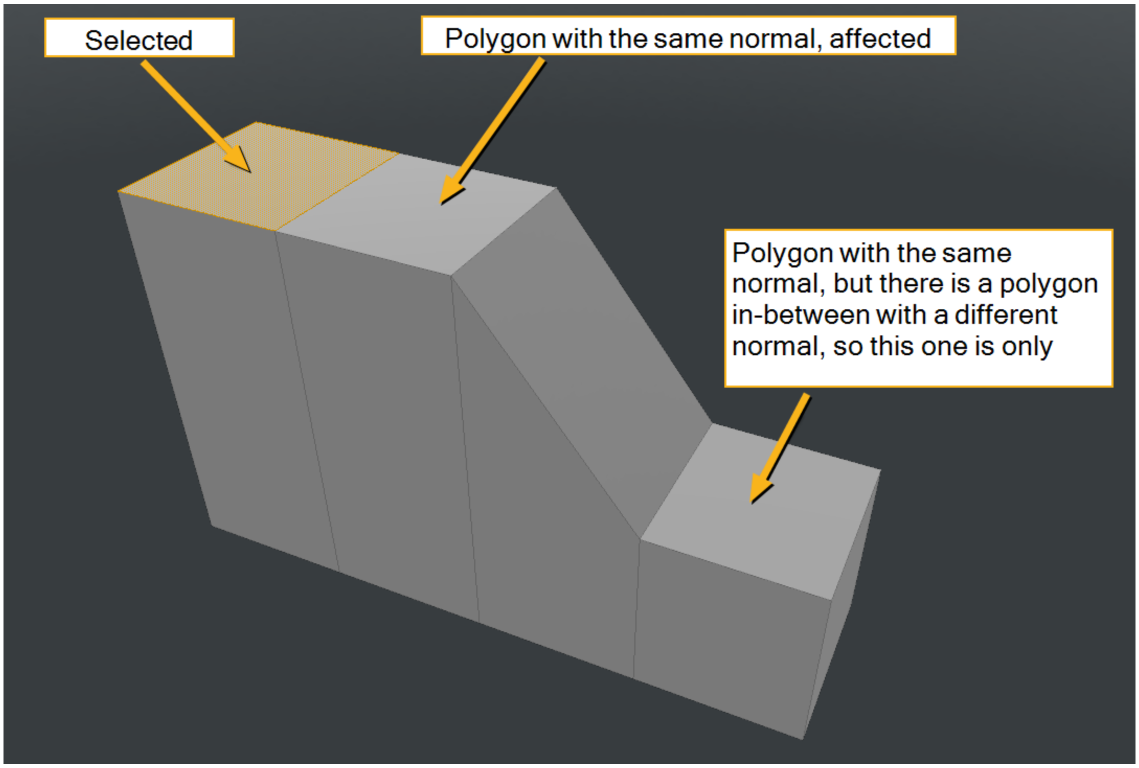
Use Connectivity

The Coplanar Falloff's **Use Connectivity** option allows you to only select connected elements within the specified range. You can find this option in the falloff's properties on the left panel.



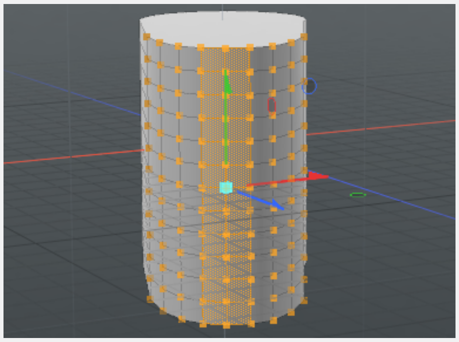
Normal None
Coplanar Falloff
Angle 51.8 °
☑ Use Connectivity
Shape Preset Linear
In 0.0
Out 0.0
Mix Mode Multiply
Show Weight

The image below shows you what polygons are affected by the falloff with and without **Use Connectivity**.

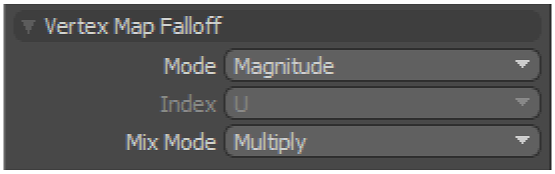

Selected
Polygon with the same normal, affected
Polygon with the same normal, but there is a polygon in-between with a different normal, so this one is only

The Coplanar Falloff has the following properties:

Angle	Determines the range outside of the selected edge to attenuate the falloff. You can change the value by right-clicking and dragging on it.
Use Connectivity	When enabled, only affects single-surface/connected elements. Unconnected elements within range are ignored.
Shape	The strength of the falloff's influence can be controlled along the extent using a shape preset. The following options are available: • **Linear** - The falloff attenuates evenly across its range. • **Ease-In** - The falloff is stronger toward the **Start** position. • **Ease-Out** - The falloff is stronger toward the **End** position. • **Smooth** - The falloff is stronger toward the center. • **Custom** - You can use the **In**/**Out** options to fine-tune the strength of the falloff.
In/Out	The **In** value determines the strength of the falloff nearer to the **Start** position, and the **Out** value determines its strength nearer to the **End** side of the falloff.
Mix Mode	When there are multiple falloffs applied to a transform (by using the **Add**

	option of the **Falloff** menu), **Mix Mode** defines how each falloff interacts with the other.
Show Weights	
Show Vertex/Show Edge	Enable or disable geometry highlights for vertices and edges, giving you a visual reference to the application of the Coplanar falloff. On the image below, the **Show Vertex** option is enabled. 

Vertex Map Falloff



Mode: Magnitude
Index: U
Mix Mode: Multiply

The Vertex Map falloff converts values of the currently selected Vertex Map to values representing falloff strength. While generally a Weight Map is used, being the most intuitive approach, you may also use values of other Vertex Map types as a falloff as well. The falloff is determined by the values of an existing Vertex Map. This allows you to create complex falloffs that actually can have a inverse negative in instances where negative values can be applied.

If a user chooses "Component" for the 'Mode' option, the falloff tool uses the component value specified by the index option. For example, if the current map is a Morph Map and the index is X, then the falloff sets the X value of morph position to the weight value. **Magnitude** is effective for morph, RGB, and RGBA maps. If the current map selected is a morph map, the falloff returns the length of the morph delta position. If it is RGB or RGBA, the falloff returns the luminosity of color vector.

Mode	The following falloff modes are available: • **Magnitude** - Values of the Vertex Map are converted to strength values and applied to the affected surface(s). • **Component** - You can select individual components of a map: Weight Map (1 component- weight value), RGB Map (3 components- R, G or B), RGBA Map (4 components- R, G, B or A), Morph Map (3 components- X, Y or Z), Pick Map (1 component).
Index	When **Component** mode is selected, you can determine the specific component using this option selector.
Mix Mode	In instances where there are multiple falloffs applied to a transform (using the **Add** option of the **Falloff** menu), the mix mode defines how each falloff interacts with the other.

MeshFusion

MeshFusion offers a sophisticated set of boolean operations for creating new geometry. With MeshFusion, you can create meshes through the non-destructive union, subtraction, and intersection of objects.

Although similar in concept to existing Boolean operations, MeshFusion is significantly more advanced, supporting real-time update of operations and a high level of control over blending between the resulting surfaces.

Note: For more information on Boolean operations, see Booleans.

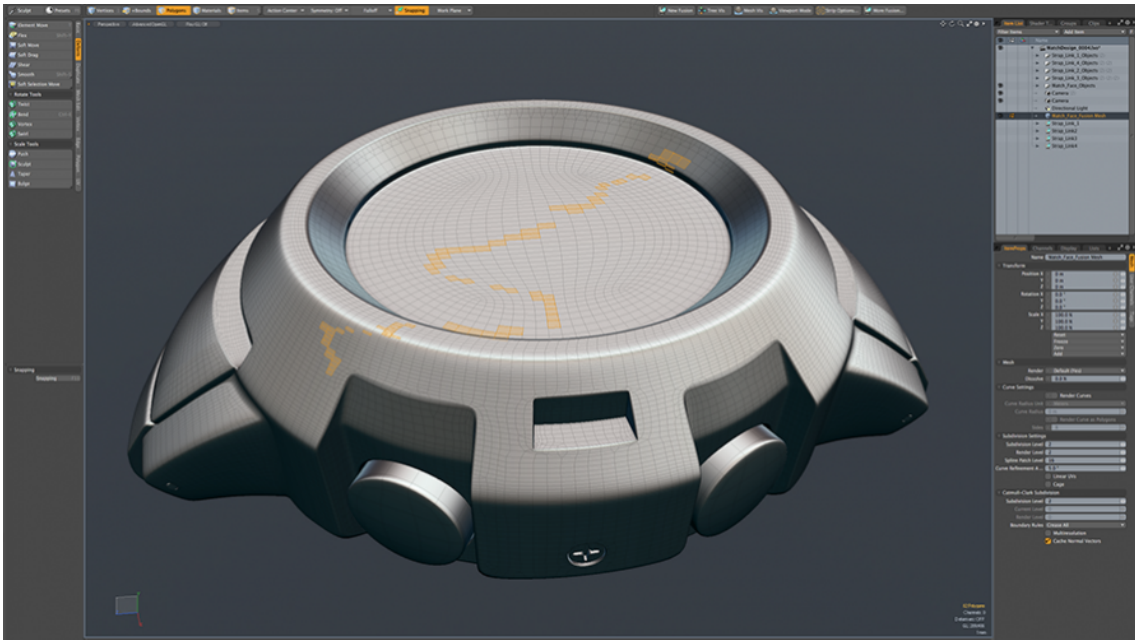
MeshFusion Prerequisites

MeshFusion works reliably with meshes that are composed of quads (four-sided polygons). Any other type of polygons are converted to quads before a fusion operation.

N-gons may produce less desirable Fusion mesh topology when involved in seams. If Fusion strips pass through polygons that were n-gons in the original mesh, the quality of those strips may be adversely affected. In such cases, you can break up the n-gons into quads and triangles in the original mesh.

Mesh Roles and Relationships

Modeling in MeshFusion is somewhat different from traditional Boolean modeling approaches.

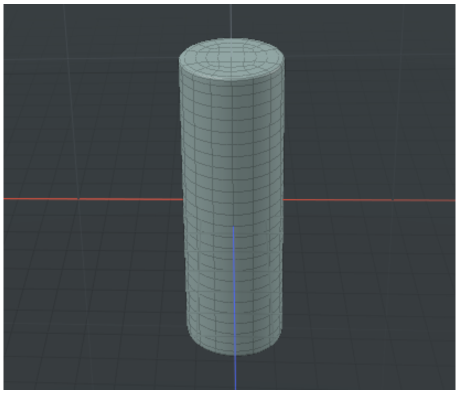


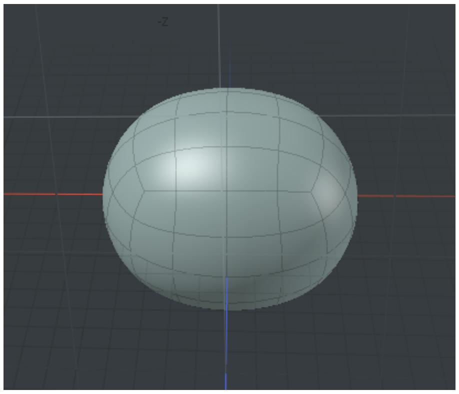
You can have two kinds of meshes:

- **Primaries** - Positive meshes
- **Trims** - Subtraction meshes. They can be of two types:
  - Subtractive Trims, which cut away part of an existing geometry that overlaps with it.
  - Intersection Trims, which create new geometry from parts of existing geometry that overlap.

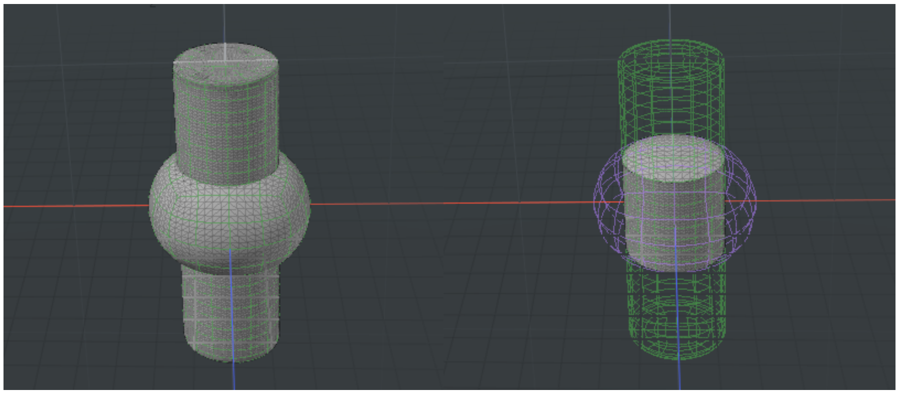
When setting up your meshes, you need to set the mesh's role and its relationship to other meshes.

For a simple demonstration, consider the following two meshes:



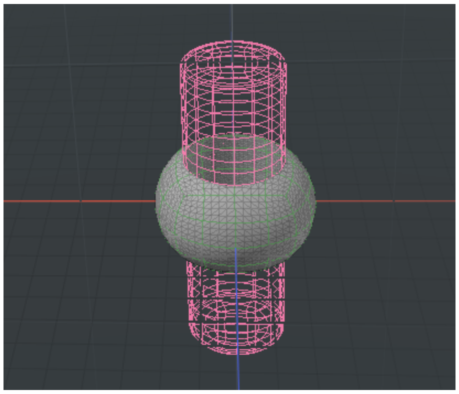


If the meshes are each placed at the origin, the images below show the resulting mesh for each boolean operation.



Union

Intersection



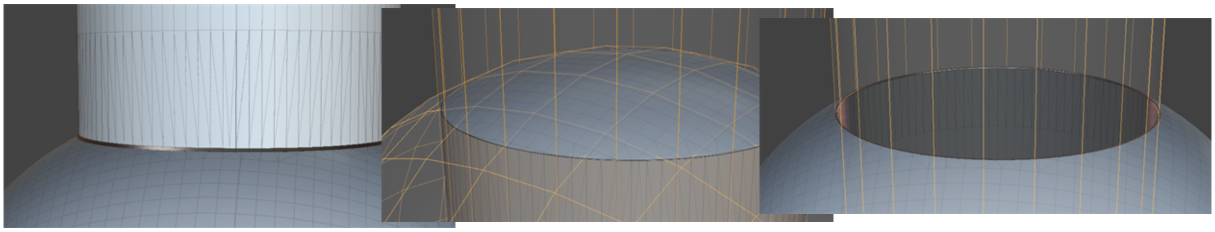
Subtractive Trim (ellipsoid with cylinder)

Note: The source meshes were selected from the Fusion Qbic Presets.

For more detail on Fusion operations, see Mesh Roles and Relationships.

Fusion Strips

When two objects are combined, new corners are created at the junction of the objects. The line around the rim of these corners is known as a strip. The images below highlight a strip formed by each operation. The strips are highlighted in yellow.

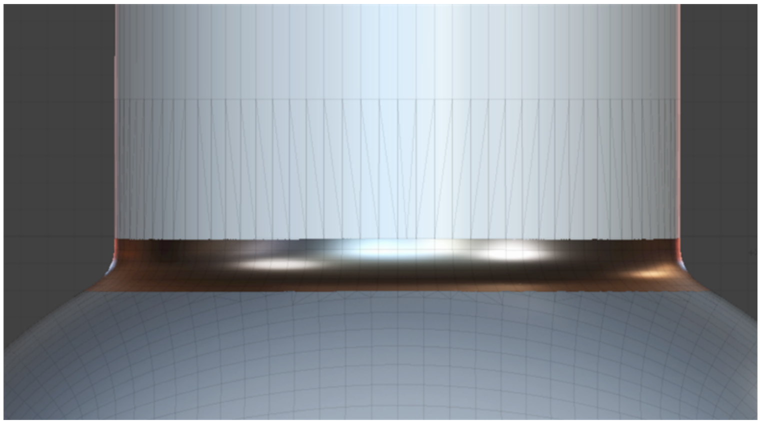


Strip at junction of union

Strip at junction of intersection

Strip at junction of trim

Strips are an important feature of MeshFusion, as they allow you to customize the geometry at the strip. For example, you can adjust the strip width and smoothing to reduce the sharpness of the join. The image below shows an adjustment to the strip width for the union example pictured above.

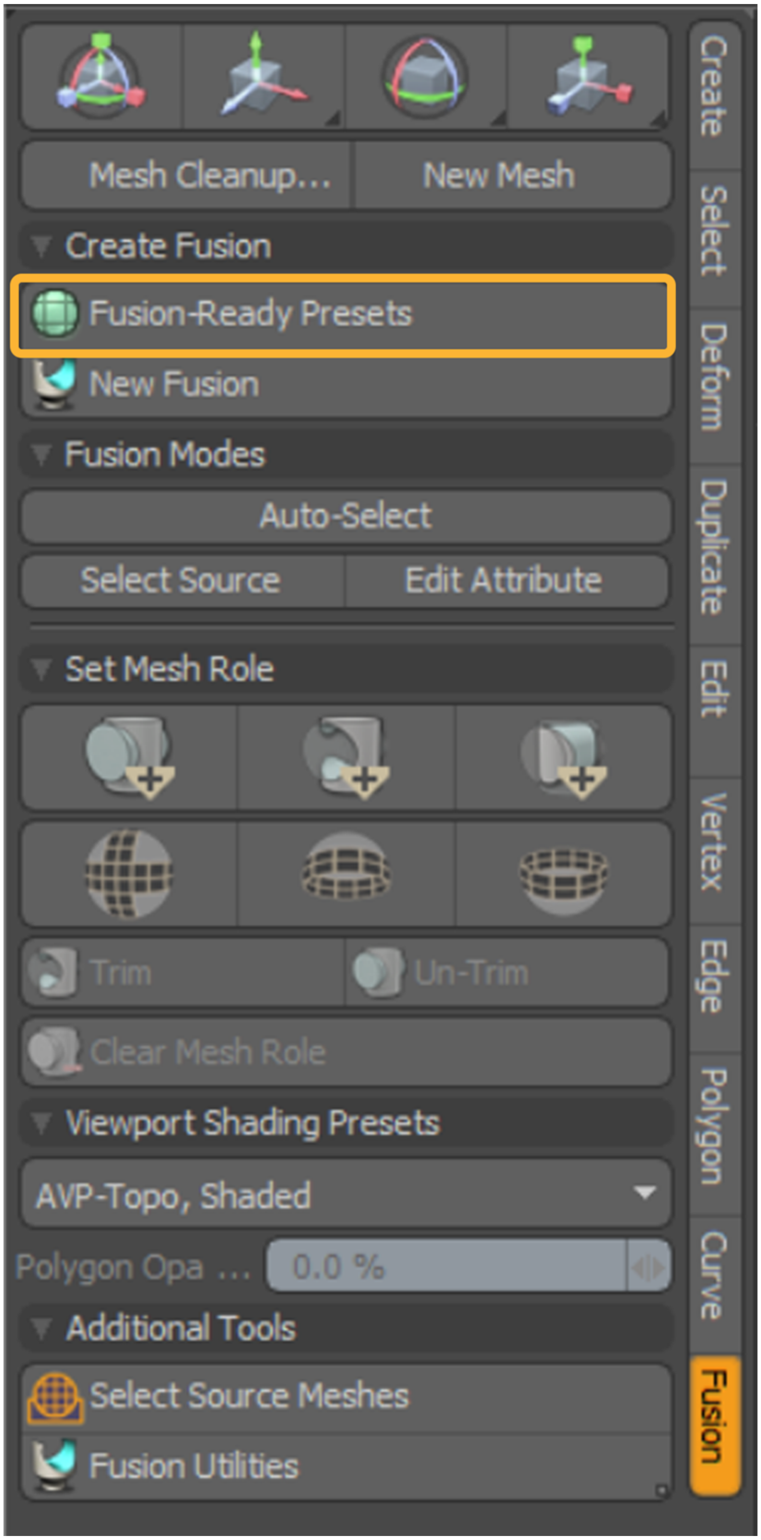


As with the other results of Fusion operations, if you make any modifications to the source mesh, strips are updated in real-time.

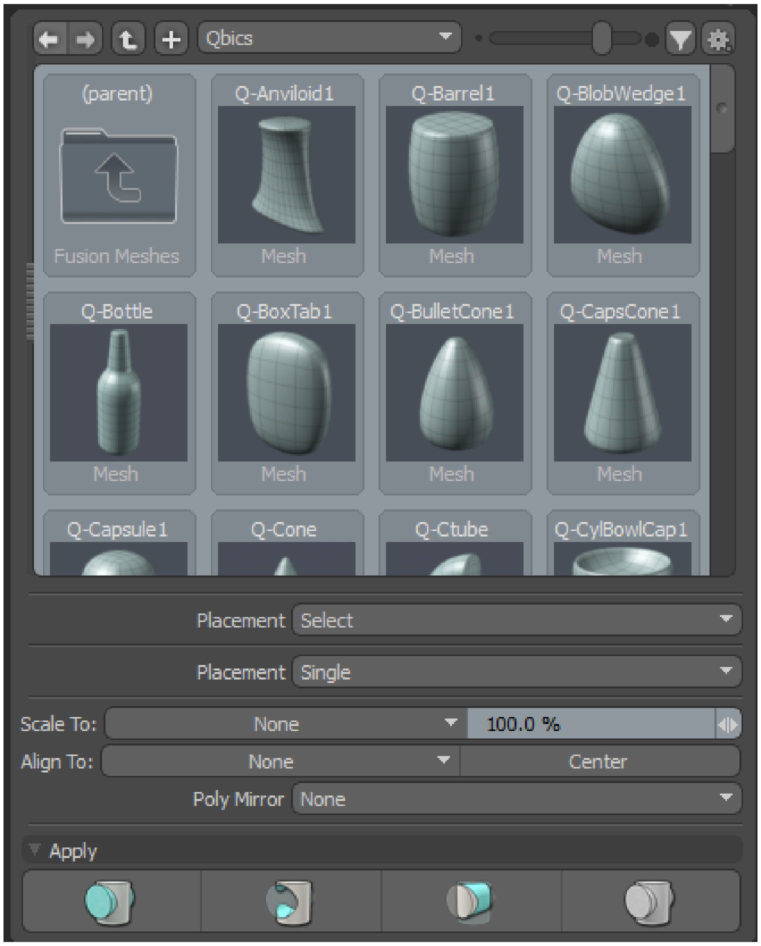
For more information, see Fusion Surface Strips.

Fusion Qbic Presets

Modo has a wide range of over fifty ready-made Qbic primitives to help you build Fusion meshes. The primitives have been designed to offer a broad variety of modeling building blocks. You can see the meshes by clicking the **Create Fusion** > **Fusion-Ready Presets** button on the **Fusion** tab.


Mesh Cleanup...
New Mesh
Create Fusion
Fusion-Ready Presets
New Fusion
Fusion Modes
Auto-Select
Select Source
Edit Attribute
Set Mesh Role
Trim
Un-Trim
Clear Mesh Role
Viewport Shading Presets
AVP-Topo, Shaded
Polygon Opa ...
0.0 %
Additional Tools
Select Source Meshes
Fusion Utilities
Create
Select
Deform
Duplicate
Edit
Vertex
Edge
Polygon
Curve
Fusion

The **Fusion Qbic Presets** browser opens.


Qbics
(parent)
Fusion Meshes
Q-Anviloid1
Mesh
Q-Barrel1
Mesh
Q-BlobWedge1
Mesh
Q-Bottle
Mesh
Q-BoxTab1
Mesh
Q-BulletCone1
Mesh
Q-CapsCone1
Mesh
Q-Capsule1
Q-Cone
Q-Ctube
Q-CylBowlCap1
Apply

Placement Select

Placement Single

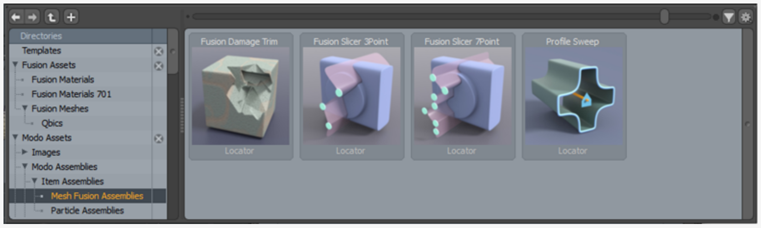
Scale To: None 100.0 %

Align To: None Center

Poly Mirror None

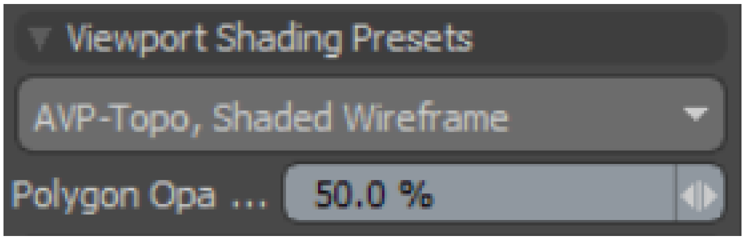
Alternatively, you can access the Qbic presets by clicking the **Presets** button on the top of the Toolbox and expand **Fusion Assets** > **Fusion Meshes** > **Qbics**. For more information, see Fusion Qbic Presets.

Tip: Modo also provides MeshFusion Assemblies presets. Click the **Presets** button on the top of the Toolbox and expand **Modo Assets** > **Modo Assemblies** > **MeshFusion Assemblies**.


Directories
Templates
Fusion Assets
Fusion Materials
Fusion Materials 701
Fusion Meshes
Qbics
Modo Assets
Images
Modo Assemblies
Item Assemblies
Mesh Fusion Assemblies
Particle Assemblies
Fusion Damage Trim
Locator
Fusion Slicer 3Point
Locator
Fusion Slicer 7Point
Locator
Profile Sweep
Locator

Fusion Viewport settings

You can optimize the 3D viewport for MeshFusion modeling with the **Viewport Shading Presets** menu found in the **Fusion** tab.

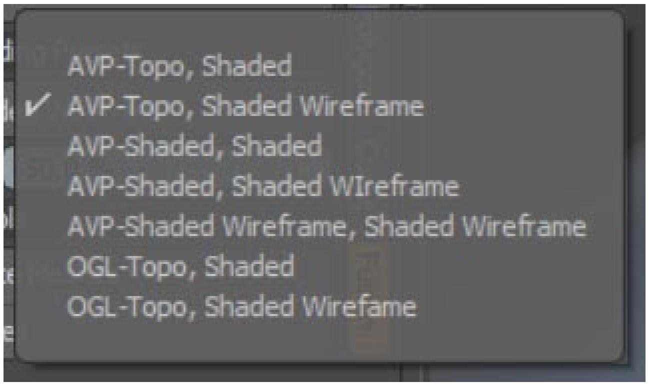


Viewport Shading Presets

AVP-Topo, Shaded Wireframe

Polygon Opa ... 50.0 %

Within the **Viewport Shading Presets** dropdown menu you can find a number of display options.

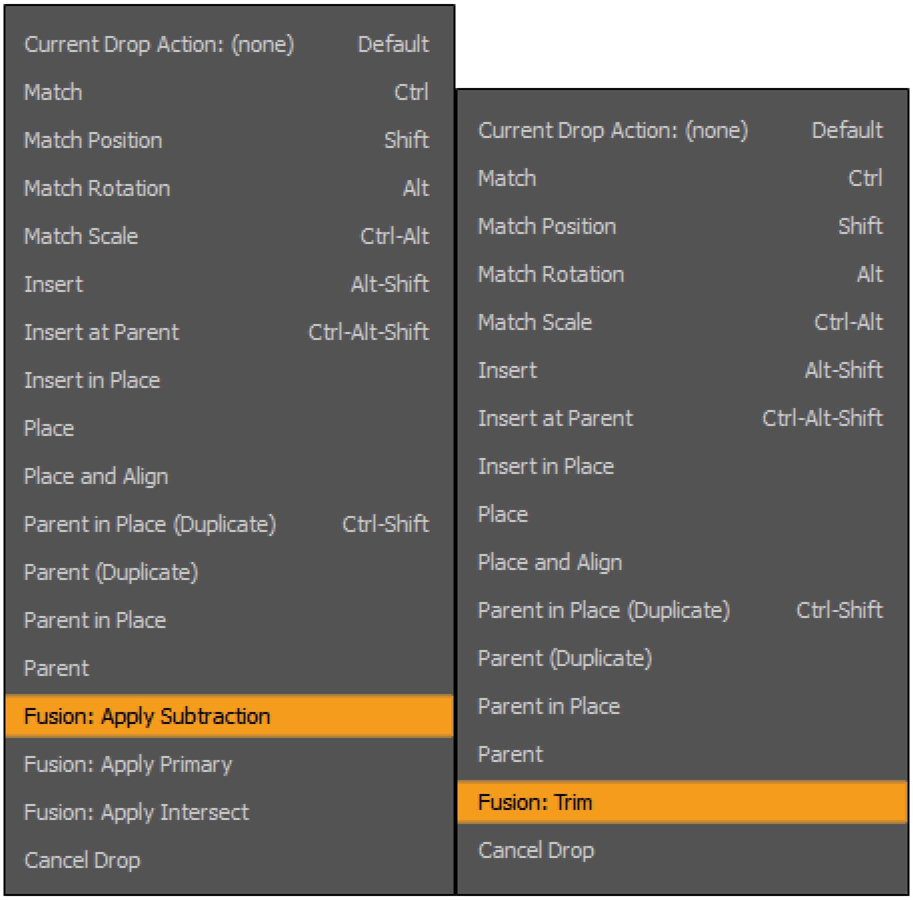

AVP-Topo, Shaded
✓ AVP-Topo, Shaded Wireframe
AVP-Shaded, Shaded
AVP-Shaded, Shaded WIreframe
AVP-Shaded Wireframe, Shaded Wireframe
OGL-Topo, Shaded
OGL-Topo, Shaded Wirefame

Polygon Opacity allows you to adjust the opacity of Fusion Objects visible with **Ghost Mode**.

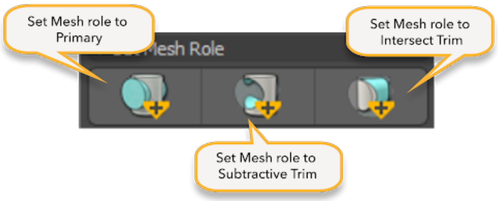
Note: For more information on **Ghost Mode**, see Viewport Toggle Buttons.

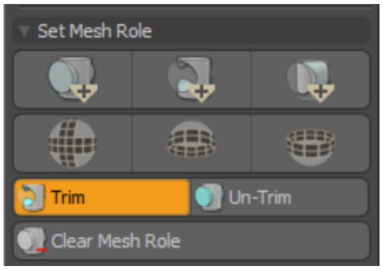
Drag and Drop Workflow

Modo has a contextual menu dedicated to building Fusion models for users who have the Model layout Toolbox minimized. Use this menu to improve your workflow while working in the 3D viewport, which is set to the maximum size. To see the contextual menu, you must have a Fusion Item in your scene and other Fusion Qbic Presets. Select the Qbic preset element, drag it onto the Fusion item, and then hold. The contextual menu displays, allowing you to select any of the following Fusion functions:


Current Drop Action: (none) Default
Match Ctrl
Match Position Shift
Match Rotation Alt
Match Scale Ctrl-Alt
Insert Alt-Shift
Insert at Parent Ctrl-Alt-Shift
Insert in Place
Place
Place and Align
Parent in Place (Duplicate) Ctrl-Shift
Parent (Duplicate)
Parent in Place
Parent
Fusion: Apply Subtraction
Fusion: Apply Primary
Fusion: Apply Intersect
Cancel Drop
Current Drop Action: (none) Default
Match Ctrl
Match Position Shift
Match Rotation Alt
Match Scale Ctrl-Alt
Insert Alt-Shift
Insert at Parent Ctrl-Alt-Shift
Insert in Place
Place
Place and Align
Parent in Place (Duplicate) Ctrl-Shift
Parent (Duplicate)
Parent in Place
Parent
Fusion: Trim
Cancel Drop

These functions are also found in the **Fusion** tab located in the Model layout Toolbox.


Set Mesh role to Primary
Set Mesh Role
Set Mesh role to Intersect Trim
Set Mesh role to Subtractive Trim


Set Mesh Role
Trim
Un-Trim
Clear Mesh Role

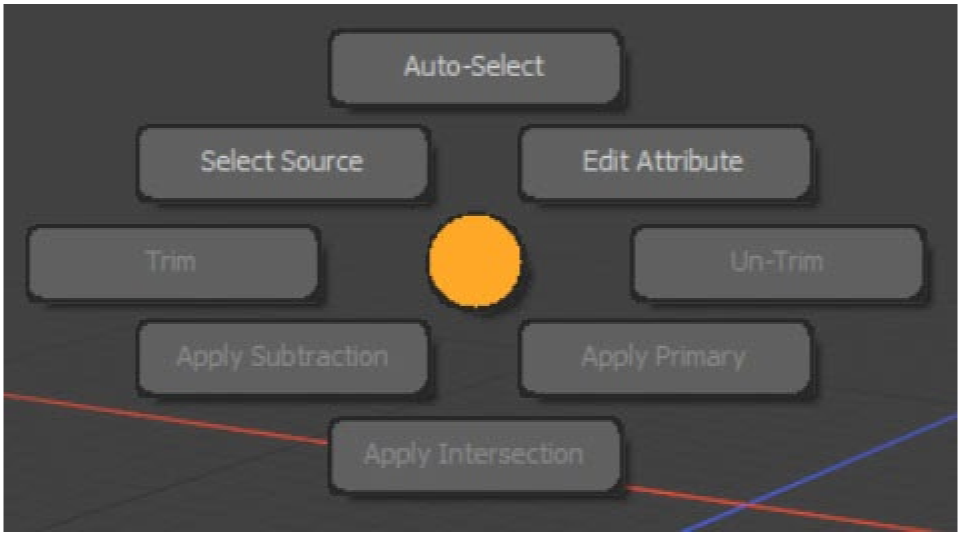
For more information, see Drag and Drop Workflow.

Tip: For information about hiding the Toolbox and other interface elements, see Customizing Your Layout.

MeshFusion Pie Menu

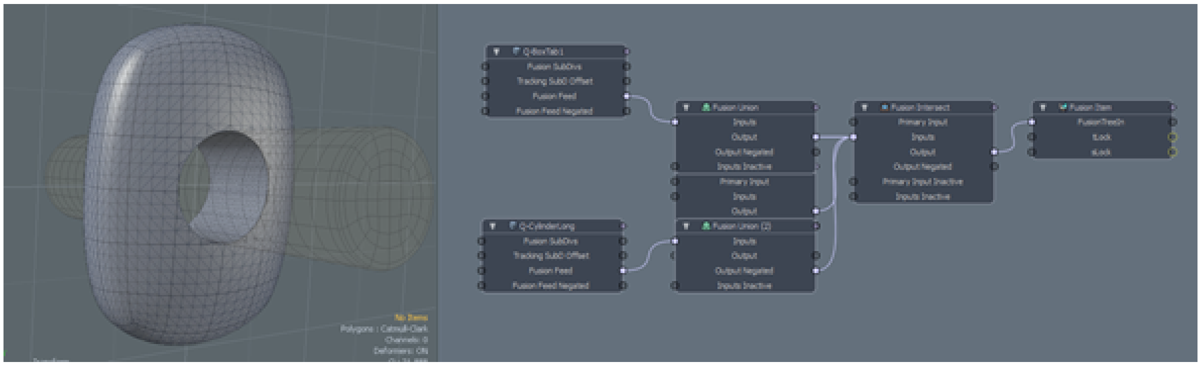
Alternatively to using the contextual menus, you can use the pie menu to quickly select common MeshFusion options when working in the 3D viewport.

Select the appropriate **Fusion Item** and then press **Ctrl** + **F** in the viewport to display the MeshFusion pie menu.


Auto-Select
Select Source
Edit Attribute
Trim
Un-Trim
Apply Subtraction
Apply Primary
Apply Intersection

Schematic Fusion

Modo has a layout dedicated to building Fusion models using the Schematic Viewport. This offers an alternative interface to using the **Fusion** tab.

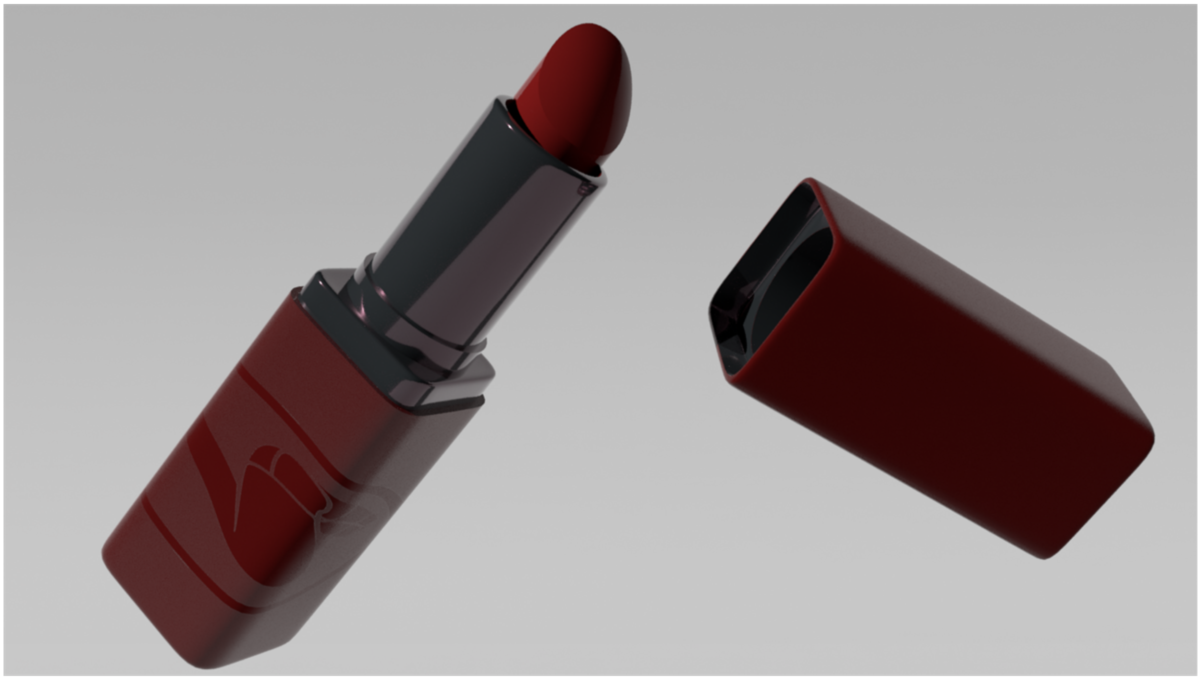

No Items
Polygons : Catmull-Clark
Channels: 0
Deformers: ON
Q-BoxTab1
Fusion SubDivs
Tracking SubD Offset
Fusion Feed
Fusion Feed Negated
Fusion Union
Inputs
Output
Output Negated
Inputs Inactive
Primary Input
Inputs
Output
Fusion Intersect
Primary Input
Inputs
Output
Output Negated
Primary Input Inactive
Inputs Inactive
Fusion Item
FusionTreeIn
tLock
sLock
Q-CylinderLong
Fusion SubDivs
Tracking SubD Offset
Fusion Feed
Fusion Feed Negated
Fusion Union (2)
Inputs
Output
Output Negated
Inputs Inactive

For more information, see Modeling in the Schematic Viewport.

Fusion Example

As a quick introduction to MeshFusion, let's work through a simple example that combines Qbic primitives. This example uses the 3D Tree to work with the Fusion (as opposed to the Schematic).

This example shows you how to model a lipstick case and bullet.



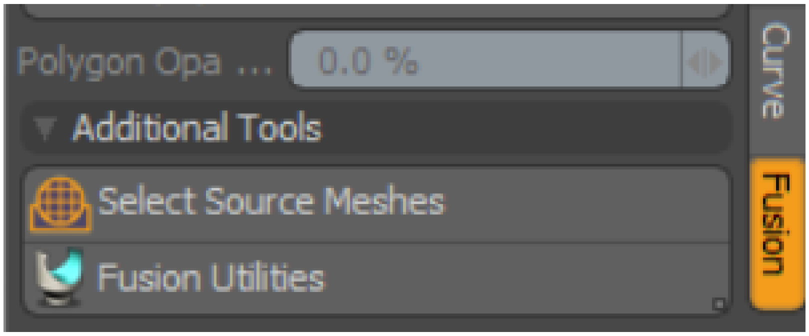
Making the Case: Primary Trims and Embossing

To make the lipstick case, we will be using **Primary Trims** to create the lower half that houses the lipstick bullet, and **Subtractive Trims** to create a hollow in the lid and lipstick tube for the bullet to sit in. To begin with, we will be making the lower half.

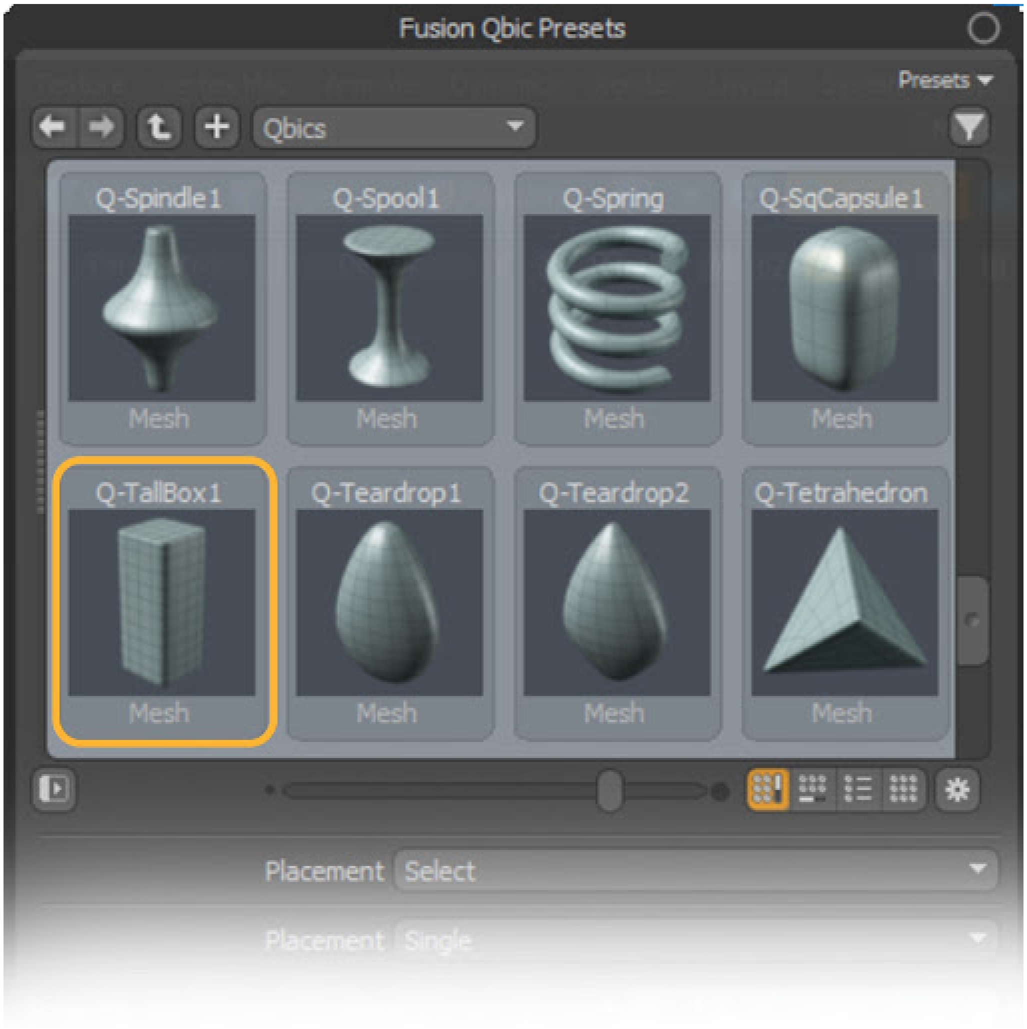
While working with MeshFusion, you may find it beneficial to enable **Auto-Select** and **Select Source** in the **Fusion** tab in the window to the left of your screen. **Auto-Select** automatically disables the visibility of any items that aren't currently selected, and **Select Source** allows you to select and edit the source mesh of a fusion item by clicking on it whilst viewing a Fusion Mesh.

To start the example, make sure you're in the **Model** layout with an empty scene.

1. Click the **Fusion** tab in the modeling toolbox on the left panel



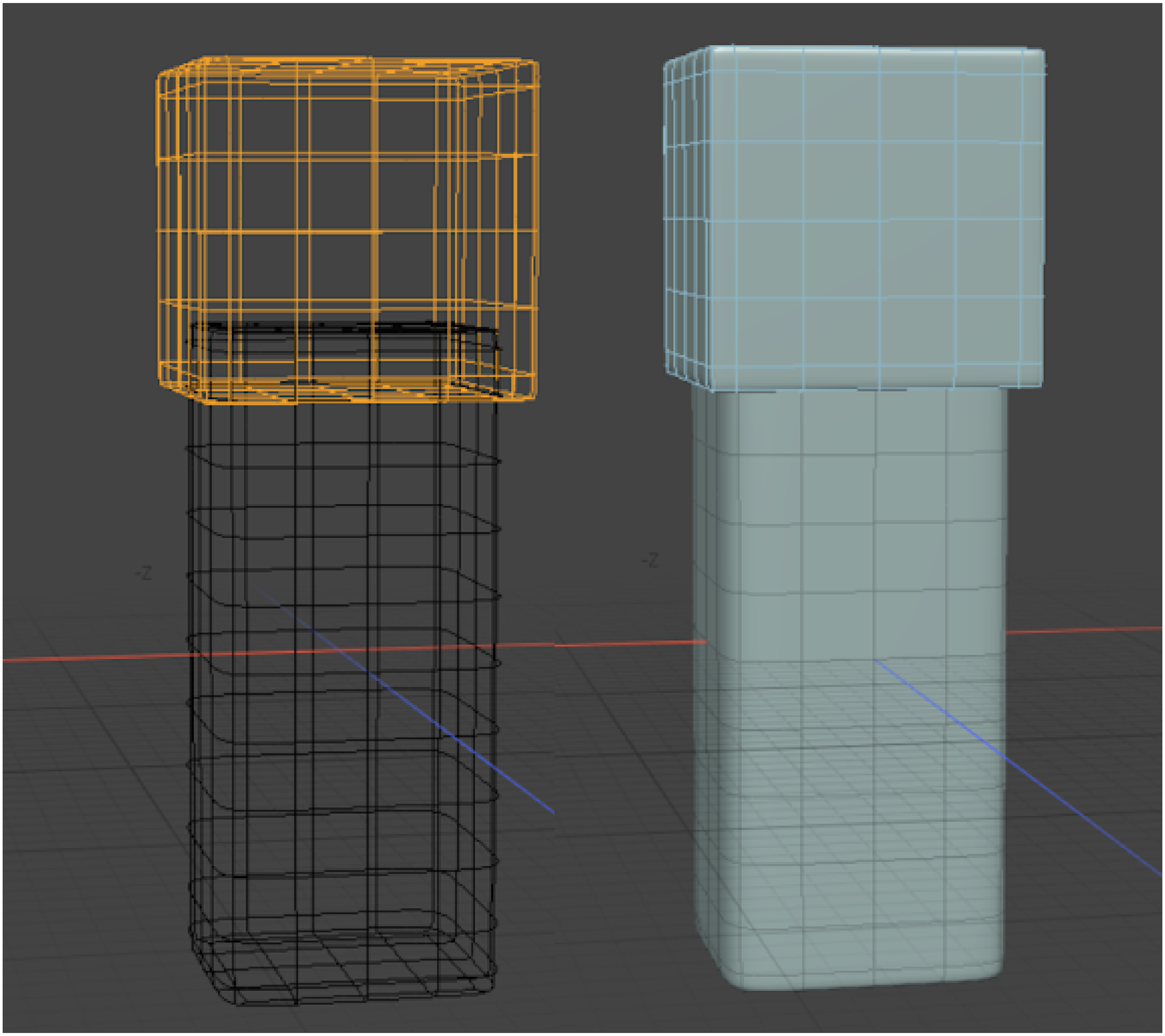
2. Navigate to the **Fusion** tab and open the **Fusion Ready Presets** window
3. Double-click the **Q-TallBox 1** mesh in the **Fusion Qbic Presets** window to add it to the scene.


Fusion Qbic Presets
Presets
Qbics
Q-Spindle1
Mesh
Q-Spool1
Mesh
Q-Spring
Mesh
Q-SqCapsule1
Mesh
Q-TallBox1
Mesh
Q-Teardrop1
Mesh
Q-Teardrop2
Mesh
Q-Tetrahedron
Mesh

Placement Select

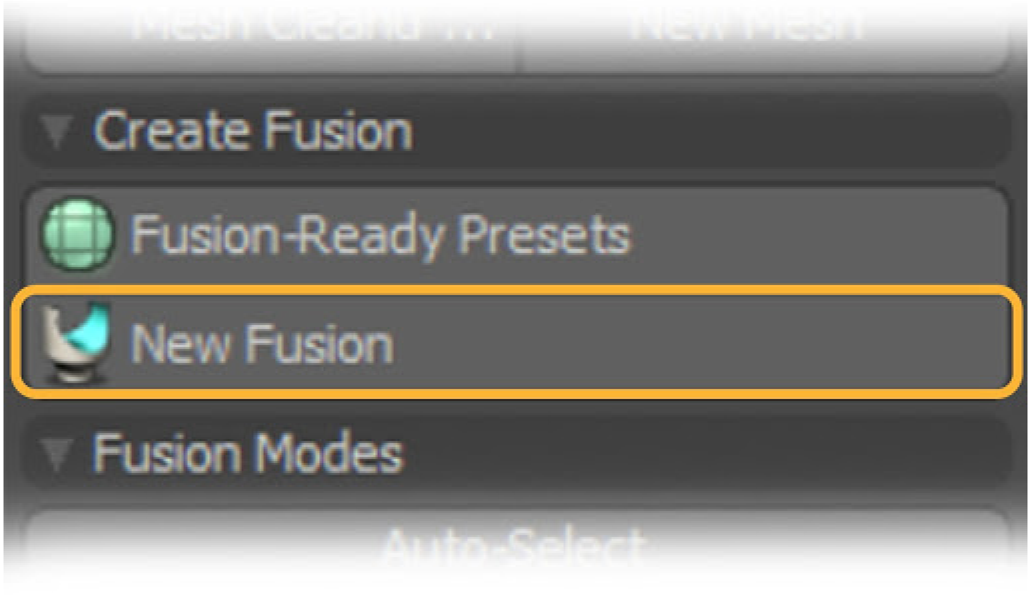
Placement Single

4. Once the box is loaded into the viewport, create a **Q-Medium Cube**. This cube will be used to emboss the base of the lipstick case.
5. Scale the Cube so it's slightly larger in the **X** and **Z-axis**. The **Y-axis** can be left as is. Once the cube has been scaled, move it so that it is intersecting slightly with the top of your first box.

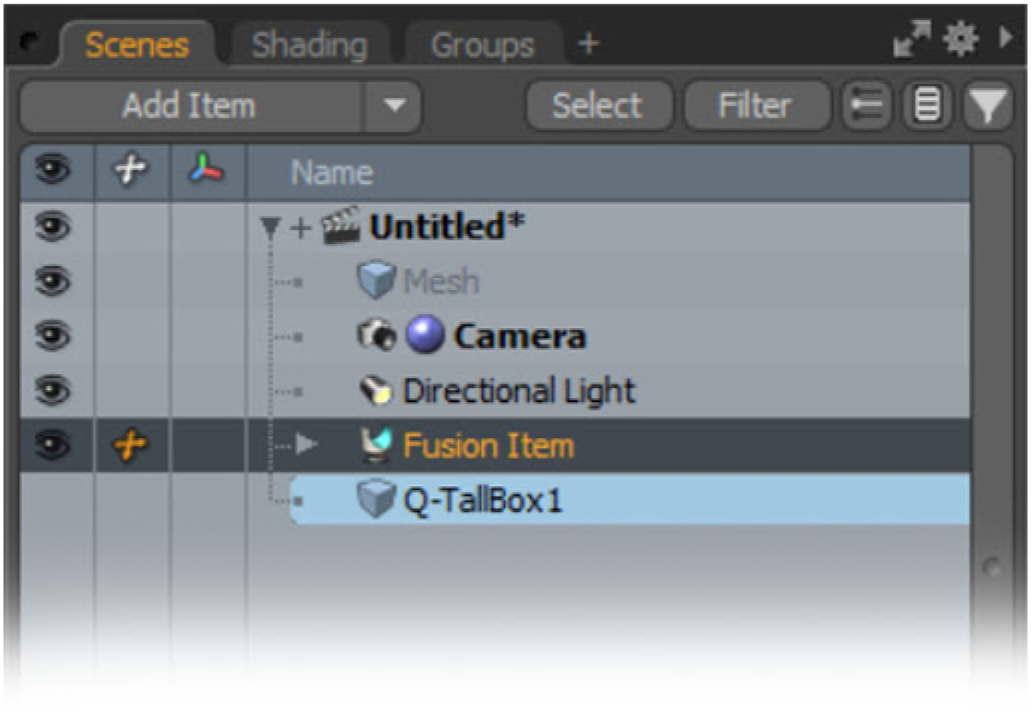


Note: You can change the scale of objects in their axis by adjusting **Scale XYZ** under the **Transform** menu in the **Mesh** tab of the **Properties** window. Scaling, rotating and translating in the **Properties** window is helpful when modeling to measurements or specifications.

6. Select the original box and create a new Mesh Fusion. To do this, navigate to the **Fusion** tab in the panel to the left of the viewport and then press **New Fusion**.

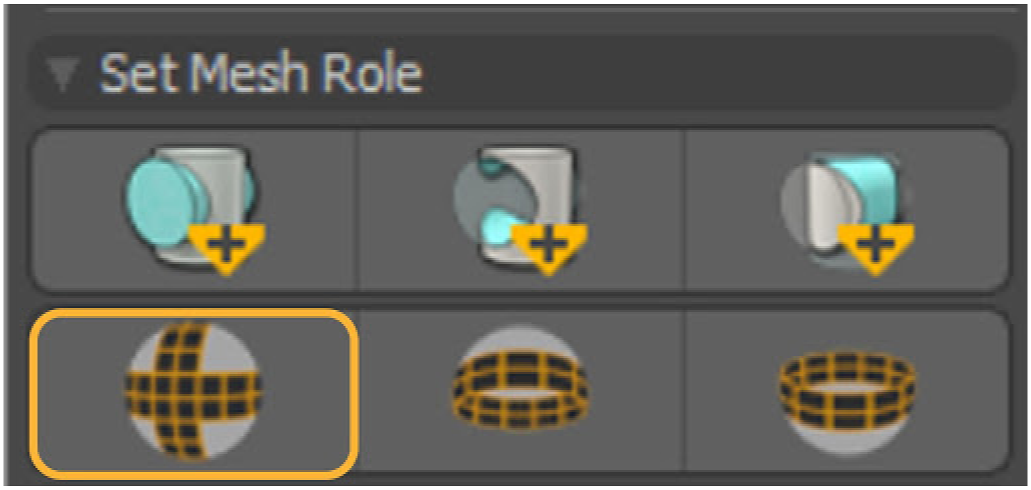

Mesh Cleanup
New Mesh
Create Fusion
Fusion-Ready Presets
New Fusion
Fusion Modes
Auto-Select

Once your mesh has been converted to a Fusion Item, a new item type called **Fusion Item** will appear in the **Items List** to the right of the viewport. This mesh will now be your **Primary** mesh.


Scenes
Shading
Groups
+
Add Item
Select
Filter
Name
Untitled*
Mesh
Camera
Directional Light
Fusion Item
Q-TallBox1

> **Note:** As you add more meshes to your scene, it's good practice to begin naming your items to keep your **Items list** organized and easier to navigate. You can do this by double right-clicking on your mesh or Fusion Item in the **Items List**, and typing in a name.

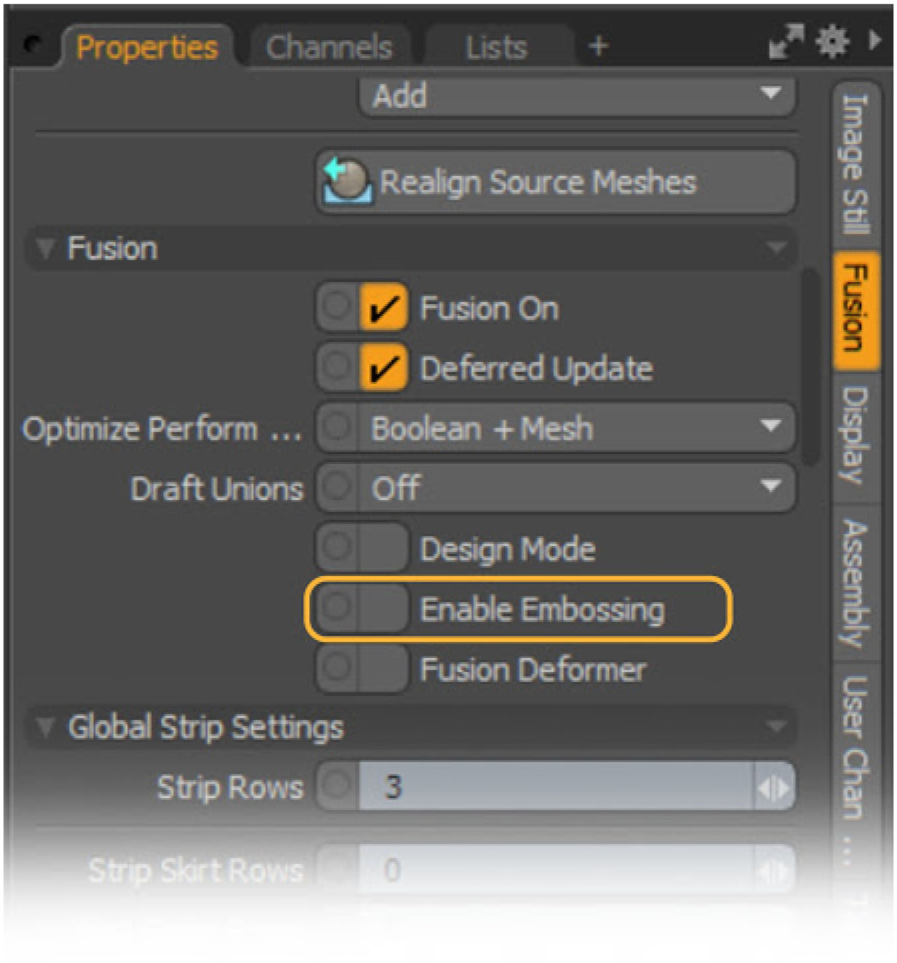
7. With both meshes selected, navigate to the **Fusion** tab on the left and then under **Set Mesh Roles**, click the **Two Sided Subtract Grid** icon to emboss the mesh.



This will create a **Strip** around your mesh, which we will use to emboss the box in the next step.

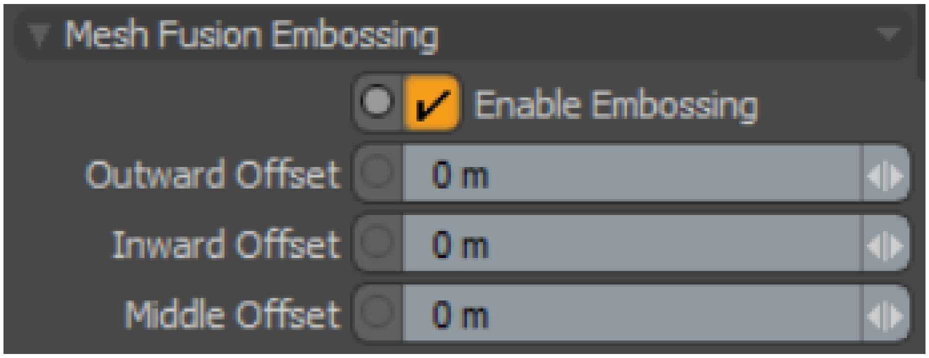
8. Enable embossing in both the **Primary** and **Trim** meshes.

   Do this by selecting **Fusion Item** in the **Items List**, then enable **Embossing** within the **Properties** window on the right side of the 3D Viewport. **Enable Embossing** can be found under the **Fusion** submenu of the **Fusion** tab in the **Properties** window.


Properties
Channels
Lists
+
Add
Realign Source Meshes
Fusion
Fusion On
Deferred Update
Optimize Perform ...
Boolean + Mesh
Draft Unions
Off
Design Mode
Enable Embossing
Fusion Deformer
Global Strip Settings
Strip Rows
3
Strip Skirt Rows
0
Image Still
Fusion
Display
Assembly
User Chan ...

> **Note:** Embossing must be enabled for the fusion item and any items that are being used to emboss the fusion item. Without enabling embossing, you will not be able to emboss your fusion item.

9. Select the **Q-MediumCube** in the **Items List** and repeat this process.
10. Once embossing has been enabled on the Cube mesh, the **Mesh Fusion Embossing** menu will become available.



Mesh Fusion Embossing

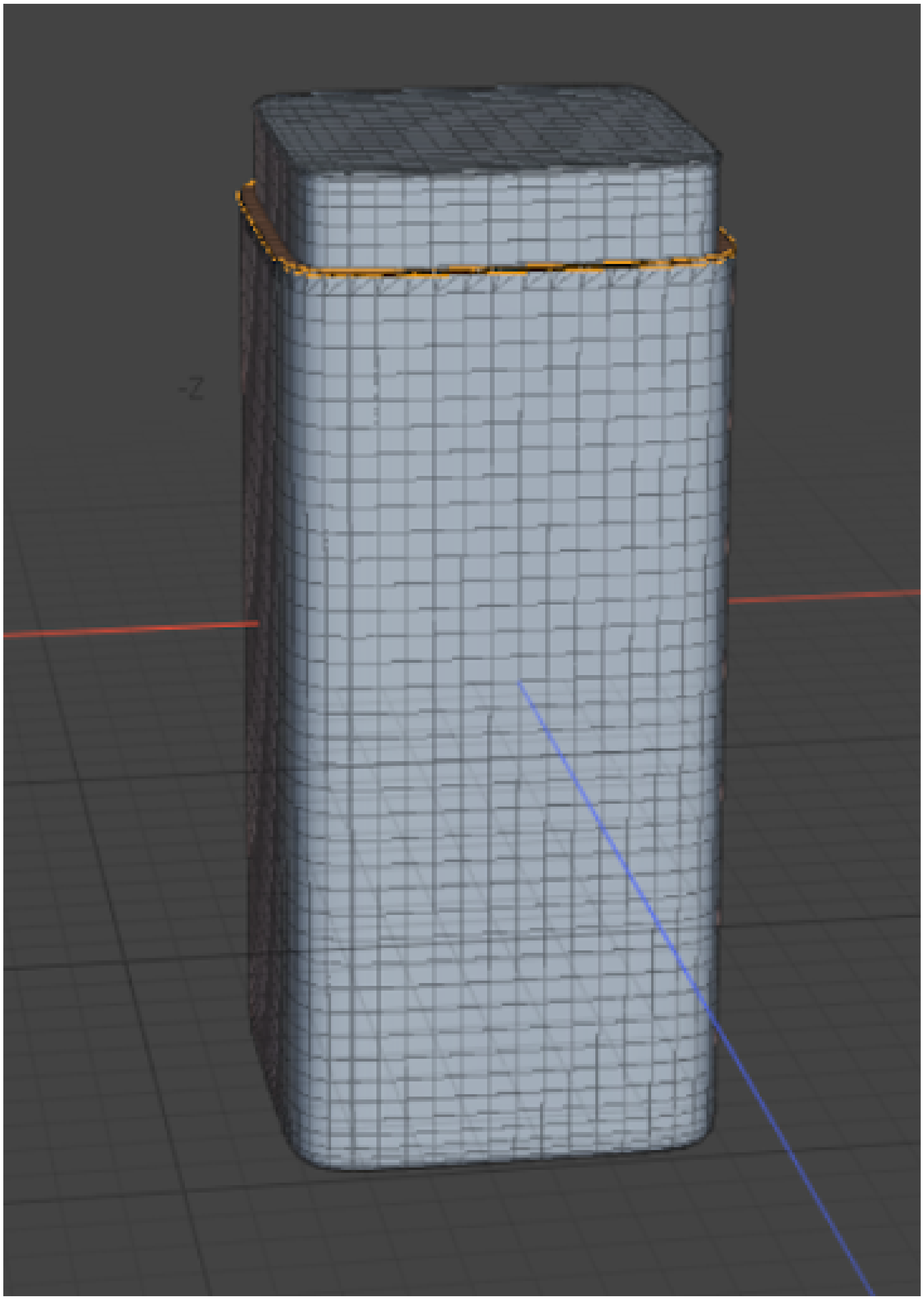
☑ Enable Embossing

Outward Offset 0 m

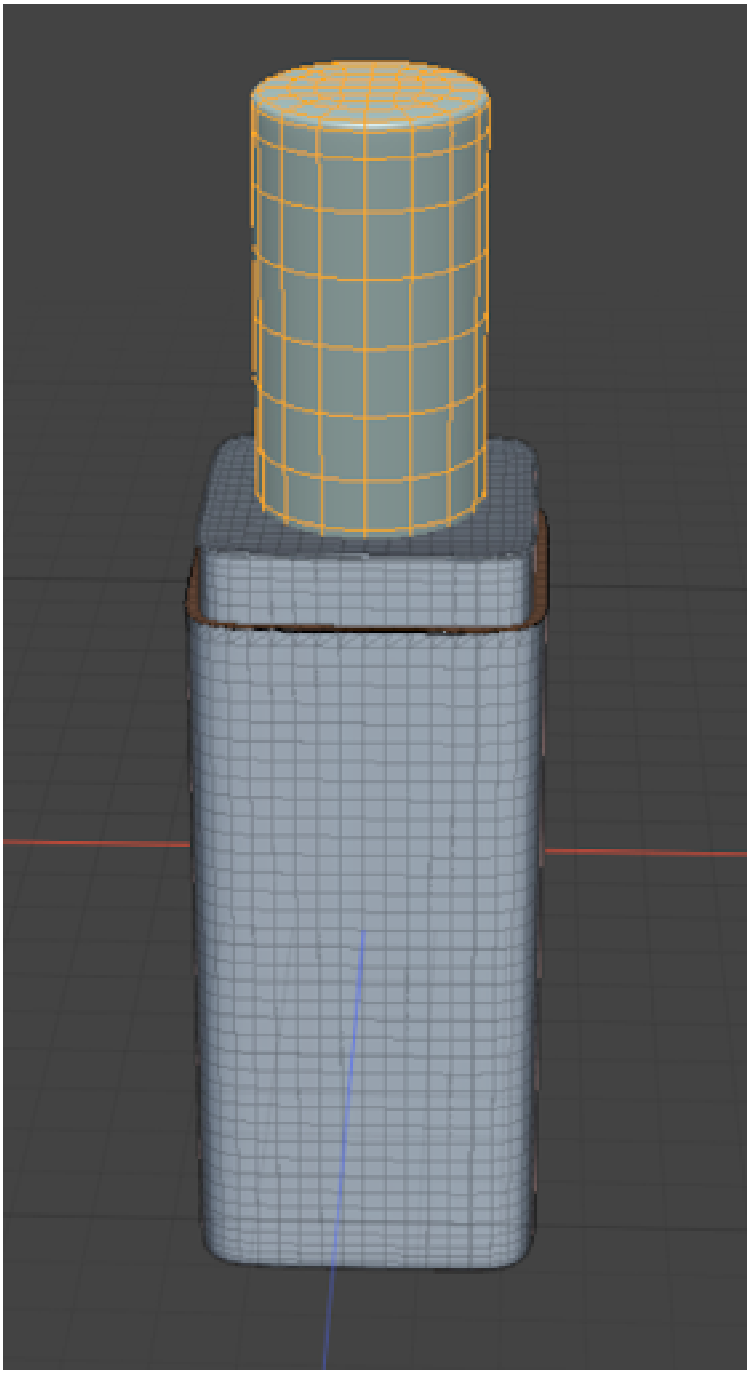
Inward Offset 0 m

Middle Offset 0 m

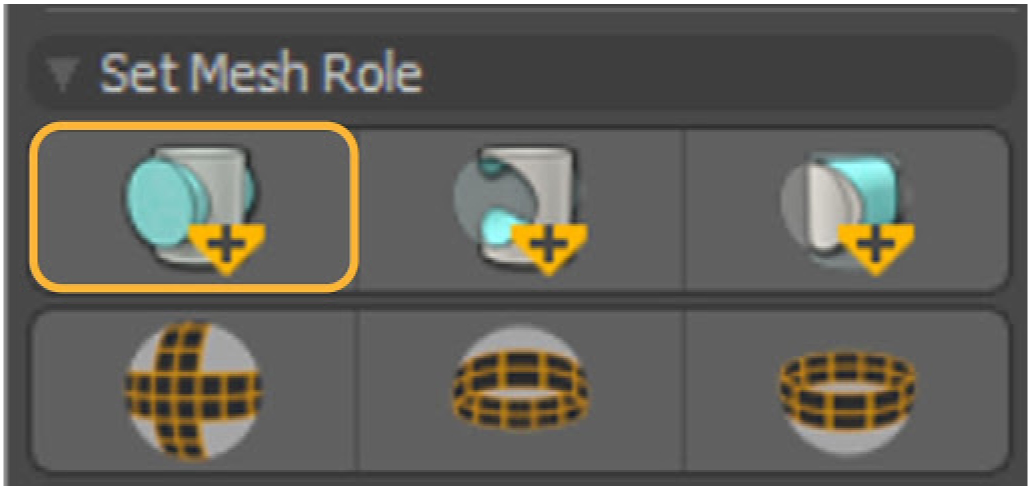
11. Set the **Outward Offset** to 25 mm. This will extrude anything below the embossing strip outward.



12. Create a **QcylinderMed** by selecting it from the **Fusion Ready Presets** window. This is the base for the tube the lipstick bullet sits in.
13. Scale the cylinder so that it sits on top and in the center of the embossed cube. Press **R** to activate the scale tool and then scale the Cylinder 50% in the **X & Z-Axis** and 60% in the **Y-Axis**.
14. After the cylinder has been scaled, press **W** and move the cylinder in the **Y-Axis** so it's intersecting the top of the box slightly.

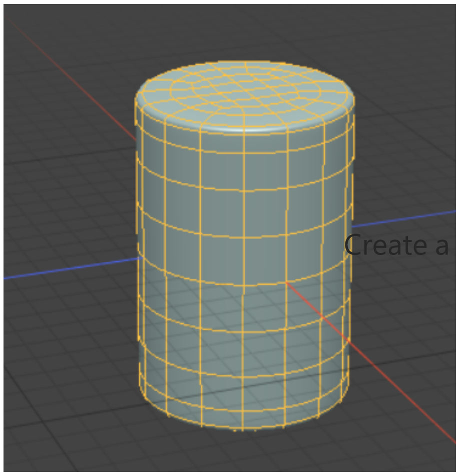


15. Once the cylinder has been positioned, select both the cylinder and the box in the **Items** menu.
16. On the left panel, in the **Fusion** tab, under **Set Mesh Role** click the **Apply Primary** icon.

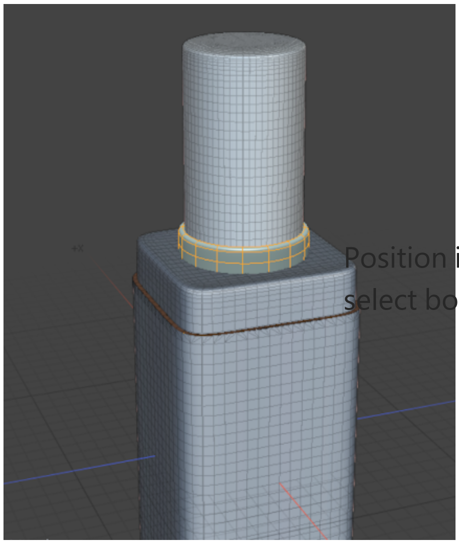


The two meshes are now fused.

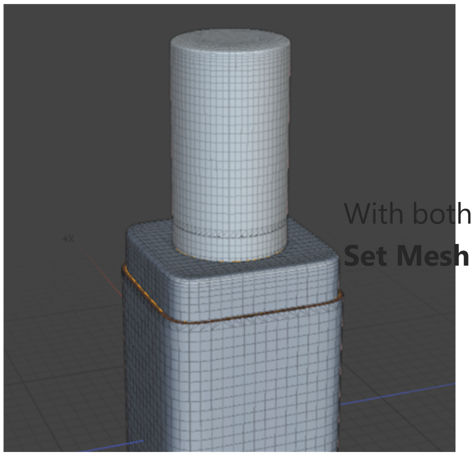
17. Now that the two meshes have been fused, use the Emboss function again to create a rim around the lipstick tube.



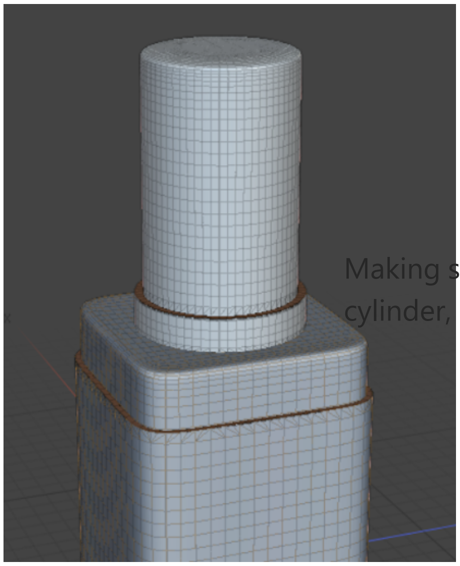
Create a **Q-CylinderMedium** and scale it 55% in all Axis.



Position it so it's intersecting with the bottom of your lipstick tube and then select both cylinders in the **Items List**.



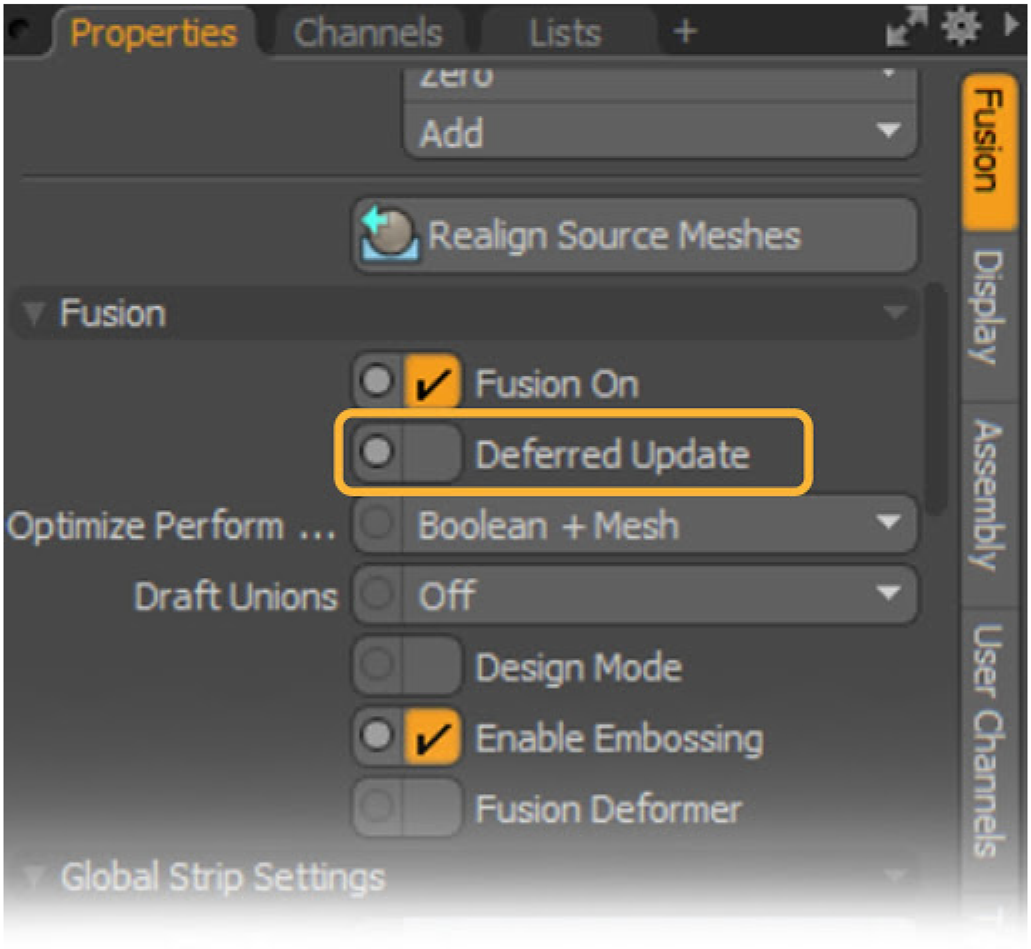
With both cylinders selected, click the **Two-Sided Subtract Grid** button in the **Set Mesh Role** menu.



Making sure that **Embossing** is enabled on the fusion item and your smaller cylinder, set the cylinder's **Inward Offset** to 35 mm.

In the next section, we will be using **Subtractive Trims** to hollow out the lipstick tube, as well as create a cap for the piece we just made. However, should you want to, you can now re-size or reposition your mesh, as well as change the offset of your embossings to change the look of the lipstick as you like. By default in Modo 15 or later, **Deferred Update** is enabled, meaning that any changes to the **Fusion Item** will not update until you release the tool.

This can be turned off and changed to real-time updates by selecting your **Mesh Item** in the **Items** list, and then unchecking **Deferred Update** in the **Fusion** sub-menu of the **Fusion** tab in the **Properties** window.


Properties
Channels
Lists
+
Zero
Add
Realign Source Meshes
Fusion
Fusion
Display
Assembly
User Channels

☑ Fusion On

☐ Deferred Update

Optimize Perform … Boolean + Mesh

Draft Unions Off

☐ Design Mode

☑ Enable Embossing

☐ Fusion Deformer

Global Strip Settings

Making the Cap: Subtractive Trims

Now that you have created the base and tube of the lipstick case, it's time to hollow the tube and create the cap. To do both of these, we will be using **Subtractive** trims. **Subtractive** trims are the opposite of **Primary** trims; where **Primary** trims adds meshes to the **Fusion Item**, **Subtractive** trims remove mesh from a Fusion Item.

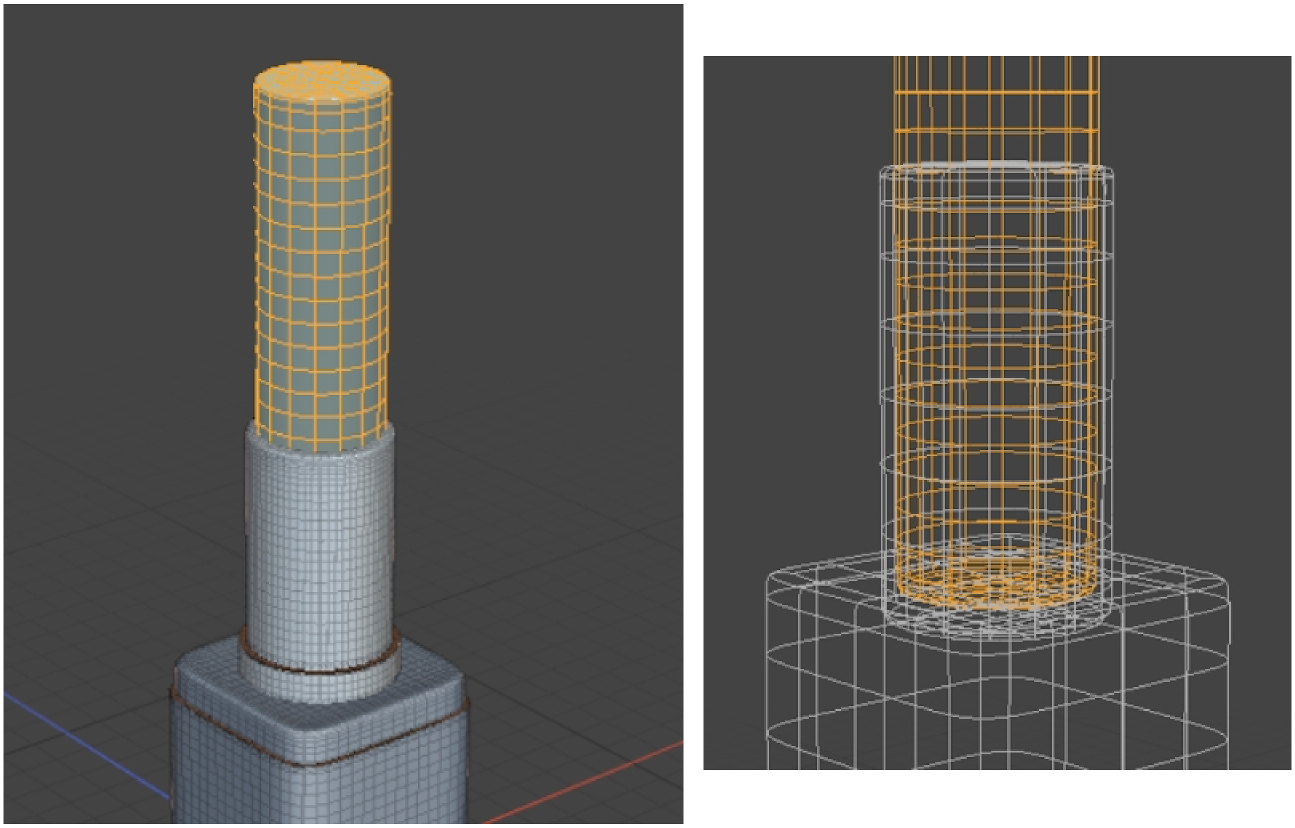
As with the base of the lipstick case, we will be using Qbic meshes to create our **Fusion Item** and **Subtractive** trims.

Hollowing the Lipstick Tube

1. Create a **Q-CylinderLong** by using the **Fusion Qbic Presets** window.

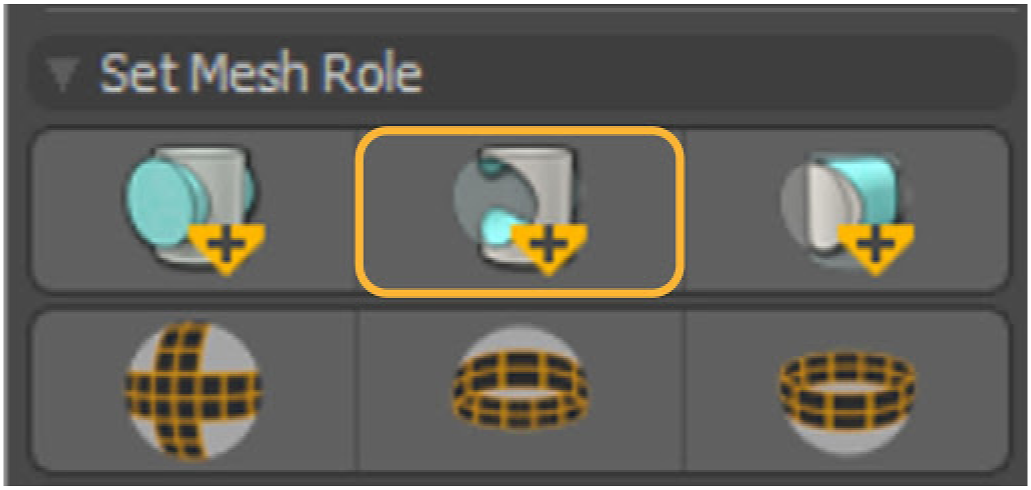
2. Scale the cylinder inwards so it's slightly smaller than the lipstick tube. Once scaled so that it fits within the tube, translate your cylinder in the **Y-Axis** so that it's sat slightly above the square base of the lipstick case.

> **Note:** It's easier to see if your mesh is intersecting or not if you set your view to **Wireframe**. To do this, select the dropdown menu in the top left of your viewport and change it from **Default** to **Wireframe**.

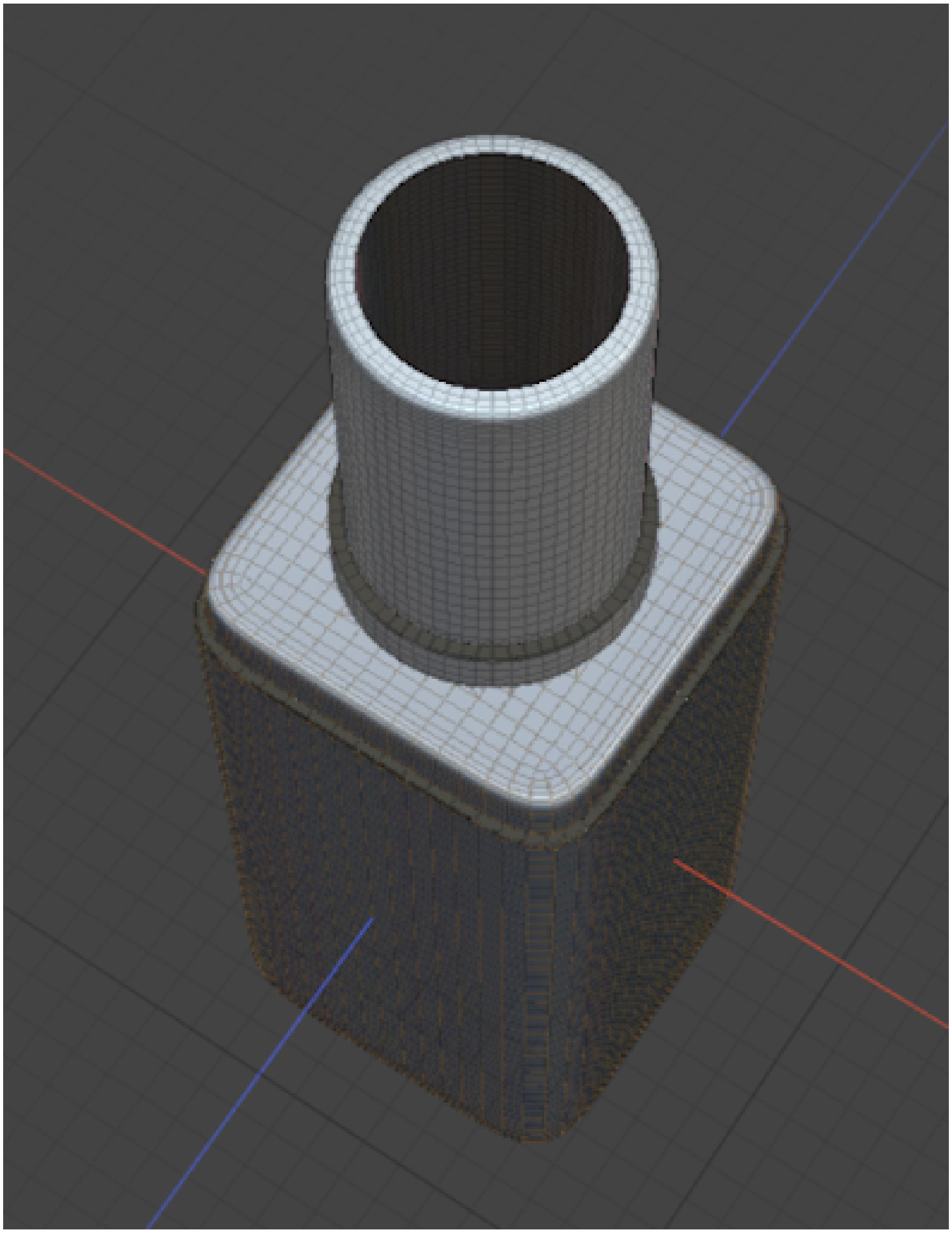


Example of how the subtractive mesh should be positioned in the lipstick tube.

3. Select the cylinder that you've just placed and the lipstick tube in the **Items List**.
4. With both meshes selected, navigate to the **Set Mesh Roles** submenu of the **Fusion** tab and click the **Apply Subtractive Trim** icon.



5. With the subtraction in place, the bottom half of the lipstick tube is now finished. The next section will cover making the cap for the lipstick.

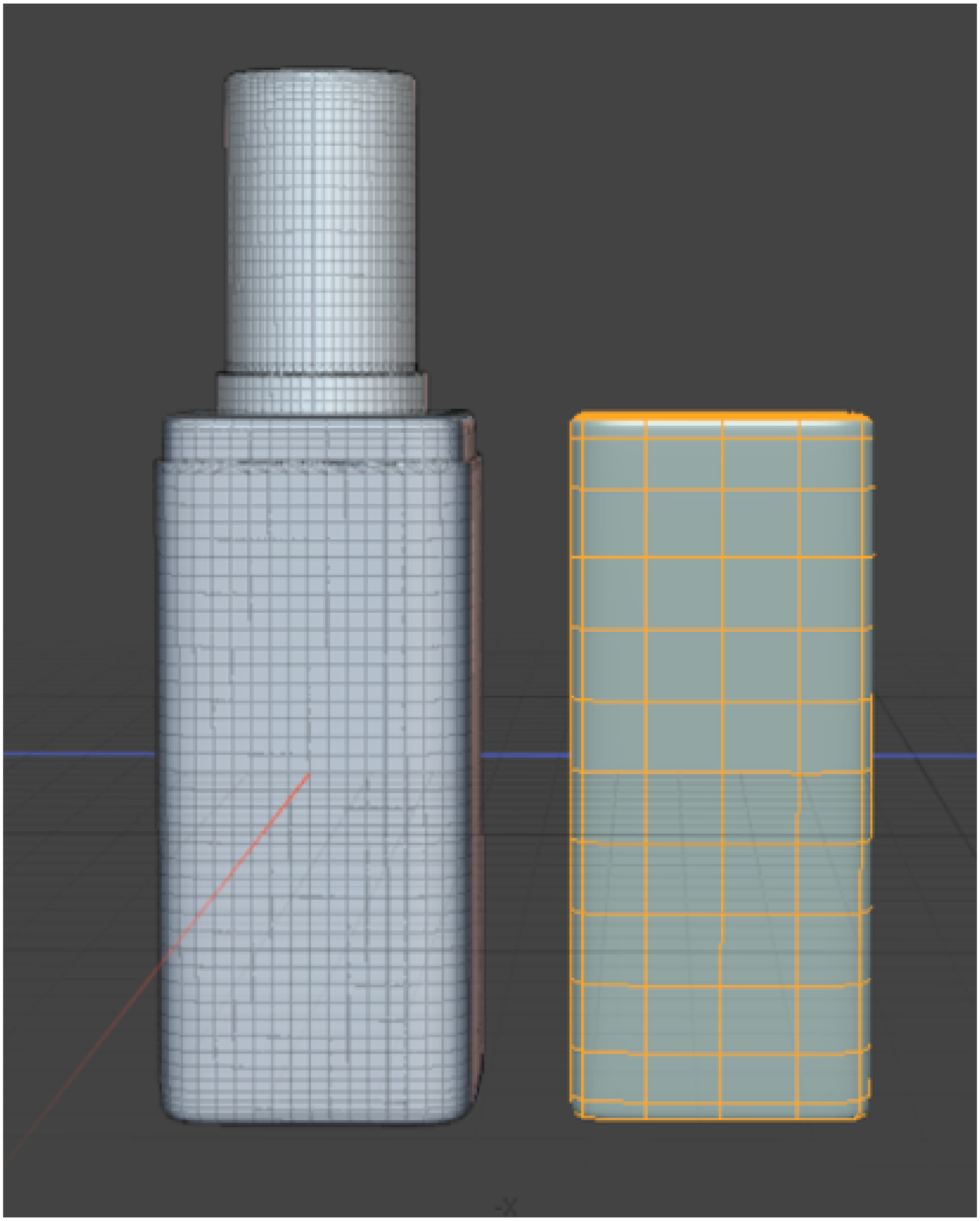


Making the Cap

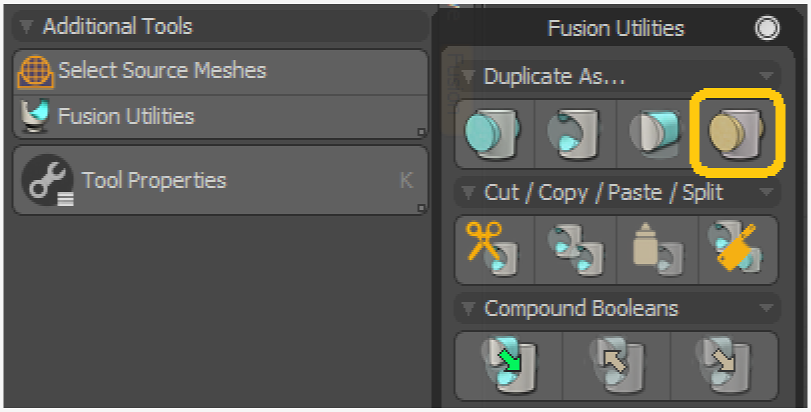
To make the cap for the lipstick case, create a new Fusion Item with Qbic meshes and then hollow it out by layering Subtractive Trims.

1. Create another **Q-TallBox-1** by navigating to the **Fusion Qbic Presets** window once, and double click to bring the Mesh in to the 3D viewport.

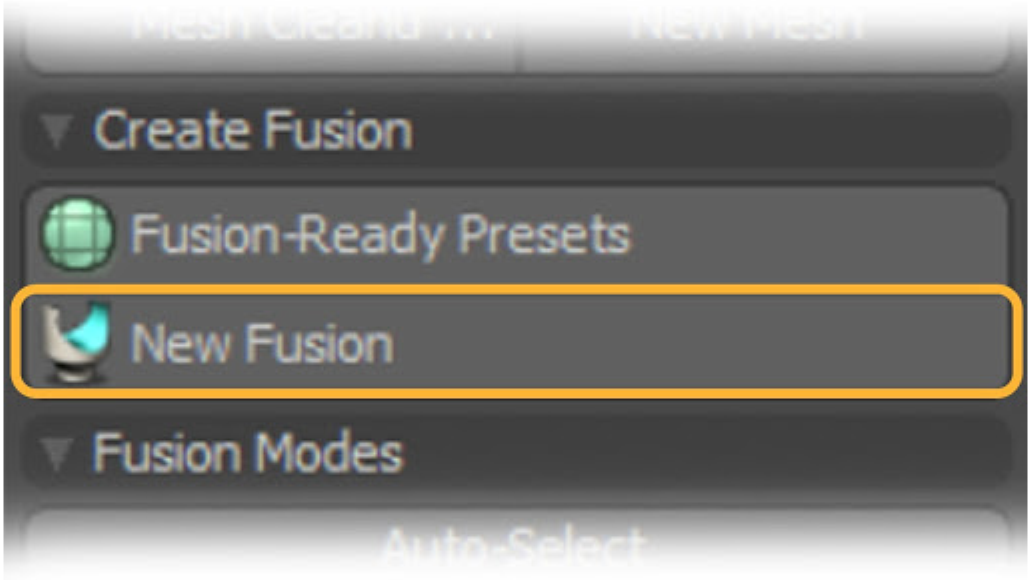
2. Make sure the new **Q-TallBox-1** is the same size as the box we used to create the base of the lipstick case.



Note: You can duplicate an existing mesh so that the two pieces of the lipstick case are consistent. To do this, select the mesh you want to duplicate and then click on the **Fusion Utilities** button in the **Fusion** tab of the left window. Click the **Duplicate as non-Fusion Mesh** button in the **Fusion Utilities** window and your mesh is duplicated.

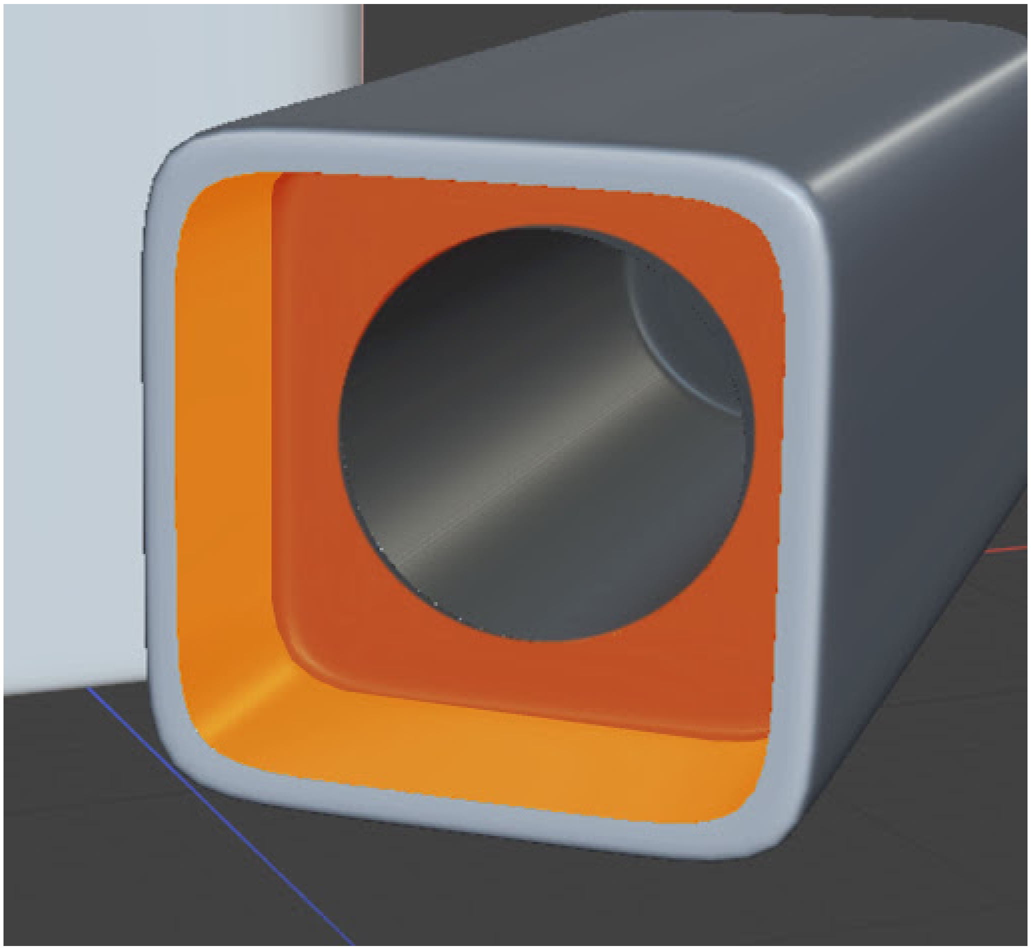

Additional Tools
Select Source Meshes
Fusion Utilities
Tool Properties
K
Fusion Utilities
Duplicate As...
Cut / Copy / Paste / Split
Compound Booleans

3. Select the **Q-TallBox-1** you just created, and click the **New Fusion** button that can be found in the **Fusion** tab in the window to the left.


Mesh Cleanup...
New Mesh
Create Fusion
Fusion-Ready Presets
New Fusion
Fusion Modes
Auto-Select

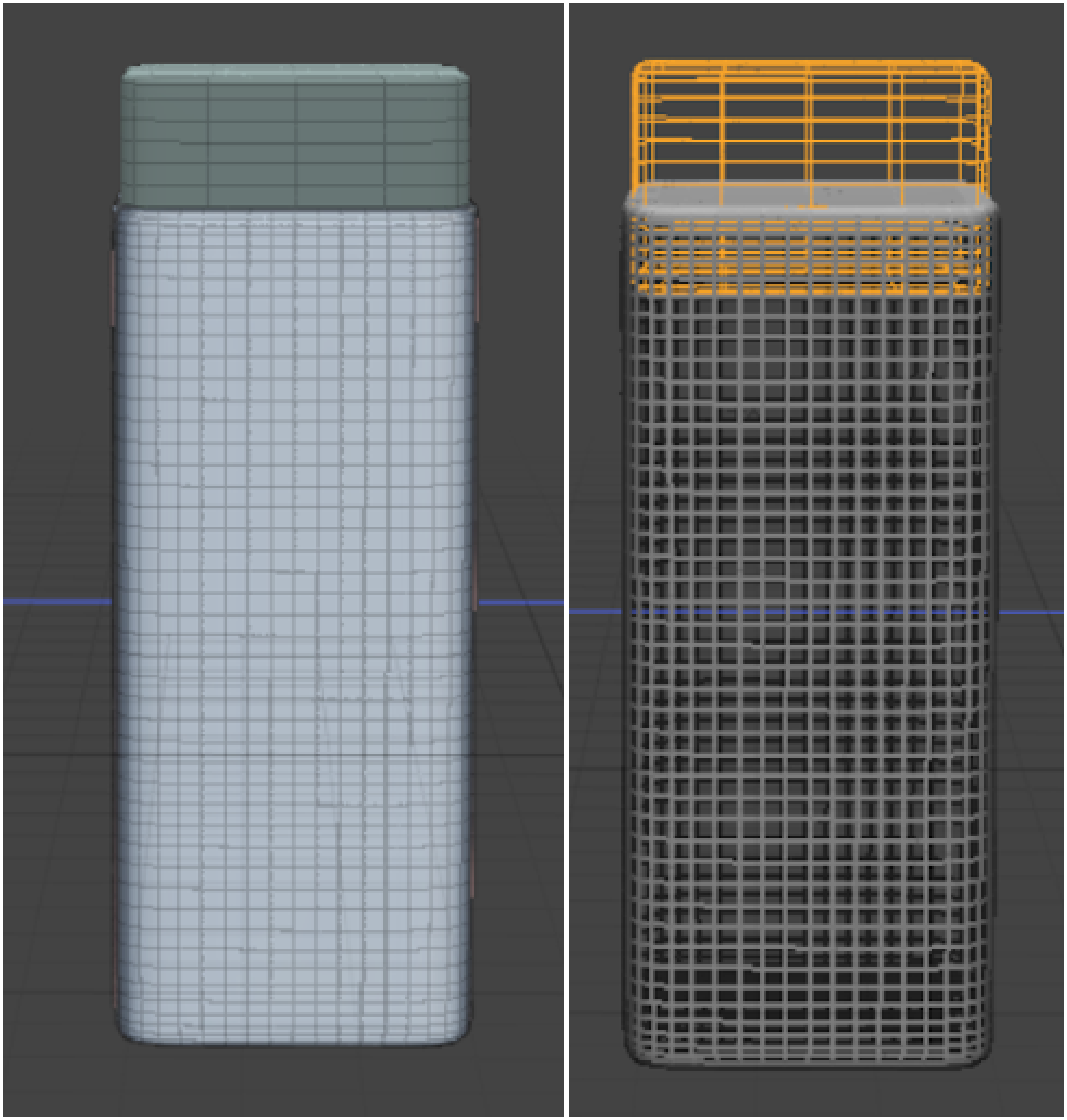
The reason we're creating another **Fusion Item** is because the cap is an object separate from the lipstick base that we were just working on. This means that in order to add trims or strips to it, it needs to be its own **Fusion Item**.

4. Select the new **Fusion Item** to begin hollowing out the inside of the cap. The first subtraction we are going to make will create the recess in the cap that is highlighted in the image below.

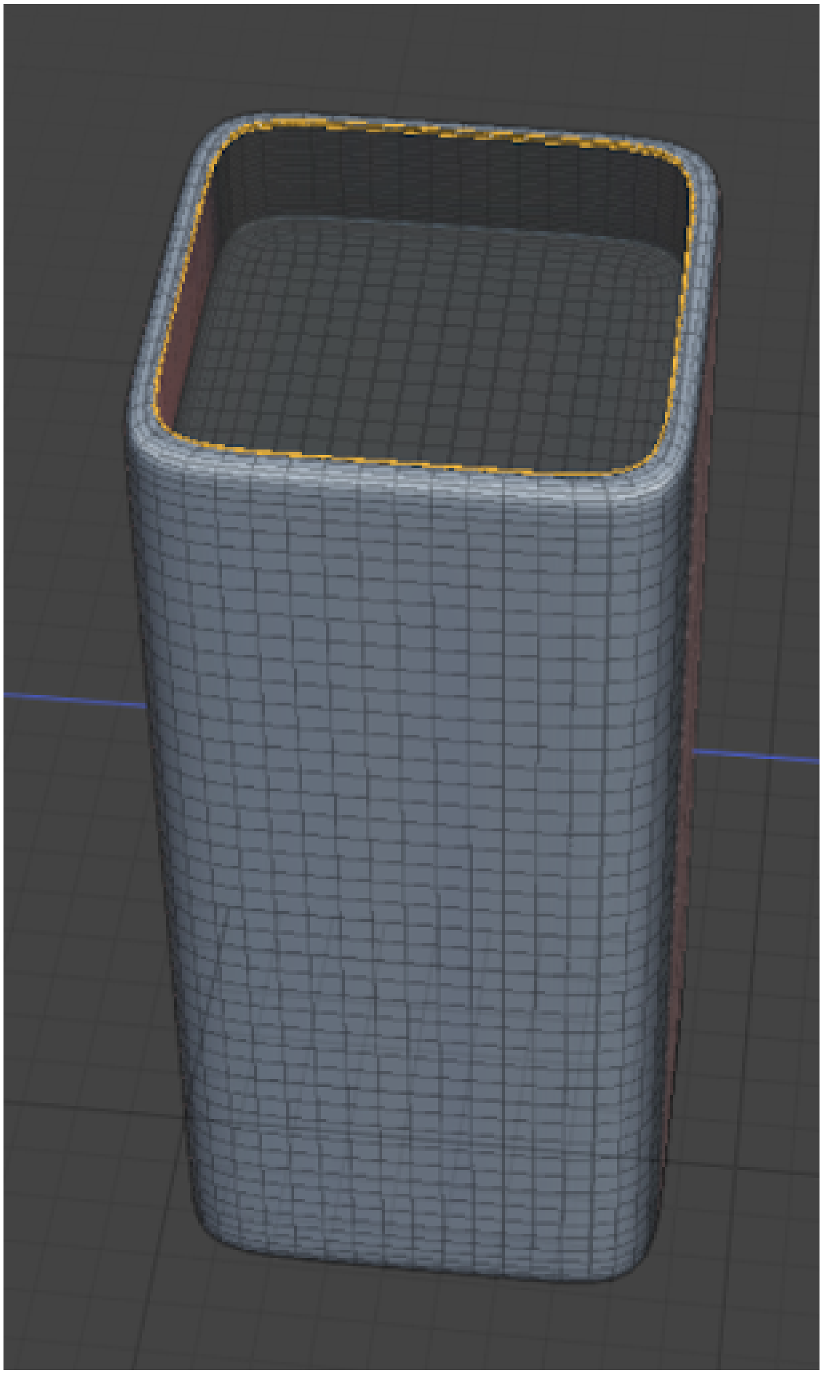


In a tube of lipstick, this is where the embossed edges would click into place to create a locking mechanism and prevent the cap from falling off accidentally.

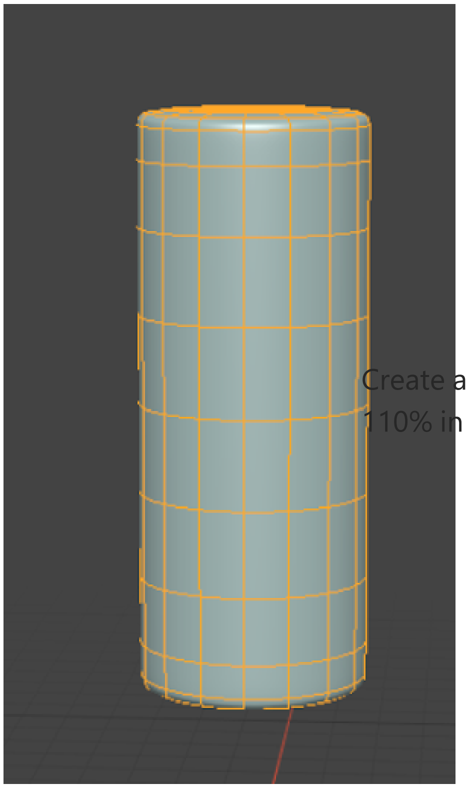
5. Create another **Q-Tallbox** and then change the scale of the cube in the **X** and **Z-Axis** to 90% and to 25% in the **Y-Axis**. This will be used to create the recess highlighted in the example shown.
6. Once the cube has been scaled, move it in the **Y-Axis** so that it's sat slightly within the cap as shown in the example.



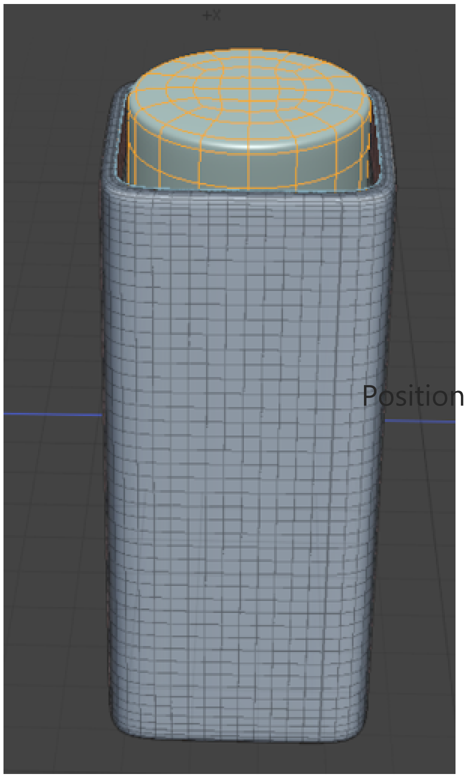
7. Select both the cube and the mesh for the cap in the **Items List**.
8. Navigate to the **Set Mesh Roles** submenu of the Fusion tab and click the **Apply Subtractive Trim** icon.
9. With the **Subtractive Trim** added, you will now have a mesh that looks like the image below.



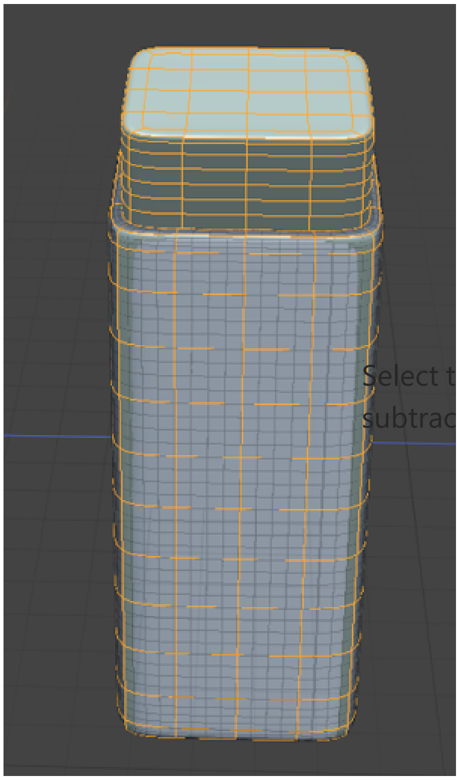
10. To finish off the cap, we need to hollow it out further with a cylinder. This will allow the tube of the lipstick to sit within the cap. The process is the same as the one we have just covered, but will be outlined in the image sequence below.



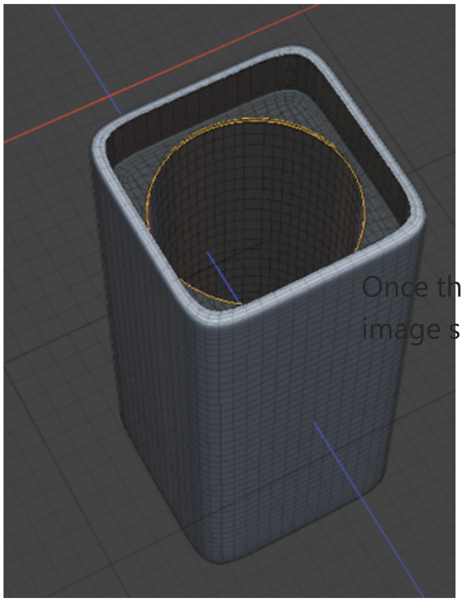
Create a **Q-CylinderMedium** and scale it so it’s 65% in the **X** and **Z-Axis** and 110% in the **Y-Axis**.



Position the Cylinder so that it’s inside the cap.

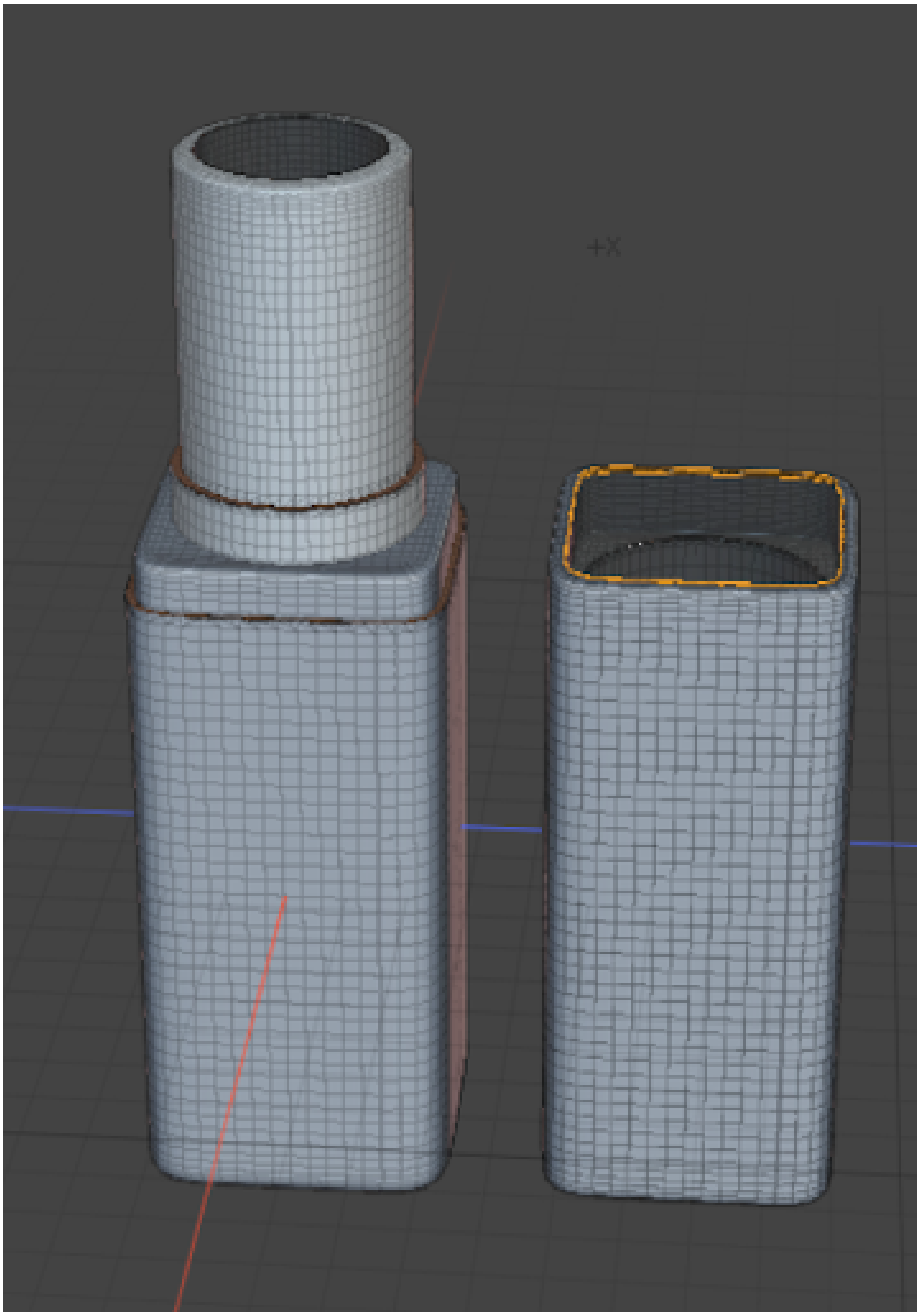


Select the Cylinder, cap Fusion Item, and the cube that has already been subtracted and then apply a **Subtractive Trim**.



Once the **Subtractive Trim** is applied, you are left with a cap that looks like the image shown.

We now have a lipstick tube and cap that should look like the image below.

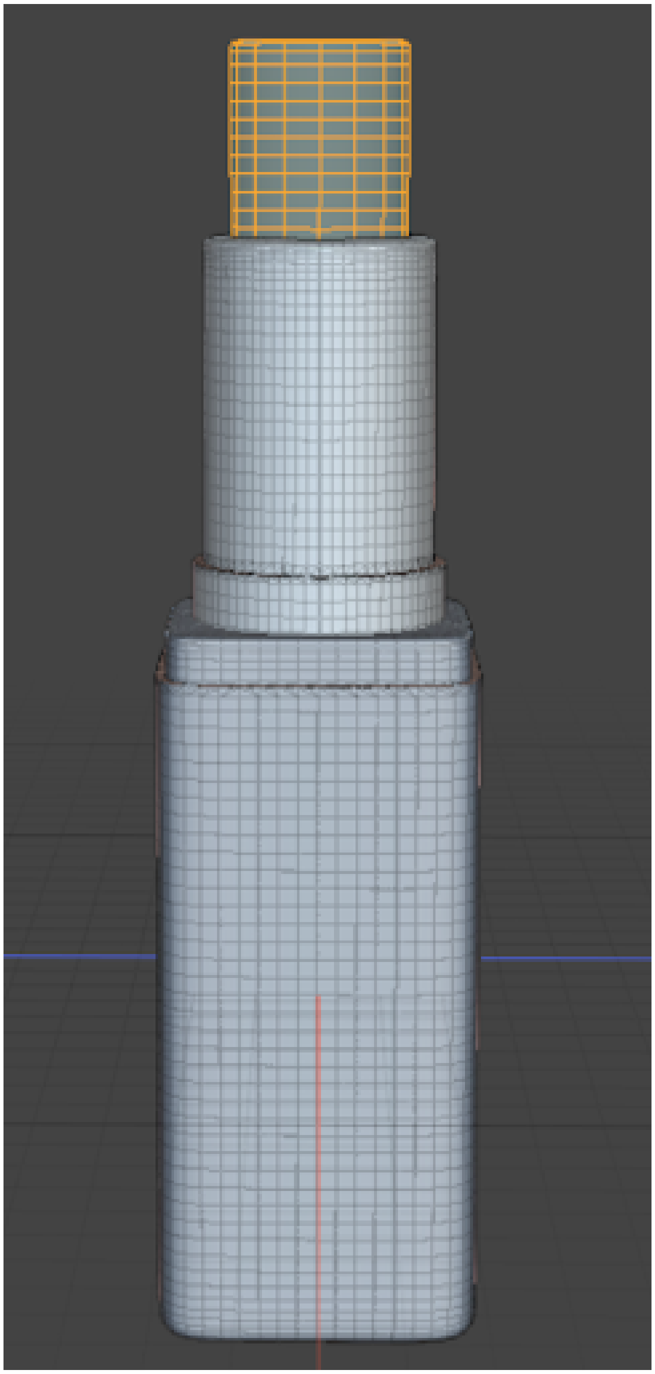


The next section will cover how to use **Intersect Trims** to create the lipstick bullet, as well as how to edit strips and patches and how to convert your **Fusion Items** into finished meshes.

Making the Lipstick Bullet: Intersection Trims

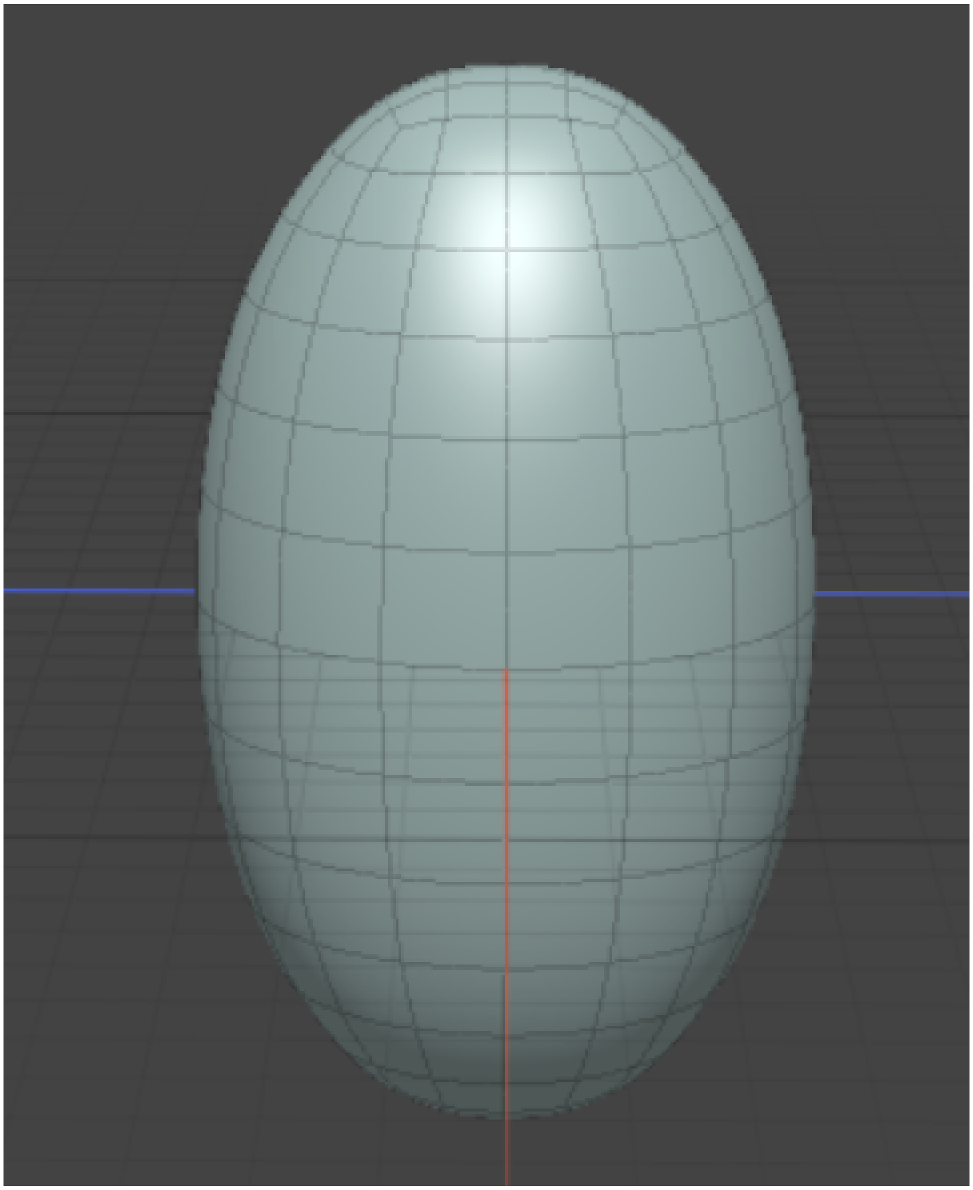
The final piece of mesh we will be making is the lipstick bullet. To make the bullet, we will use two Qbic meshes and an **Intersection Trim**. When applied to a mesh, anything inside of the Intersecting mesh is kept, while anything outside of the mesh is removed.

1. Create a **Q-CylinderLong**.
2. Scale it so it fits within the lipstick tube.



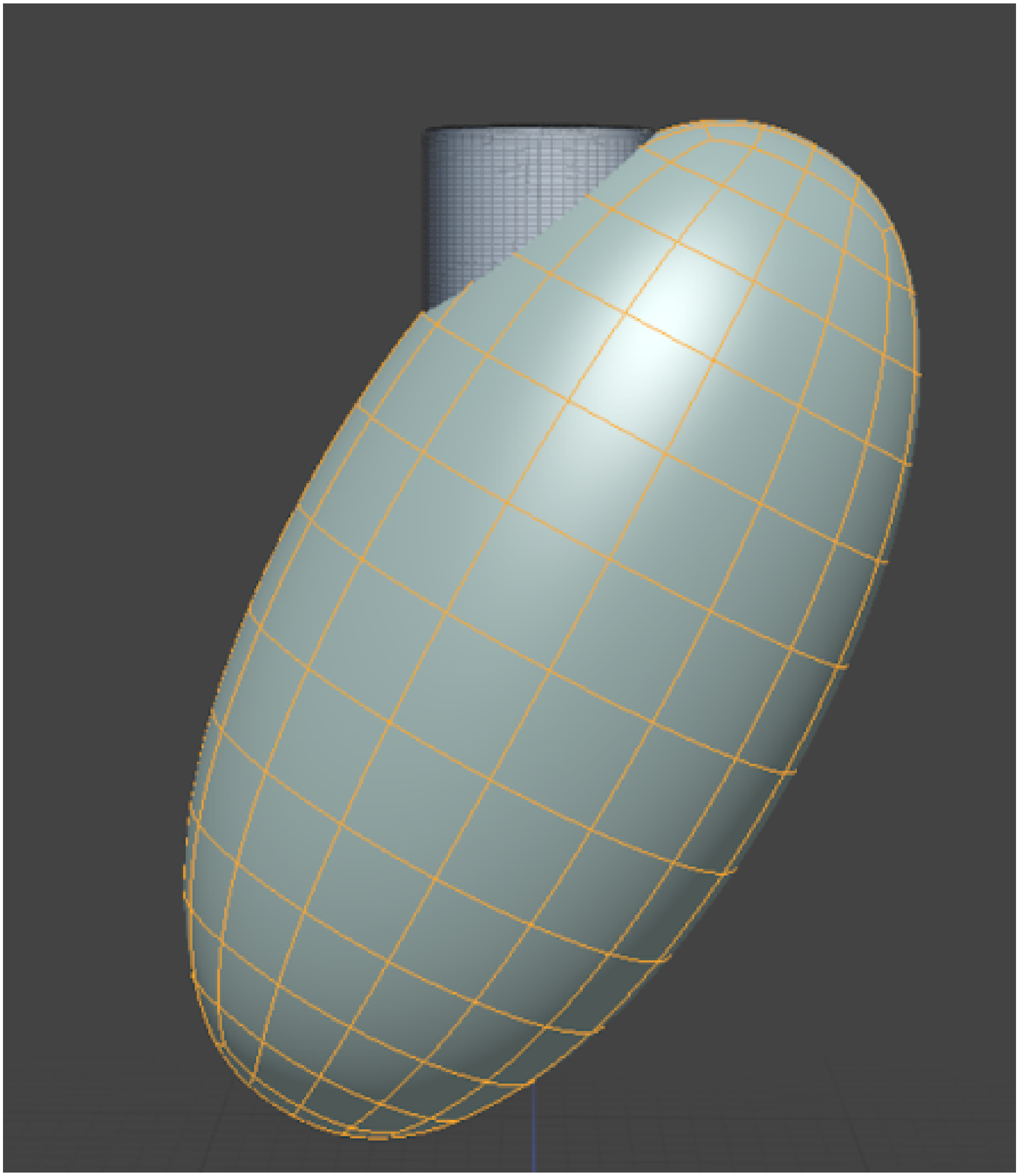
3. Click **New Fusion** in the **Fusion** tab to make the lipstick bullet a stand-alone **Fusion Item**, similar to the cap mesh.
4. Once the cylinder has been converted to a **Fusion Item**, we need to generate the mesh that will be used for the **Intersection Trim**.

Create a **Q-Ellipsoid-2** by navigating to the Fusion **Qbic Fusion Presets** window and double-clicking the mesh.

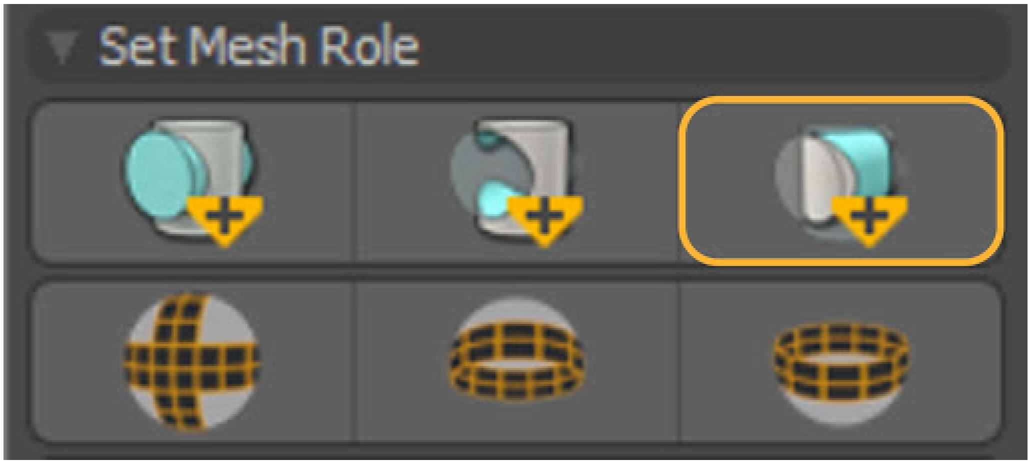


The curve of the ellipsoid will be used to cut the top off the cylinder and create a slightly curved, angled tip on the lipstick.

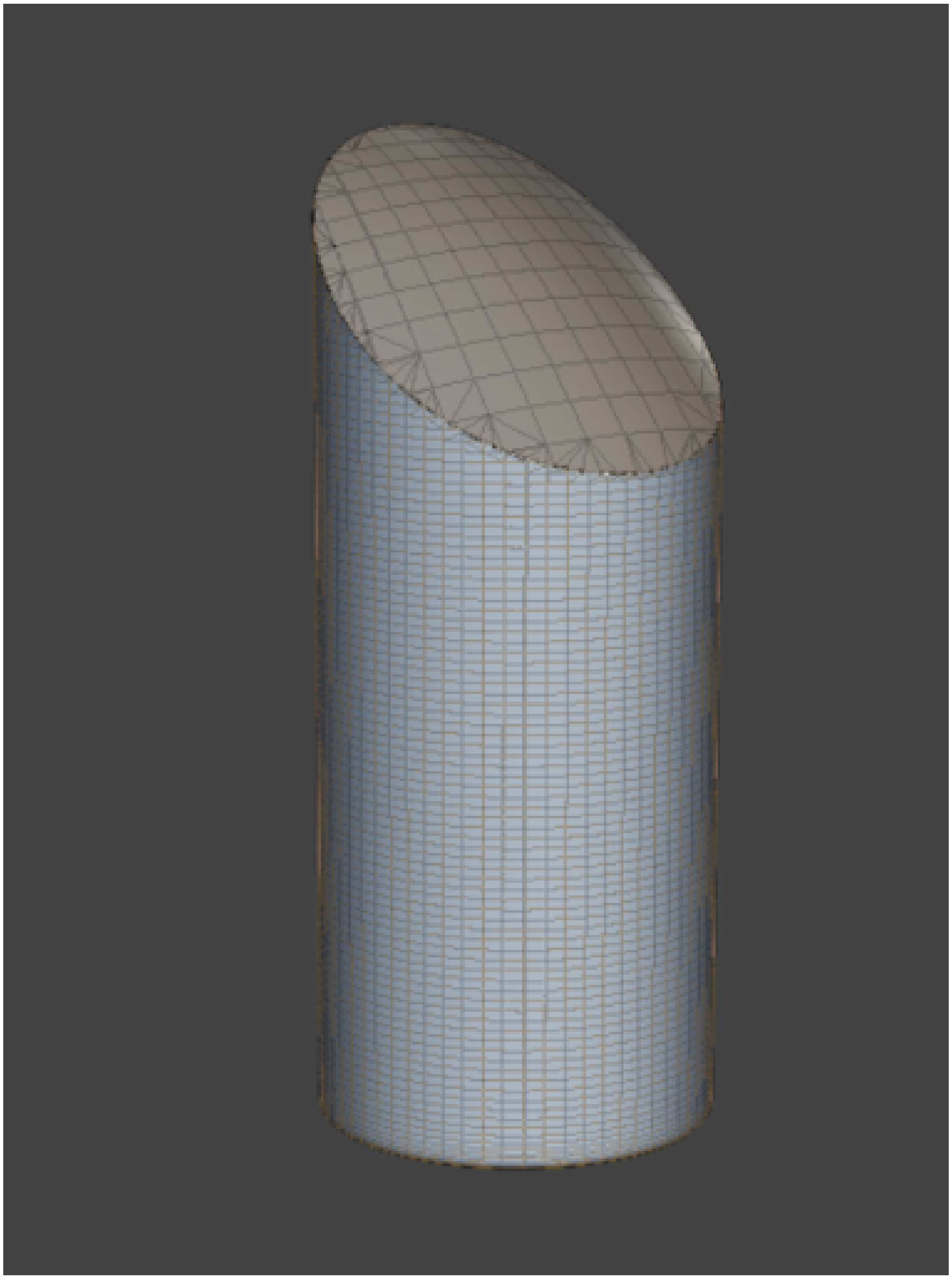
5. Translate and rotate the ellipsoid so that it'll create a diagonal edge at the top of the cylinder. In the image below, the ellipsoid is rotated 28° in the **Z-Axis**, but feel free to play around with rotation, scale and how high up you want your cut to be.



6. Select both the ellipsoid and the Cylinder in the **Items List**.
7. With both meshes selected click the **Apply Intersect Trim** icon, found in the **Set Mesh Roles** submenu of the **Fusion** tab.

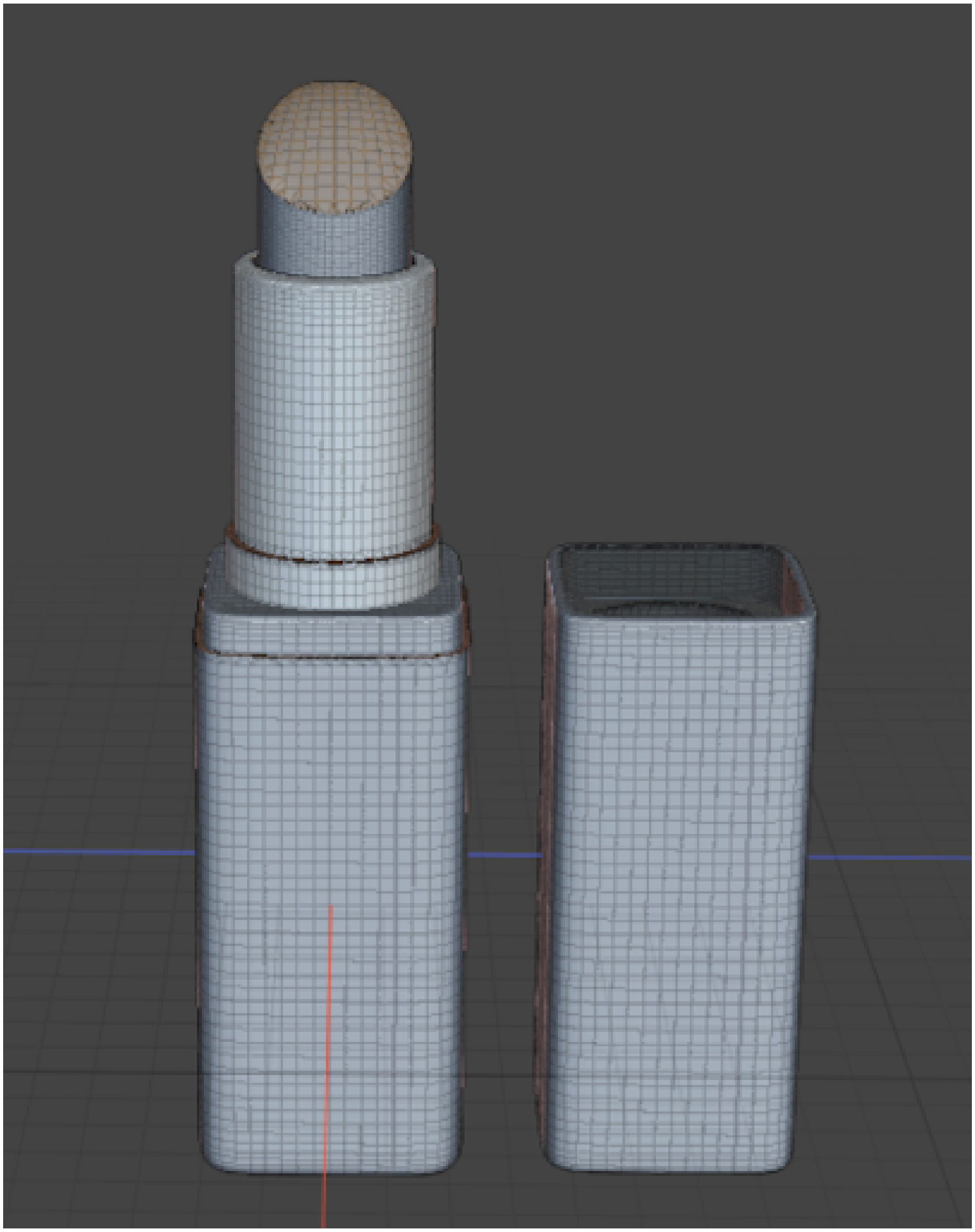


8. Once the trim is applied, you are left with a mesh that should look similar to the one in the image below.



9. Now that all three individual components have been created, you can Enable visibility on all three of your **Fusion Items** and check the model.

   You should have a set of **Fusion Items** that look similar to the image below.

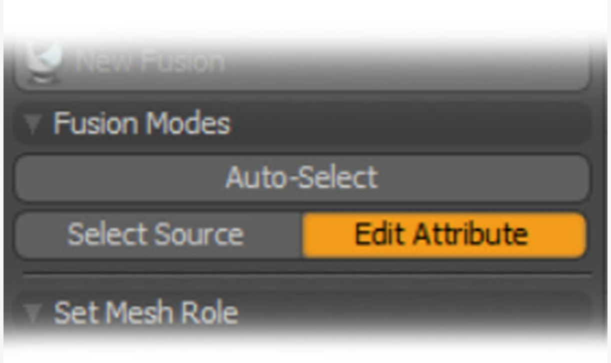


At this point, feel free to change the depths of any embossing, or the heights or sizes of any of the lipstick components. The next two topics will cover how to fine tune a model by editing **Surface Strips**, as well as how to convert a Fusion Item into a mesh.

Fine Tuning: Editing Surface Strips

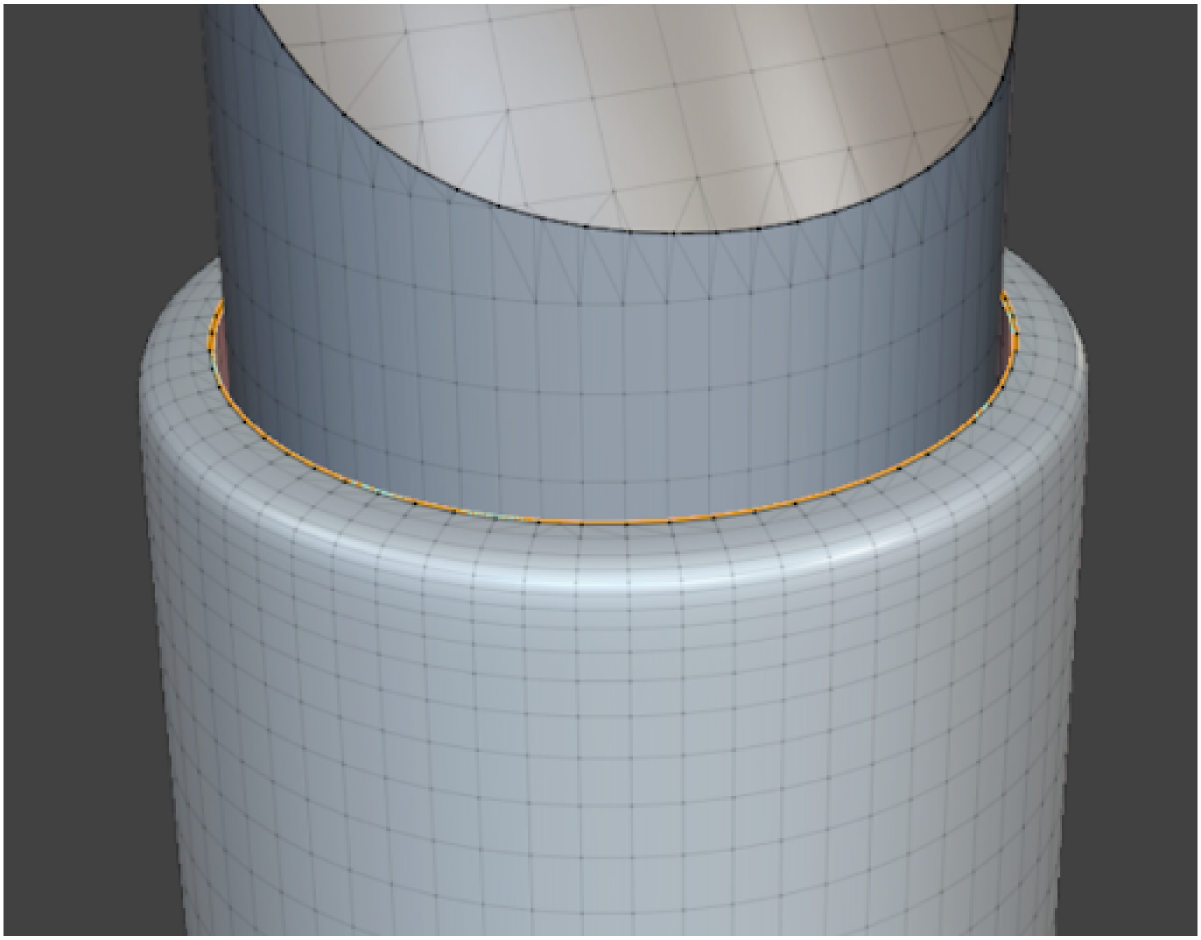
Before converting our Fusion Items to meshes, we can edit our strips to add some smoothness to the edges of our objects. Let's edit the edge of the lipstick tube to see how this is done.

Note: In order to edit **Surface Strips**, **Edit Attributes** must be enabled. To do this, click the **Edit Attributes** button found under the **Fusion Modes** submenu.

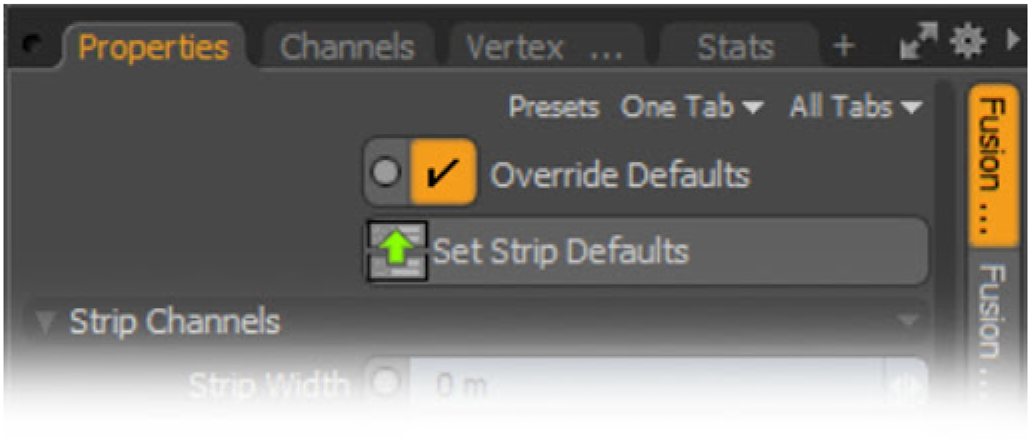


1. Once that **Edit Attribute** has been enabled, it’s time to select our strips. There are a couple ways to do this.
   1. The first way is to select your **Fusion Item** in the **Items List**. Once selected, you can select single strips by zooming into the strip you want to select in the 3D view port and left-clicking it once.
   2. You can also select all strips on your **Fusion Item** by double clicking any strip on the mesh.

   The strip that we can select and edit is the one at the rim of the lipstick tube, highlighted in orange in the image below.



2. Enable **Override Defaults** by navigating to the **Properties** window and click the checkbox next to **Override Defaults**. If this is not enabled, any changes you make to **Fusion Strips** will not update.

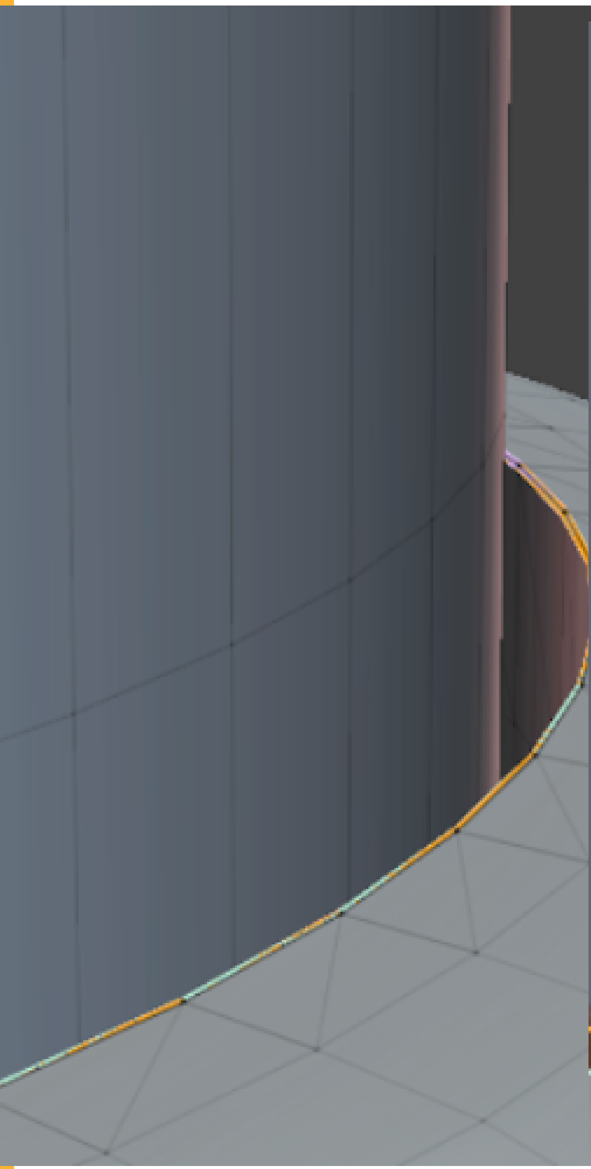


Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Fusion ...
Fusion ...
Set Strip Defaults
Strip Channels

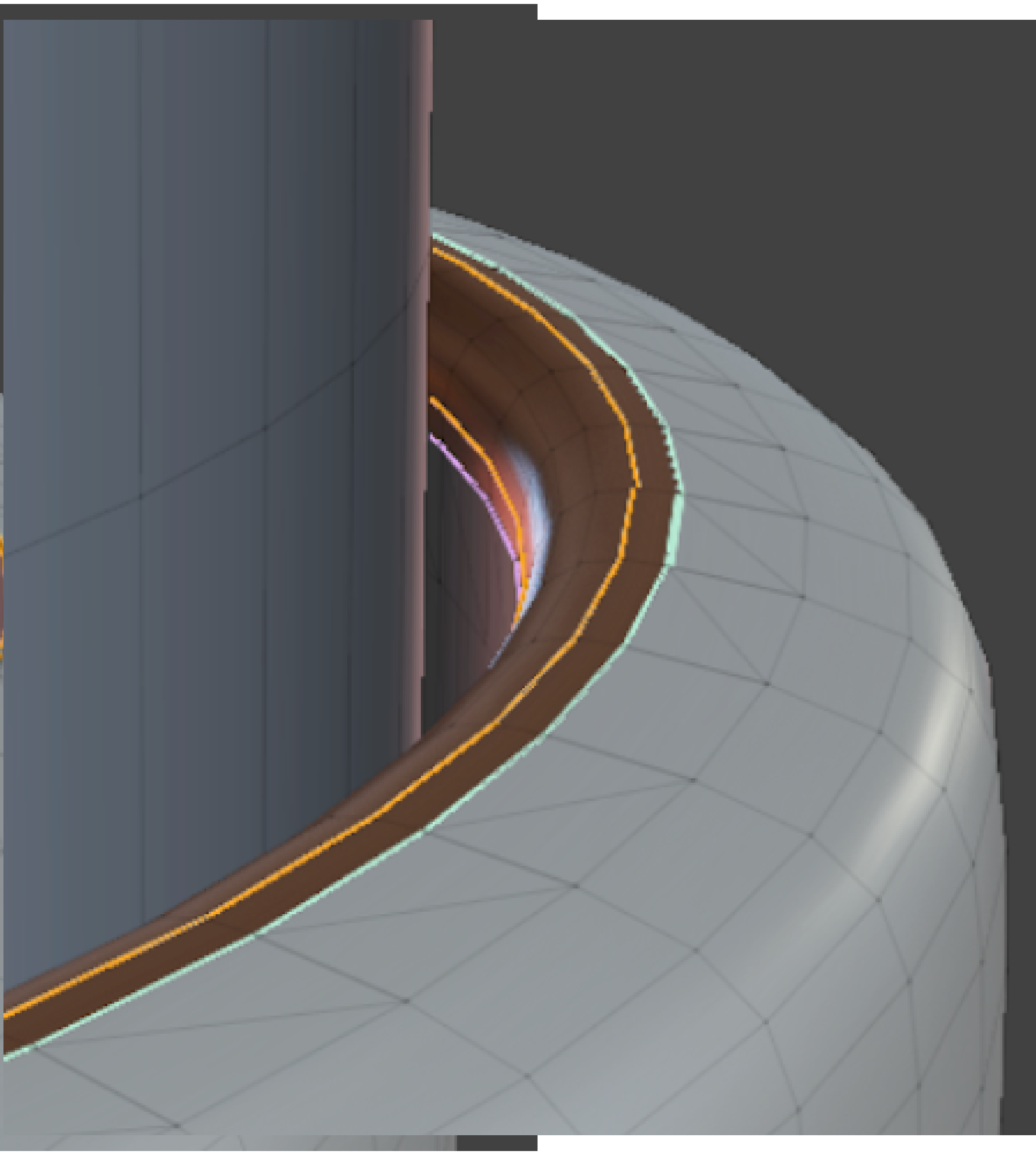
☑ Override Defaults

Strip Width 0 m

3. Now that we have **Override Defaults** enabled, we can edit our strip. To do this, in the **Properties** window, change the **Strip Width** to 35 mm. This widens the strip and has a smoothing effect.



Strip Width set to 0 mm.



Strip Width set to 35 mm.

As you can see in the images above, by increasing the width of the strips we have also increased how curved the edge is.

You can now edit other surfaces strips on your **Fusion Items**. Also try experimenting with the **Strip Profile** setting, which controls the roundness of the strip. A **Strip Profile** of 0% makes a flat bevel, at 100% it is fully rounded.

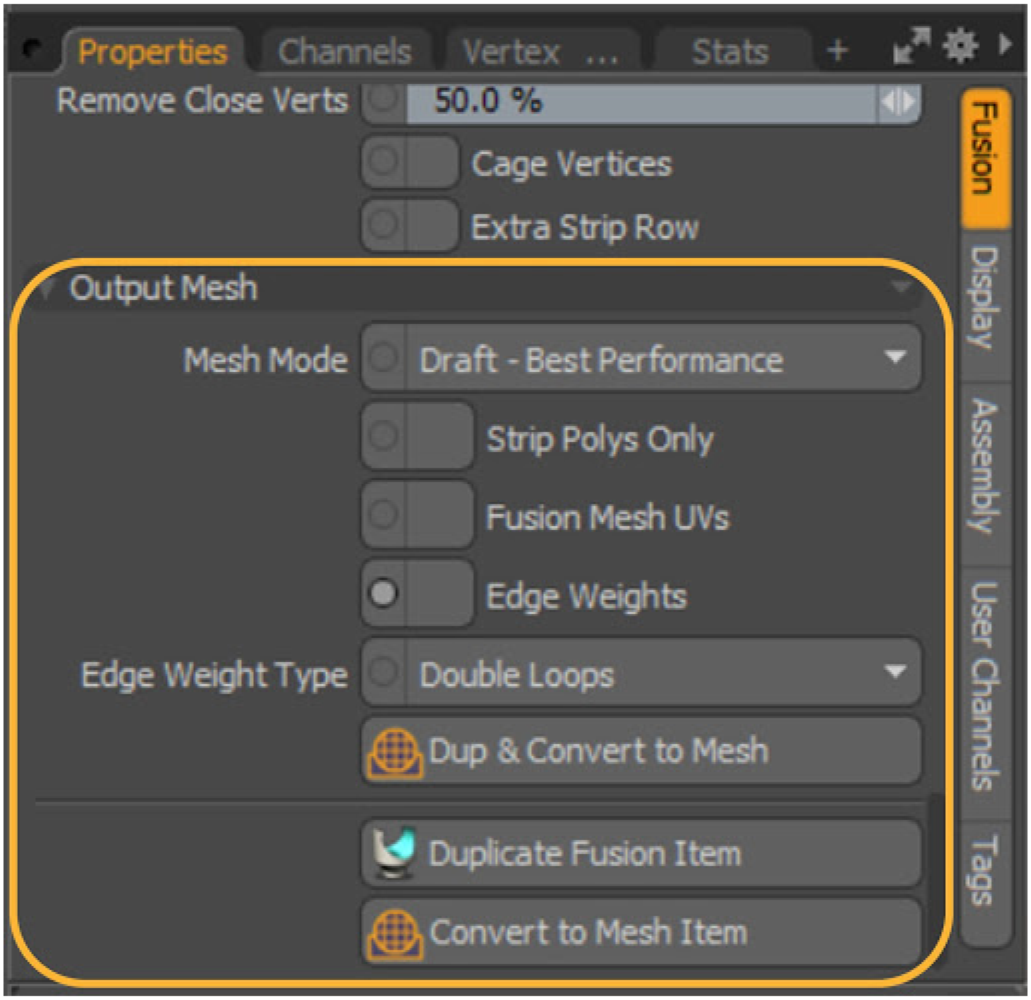
Note: For more information on editing Surface Strips, see Fusion Surface Strips.

Converting to a Mesh

When you've finished creating a Fusion mesh, you can convert it into a regular mesh. This prepares it for 3D printing or for editing the geometry using standard geometry tools. Each **Fusion Item** in the **Item List** can be converted into a mesh using the **Convert to Mesh Item** button on the item's **Properties** panel.

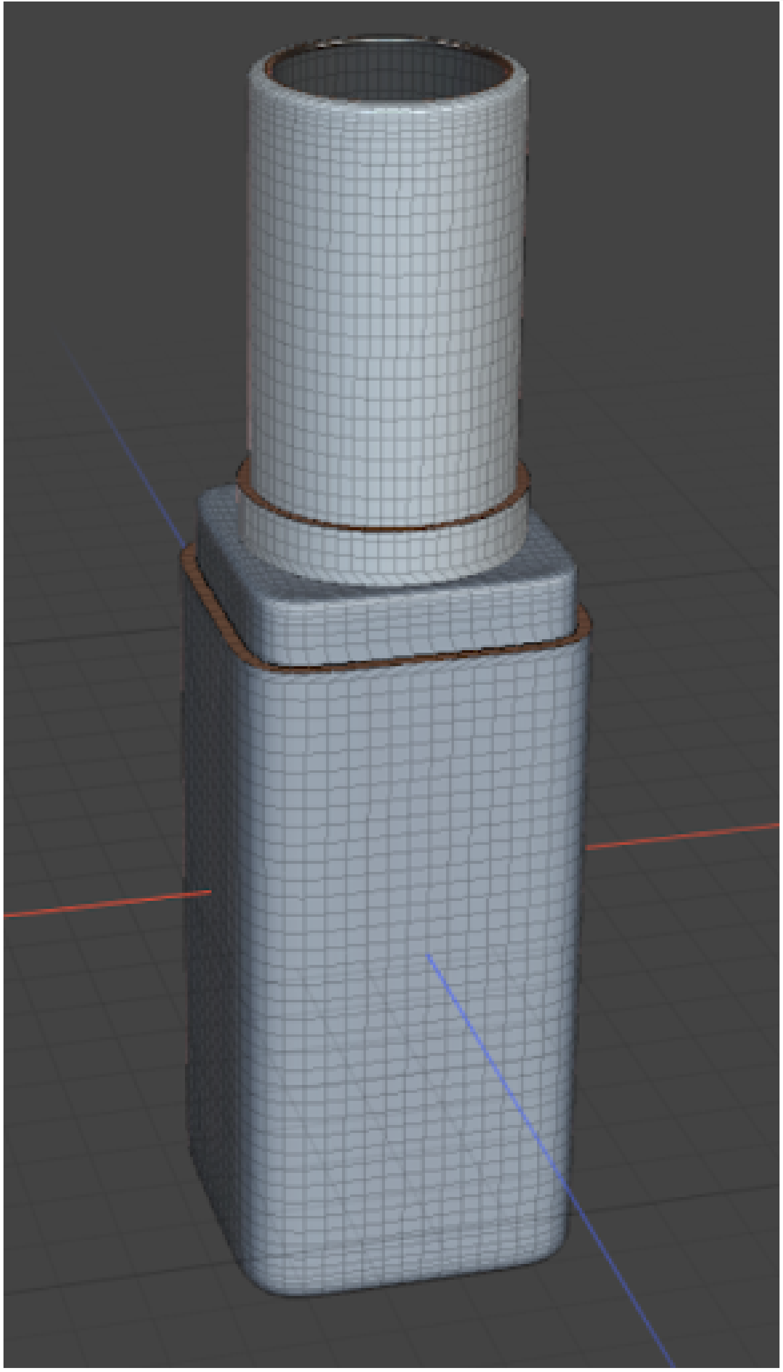
Let's convert the lipstick tube first.

1. Select the **Fusion Item** for the lipstick tube in the **Items List**.
2. In the **Properties** panel ensure the **Fusion** tab is enabled, and scroll down to the **Output Mesh** menu.


Properties
Channels
Vertex ...
Stats
Fusion
Display
Assembly
User Channels
Tags
Output Mesh
Dup & Convert to Mesh
Duplicate Fusion Item
Convert to Mesh Item

Remove Close Verts 50.0 %
○ Cage Vertices
○ Extra Strip Row
Mesh Mode: Draft - Best Performance
○ Strip Polys Only
○ Fusion Mesh UVs
● Edge Weights
Edge Weight Type: Double Loops

3. In the **Output Mesh** menu, change the **Mesh Mode** to **Airtight Final**.



4. Click **Convert to Mesh Item**.
5. Click **OK** to replace the **Fusion Item** with a **Mesh Item**.
6. Click **OK** on the next dialog.

   The **Fusion Item** has now been converted into a **Mesh**.

   You can now convert the bullet and cap to meshes and position both as needed.

Mesh Roles and Relationships

Modeling with MeshFusion is slightly different than modeling with other boolean modeling applications. You can create unions, subtractions, and intersections by using simple commands. When modeling with MeshFusion, there are two things you need to define for each mesh you add to your scene, its role and its relationship to other meshes.

Mesh Roles

Meshes fall into two main categories: Primaries and Trims. Primaries are additive meshes, while Trims are subtractive. You can create a subtraction or an intersection using Trims.

Primaries

Primaries are positive meshes, which provide the base geometry of your scene. You can create unions by combining Primaries or trim them using other meshes to create new shapes.

All Primary meshes are unioned by default.

When you set the role of a mesh to Primary, it is drawn in green wireframe in the 3D viewport, and highlighted with green in the **Item List**.

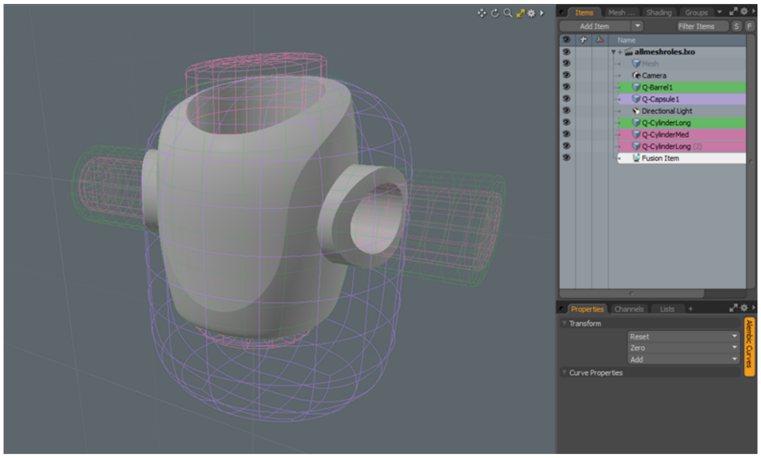
Trims

Trims are meshes you use to create subtractions from Primaries. You can create two types: Subtractive and Intersection Trims.

When you apply a Subtractive Trim to a Primary, the overlapping area of the two meshes is removed. Trims are highlighted with a magenta color in the viewport and the **Item List**.

Intersections are Trims that preserve only the part of the Primary that overlaps with the Trim. When you set the role of a mesh to intersection, it is highlighted with purple in the 3D viewport and the **Item List**.

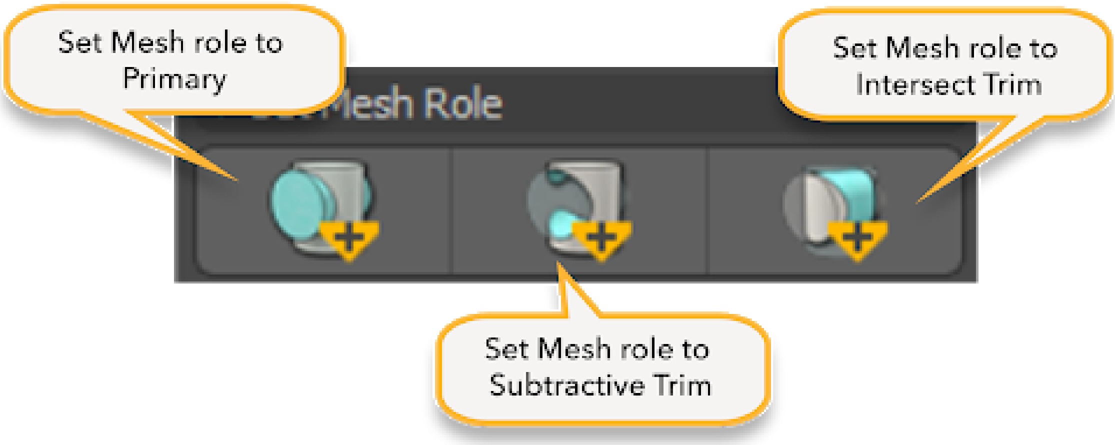
The image below shows how different meshes are colored in the 3D viewport and in the **Item List**. Click the image to enlarge it.


allmeshroles.lxo
Camera
Q-Barrel1
Q-Capsule1
Directional Light
Q-CylinderLong
Q-CylinderMed
Q-CylinderLong (2)
Fusion Item

Assigning Mesh Roles

You can set the mesh roles in the **Fusion** tab.

Under **Set Mesh Role**, press **Alt** and click the appropriate button.


Set Mesh role to Primary
Set Mesh Role
Set Mesh role to Intersect Trim
Set Mesh role to Subtractive Trim

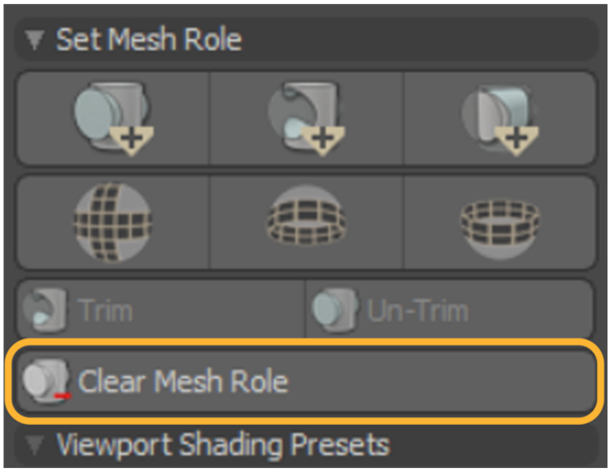
The mesh is color-coded in the 3D viewport and the **Item List**, indicating its new role.

Note: You can adjust the wireframe's opacity in the **Defaults Preferences**, under **Procedural Modeling** > **MeshFusion** > **MeshFusion Wire Opacity**. For more information, see Procedural Modeling.

Changing or Deleting Mesh Roles

You can also change or delete the role of a mesh anytime. Select your mesh and click the button of the role you want to assign.

To delete the role of a mesh, click the **Clear Mesh Role** button:

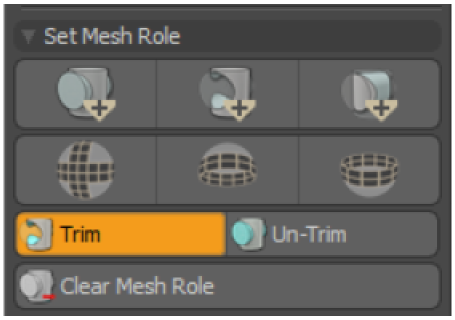

Set Mesh Role
Trim
Un-Trim
Clear Mesh Role
Viewport Shading Presets

Specifying Relationships

Once you've assigned a role to a mesh, you have to define its relationship to other meshes. For instance, when you set a mesh to be a Trim, it doesn't automatically trim any Primary geometry, because you haven't yet specified what to trim.

To set up a mesh's relationship to another mesh:

1. Select the Trim mesh and the Primary mesh in the 3D viewport or the **Item List**.
2. In the **Fusion** sub-tab on the left, under **Set Mesh Roles**, click the **Trim** button.


Set Mesh Role
Trim
Un-Trim
Clear Mesh Role

The Trim is applied to the selected Primary mesh.

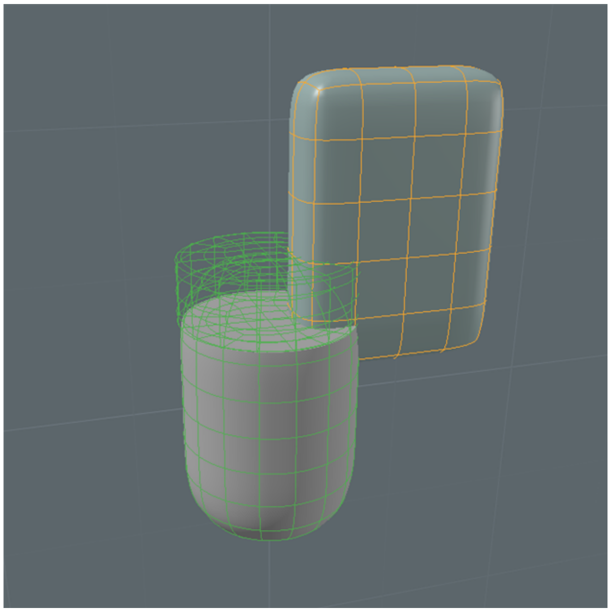
Tip: You can also drag and drop a mesh on a Fusion Item to assign a role to it and trim or untrim the Fusion Item.

Using Compound Trims

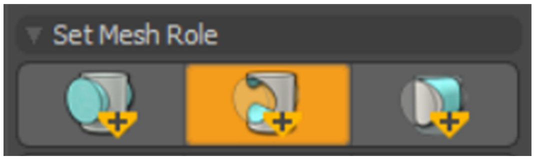
You can create more unique trimming shapes by combining multiple meshes to create Compound Trims. Here's how to do this:

1. Add the new MeshFusion trimming mesh to your scene. The example below uses the Fusion-Ready Presets assets available in the **Fusion** tab of the modeling toolbox, under **Fusion-Ready Presets**.

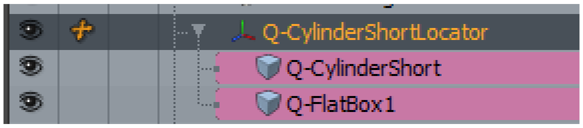
Note: If you need more information on the basics of modeling with MeshFusion, see Fusion Example.



2. Select your Trim mesh and the new mesh you want to combine it with.
3. In the **Fusion** sub-tab of the modeling toolbox, under **Set Mesh Role**, click the button for Subtractive Trims.

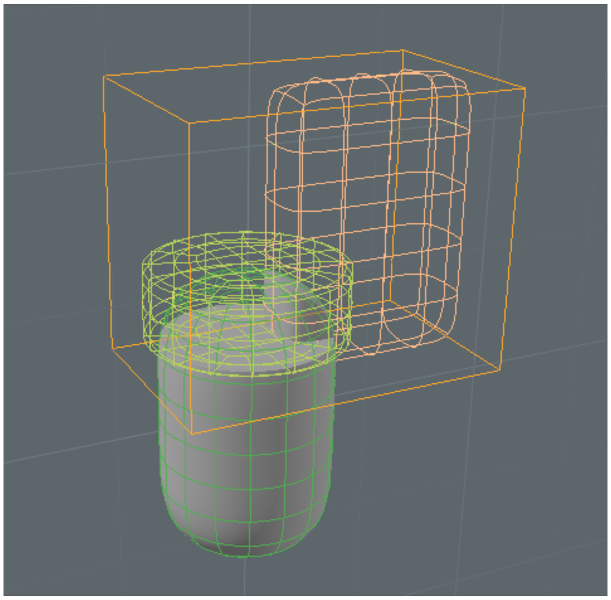


A Locator item is created, containing your trimming meshes. If you expand the Locator item in the **Item List**, you can see the Trims included in the compound.



Q-CylinderShortLocator
Q-CylinderShort
Q-FlatBox1

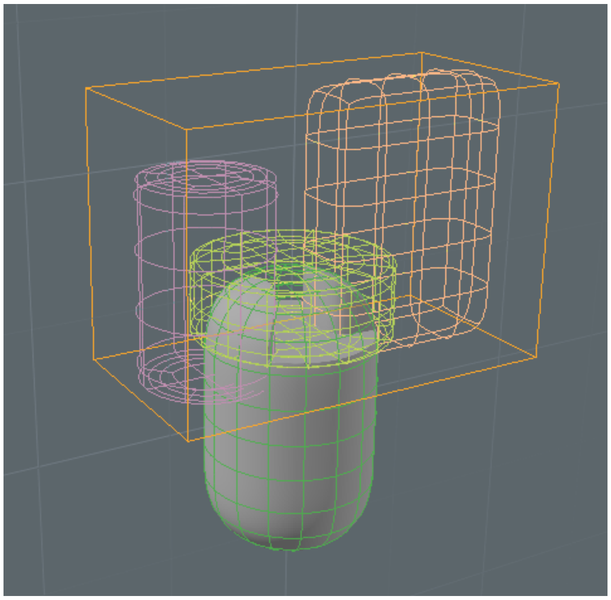
In the 3D viewport, a bounding box is created around your trimming meshes.



The Compound Trim behaves as a single element. You can move, scale, and rotate it as one unit, or transform each of its meshes individually.

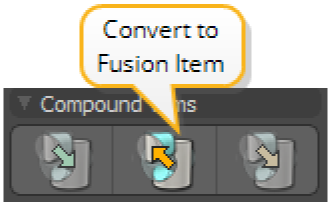
Note: When you select either of the meshes, the whole Locator item is selected. To select just one of the meshes, click the mesh again.

You can further trim the Compound Trim the same way: select the new mesh and the Compound Trim, then in the **Set Mesh Roles** section, click the (Apply Subtractive Trim) button. The new mesh becomes part of the Compound Trim, and you can see the Primary affected by the new Trim.

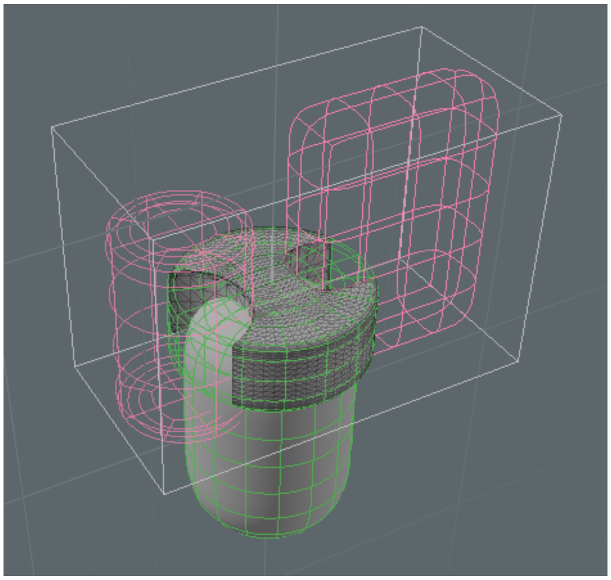


If you want to add additional Primaries to the Compound Trim, you need to convert it into a Fusion Item first. Here's how to do this:

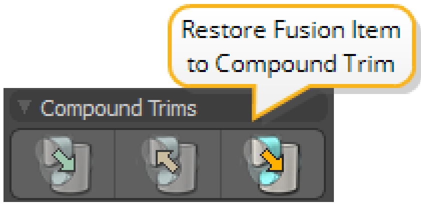
1. Select your Compound Trim in the 3D viewport or in the **Item List**.
2. In the **Compound Boolean** section found under the **Fusion Utilities** menu in the **Fusion** tab, click the (Convert to Fusion) button.



The Trim is transformed into a Fusion Item.

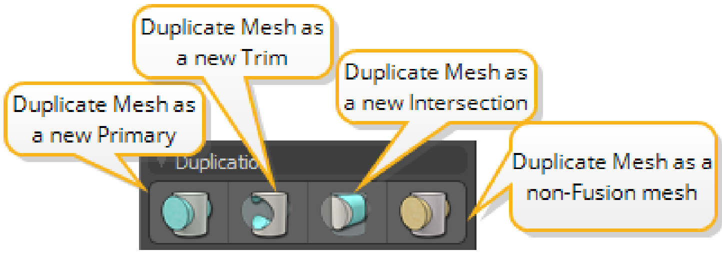


To convert the Fusion Item back to a Trim, click the (Restore Fusion Item to Compound Boolean) button.


Restore Fusion Item to Compound Trim
Compound Trims

Duplicating Meshes

You can duplicate a mesh using the controls in the **Fusion Utilities** menu in **Fusion** sub-tab. Select your mesh and click the appropriate button to duplicate it.

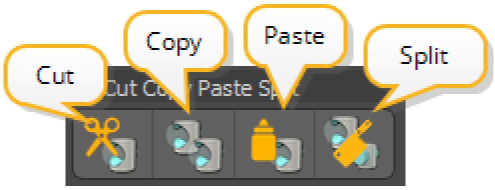

Duplicate Mesh as a new Trim
Duplicate Mesh as a new Primary
Duplicate Mesh as a new Intersection
Duplicatio
Duplicate Mesh as a non-Fusion mesh

Note: When duplicating Trims or Intersections, the duplicate mesh doesn't yet affect any Primaries. For that you need to specify what mesh to trim.

Cutting, Copying, Pasting, and Splitting in MeshFusion

You can cut, copy, and paste elements of a Fusion Item. This can be useful if you want to include parts of one Fusion Item in another. You can also divide a complicated Fusion Item into multiple items, using the Split tool.

You can access the tools by clicking the **Fusion Utilities** button, which can be found in the **Fusion** sub-tab of the Modeling toolbox, which is the left-side panel of the **Model** layout.



Note: Standard cut, copy, and paste tools and their shortcuts don't apply to Fusion Items.

The elements you can edit with these tools are called Fusion Branches. A Fusion Branch is a Primary mesh and any Trims affecting it. For more information on Primaries and Trims, see Mesh Roles and Relationships.

Cutting and Pasting Elements

To cut and paste from a Fusion Item to a Fusion Branch:

1. Select the Primary you want to cut.
2. Click the Cut button in the **Fusion** sub-tab.

   The selected Primary and its associated Trim is removed from the source Fusion Item.

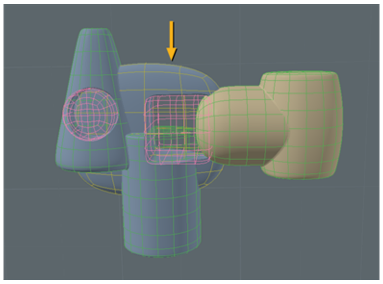
Note: The Trim affecting the Primary also remains in the source item.

3. Select the destination Fusion Item either in the **Item List**, or by selecting any of its Primary meshes in the 3D viewport.

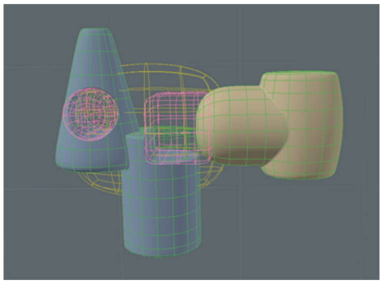
4. Click the Paste button.

   The selected Primary is attached to the destination Fusion Item.

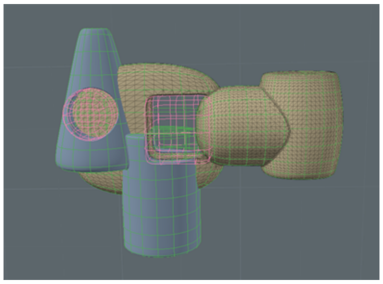
In the following example, an element is cut and pasted from the blue Fusion Item to the yellow Fusion Item.



The wireframe of the selected Primary is highlighted in yellow.



The Primary is cut.



The Primary is copied onto the destination Fusion Item.

Copying and Pasting Elements

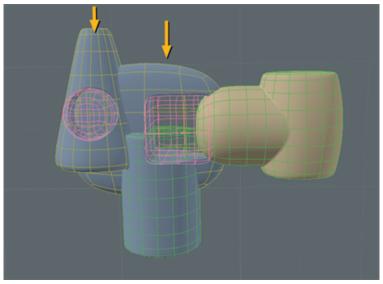
Copying elements works similarly to cutting and pasting.

1. Select the Primary you want to copy.
2. Click the Copy button in the **Fusion Utilities** menu in the **Fusion** sub-tab.
3. Select the destination Fusion Item either in the **Item List**, or by selecting any of its Primary meshes in the 3D viewport.
4. Click the Paste button.

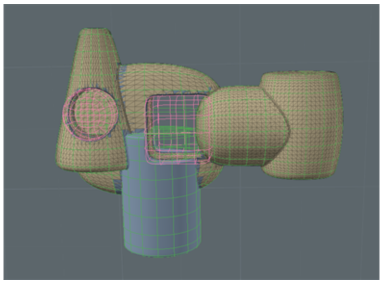
   A duplicate of the selected Primary is attached to the destination Fusion Item. The copied elements overlap with the source elements. You can reposition them using the standard Transform tools.

Note: For more information on the standard Transform tools, see Transforming Items.

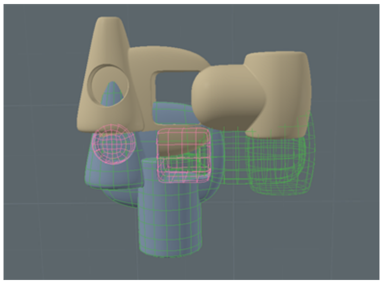
In the following example, two elements are copied and pasted from the blue source Fusion Item to the yellow destination Fusion Item.



The wireframe of the selected Primary is highlighted in yellow.



The Primaries are pasted onto the destination Fusion Item.



The destination Fusion Item is repositioned for clearer visibility.

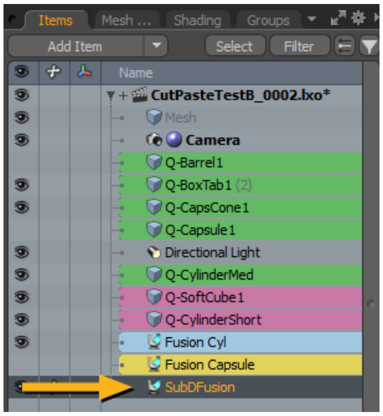
Splitting Elements

You can divide Fusion Items into multiple elements. This allows you to simplify complex Fusion Items to improve performance, or to make the modeling process more flexible.

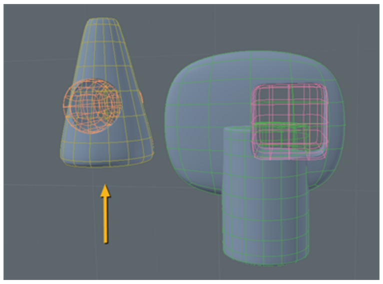
To split a Fusion Item:

1. Select the Primary you want to detach from the item.
2. Click the Split button.

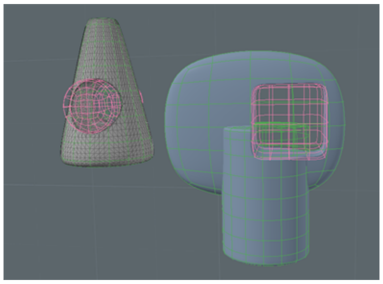
   The selected Primary is separated from the original Fusion Item, and a new **SubDFusion** item is created in the **Item List**.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
CutPasteTestB_0002.lxo*
Mesh
Camera
Q-Barrel1
Q-BoxTab1 (2)
Q-CapsCone1
Q-Capsule1
Directional Light
Q-CylinderMed
Q-SoftCube1
Q-CylinderShort
Fusion Cyl
Fusion Capsule
SubDFusion

In the following example, the cone is split from the Fusion Item.



The wireframe of the selected Primary is highlighted in yellow.

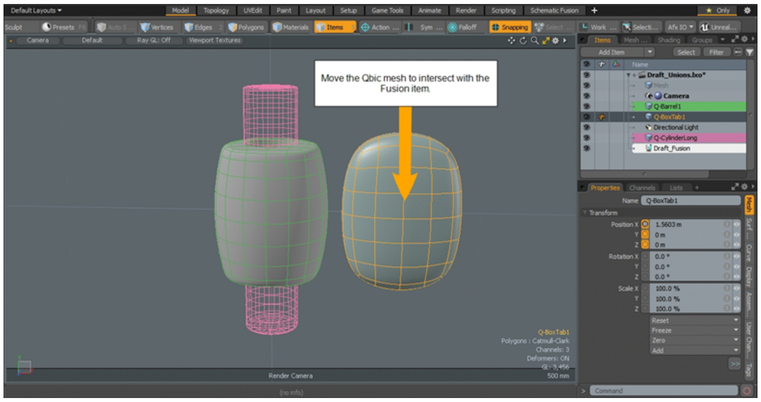


The cone is detached from the Fusion Item and it no longer inherits its material properties.

Drag and Drop Workflow

Use the drag-and-drop contextual menus to quickly building Fusion models. These menus are available in the 3D Viewport when you have a Fusion Item in your scene and other Fusion Qbic Presets.

Click on the image below to view an animation.



Move the Qbic mesh to intersect with the Fusion item.

Name Q-BoxTab1

Position X 1.5603 m

Y 0 m

Z 0 m

Rotation X 0.0 °

Y 0.0 °

Z 0.0 °

Scale X 100.0 %

Y 100.0 %

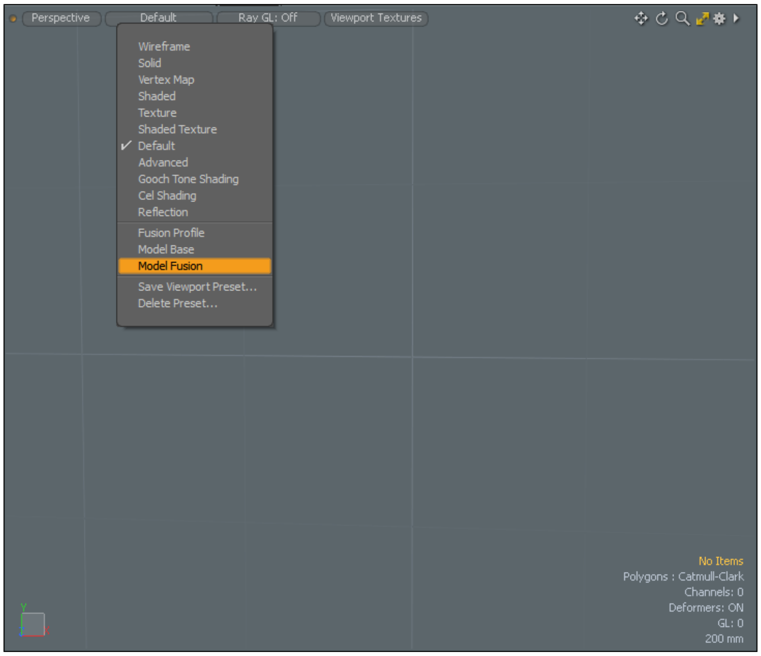
Z 100.0 %

Default Layouts
Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
Schematic Fusion
Only
Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Draft_Unions.lxo*
Mesh
Camera
Q-Barrel1
Q-BoxTab1
Directional Light
Q-CylinderLong
Draft_Fusion
Properties
Channels
Lists
Transform
Reset
Freeze
Zero
Add
Render Camera

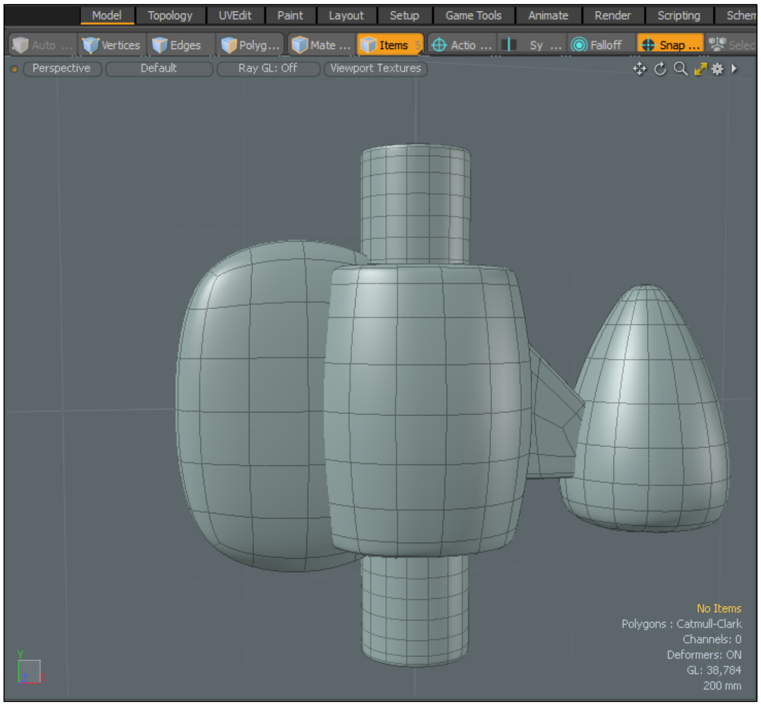
The following is an example of using the contextual Fusion menu items.

To use the contextual menus:

1. Open the **Model** layout, click the **Default** button on the top of the 3D Viewport, and select **Model Fusion**.

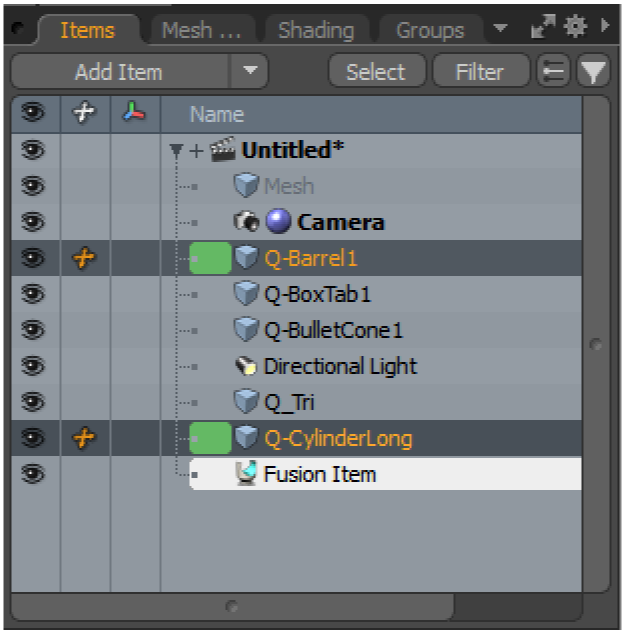

Perspective
Default
Ray GL: Off
Viewport Textures
Wireframe
Solid
Vertex Map
Shaded
Texture
Shaded Texture
✓ Default
Advanced
Gooch Tone Shading
Cel Shading
Reflection
Fusion Profile
Model Base
Model Fusion
Save Viewport Preset...
Delete Preset...
No Items
Polygons : Catmull-Clark
Channels: 0
Deformers: ON
GL: 0
200 mm

2. In the Toolbox **Fusion** tab, click **Qbic Presets**.
3. Double-click on five Qbic Presets to add them to your scene. In this example we are using **Q-Barrel1**, **Q-BoxTab1**, **QBulletCone1**, **Q-tri**, and **Q-CylinderLong**.
4. In the **Items** tab, on the left panel, select each Qbic preset, press **W**, and move them into place.


Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
Schem
Auto ...
Vertices
Edges
Polyg...
Mate ...
Items
Actio ...
Sy ...
Falloff
Snap ...
Selec
Perspective
Default
Ray GL: Off
Viewport Textures
No Items
Polygons : Catmull-Clark
Channels: 0
Deformers: ON
GL: 38,784
200 mm

5. Select **Q-Barrel1** and **Q-CylinderLong**and then in the Toolbox Fusion tab, click **New Fusion** > **New Fusion with Selected Meshes**.

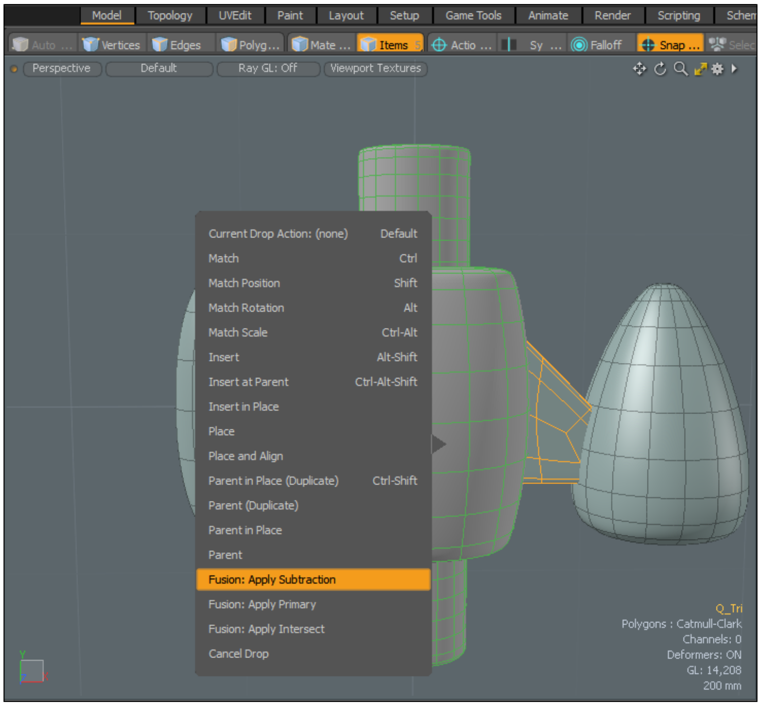
   A Fusion Item is added to you scene.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Untitled*
Mesh
Camera
Q-Barrel1
Q-BoxTab1
Q-BulletCone1
Directional Light
Q_Tri
Q-CylinderLong
Fusion Item

6. In the 3D viewport, select **Q_Tri** and drag-and-drop that item onto the **Q-Barrel1** Item.

   The contextual menu displays.

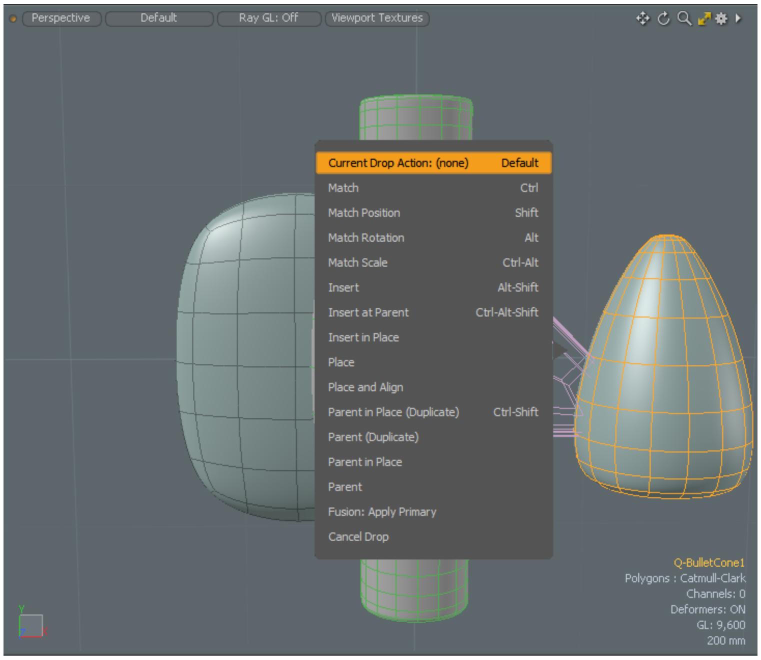
7. Select **Fusion: Apply Subtraction**.


Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
Schem
Auto ...
Vertices
Edges
Polyg...
Mate ...
Items
Actio ...
Sy ...
Falloff
Snap ...
Selec
Perspective
Default
Ray GL: Off
Viewport Textures
Current Drop Action: (none)	Default
Match	Ctrl
Match Position	Shift
Match Rotation	Alt
Match Scale	Ctrl-Alt
Insert	Alt-Shift
Insert at Parent	Ctrl-Alt-Shift
Insert in Place
Place
Place and Align
Parent in Place (Duplicate)	Ctrl-Shift
Parent (Duplicate)
Parent in Place
Parent
Fusion: Apply Subtraction
Fusion: Apply Primary
Fusion: Apply Intersect
Cancel Drop
Q_Tri
Polygons : Catmull-Clark
Channels: 0
Deformers: ON
GL: 14,208
200 mm

8. Select the **Q-BulletCone1** item on the left and drag-and-drop it onto the **Q-Barrel1** item.

   A contextual menu displays.

9. Click **Fusion: Apply Primary**.


Perspective
Default
Ray GL: Off
Viewport Textures
Current Drop Action: (none) Default
Match Ctrl
Match Position Shift
Match Rotation Alt
Match Scale Ctrl-Alt
Insert Alt-Shift
Insert at Parent Ctrl-Alt-Shift
Insert in Place
Place
Place and Align
Parent in Place (Duplicate) Ctrl-Shift
Parent (Duplicate)
Parent in Place
Parent
Fusion: Apply Primary
Cancel Drop
Q-BulletCone1
Polygons : Catmull-Clark
Channels: 0
Deformers: ON
GL: 9,600
200 mm

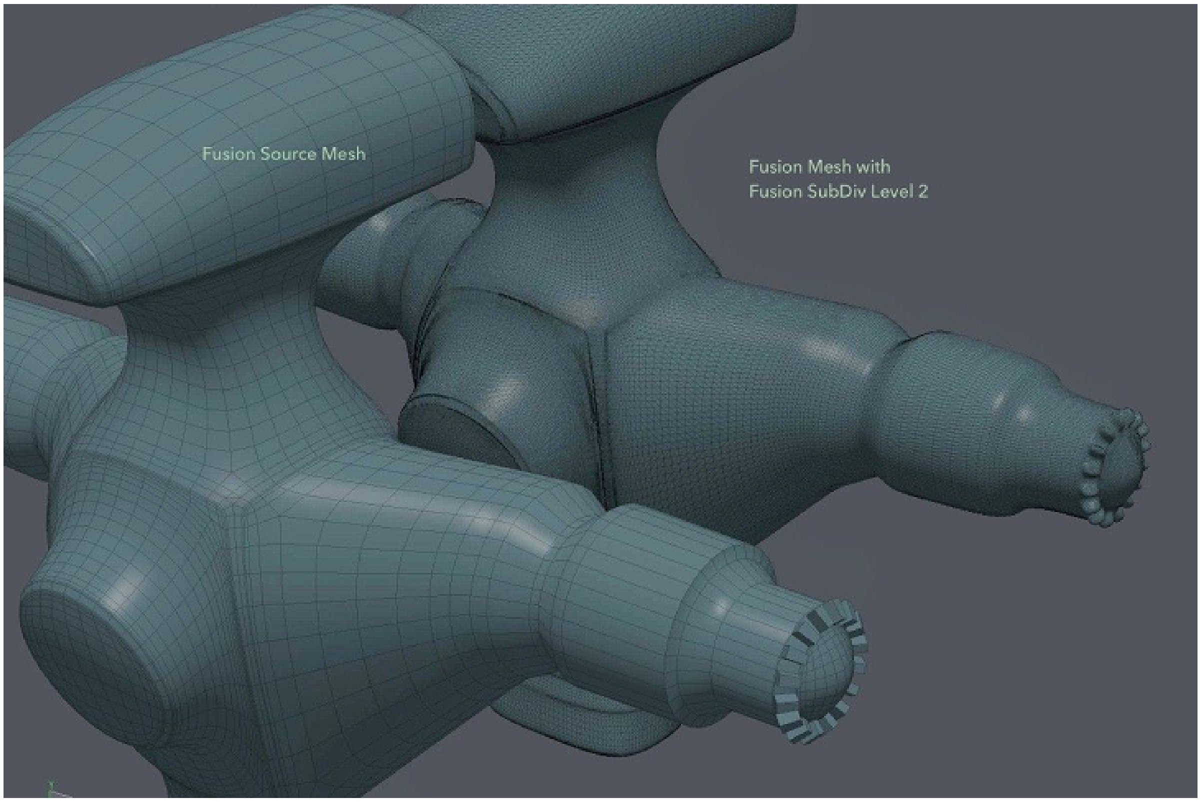
10. Select the **Q-Tri** item and drag-and-drop it on the **Q-BulletCone1** item.

    A different contextual menu displays.

11. Click **Fusion: Trim**.

Using MeshFusion with Zero Subdivision

MeshFusion converts all of its source meshes to Catmull-Clark geometry, which is then converted to a polygonal mesh using the Fusion source mesh's **Fusion SubDivs** property as the subdivision level. Generally, MeshFusion uses one or two levels of subdivision. This is required by certain topologies, such as triangles, n-gons, and vertices with three edges converging on them, that MeshFusion can't handle without subdivision. If your source mesh was already quite dense to start with, this can cause problems.


Fusion Source Mesh
Fusion Mesh with
Fusion SubDiv Level 2

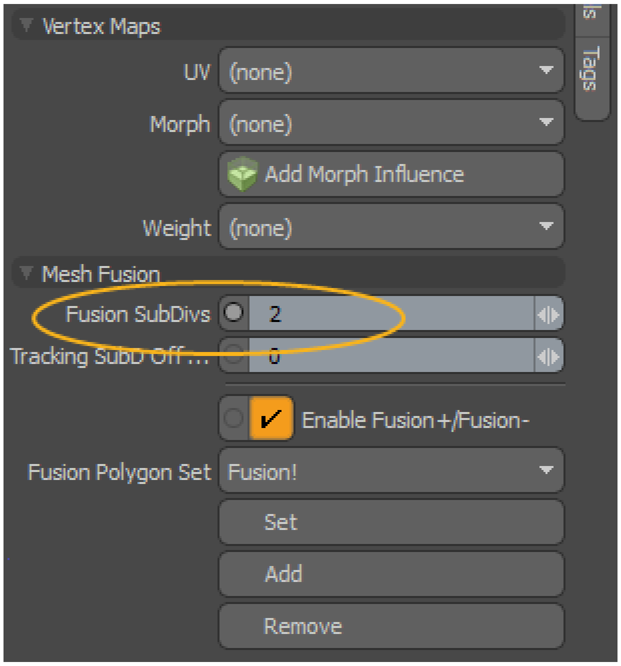
In such cases you can set the subdivision level to 0. You can do this when creating your Fusion meshes, in the **New Fusion** dialog. For more information, see Creating a New Fusion Item.

If you already have a fusion mesh, you can modify the subdivision level in its properties. To do this:

1. Select your Fusion source mesh in the 3D viewport or the **Items** list.

   This opens its **Properties** on the lower right panel.

2. Under **Mesh Fusion**, set **Fusion SubDivs** to 0.



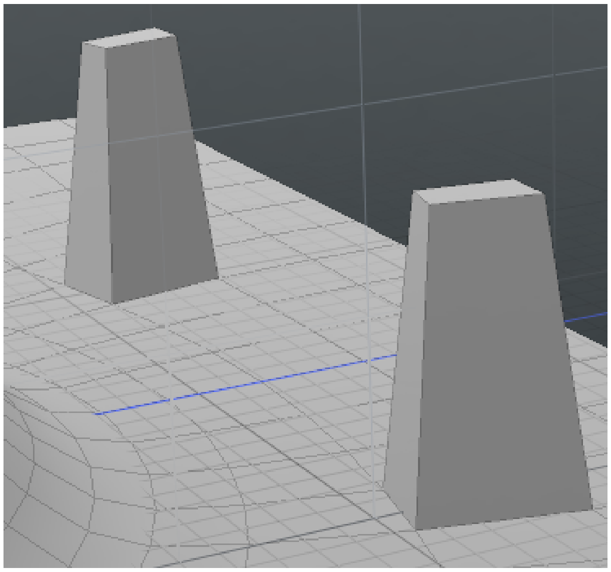
Vertex Maps
UV (none)
Morph (none)
Weight (none)
Mesh Fusion
Fusion SubDivs 2
Tracking SubD Off… 0
☑ Enable Fusion+/Fusion-
Fusion Polygon Set Fusion!

Troubleshooting

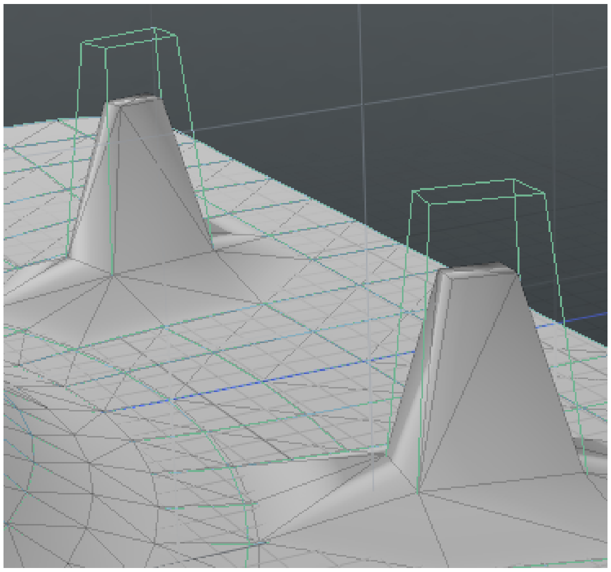
Using 0 level of subdivision can lead to some issues with the geometry and shading. This section walks you through how you can resolve them with the existing tools in Modo.

Geometry

As a result of converting the mesh to Catmull-Clark, the nature of the geometry changes, even with **Fusion SubDivs** set to 0. The images below show you an example of this problem:



Original Mesh



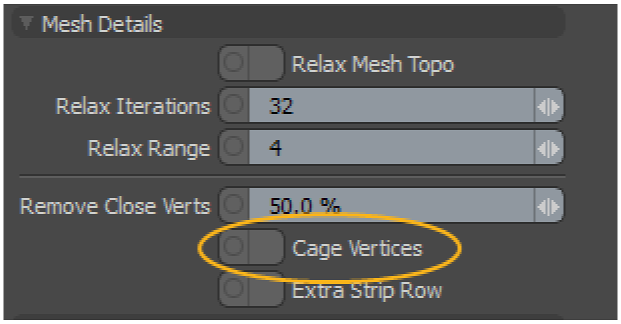
Fusion Item with 0 Subdivision

You can fix this by enabling the **Cage Vertices** option in the Fusion Item's properties:

1. Select your **Fusion Item** in the **Items** list.

   This opens its **Properties** on the lower right panel.

2. Under **Mesh Detail**, enable **Cage Vertices**.



Mesh Details

○ Relax Mesh Topo

Relax Iterations 32

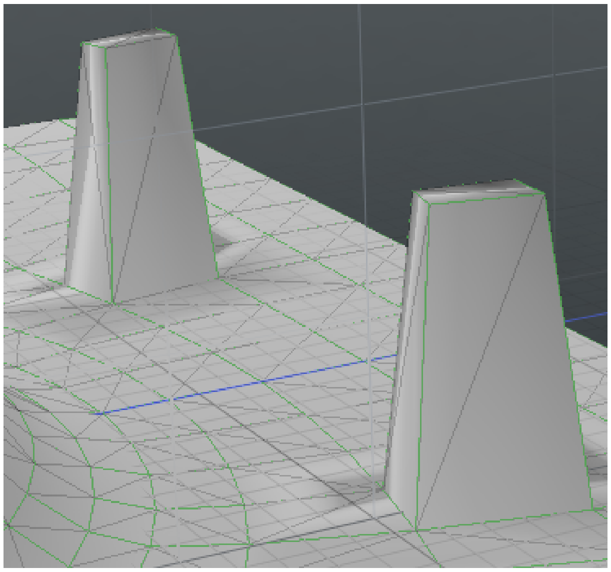
Relax Range 4

Remove Close Verts 50.0 %

○ Cage Vertices

○ Extra Strip Row

This uses the cage vertices of the source meshes, making the geometry structurally identical to the original:



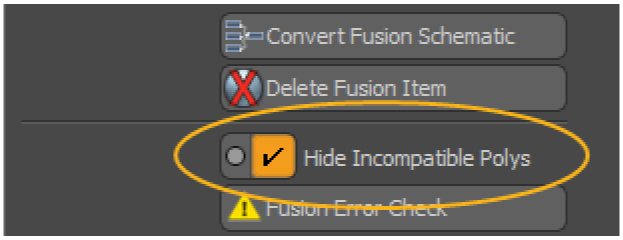
Incompatible Topology

Topology that triggers MeshFusion to use a subdivision level of 2 remains incompatible with MeshFusion at zero level subdivision. This means that you can't have seams and the strips they generate pass through them.

You can identify problematic areas by hiding any incompatible polygons:

1. Select your **Fusion Item** in the **Items** list.

   This opens its **Properties** on the lower right panel.
2. Under **Modeling Utilities**, enable **Hide Incompatible Polys**.


Convert Fusion Schematic
Delete Fusion Item
Hide Incompatible Polys
Fusion Error Check

This hides incompatible polygons.

Note: This option hides the affected geometry from the output too, so make sure you only use this method for diagnostic purposes.

N-gons

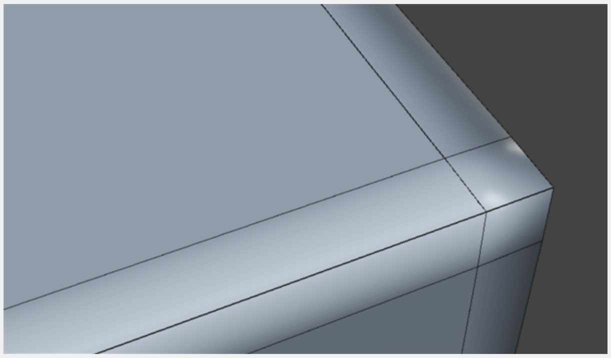
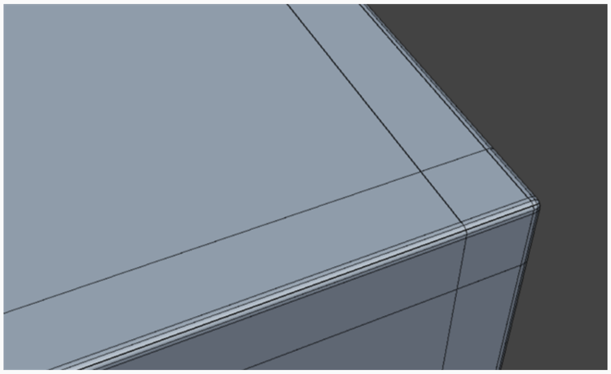
MeshFusion can't handle n-gons automatically. To deal with them, you can utilize the **Quadruple** Polygon tool to divide them. For detailed information on these tools, see Polygon Tools.

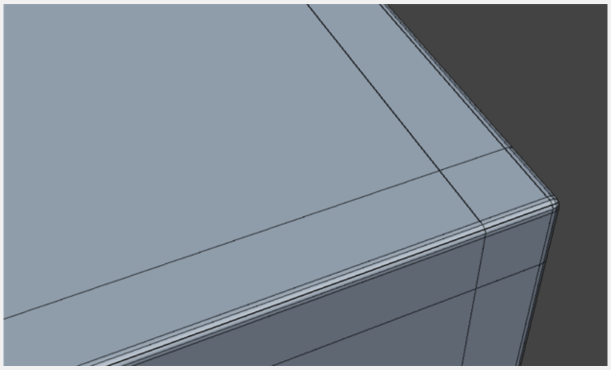
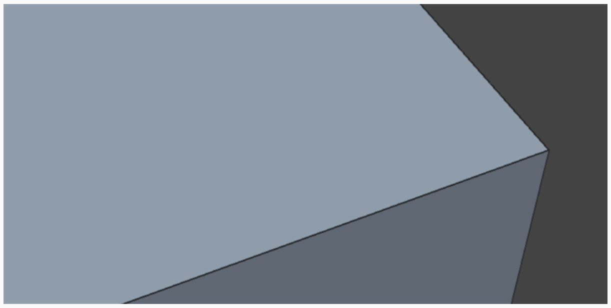
If your n-gon is non-planar, the safest option is to apply Triple to it.

Edge Weighting

Edge Weighting is a useful tool for creating highly detailed, zero-subdivision meshes with low poly counts. Keeping a low poly count is useful for instances where your model is placed in a real time environment, a game asset or used for practical applications such as 3D printing.

There are four different **Edge Weighting** types in MeshFusion

Single Loops	Single loops add one loop on either side of each weighted edge. 
Double Loops	Double loops add two loops on either side of each weighted edge.  Double Loops are useful for keeping edge integrity and clean topology while working

	with zero SubDivisions.
Adaptive Loops	Adaptive Loop automatically adds a single or double loop based on the topology of the mesh. 
Accurate Edge Weighting	Displays the actual edge weighting of a mesh based on its subdivision level. This means that no additional edge loops are added to create edge integrity. 

Note: For information on how to apply weighted edges to a mesh, see Edge Weight Tool

Enabling Edge Weighting

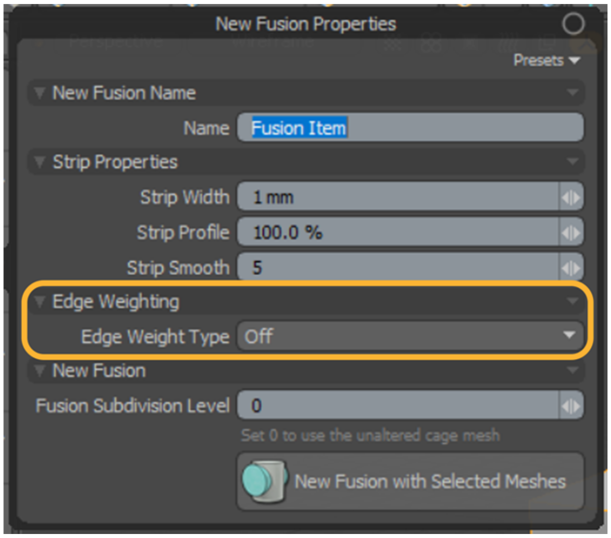
MeshFusion does not apply edge weighting by default, but it can be applied in one of two ways.

The first way is to enable it from the **New Fusion Properties window**.

1. Select your mesh and go to the **Fusion** tab in the window to the left of the screen.
2. Press and hold the **Alt** key.

   The **New Fusion** button in the Fusion tab changes to **New Fusion Properties**.
3. Click **New Fusion Properties** to display its controls.

   In the **New Fusion** properties window, locate the **Edge Weighting** section.

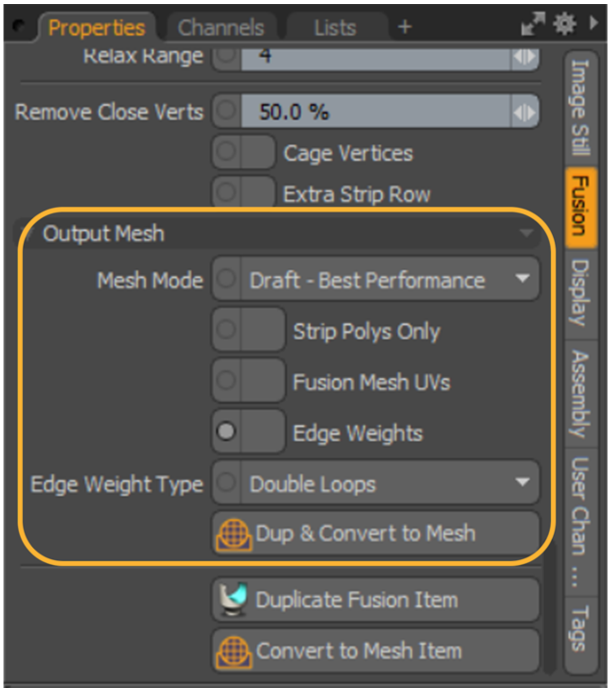

New Fusion Properties
Presets
New Fusion Name
Strip Properties
Edge Weighting
New Fusion
New Fusion with Selected Meshes

Name: Fusion Item
Strip Width: 1 mm
Strip Profile: 100.0 %
Strip Smooth: 5
Edge Weight Type: Off
Fusion Subdivision Level: 0
Set 0 to use the unaltered cage mesh

From here, you can enable **Edge Weighting** on your **Fusion Item** and assign which type of weighting is suitable for your model.

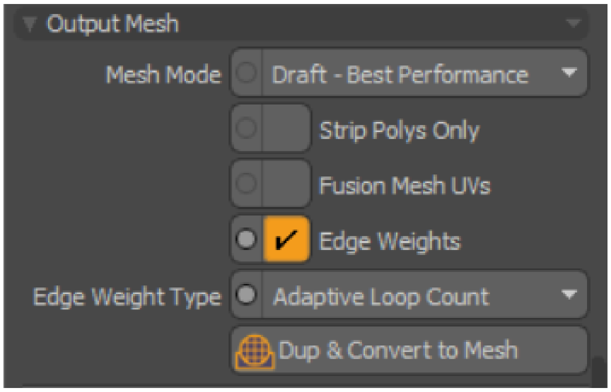
The second way to enable edge weighting, is to enable **Edge Weights** and then assign an **Edge Weight Type** to your Fusion Item in the Fusion tab of the **Properties** window. This option is useful for applying edge weighting to pre-existing Fusion items.

1. Select your **Fusion Item** in the **Items List**.
2. Navigate to the **Properties** window below the **Items List**, and scroll down in the **Fusion** tab to the **Output Mesh** section.


Properties
Channels
Lists
+
Image Still
Fusion
Display
Assembly
User Chan ...
Tags
Output Mesh
Dup & Convert to Mesh
Duplicate Fusion Item
Convert to Mesh Item

Relax Range: 4
Remove Close Verts: 50.0 %
☐ Cage Vertices
☐ Extra Strip Row
Mesh Mode: Draft - Best Performance
☐ Strip Polys Only
☐ Fusion Mesh UVs
☑ Edge Weights
Edge Weight Type: Double Loops

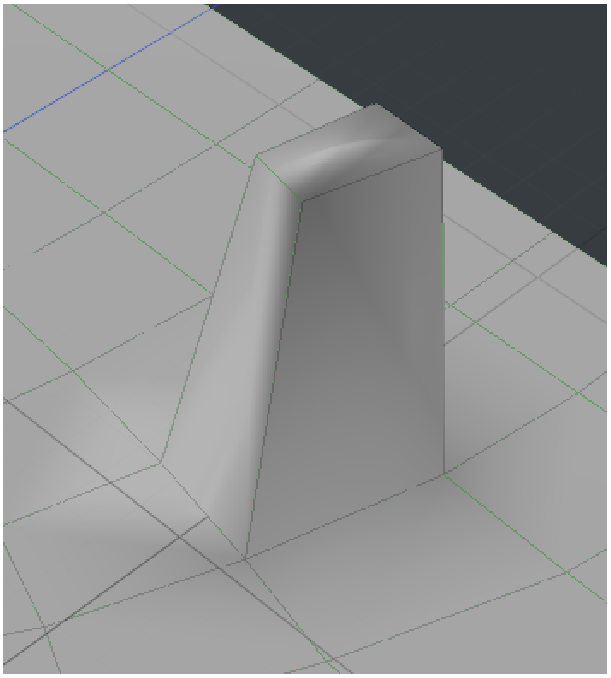
3. Click the checkbox next to **Edge Weights** so that it turns orange. This enables edge weighting.


Output Mesh
Mesh Mode
Draft - Best Performance
Strip Polys Only
Fusion Mesh UVs
Edge Weights
Edge Weight Type
Adaptive Loop Count
Dup & Convert to Mesh

Once edge weighting is enabled, your mesh automatically updates to reflect that. From here, you can change the **Edge Weight** Type to whichever suits you and your model best.

Shading Issues

MeshFusion generates a normal map, replacing the usual shading. This can cause sharp edges to be shaded incorrectly, as shown on the image below:



You can deal with this issue by turning the normal map off for problematic areas, after converting your Fusion model into a regular mesh.

Note: See Fusion Mesh Conversion on how to convert your Fusion item.

To delete the normal map from a specific area:

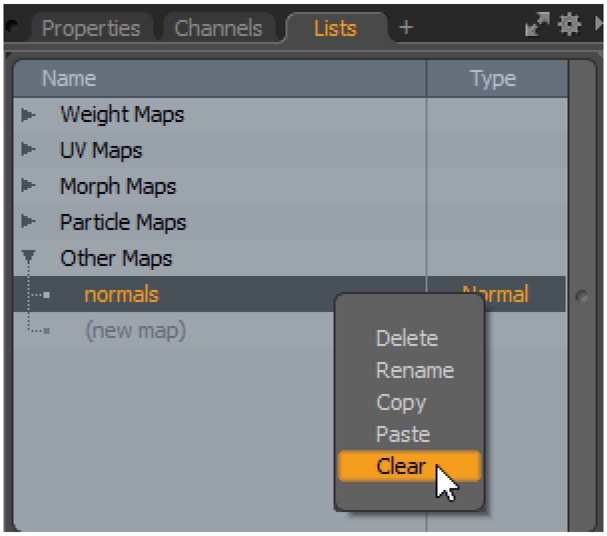
1. In Polygons selection mode, select the polygons of the problematic area.

Tip: It's best to have the boundary of the normal map and the cleared area in a place that doesn't contain much curvature.

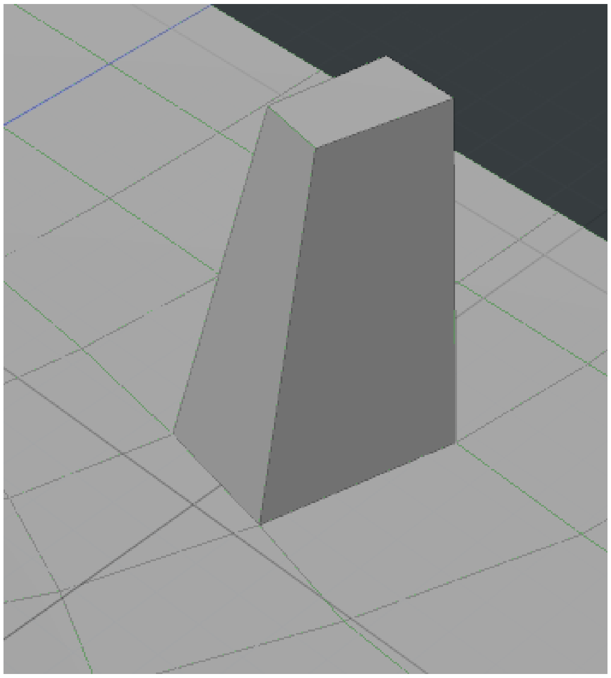
2. In the **Lists** or **Vertex Map List** viewport on the right panel, expand **Other Maps**.

   normals is the normal map generated by MeshFusion.

3. Right-click **normals** and click **Clear**.


Properties
Channels
Lists
+
Name
Type
Weight Maps
UV Maps
Morph Maps
Particle Maps
Other Maps
normals
Normal
(new map)
Delete
Rename
Copy
Paste
Clear

This clears the normal map from the selected area and it defaults to regular shading:



Fusing by Selection Set

Fusion selection sets allow you to add arbitrary amounts of complex detail to your fusion model without using unnecessary Fusion operations. It is an essential part of the Kit Fusing workflow described farther below.

For any Fusion source mesh, you can specify which of its polygons should be seen by MeshFusion for the purposes of generating intersections, trims, and strips.

MeshFusion uses the following three polygon selection sets in the source mesh:

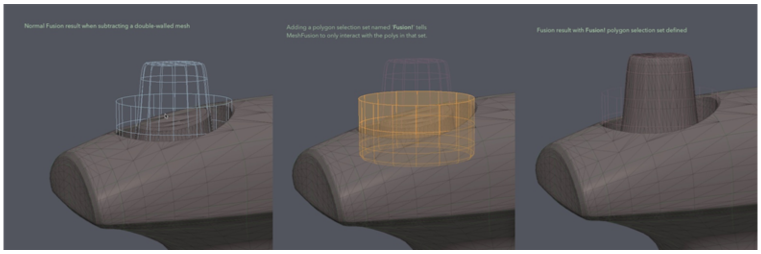
- **Fusion!**
- **Fusion+**
- **Fusion-**

Fusion!

The **Fusion!** selection set specifies which polygons of a MeshFusion source mesh should be included in MeshFusion's Boolean operations.

By default, when there is no **Fusion!** set, all source mesh polygons are of the intersect type, which means that if they encounter another source mesh in space and that other source mesh was designated as interacting with this source mesh, they are cut by it, creating an intersection seam. The **Fusion!** set allows you to mark only some polygons of the source mesh as intersection-producing. The remaining polygons are still present in the Fusion output, but they are present only as a whole, never cut by other surfaces.

A simple example of using this feature is sticking a double-walled tube into a flat surface. Only the outer wall of the tube needs to be welded to the flat surface with a Fusion strip, but in a straightforward Boolean operation the inner wall is welded to the surface too. To avoid this, create a **Fusion!** set on the tube containing only the polygons of the outer wall.


Normal Fusion result when subtracting a double-walled mesh
Adding a polygon selection set named 'Fusion!' tells MeshFusion to only interact with the polys in that set.
Fusion result with Fusion! polygon selection set defined

Fusion+ and Fusion-

The **Fusion+** and **Fusion-** selection sets define which polygons in the source mesh should be used by MeshFusion. By default all polygons are used:

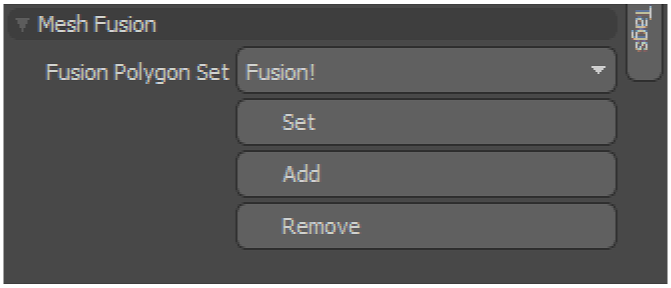
- If a **Fusion+** set is present, only polygons in that set are used.
- If a **Fusion-** set is present, only polygons not in that set are used.
- If both sets are present, MeshFusion uses polygons in the **Fusion+** set, excluding those that are also in the **Fusion-** set.

Unlike in the case of **Fusion!**, polygons excluded by **Fusion+**/**Fusion-** are fully excluded, as if they were not present in the source mesh at all. These selection sets might be useful when your source mesh contains areas that are incompatible with MeshFusion, such as edges with three polygons attached to them. You can use these selection sets to exclude problematic areas from MeshFusion.

Specifying the Selection Sets

To specify a selection set on a mesh:

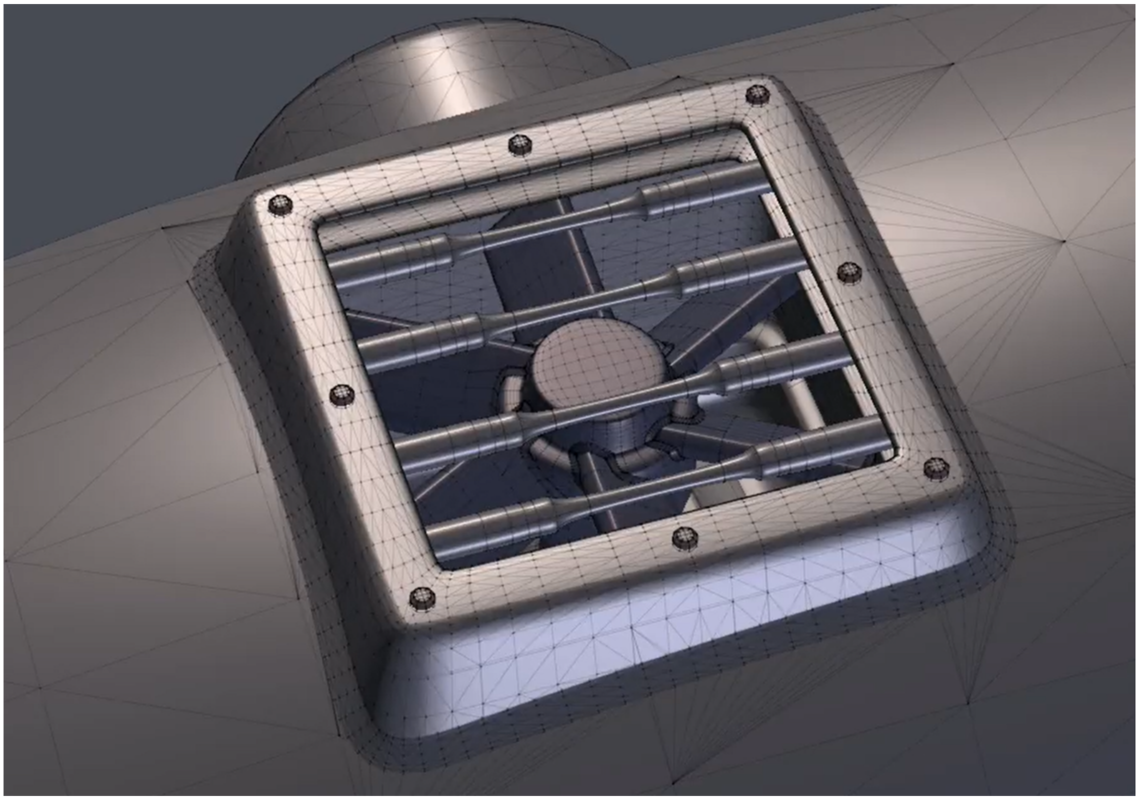
1. Select the polygons to include.
2. In the item's **Properties** panel on the right, make sure you're in the Mesh sub-tab.
3. Under **MeshFusion**, set **Fusion Polygon Set** to the required selection set (**Fusion!**, **Fusion+**, or **Fusion-**), then click **Set**.



Fusion Polygon Set: Fusion!

The selected polygons are marked by the selection set. You can add more polygons by selecting them and clicking **Add**, or remove polygons by clicking **Remove**.

Kit Fusing



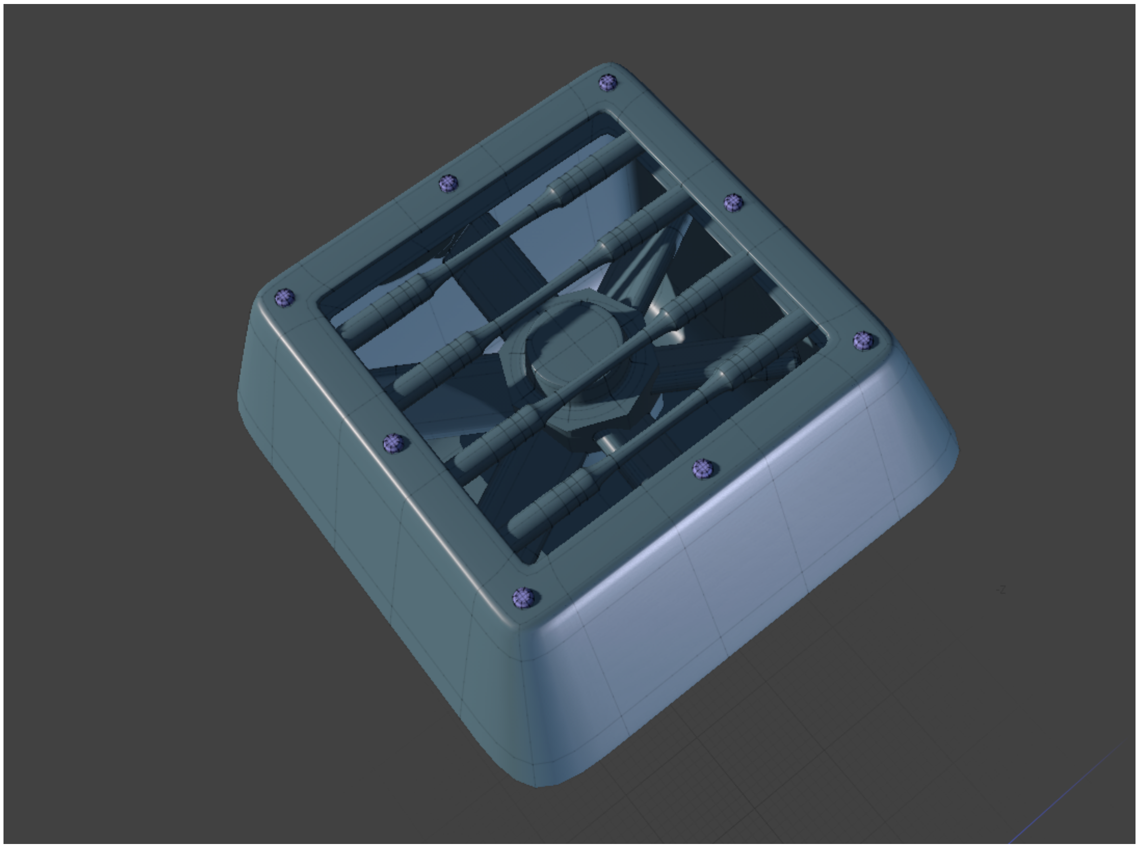
Kit Fusing is a workflow that allows you to add a new feature containing a detailed mesh to your model, using very simple MeshFusion operations.

You can specify only part of your mesh to attach to the main mesh. Typically, you can use some sort of an outer sleeve for that. The Fusion! polygon selection set can be used to designate that outer sleeve to interact with MeshFusion. This way, only that part of the newly added mesh interacts with the main mesh, producing

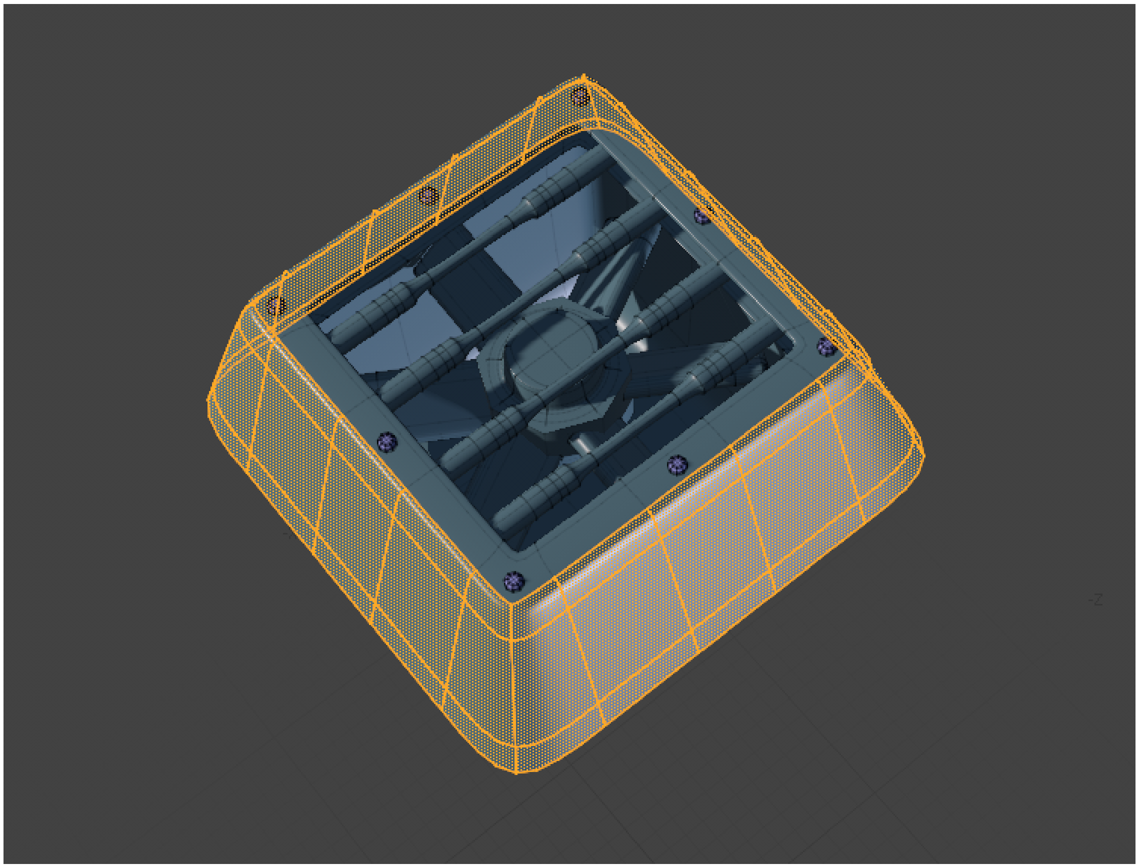
Fusion strips. All the other parts of the new mesh remain as they are, not cut by the main mesh, even though it might go through them in space.

The newly added feature might also have additional separate meshes, which are grouped with the mesh that is interacting with the main mesh. This way you can move them together, but these additional meshes are not part of the Fusion setup.

An example for this is adding the following vent to the main mesh:



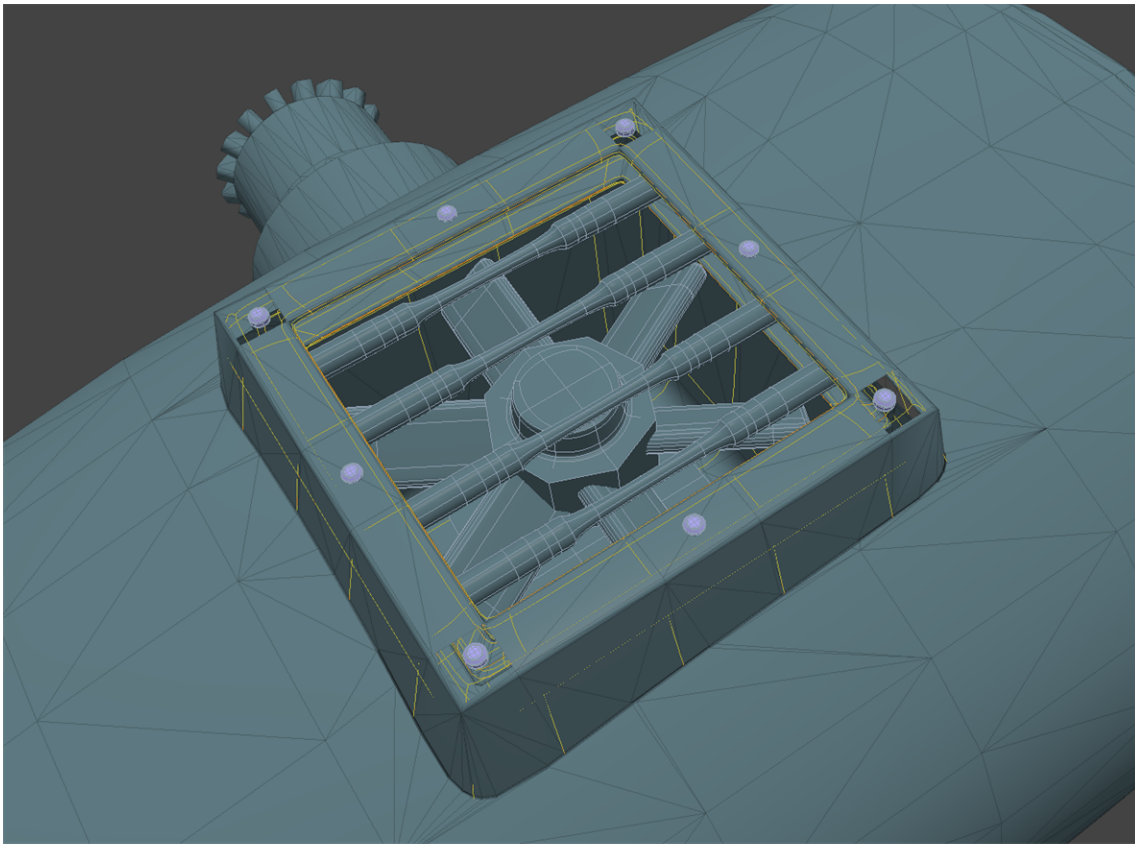
This feature consists of multiple meshes, such as the bars, the fan, and other details on the inside, which are all parented under the mesh that makes up the sleeve. We only want this outer mesh to participate in MeshFusion, so those polygons are added to the **Fusion!** selection set.



To add this new feature, select the main mesh along with the new mesh, and create a new Fusion Item.

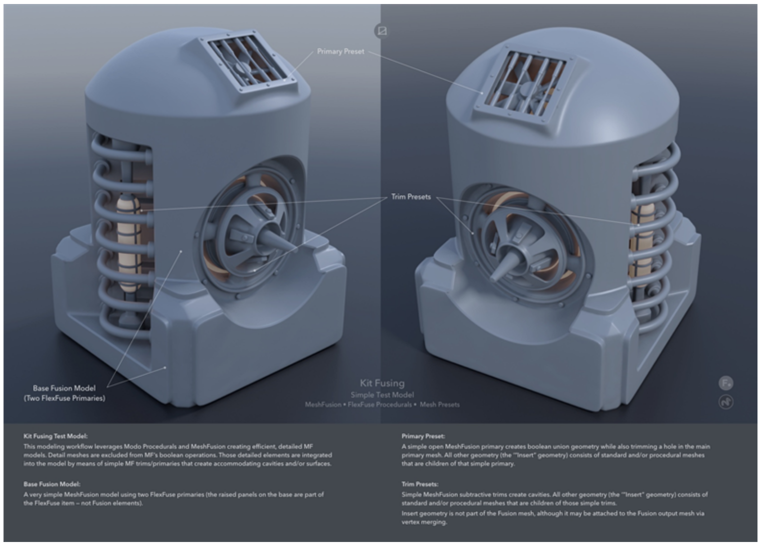
Tip: For more information on creating Fusion items, see Creating a New Fusion Item.

The new feature is added to the main mesh using a simple intersection from MeshFusion's point of view.



Other Kit Fusing Examples

The following images show you some additional examples of how Kit Fusing can be used to create more detailed meshes.


Primary Preset
Trim Presets
Base Fusion Model
(Two FlexFuse Primaries)
Kit Fusing
Simple Test Model
MeshFusion • FlexFuse Procedurals • Mesh Presets

Kit Fusing Test Model:

This modeling workflow leverages Modo Procedurals and MeshFusion creating efficient, detailed MF models. Detail meshes are excluded from MF's boolean operations. Those detailed elements are integrated into the model by means of simple MF trims/primaries that create accommodating cavities and/or surfaces.

Base Fusion Model:

A very simple MeshFusion model using two FlexFuse primaries (the raised panels on the base are part of the FlexFuse item – not Fusion elements).

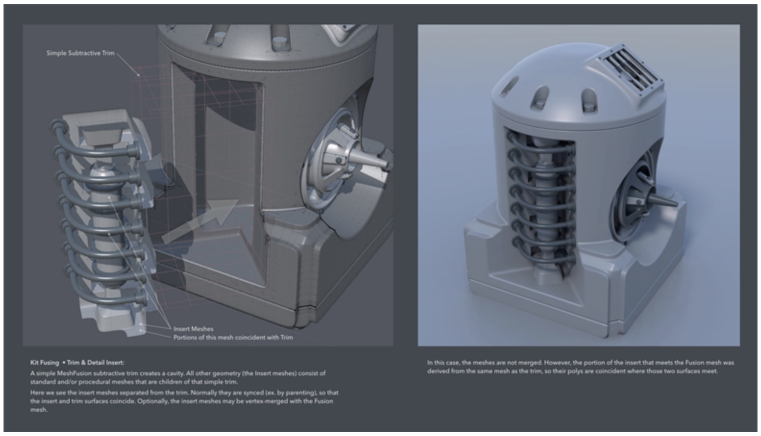
Primary Preset:

A simple open MeshFusion primary creates boolean union geometry while also trimming a hole in the main primary mesh. All other geometry (the ""Insert" geometry) consists of standard and/or procedural meshes that are children of that simple primary.

Trim Presets:

Simple MeshFusion subtractive trims create cavities. All other geometry (the ""Insert" geometry) consists of standard and/or procedural meshes that are children of those simple trims.

Insert geometry is not part of the Fusion mesh, although it may be attached to the Fusion output mesh via vertex merging.



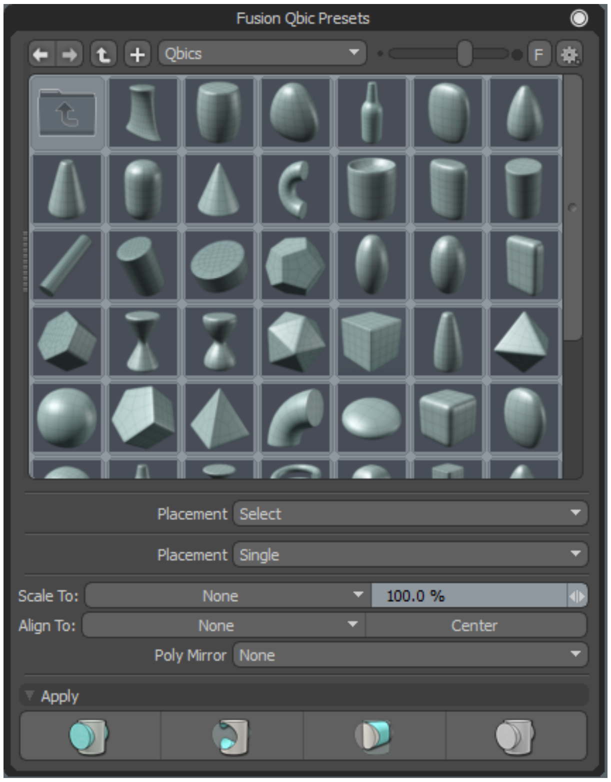
Kit Fusing • Trim & Detail Insert:

A simple MeshFusion subtractive trim creates a cavity. All other geometry (the Insert meshes) consist of standard and/or procedural meshes that are children of that simple trim.

Here we see the insert meshes separated from the trim. Normally they are synced (ex. by parenting), so that the insert and trim surfaces coincide. Optionally, the insert meshes may be vertex-merged with the Fusion mesh.

In this case, the meshes are not merged. However, the portion of the insert that meets the Fusion mesh was derived from the same mesh as the trim, so their polys are coincident where those two surfaces meet.

Fusion Qbic Presets



Placement: Select

Placement: Single

Scale To: None 100.0 %

Align To: None Center

Poly Mirror: None

Qbic meshes are specifically designed to help with building Fusion models. They are composed of quadrangles, which guarantees smooth operation in MeshFusion. Modo offers a broad palette of Qbic meshes, which you can access by one of the following routes:

- In the **Model** layout, on the **Fusion** sub-tab of the modeling toolbox, click **Fusion ready Presets** to open the **Fusion Qbic Presets** window. The Qbics are found under **Fusion Assets** > **Fusion Meshes** > **Qbics**.

- In the **Schematic Fusion** layout, the presets are part of the layout by default, in the lower-right panel.
- In the Preset Browser, you can find the Qbics under **Fusion Assets** > **Fusion Meshes** > **Qbics**. Press **F6** to open the Preset Browser.

Adding a Qbic

Double-click on a primitive to add it to the scene. Alternatively, before adding the primitive, you can adjust the placement settings and click one of the **Apply** buttons to choose how to add it.

There are four buttons on the bottom of the preset window as you can see in the image above.

From left-to-right, the buttons are:

- **Add Preset as Fusion Primary** - Add the preset as a primary.
- **Add Preset as Fusion Subtractive Trim** - Add the preset as a subtractive trim.
- **Add Preset as Fusion Intersect Trim** - Add the preset as an intersection trim.
- **Add Preset to scene without adding to Fusion Item** - Simply add the preset to the scene. You can then add it to the fusion manually if required.

Note: For more information on mesh roles, see Mesh Roles and Relationships.

Note: To use the first three buttons you must have a Fusion Item in the scene and a selected item.

The following placement parameters are available:

Fusion Qbic Presets	
Placement Presets	Choose a placement preset: this populates the placement parameters. Alignment options: • **Select** - Don't use a placement preset. • **Single No Align/Scale** - Use the **Single** setting for **Placement**, and None for **Align To**. • **Single Align/Scale To Elements** or **Meshes** - Use the **Single** setting for **Placement**, and scale the preset to the selection. • **Combine Align/Scale To Elements** or **Meshes** - Use the **Combined** setting for **Placement**, and scale the preset to the selection.

	• **Multi Align/Scale To Elements** or **Meshes** - Use the **Multiple** setting for **Placement**, and scale the preset to the selection.
Placement	Sets the placement type: • **Single** - Places a single copy of the selected preset. • **Combined** - When multiple targets are selected (multiple mesh elements or multiple meshes), one copy of the preset is placed relative to each of those targets, then combined into a single mesh item • **Multiple** - Same as **Combined**, except preset copies remain separate mesh items after placement. **Note:** Combining meshes that are to be used as fusion sources is efficient from a performance standpoint, however, they must not overlap. Overlapping makes for a self-intersecting mesh, which fusion can't process. On the other hand, the **Multiple** option is good from an editing standpoint as each placed preset copy is properly transformed (except when alignment options are used) and can often be conveniently adjusted after placement by using the **Local** Action Center.
Scale To:	Scales the Qbic at placement. Choose from: • **None** - Do not scale. • **Selected Meshes** - Adjust height to match the selected mesh(es). If more than one mesh is selected, it uses the first mesh in the **Item List**. • **Selected Element** - Adjust height to match the selected elements. • **Unit** - Set the height manually. Select one of the options above and then set the amount.
Align To:	Choose where to align the Qbic: • **Selected Meshes** - Align to the selected mesh(es). If more than one mesh is selected, it uses the first mesh in the **Item List**. • **Selected Elements** - Align to the selected elements (such as polygons) on the current item. • **X, Y, Z, -X, -Y, -Z** - Align to one of the axes. A negative value reverses the direction of the Qbic along that axis. If you have selected meshes or selected elements, select **Center** to position the Qbic

	at the center of the selection.
Poly Mirror	Creates two of the selected Qbic, with the second mirrored around the selected axis.

Working with MeshFusion Assemblies

Modo provides a number of procedural **MeshFusion Assemblies**, which allows you to work with simple geometries and familiar direct modeling workflows and tools. For example, to improve your Game Engine workflow, use these MeshFusion assemblies with high polygon assets to quickly design your models. These assemblies provide the ability to fuse complex procedural geometry to broaden the applicability of that geometry. You can freely combine procedurals and direct modeling geometry into complete models that would otherwise be difficult to achieve using direct modeling techniques.

Tip: The MeshFusion Item Assemblies are no longer included with Modo but can be downloaded from the Preset Browser **Cloud Assets** > **AddOns** > **Foundry** section. Moving the assemblies outside the application, allows us to update and improve them independently of the Modo release schedule.

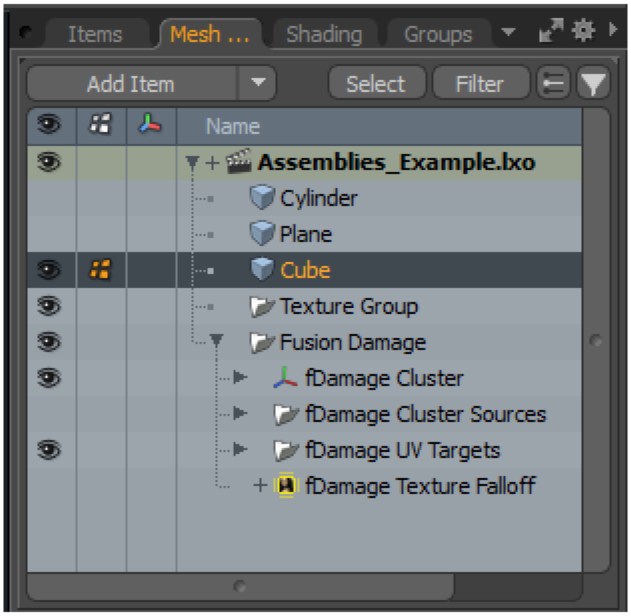
Adding a MeshFusion Assembly

Tip: To improve the performance, you can enable the MeshFusion property **Deferred Updating**. Updating is then only seen when you move your mouse upwards in the **3D Viewport**. For more information, see Fusion Item.

To add a MeshFusion assembly:

1. From the **Preset Browser**, click **Cloud Assets** > **AddOns** > **Foundry** and download the assemblies.
2. Open the **Model** layout.
3. In the right panel, open the **Items** list, and select a Fusion Item or mesh item.
4. In the right panel, click **Add Item**, and expand **Assemblies** > **Item Assemblies** > **MeshFusion Assemblies**.
5. Double-click on an assembly to add it to your scene.

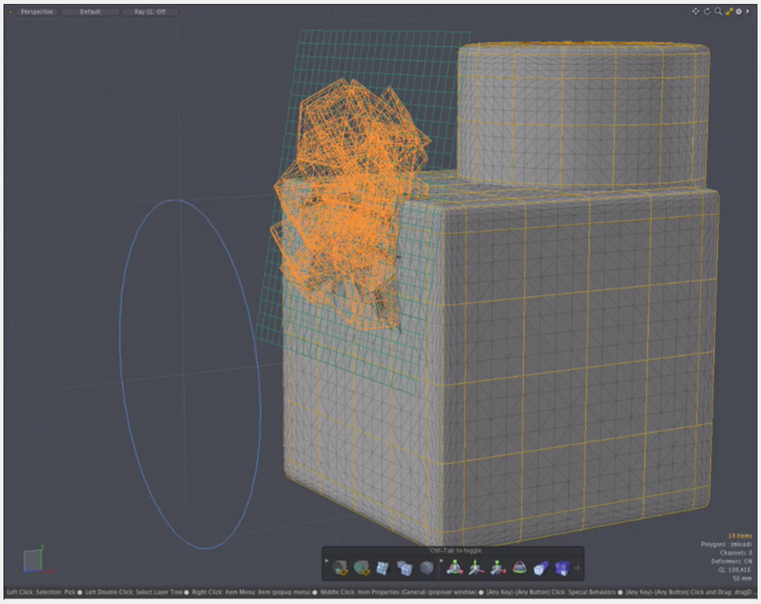
6. Open the **Mesh Ops** tab to view the assembly components. In this example, I have added the **Fusion Damage** MeshFusion assembly to the **Cube** mesh item.

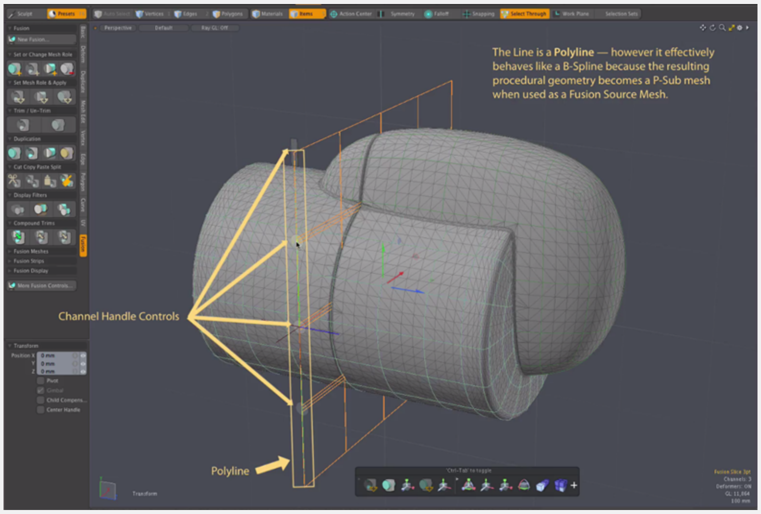
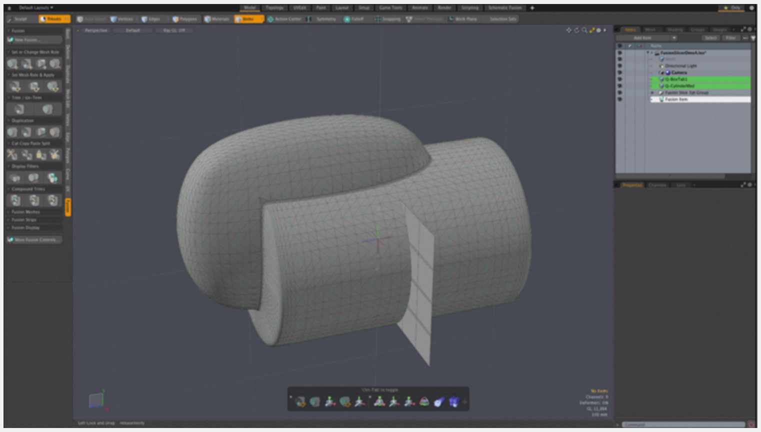


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Assemblies_Example.lxo
Cylinder
Plane
Cube
Texture Group
Fusion Damage
fDamage Cluster
fDamage Cluster Sources
fDamage UV Targets
fDamage Texture Falloff

Procedural MeshFusion Assemblies

Assembly	**Description**
ElboPipe	Creates piping consisting of straight segments and relatively short bends (with optional elbow joint geometry at the bends). Uses **Pen Extrude** Mesh Operation to create your ElboPipe curves. You can create a path as a polyline to create elbows around the geometry. As a result, the geometry is automatically added at each vertex of the polyline. **Tip:** While this assembly was designed with Fusion-friendly source meshes in mind, it is also useful when designing non-Fusion models.
Fusion Damage Trim	Combines individual meshes that act as subtractive Fusion Trims and a plane that acts as the target for a procedural UV transform. The individual meshes are being mapped into the UV space of the plane. The plane has a procedural Bend Effector Mesh Operation applied to it, which allows you to shape the trims to fit to the geometry of the

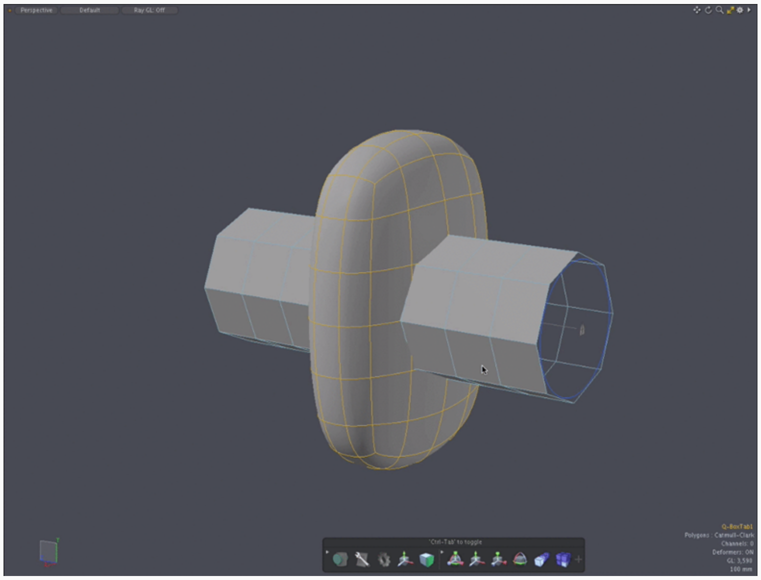
	Fusion Item. This assembly creates trims from the Primaries. Click the image below to view an animation.  **Tip:** You can relocate individual meshes in the **Damage Cluster Mesh Operations** list by selecting them and applying the transform tools. Therefore, affecting where they fall on the UV transform target plane. Select the plane in **Item** mode and then use the transform tool to move it onto your **Fusion Item**. You then turn these meshes into fusion trims by first expanding the **Fusion Damage** > **Damage Cluster** in the **Items** tab on the right panel and selecting all mesh items. Then select the primaries you wish to trim from your Fusion Model.
Fusion Slicer 3Point	Creates a thickened plane with three control points that allows you to set the path of your slice. This assembly has a Locator that is used to transform the assembly, Channel Handle controls, a B-Spline, and a polyline used to slice. All of the components of this assembly are seen in the **Mesh Operations** list.

Fusion Slicer 7Point	Creates a thickened plane with seven control points that allows you to set the path of your slice. This assembly has a Locator Item that is used to transform the assembly, Channel Handle controls, a B-Spline, and a polyline used to slice. All of the components of this assembly are seen in the **Mesh Operations** list.  The Line is a **Polyline** — however it effectively behaves like a B-Spline because the resulting procedural geometry becomes a P-Sub mesh when used as a Fusion Source Mesh. Once added to your scene, use the Transform tools to move and scale the slicer to the appropriate position on your model. Select the primaries and the slicing plane to trim. In the Toolbox, on the left panel, click **Set new mesh to Subtractive Trim and apply to selected meshes**. The geometry is sliced. Click the image below to view an animation. 
Profile Sweep	Creates a procedural profile that you can apply to your **Fusion Item**

or other modeled geometry. This assembly also supports a mesh to use as the profile as opposed to the Spline based profile. In addition, from the **Model** layout Toolbox **Fusion** sub-tab, you can change the profile to use one of the many **Fusion Meshes** > **Fusion-Ready Presets** > **pFusion Profiles**.

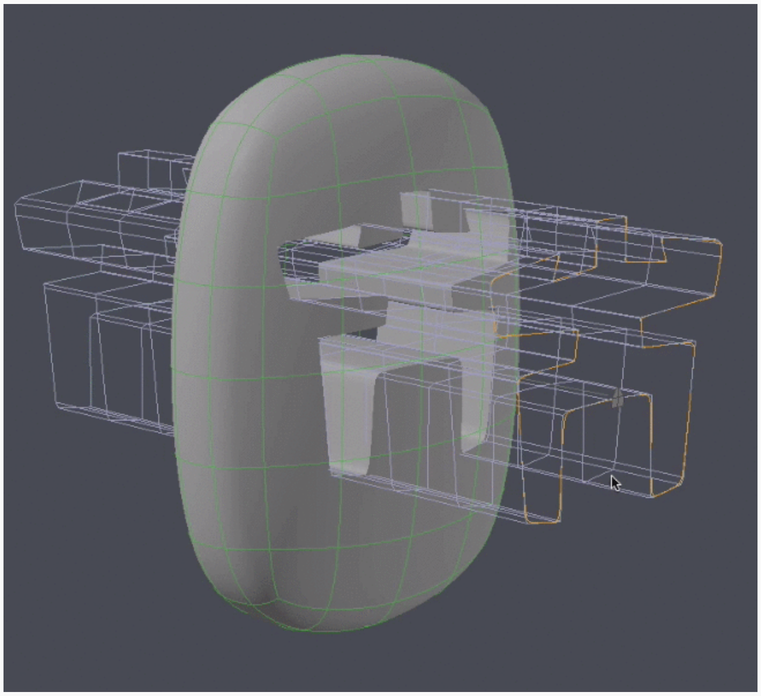
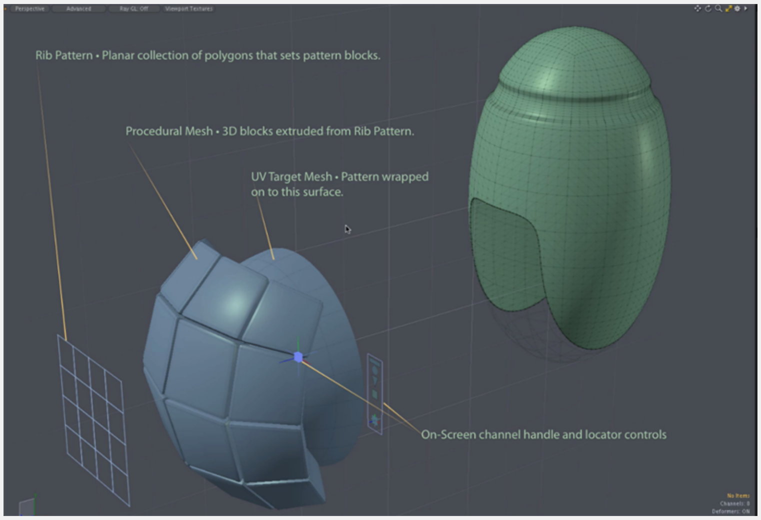
To add the **Profile Sweep** assembly to your scene, drag-and-drop it onto the **Fusion Item** source mesh and select the type of fusion you would like to perform from the contextual menu. For example **Fusion: Apply Subtraction**.

Click the image below to view an animation.

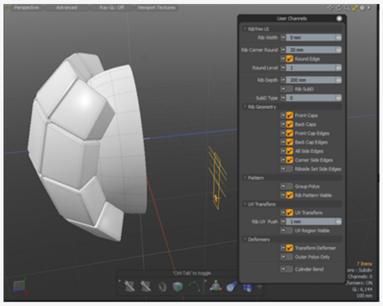


To change the assembly profile to use another **Fusion-Ready Presets Preset**, open the Toolbox **Fusion** sub-tab on the left panel. Then click **MeshFusion** > **Fusion-Ready Presets**. Navigate to **pFusion Profiles** and select a new profile. Then select a source item (polygon, item, or mesh) and the **Fusion Item** from the **Mesh Operations** list. Right-click on the assembly widget to select **Import Browser Profile** from the contextual menu. The new profile is applied.

Click on the image below to view an animation.

	
RibTrim	Automates the creation of frame, rib, and tile structures. This assembly creates patterns of edge looped blocks that may be used as Fusion Trims or Mesh Roles and Relationships. The blocks are aligned to selected geometry using a procedural UV Transform and / or a procedural **Deformer**. You can create and edit patterns as simple 2D meshes by creating planes with an arbitrary arrangement of polygons. For more information about Deformers, see Applying Deformers.  Rib Pattern • Planar collection of polygons that sets pattern blocks. Procedural Mesh • 3D blocks extruded from Rib Pattern. UV Target Mesh • Pattern wrapped on to this surface. On-Screen channel handle and locator controls

Tip: You can transform the **RibTrim** channel controls to a new location in your scene and double-click it to view the **User Channel** options.



User Channels

RibTrim UI

Rib Width: 5 mm

Rib Corner Round: 20 mm

☑ Round Edge

Round Level: 1

Rib Depth: 200 mm

☐ Rib SubD

SubD Type: 0

Rib Geometry

☑ Front Caps

☑ Back Caps

☑ Front Cap Edges

☑ Back Cap Edges

☑ All Side Edges

☑ Corner Side Edges

☐ Ribside Set Side Edges

Pattern

☐ Group Polys

☑ Rib Pattern Visible

UV Transform

☑ UV Transform

Rib UV Push: 1 mm

☐ UV Region Visible

Deformers

☑ Transform Deformer

☐ Outer Polys Only

☐ Cylinder Bend

Using Curves with MeshFusion

Curves, Beziers, and B-Splines can serve as input meshes in MeshFusion modeling. They can be used the same way as regular polygonal meshes and produce primaries, trims, and surface strips.

B-Splines are native to MeshFusion, but Curves and Beziers are automatically converted to B-Splines when used by MeshFusion. Bezier curves can only be converted to B-Spline approximately, but you can increase the number of B-Spline segments for greater accuracy.

MeshFusion works with open, closed, flat, and 3D curves. You can also use curves produced by the Text tool with the Bezier **Output Type**.

Using a Curve as an Input Mesh

To use a Curve as an input mesh, make sure you have a scene containing a mesh item in one layer and a curve in another mesh layer.

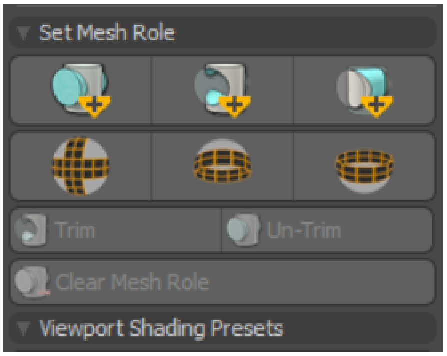
Note: For more information on adding different types of curves, see Adding Curves, Bezier, or B-Spline.

1. Select your mesh item in the Item List, and create a new fusion item, as described in Creating a New Fusion Item.

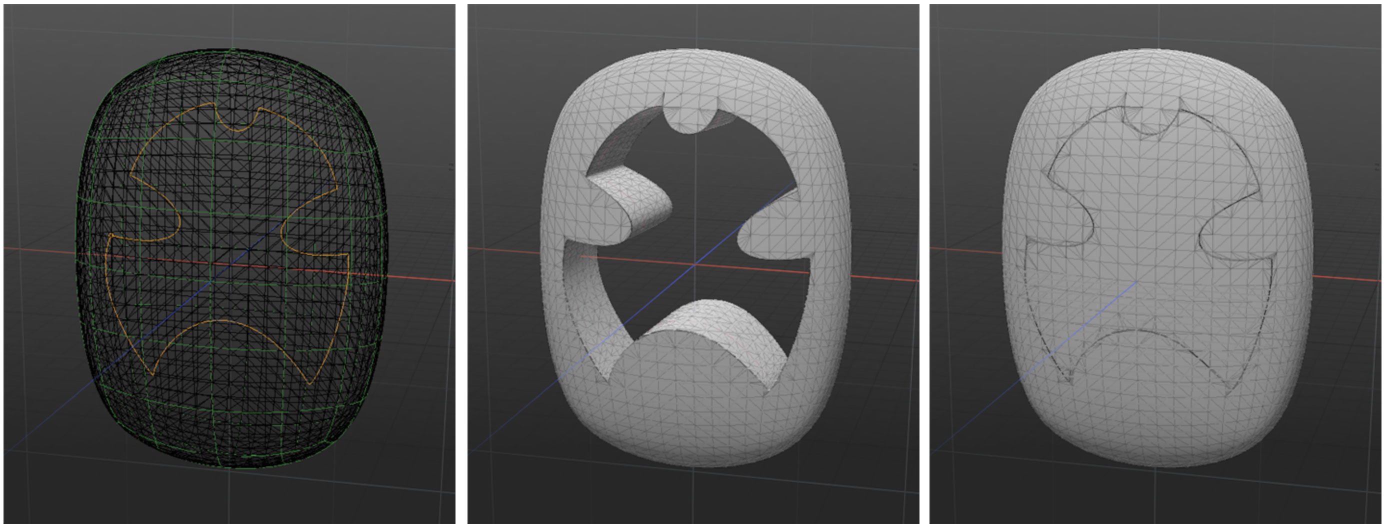
   Your mesh item becomes a primary and it's marked with green in the Item List and the 3D viewport. A Fusion Item is also created.
2. In the Item List, select the mesh layer containing your curve, and the mesh layer containing your primary.

3. In the **Fusion** sub-tab on the left panel, click the required button, depending on the kind of trim or surface strip you want to create. For example, to create the surface strip example below, click the Full Strip with Grid Crossing button under **Apply Surface Strip**.

   For more information on mesh roles and surface strips, see Mesh Roles and Relationships, and Fusion Surface Strips.



This example shows a curve applied to one of the Fusion Qbic Presets first as a Subtractive Trim, then as a Surface Strip.



Original

Subtractive Trim

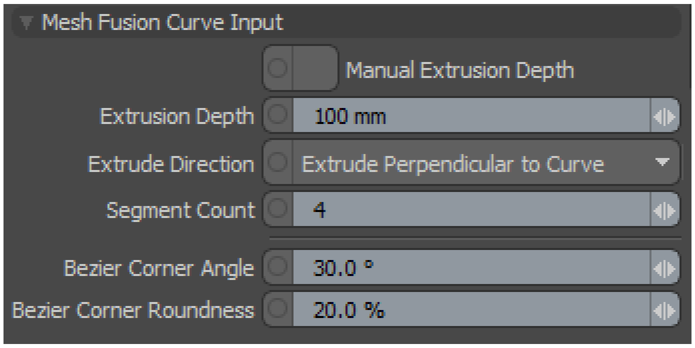
Surface Strip

Curve Input Properties

The Curve input settings help you adjust how your curve is converted to B-Spline and how geometry is extruded during MeshFusion. To access them:

1. Select your curve in the Item List.

   This opens the **Properties** tab on the lower right panel.
2. In the **Mesh** sub-tab, scroll down to the **Mesh Fusion Curve Input** section.



Manual Extrusion Depth

Extrusion Depth: 100 mm

Extrude Direction: Extrude Perpendicular to Curve

Segment Count: 4

Bezier Corner Angle: 30.0 °

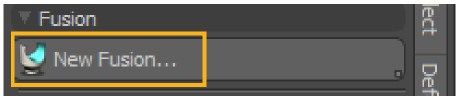
Bezier Corner Roundness: 20.0 %

Manual Extrusion Depth	When enabled, you can specify the **Extrusion Depth** value. When disabled, Modo uses the default.
Extrusion Depth	Specifies how far meshes are extruded from the curves when **Manual Extrusion Depth** is enabled.
Extrude Direction	Specifies the way meshes are extruded. The options are: • **Extrude Perpendicular to Curve** - The surface is extruded perpendicular to the plane of the curve, to a specified distance. This works well if the curve is flat, that is, it doesn't deviate from a plane. • **Extrude Along Surface Normal** - The curve is treated as a 3D curve and the surface produced from it is used as a trim. The cutting surface is extruded in the direction perpendicular to the surface being cut. This works for creating a surface strip on any curved surface by laying down the curve on that surface using the Mesh Constraints tools.
Segment Count	Specifies the number of B-Spline segments used when converting Bezier curves to B-Spline. Bezier curves can only be converted to B-spline approximately. The greater the number of B-Spline control points, the closer the approximation. Increasing the number of B-spline segments provides greater accuracy, but too many segments can reduce performance and increase output mesh density. For smooth Bezier curves without sharp angles, even a limited number of B-Spline segments is enough.
Bezier Corner Angle	The corners on a Bezier curve are automatically rounded when it's converted to a B-Spline. This specifies a threshold for rounding corners. Only the sharp corners where the change in direction is significant enough are rounded. Shallower corners are rounded by a very smooth curve, essentially becoming non-corners.
Bezier Corner Roundness	This value specifies the strength of the rounding effect.

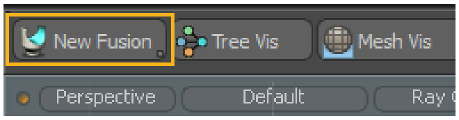
Creating a New Fusion Item

To build a fusion, you select your starting meshes and use the **Fusion New** window to create a Fusion Item. You can open the window by clicking the **New Fusion** button. This is found in the modeling toolbox's **Fusion** sub-tab in the **Model** layout, and in the top-left of the **Schematic Fusion** layout.

Model Layout



Schematic Fusion Layout



Using the Fusion New Window

Once opened, follow these steps to create a new fusion:

1. In the **Name** field, enter the name of the new item.
2. Before creating a new Fusion item, you can optionally edit the following strip properties, found under **Strip Properties**:

Note: For more information on working with strips, see Fusion Surface Strips.

- **Strip Width** - Sets the width of the strips in the new fusion.
- **Strip Profile** - Sets the **Profile** amount. The smaller the value, the flatter the strip. When working with surface strips, set this value to 100% for best results.
- **Strip Smooth** - Sets the smoothing value for the strips.

Note: You can edit these values later in the strip's **Properties**.

3. Set any of the following **Edge Weight Type** settings, found under **Edge Weighting**.

Note: MeshFusion simulates Catmull-Clark edge-weighting by internally adding edge loops to the source meshes (original source meshes are unchanged).
When selecting an **Edge Weight Type**, the setting persists for any subsequent new Fusion items for all Modo sessions.

- **Off** - Disables Edge Weighting.
- **Double Loops** - Adds two edge loops on either side of each weighted edge. Provides best fidelity to the CC surface, but produces denser Fusion meshes that may slow performance.
- **Single Loops** - Adds a one edge loop on either side of each weighted edge. This option produces a surface that is often indistinguishable from that created with Double Loops. It produces cleaner, less dense, Fusion mesh items.

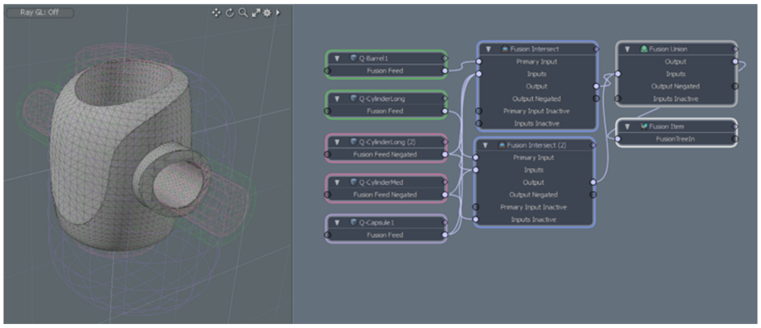
Note: With certain source mesh topologies, fidelity to actual CC surface may be lower and greater Fusion SubDiv levels may be automatically enforced (effectively canceling the lower density benefits).

- **Adaptive Loop Count** - Determines the best loop count (single or double loops) based on the nature of source mesh topology. Double loops are only created on those edges where single looping have fidelity and density requirement disadvantages.

4. If you're working with a dense source mesh, you can decrease the **Fusion Subdivision Level**. This may cause problems with other aspects of the geometry. For details on how to address those, see Using MeshFusion with Zero Subdivision.
5. Create your Fusion item by clicking **New Fusion with Selected Meshes**.

   A new **Fusion Item** is added to the **Item List**. The **Fusion Item** is the basis of your Fusion mesh and enables further editing of the result. For more information, see Fusion Item.

Modeling in the Schematic Viewport


Ray GL: Off
Q-Barrel1
Fusion Feed
Q-CylinderLong
Fusion Feed
Q-CylinderLong (2)
Fusion Feed Negated
Q-CylinderMed
Fusion Feed Negated
Q-Capsule1
Fusion Feed
Fusion Intersect
Primary Input
Inputs
Output
Output Negated
Primary Input Inactive
Inputs Inactive
Fusion Intersect (2)
Primary Input
Inputs
Output
Output Negated
Primary Input Inactive
Inputs Inactive
Fusion Union
Output
Inputs
Output Negated
Inputs Inactive
Fusion Item
FusionTreeIn

You can build Fusion models using Modo's Schematic Viewport instead of using the 3D viewport. The **Schematic Fusion** layout supports the full range of Boolean expressions, and allows the use of mesh instances and sharing of meshes between multiple Fusion Items.

Schematic Fusion has been designed to be user-friendly with quick access to common features. There are a number of common shortcuts in the Schematic Pie Menu and context menus.

Note: Schematic Fusion modeling is not compatible with the new MeshFusion workflow introduced in Modo 10.2. You can still edit your model in the Schematic, but you then need to convert it to the new format. For more details on how to do this, see Legacy Fusion Items.

Note: Drag-and-drop uses the new workflow commands, so it's not compatible with legacy Schematic Fusion editing.

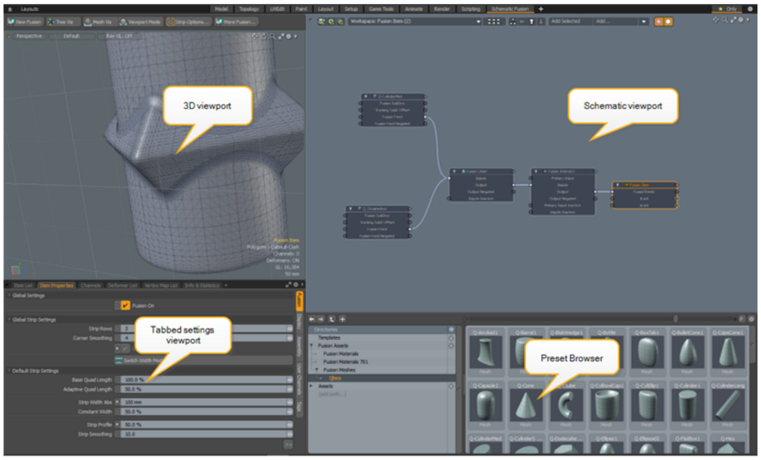
Creating a Schematic Fusion

In order to use Schematic Fusion, you must first create a Fusion Item using the **New Fusion** button. See Creating a New Fusion Item on how to do this.

A Schematic Fusion Item is created automatically. You can see it when you switch to the **Schematic Fusion** layout.

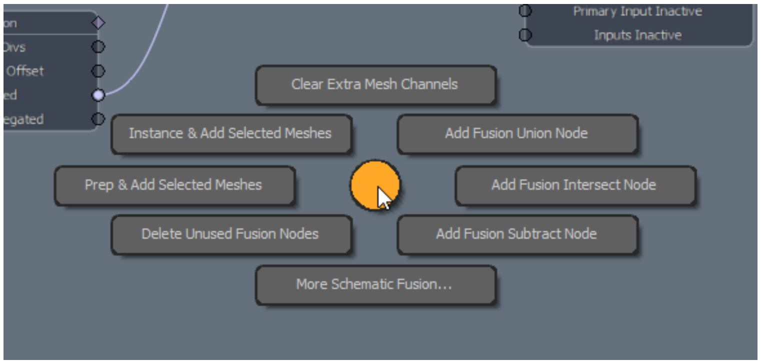
Schematic Fusion Layout

The default Schematic Fusion layout is divided into four viewports. The image below shows how the layout is organized. Click the image to enlarge it.


3D viewport
Schematic viewport
Tabbed settings viewport
Preset Browser

Schematic Pie Menu

The Schematic pie menu provides a shortcut to the most common Fusion commands. Position the mouse over the **Schematic** viewport and press **Ctrl**+**F** to open the Schematic pie menu.


Primary Input Inactive
Inputs Inactive
Offset
Clear Extra Mesh Channels
Instance & Add Selected Meshes
Add Fusion Union Node
Prep & Add Selected Meshes
Add Fusion Intersect Node
Delete Unused Fusion Nodes
Add Fusion Subtract Node
More Schematic Fusion...

The menu offers the following options (clockwise from the top):

- **Clear Extra Mesh Channels** - Remove any channels not related to Fusion.
- **Add Fusion Union Node** - Add a union node to the **Schematic**.
- **Add Fusion Intersect Node** - Add an intersect node to the **Schematic**.
- **Add Fusion Subtract Node** - Add a subtract node to the **Schematic**.
- **More Schematic Fusion...** - Open additional Schematic options.
- **Delete Unused Fusion Nodes** - Remove all childless nodes.
- **Prep & Add Selected Meshes** - Prepare all selected items for use in Schematic Fusion, and add them to the **Schematic** workspace, ready to be used in the Fusion mesh.
- **Instance & Add Selected Meshes** - Create an instance of each selected item and the instance(s) to the **Schematic** workspace, ready to be used in the Fusion mesh.

Note: Any changes to the original mesh are applied to the instance.

Prepping Items

To add a mesh to the **Schematic**, you must first prepare it so that its node has the necessary inputs and outputs. If you don't prep it, the mesh node doesn't have a **Fusion Feed** output and cannot be connected to a Boolean operator.

Let's say you want to add a Qbic primitive to the **Schematic**:

1. In the **Schematic Fusion** layout, double-click a Qbic preset in the Preset Browser (lower-right) to add it to the scene.
2. With the item selected and the cursor over the **Schematic** viewport, press **Ctrl**+**F** to open the pie menu, and select **Prep & Add Selected Meshes**.

   The node appears in the **Schematic** viewport.

Note: If the mesh is already in the **Item List**, skip step 1.

You can also prep the mesh by clicking **More Fusion...** above the 3D viewport and selecting **Prep Selected**. You can then add the mesh to the **Schematic** by dragging it from the **Item List** or the **3D viewport**.

Pruning the Schematic Fusion

There may be times when you are left with nodes that have no children, which do not affect the Fusion mesh.

If you want to remove the nodes and tidy the Schematic, follow these steps:

1. With the mouse over the **Schematic** workspace, press **Ctrl**+**F** to open the pie menu.
2. Select **Delete Unused Fusion Nodes**.

Schematic Fusion Item

When you create a Fusion model, a **Fusion Item** node is added to the **Schematic** viewport. This is required to produce a Fusion model. The results of each branch are fed into the **FusionTreeIn** input.

A Fusion item has an extensive set of properties that are detailed in Fusion Item.

Fusion Nodes

There are three types of node in any Fusion graph:

- **Meshes** - The source geometry. Only the Fusion channel outputs are used and they can be connected to Fusion operation nodes.
- **Fusion Operations** - **Fusion Union**, **Fusion Intersection**, and **Fusion Subtract**. These take input from Fusion output channels of meshes and/or other Fusion operation nodes. Their output channels may connect to other operation nodes or to the Fusion item.
- **Fusion Item** - Accepts a single output from a Fusion operation node only.

You typically take the **Fusion Feed** output from a mesh and wire it into the appropriate input of the Boolean operator. Remember to prep the mesh in order to have the **Fusion Feed** output. The various branches of the node graph are fed into a **Fusion Intersect** node to combine the branches. The flow then terminates in the **Fusion Item**.

Meshes

Meshes must be prepped as Fusion Source meshes. This process adds their two Fusion channels:

- **Fusion Feed** - Adds positive version of mesh to Fusion.
- **Fusion Feed Negated** - Adds negative version of mesh to Fusion. See Fusion Feed Negated for more information.

Operation Nodes

Operation nodes determine the Boolean operation performed using their inputs. You can add an operation node to the **Schematic** by clicking **Add...** at the top of the **Schematic** workspace. They are found under **Channel Modifiers** > **Other**.

Note: To quickly locate a Boolean operation node, type part of the name into the search field after clicking **Add...**.

Fusion Union

Fusion Union makes use of the following channels:

- **Input** - Any combination of meshes and outputs of other operation nodes.
- **Output** - The Boolean union of all inputs.

- **Inputs Inactive** - The inputs are swapped here when you select **Fusion Node Options** > **Deactivate All** from the context menu. You can reinstate them by selecting **Activate All**.

Note: To open the context menu, right-click on the node title.

Fusion Intersect

Fusion Intersect makes use of the following channels:

- **Primary Input** - The primary input is used to set the initial geometry for the intersection. This is required because intersections are performed sequentially, and the order of those operations can affect the result. The input is either a mesh or the output of another operation node. All other inputs (attached to the **Inputs** channel) are applied to this initial geometry.

Note: Having something connected to the **Primary Input** channel is not required, unless you need to control the order of operation. If it is not used, intersections are simply carried out in the order they were connected to the **Inputs** channel.

- **Inputs** - Any number of meshes and/or outputs of other operation nodes.
- **Output** - The Boolean result of the inputs.
- **Output Negated** - The negated Boolean result of the inputs.
- **Primary Input Inactive** - The **Primary Input** is swapped here when you select **Fusion Node Options** > **Deactivate All** from the context menu. You can reinstate it by selecting **Activate All**.
- **Inputs Inactive** - The inputs are swapped here when you select **Fusion Node Options** > **Deactivate All** from the context menu. You can reinstate them by selecting **Activate All**.

For **Primary Input**, the best practice is to think of it as the initial volume, and all additional inputs as geometry that trims away from that volume (all intersections remove volume). This is important because of Fusion's support of open surfaces. (Traditional Booleans are restricted to closed volumes, in which the order of intersection doesn't matter.)

For example, say you have a sphere as the Primary, and several one-sided planes as inputs, slicing away parts of the sphere. If you tried to first intersect the planes, it might fail. Intersecting two finite planes does not describe a surface or volume, and subsequent intersections would have nothing to interact with. Connecting the sphere to the **Primary Input** solves this.

Note: Be careful with open surfaces as their edges must be outside of the Boolean volume.

Fusion Subtract

The Fusion subtract node uses the following channels:

- **Input A** - The initial geometry to subtract from. This must be a single item (although it can be the output of another operation, which may itself have many inputs).
- **Input B** - The geometry to subtract from **Input A**. This must be a single item (although it can be the output of another operation, which may itself have many inputs).
- **Output** - The Boolean result of the subtraction.
- **Output Negated** -The negated Boolean result of the subtraction.

Note: We recommend against the use of subtraction nodes. Although they may seem more straightforward, they are less flexible and less efficient than using intersection. A negated intersection is the same as a subtraction. See Fusion Feed Negated.

Fusion Item Node

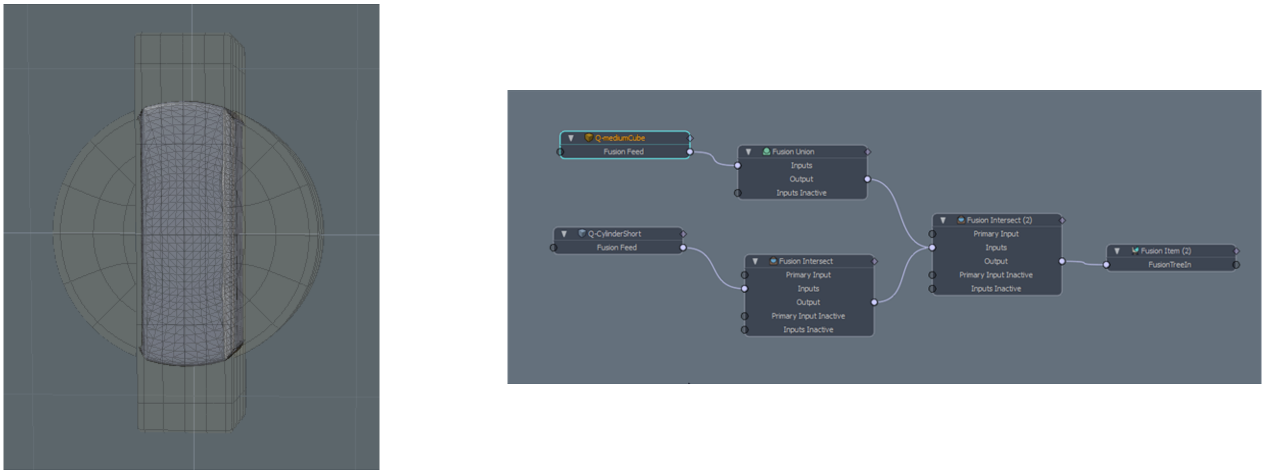
The Fusion Item node uses the following channel:

- **FusionTreeIn** - Accepts a single output from operation nodes only. This takes the output from the last operation node in the graph.

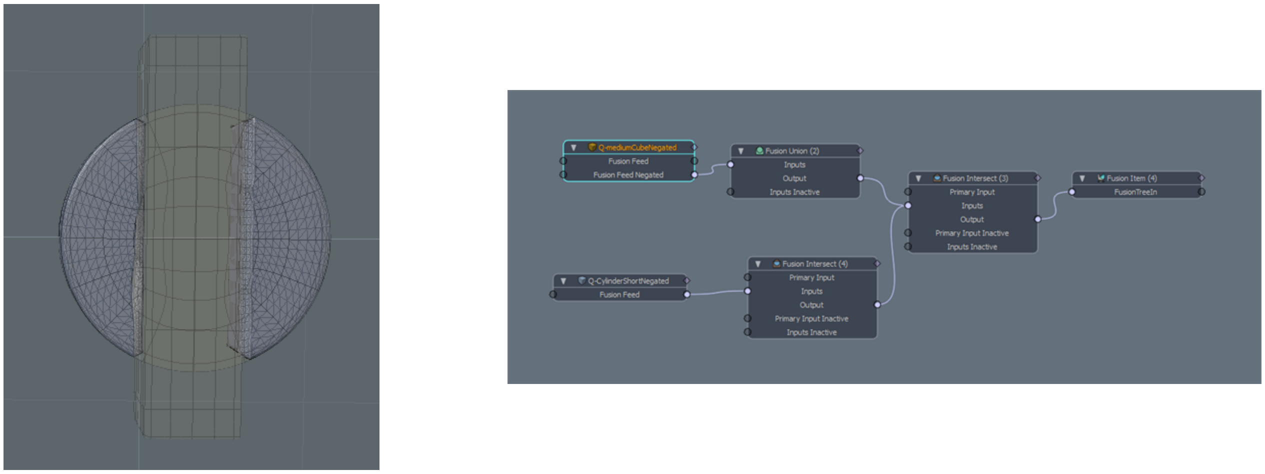
Fusion Feed Negated

Negating the feed generates a mesh with reversed geometry, similar to flipping the polygons. This is useful for when you want to have multiple subtractive elements on a mesh, which is difficult to achieve with Boolean subtraction.

A negated intersection is equivalent to a subtraction (intersections always remove geometry). With a normal intersection, all geometry is subtracted except for the geometry that is within both meshes. However, an intersection using **Fusion Feed Negated** subtracts the negated mesh and leaves the rest of the other mesh that is outside the negated mesh. The images below show the difference, the corresponding schematic is shown to the right of the result.

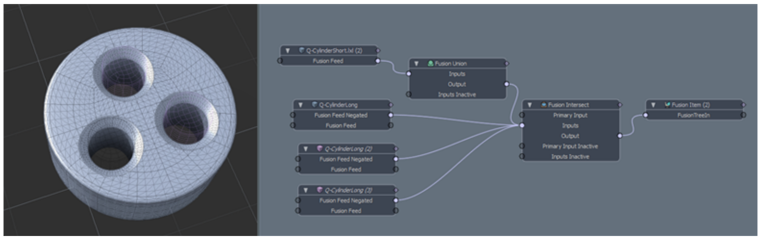

Q-mediumCube
Fusion Feed
Fusion Union
Inputs
Output
Inputs Inactive
Q-CylinderShort
Fusion Feed
Fusion Intersect
Primary Input
Inputs
Output
Primary Input Inactive
Inputs Inactive
Fusion Intersect (2)
Primary Input
Inputs
Output
Primary Input Inactive
Inputs Inactive
Fusion Item (2)
FusionTreeIn

Above: Intersection with normal rectangle.


Q-mediumCubeNegated
Fusion Feed
Fusion Feed Negated
Fusion Union (2)
Inputs
Output
Inputs Inactive
Fusion Intersect (3)
Primary Input
Inputs
Output
Primary Input Inactive
Inputs Inactive
Fusion Item (4)
FusionTreeIn
Q-CylinderShortNegated
Fusion Feed
Fusion Intersect (4)
Primary Input
Inputs
Output
Primary Input Inactive
Inputs Inactive

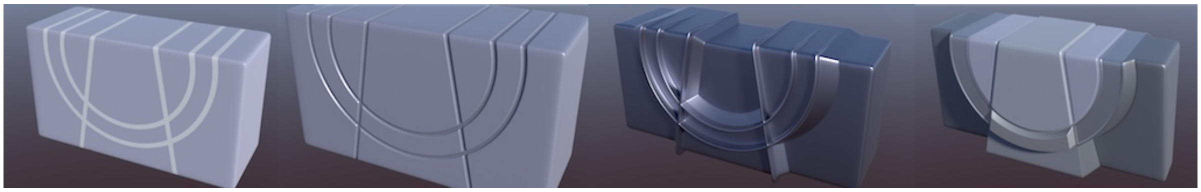
Above: Intersection with negated rectangle.

The image below shows a simple mesh that is constructed by intersecting negated cylinders (one original and two instances) with a single mesh.


Q-CylinderShort.lxl (2)
Fusion Feed
Fusion Union
Inputs
Output
Inputs Inactive
Q-CylinderLong
Fusion Feed Negated
Fusion Feed
Q-CylinderLong (2)
Fusion Feed Negated
Fusion Feed
Q-CylinderLong (3)
Fusion Feed Negated
Fusion Feed
Fusion Intersect
Primary Input
Inputs
Output
Primary Input Inactive
Inputs Inactive
Fusion Item (2)
FusionTreeIn

Note: You can also use subtraction to achieve the effect demonstrated above, by unifying the three small cylinders before subtracting the result from the larger cylinder.

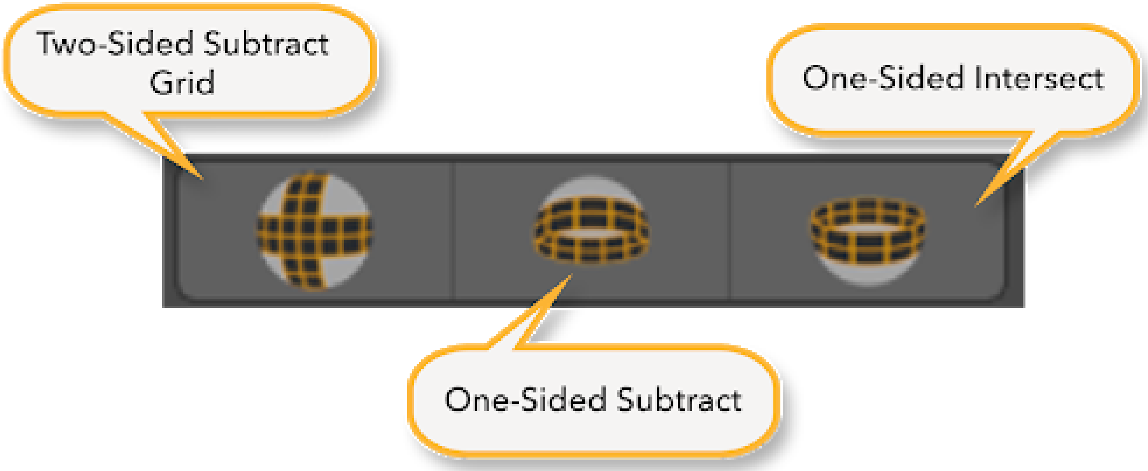
Fusion Surface Strips



Strips are a primary geometric element of a Fusion model. They are the surfaces around the intersecting edges of the source mesh. You can edit strip properties to determine the smoothness, size, or flatness of the strip. Surface Strips are also used for embossing and extruding meshes. Surface strips are useful for smoothing or beveling edges, as well as to create clean topology for when the **Fusion Item** is converted to a polygonal mesh.

Surface Strip Types

A number of surface strip types are found in the Fusion tab under Set Mesh Role.



	Creates a two-sided surface strip on the primary Fusion Item. A 90 degree grid is created between the corners of the intersection. The Fusion Item remains unchanged. Moving the mesh item moves the strip on the Fusion Item's surface.

	Tip: You can apply this to one or more primary Fusion Items with the mesh item used to create the surface strip. Selecting one or more primary items and clicking **UnTrim** removes the surface strip. Clicking **Trim** reapplies the surface strip.
	Subtracts the mesh item from the Fusion Item at the trim line and creates a single-sided surface strip. It performs the trim and leaves one half of the surface strip on the trimmed primary Fusion Item. No new surfaces are added by the trimming mesh item.
	Intersects the mesh item with the Fusion Item and creates single-sided surface strips along the cut edges. This function performs the trim and leaves one half of the surface strip on the trimmed primary Fusion Item. No new surfaces are added by the trimming mesh item.

Holding the **Alt** key and clicking these controls allows you to change the mesh's role, but maintain its relationship to Primaries.

Surface strips are created at the junction of two combined objects, and so it is also possible to use meshes as a source for surface strips.

Creating Surface Strips

Surface strips are created automatically along the seam between meshes when a trim or mesh role is assigned. The example below shows you how to use meshes to create surface strips on a **Fusion Item** as well as how to edit them.

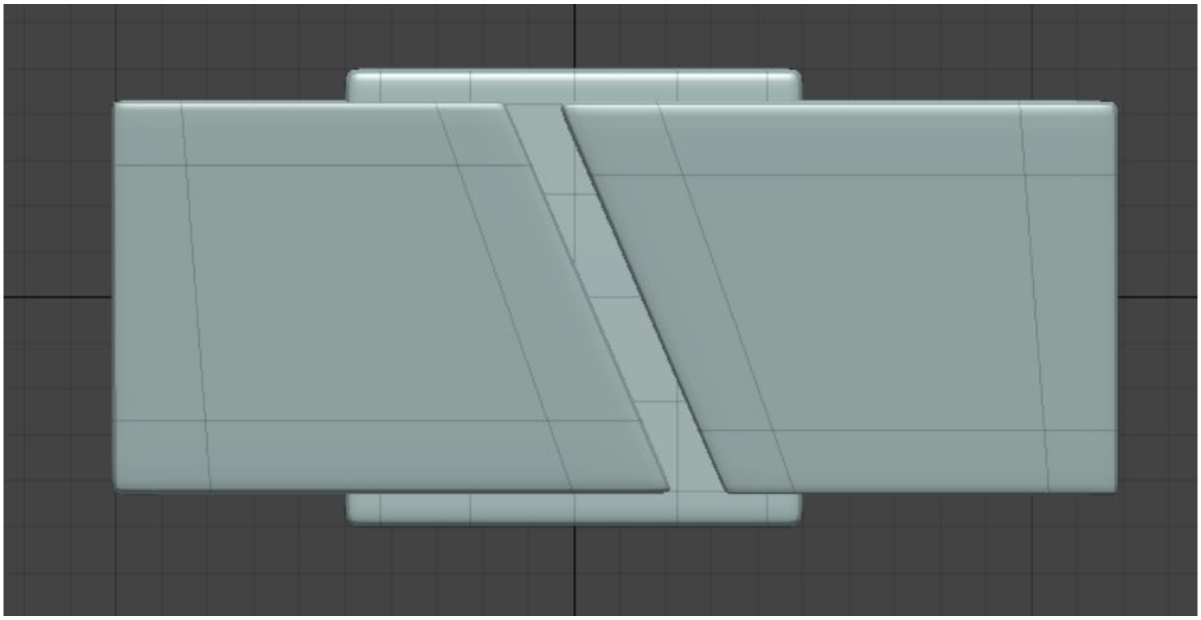
Note: The first time you create surface strips, we recommend using the **Fusion Item** properties used in the example. For a full list of **Fusion Item** properties, see Fusion Item Properties.

Note: Edits to strips will not update unless **Override Defaults** is enabled. This can be found in the **Fusion Strip** tab in the **Properties** window.

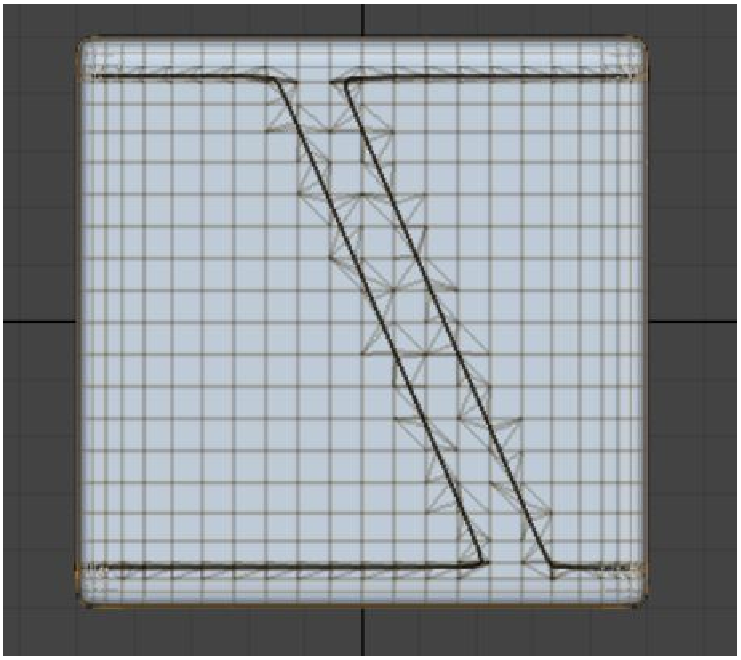
1. Use the **Fusion Ready Presets** window to load three Qbic meshes into your scene.

> **Note:** Primitive meshes can also be used to create Fusion Strips, but for the cleanest results use Qbic meshes as they are fully optimized for the assembly of fusion meshes.

2. Position the meshes so that they all overlap, as shown in the image below. The overlapping meshes create the fusion strips once fusion roles have been set.



3. Select the mesh to which you want to apply the strips from the **Items list**. In this example, the mesh we want to select is the cube at the back.
4. Navigate to the **Fusion** tab, and click the **New Fusion** button to convert the mesh to a Fusion Item.
5. Select the meshes overlapping the **Fusion Item** from the **Items** list. The overlapping meshes are used to create the surface strips.
6. Click ⊕ to create a **Two Sided Subtract Grid**. This button can be found under the **Set Mesh Roles** submenu in the **Fusion** tab.



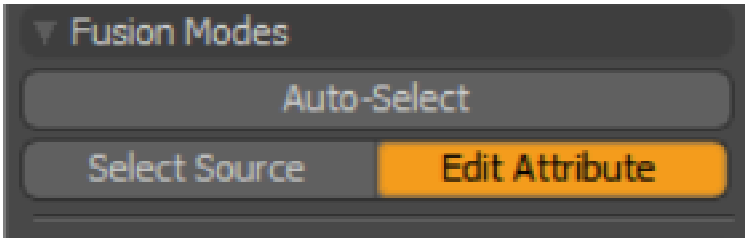
These steps produce a **Fusion Item** with two surface strips, but there's no limit to the number strips you can create.. By default, these strips will be quite thin but the next section of this page will teach you how to change the **Width** and **Profile** of these strips.

Curves, Beziers or B-Splines can also be used to create surface strips on Fusion Models.

Editing Surface Strips

Often, you might find yourself wanting to change the width, profile or density of your strip in order to create cleaner topology for your model once it's converted into a mesh item. Fusion strips can be easily edited in the **Fusion Strip** tab in the **Properties** menu. This section with go over what strip parameters can be edited and how to do it.

Before you can edit strips, you need to be able to select them. To edit the attributes of a Fusion Item, make sure that **Edit Attributes** is selected. **Edit Attributes** can be found under the **Fusion Modes** menu in the **Fusion** tab.



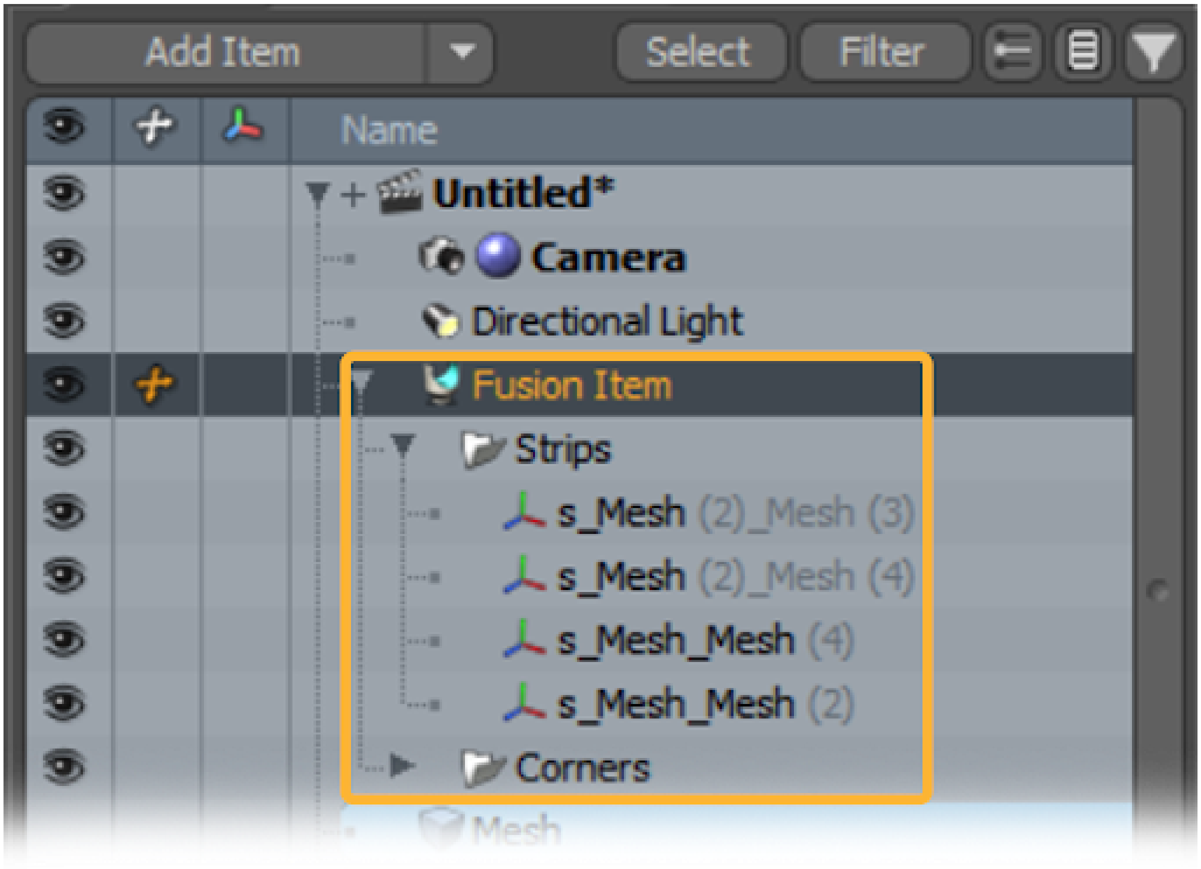
With **Edit Attribute** enabled, there are two ways to select Surface Strips:

- Double-click any strip on your mesh to select all strips in a fusion.

OR

- Select your **Fusion Item** in the **Items List** and expand it to select individual strips.

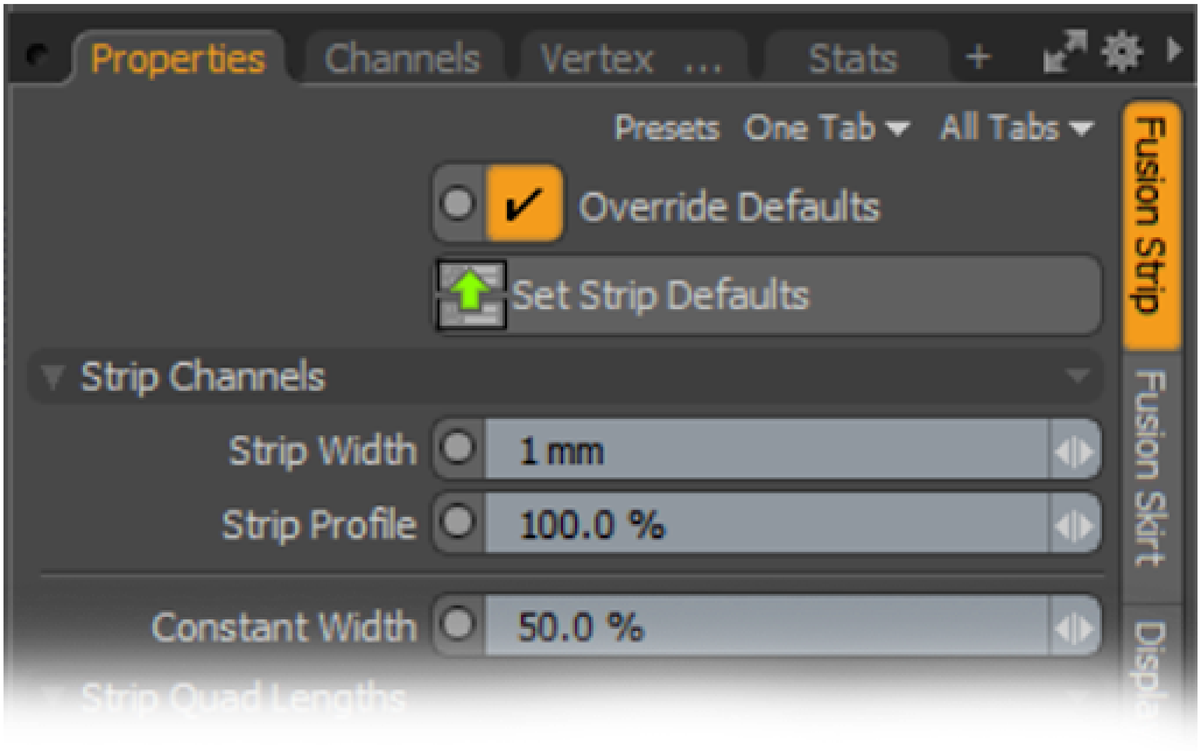
Once expanded, you will see two folders: **Strips** and **Corners**. Within the Strips folder, you can view all the strips that are a part of your mesh. From there you can individually select strips.


Add Item
Select
Filter
Name
Untitled*
Camera
Directional Light
Fusion Item
Strips
s_Mesh (2)_Mesh (3)
s_Mesh (2)_Mesh (4)
s_Mesh_Mesh (4)
s_Mesh_Mesh (2)
Corners
Mesh

Selected strips are highlighted in orange.

Note: For more information on corners, see Corner Strips and Skirts.

Selected strips can be edited in the **Fusion Strip** tab, which can be found in the **Properties** menu. Within the **Fusion Strip** tab, you are able to change the width, angle and density of your strip.


Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Fusion Strip
Fusion Skirt
Displa

☑ Override Defaults

Set Strip Defaults

Strip Channels

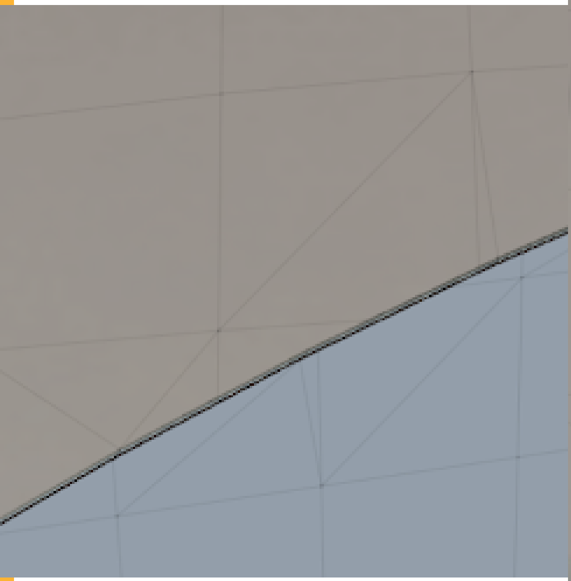
Strip Width 1 mm

Strip Profile 100.0 %

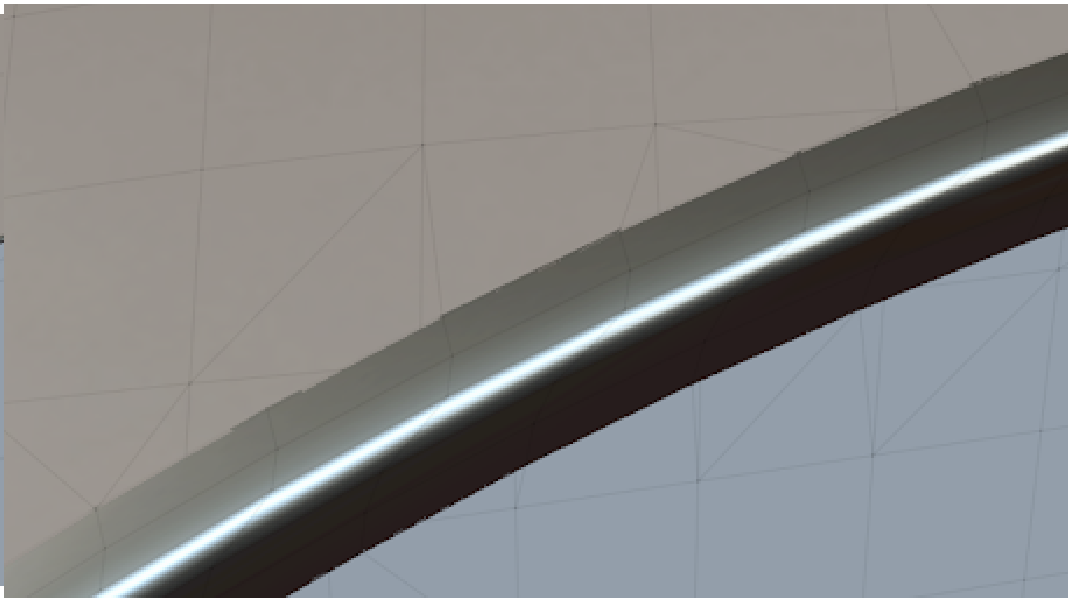
Constant Width 50.0 %

Strip Quad Lengths

Strip Width controls the width of the strip you have selected.



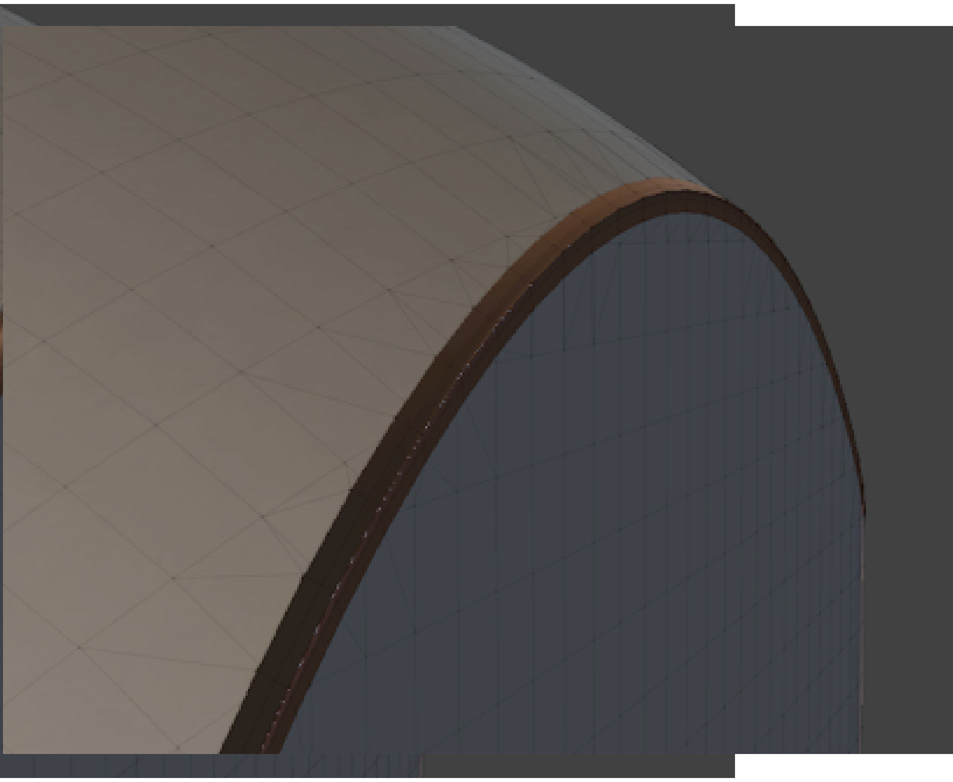
Strip set to default width.



Strip set to 30mm.

The **Strip Profile** parameter controls the roundness a strip. The lower the number, the flatter the curve and the higher the number the rounder the curve.

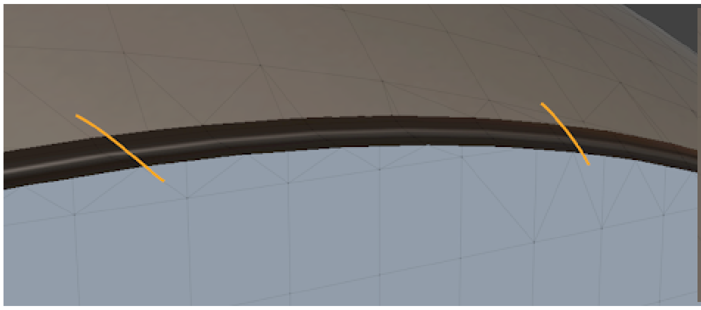
Profile set to Zero.



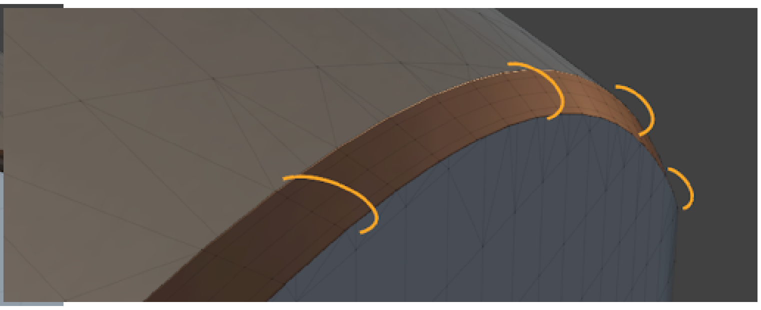
Profile set to 100.

Profile set to 200.

Tip: If you click and drag in the parameter boxes for **Strip Width** and **Strip Profile**, yellow guides will appear and give a real-time estimate of how flat/cured/wide your strip will become.

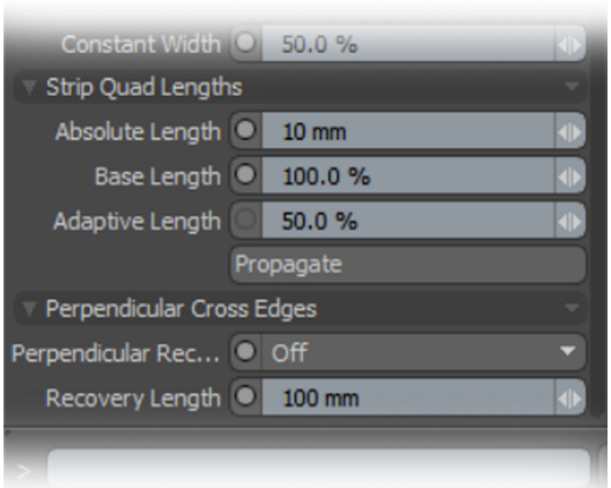


Guides for Strip Width.



Guides for Profile.

The **Strip Quad Lengths** parameters allow you to adjust the amount of quads present in the strips, as well as how close together they are. The lower the **Base Length** and **Adaptive Length**, the denser your strip will be. Higher **Base Length** and **Adaptive Length** values produce sparser strips.



Constant Width 50.0 %

Strip Quad Lengths

Absolute Length 10 mm

Base Length 100.0 %

Adaptive Length 50.0 %

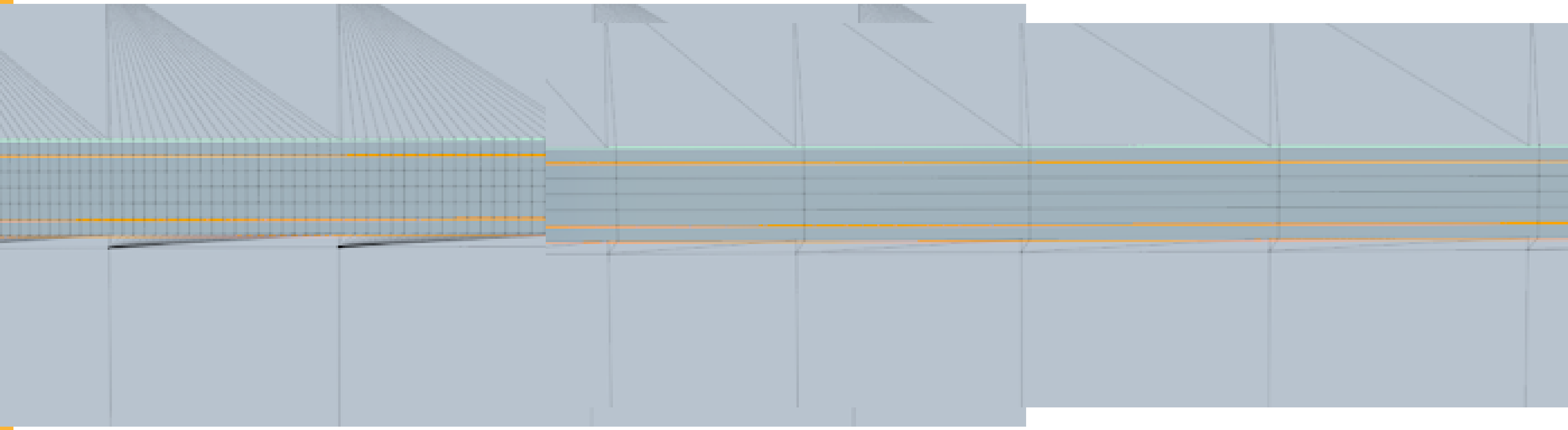
Perpendicular Cross Edges

Perpendicular Rec... Off

Recovery Length 100 mm

Strip Quad Lengths parameter menu

Absolute Length makes sure that the distance between each quad is the same throughout the whole strip.



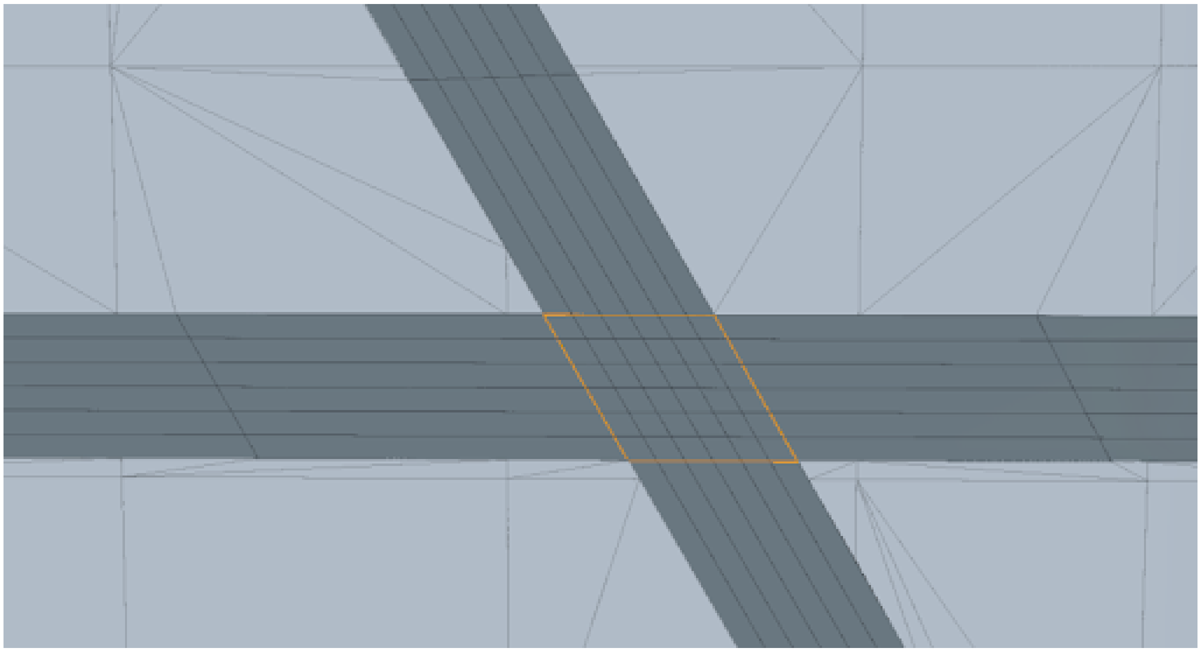
Base and **Adaptive Lengths** set to 5%.

Base and **Adaptive Lengths** set to 100%.

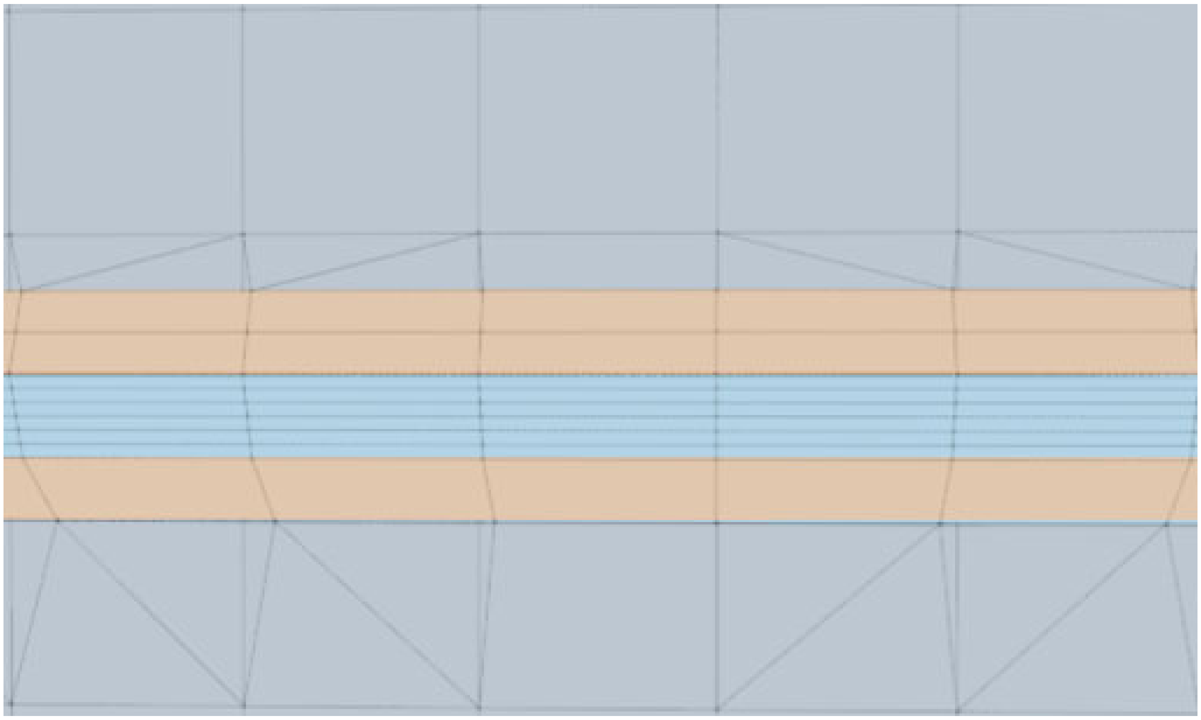
> **Note:** Edits to strips will not update unless **Override Defaults** is enabled. This can be found in the Fusion Strip tab in the Properties window.

Corner Strips and Skirts

Corner Strips are a type of surface strip created where two **Surface Strips** join or intersect. Corners are useful when editing the width of strips in specific areas of a mesh or increasing the density of a mesh at an intersection to allow for better topology on post-Fusion models.



Skirts are an additional piece of geometry that you can add to a Surface Strip to add extra width and ensure better topology for Post-MeshFusion modeling.

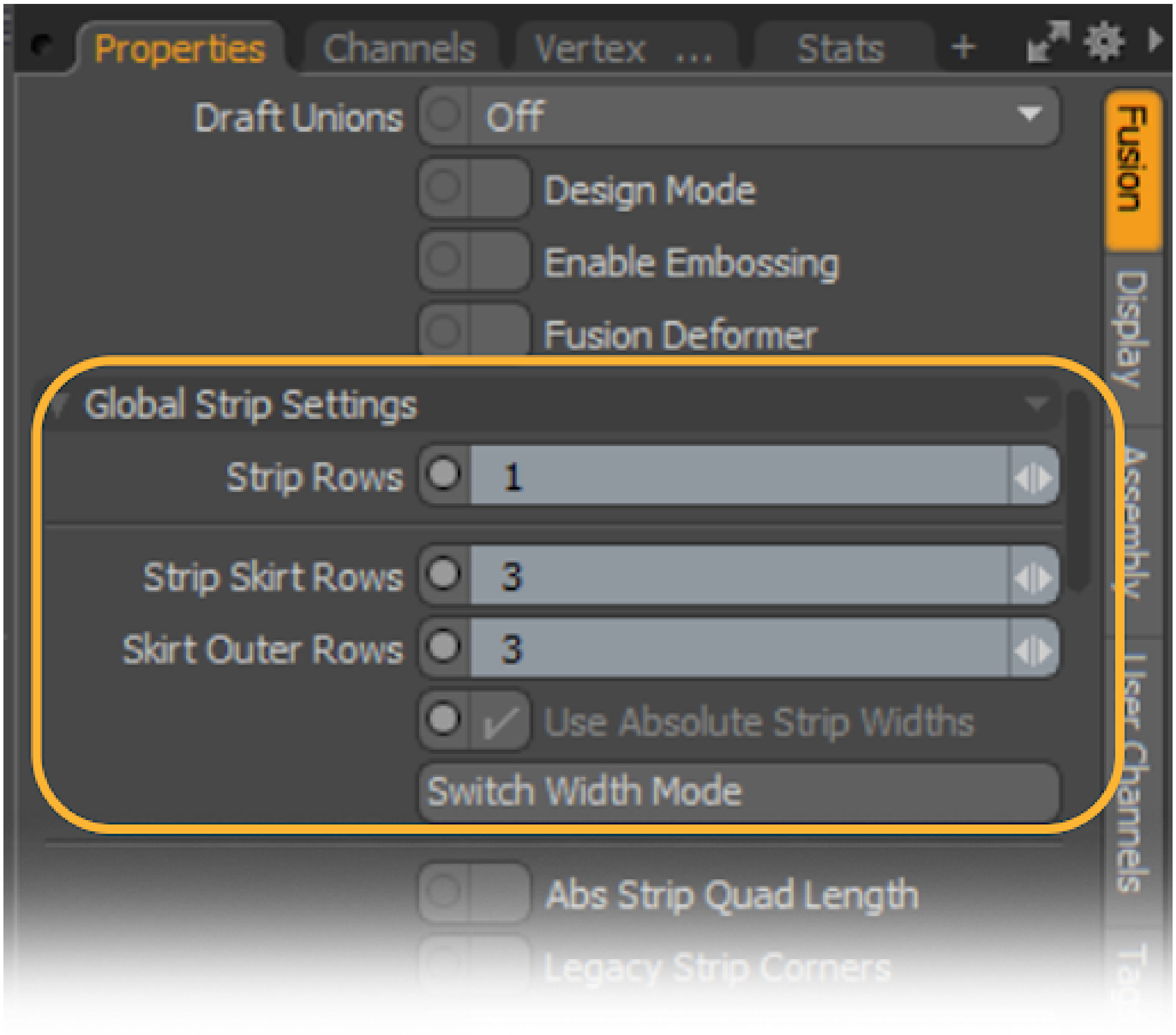


Surface Strip with a Skirt added to it. The skirt is highlighted in orange.

Adding Skirts

To add skirts to your strip, select your **Fusion Item** in the **Items List** so that the **Fusion** tab becomes available in the **Properties** window in the lower-left corner of your Modo window.

Within the **Fusion** tab in the **Properties** window, scroll down to the **Global Strips Menu** sub-menu. Here you can input how many **Skirt Rows** you want applied to your strip.



Draft Unions: Off

Design Mode

Enable Embossing

Fusion Deformer

Global Strip Settings

Strip Rows: 1

Strip Skirt Rows: 3

Skirt Outer Rows: 3

☑ Use Absolute Strip Widths

Switch Width Mode

Abs Strip Quad Length

Legacy Strip Corners

Editing skirts

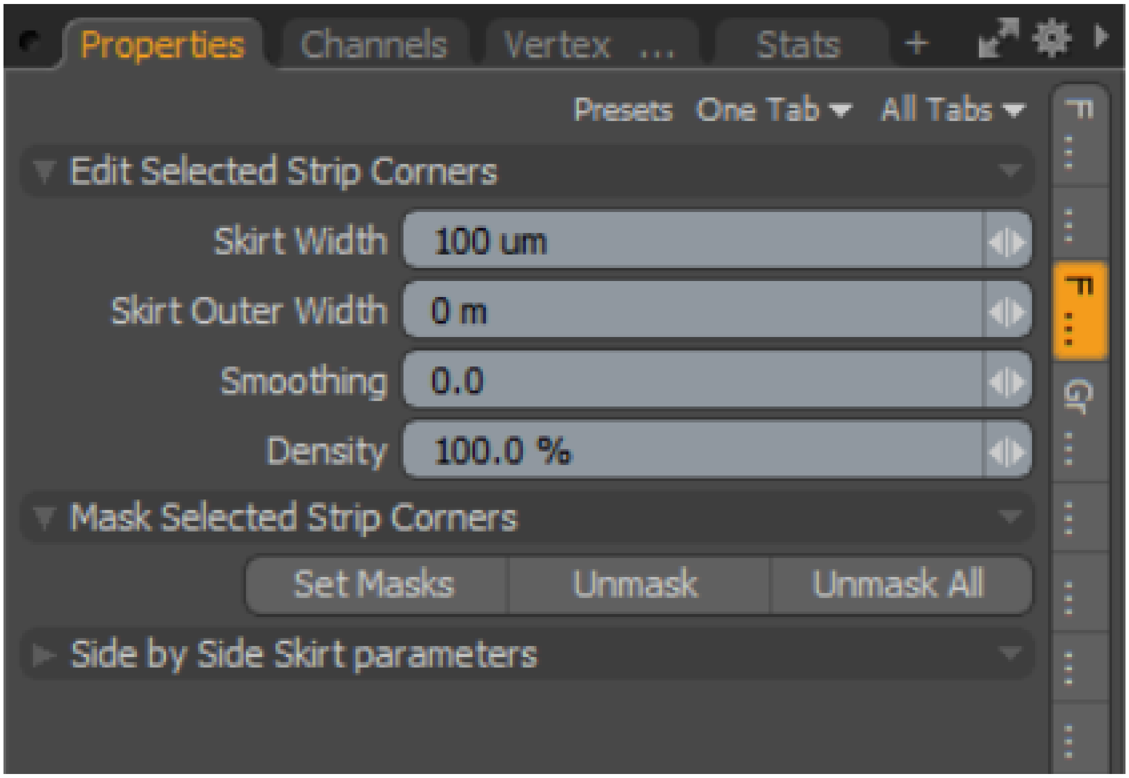
After you have your skirts set up, there are two ways you can edit them:

- If your Fusion item has any intersecting strips, select the corner of the strip to which you want to add a skirt.

OR

- Select the patch (an area of Geometry surrounded by strips) that you want to add a skirt to. Patches can be selected by double-clicking the patch. Once selected, they are highlighted in orange.

Once you have selected your patch or corner, navigate to the **Fusion Skirt** tab in the **Properties** window. In the **Fusion Skirt** tab you will be able to edit the **Width** and **Outer Width** of the skirts, as well as the skirt's **Smoothing** and **Density**.


Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Edit Selected Strip Corners

Skirt Width 100 um

Skirt Outer Width 0 m

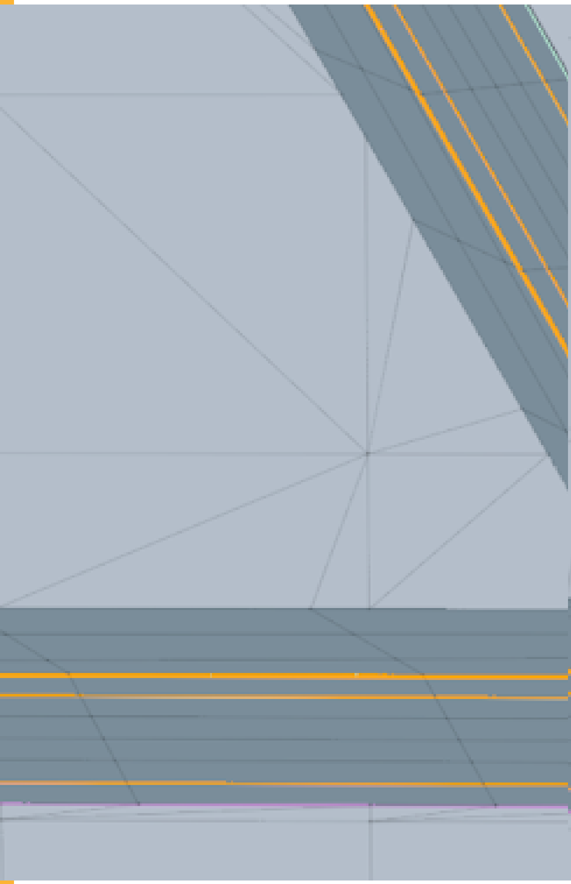
Smoothing 0.0

Density 100.0 %

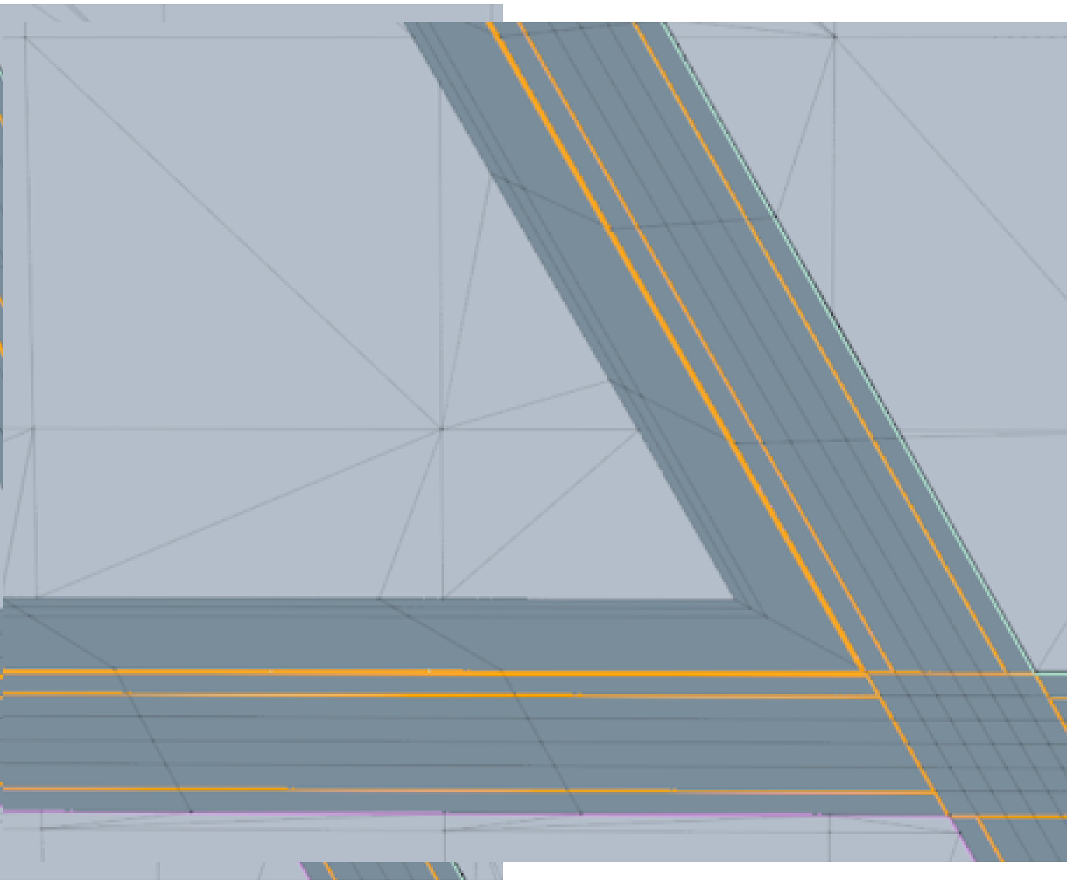
Mask Selected Strip Corners
Set Masks
Unmask
Unmask All
Side by Side Skirt parameters

Skirt Width allows you to edit the thickness of the skirt you have selected.

Within the skirt, two rows of Geometry will be produced. These help maintain edge integrity when the **Fusion Item** is converted to polygonal mesh and subdivided. **Skirt Outer Width** then allows you to set how far away from the outer edge of your skirt you want inner rows to be.

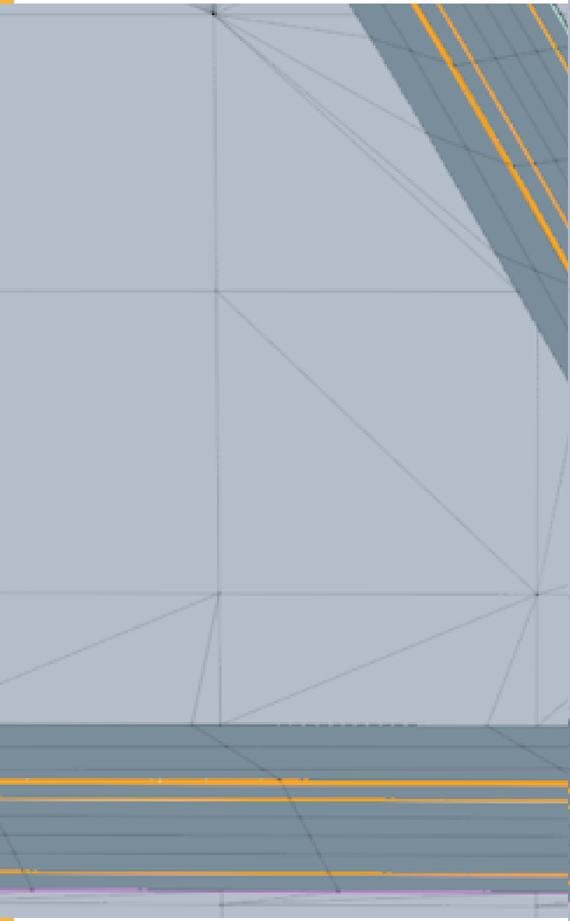


Skirt with Outer Width set to 15mm.

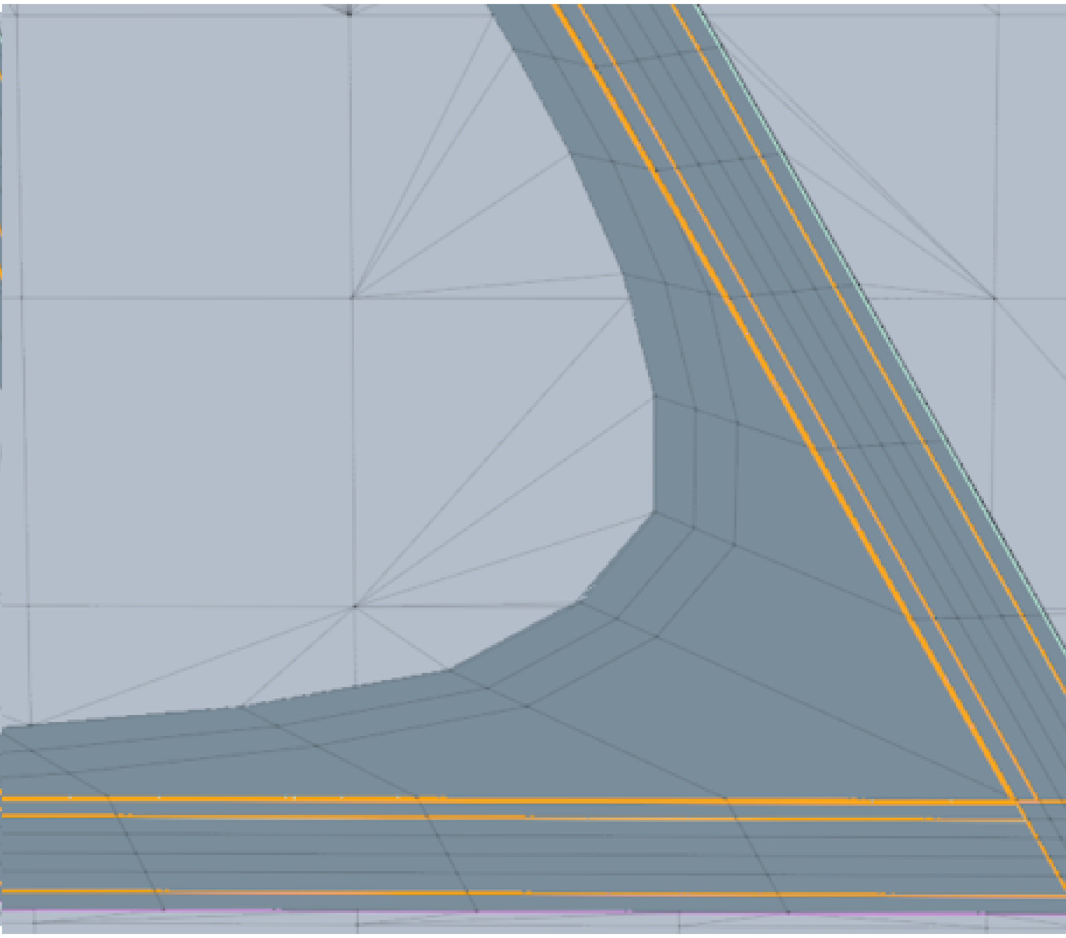


Skirt with Outer Width set to 5mm.

Smoothing allows you to reduce the sharpness of any corners or joints within your skirts.

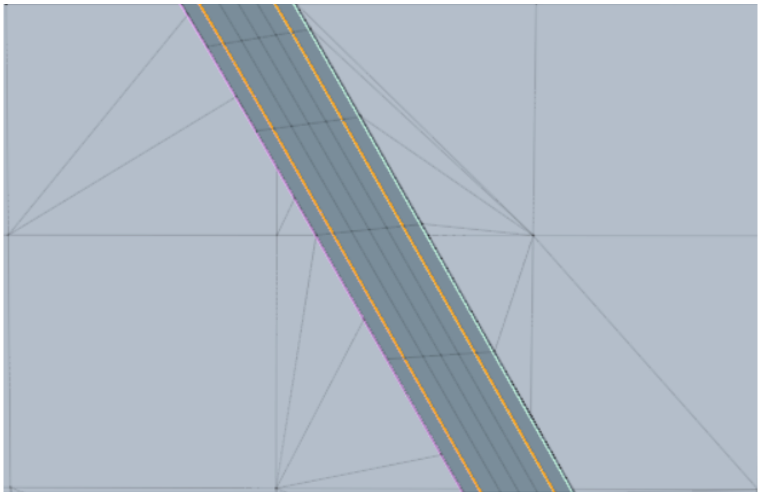


Smoothing set to 0.



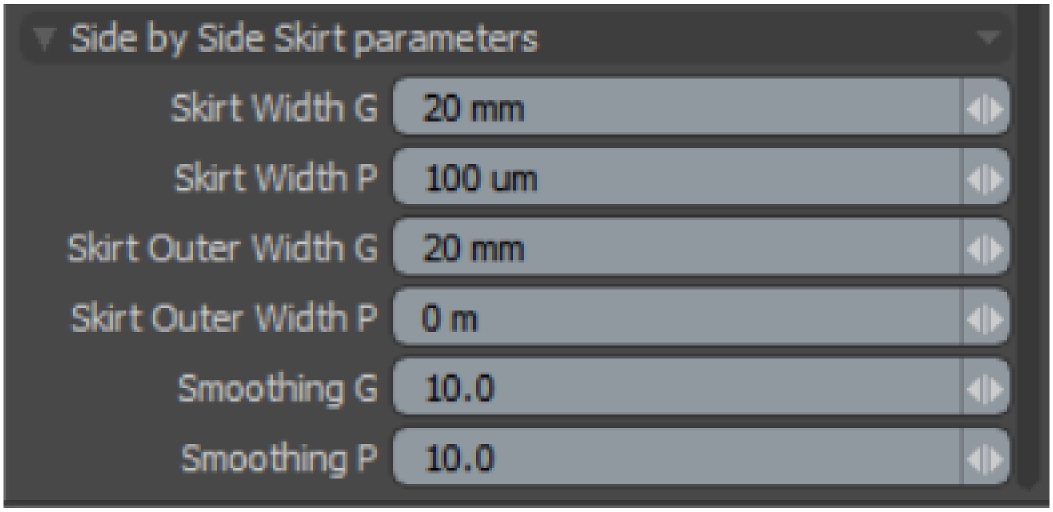
Smoothing set to 10.

Density controls how much or how little polygon density you have in your skirt and strip. As with **Adaptive** and **Base Length** in the **Fusion Strips** tab, the higher the number, the fewer polygons there are and the lower the number, the more there are.



When selected, each strip has a green side and a purple side. Within the **Side by Side Skirt Parameters** menu each side is indicated by **P** for Purple, or **G** for Green. For more in depth editing of skirts, you can also use the Side by Side Skirt Parameters menu.

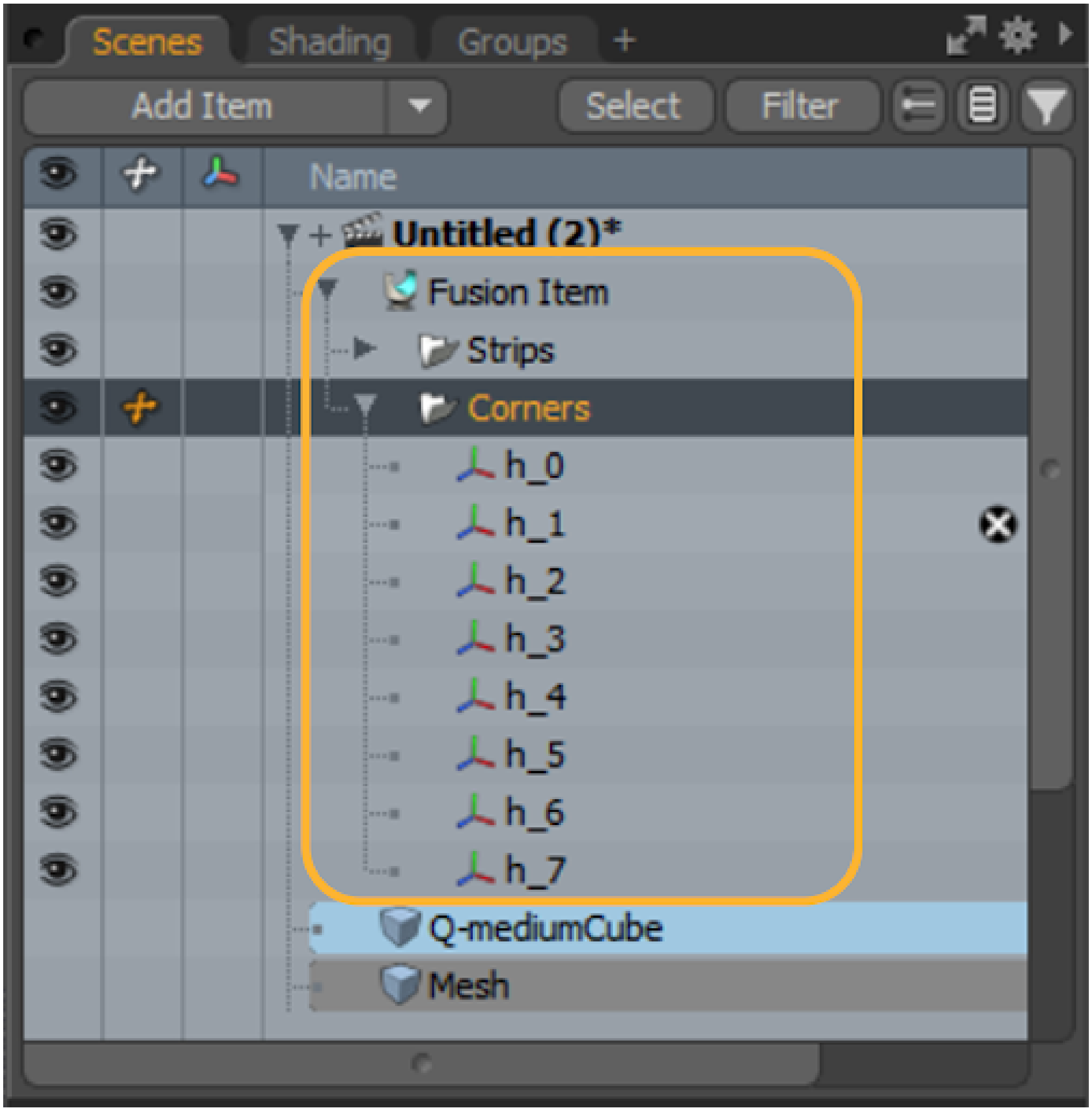
Use the **Side by Side Skirt Parameters**, to change the **Width**, **Outer Width** and **Smoothing** of each side. These parameters are the same as the ones covered earlier on in this topic, but instead of affecting the whole skirt only the side you are editing will change.



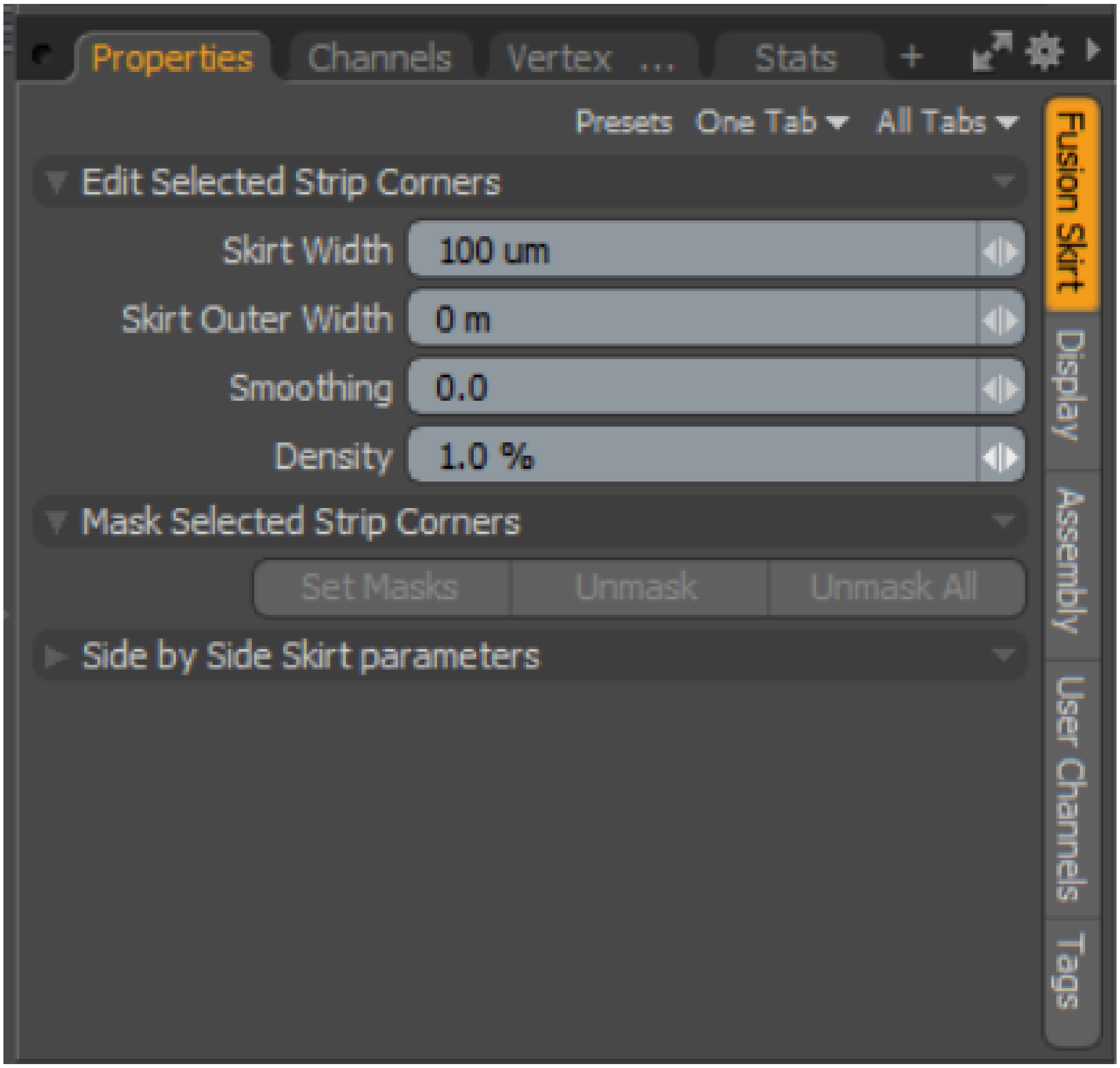
Skirt Width G: 20 mm
Skirt Width P: 100 um
Skirt Outer Width G: 20 mm
Skirt Outer Width P: 0 m
Smoothing G: 10.0
Smoothing P: 10.0

Editing Corner Strips

Corner strips are selected in the same way as surface strips are, but to select individual corners, navigate through the **Corners** folder instead of the Strips folder. Selected corner strips are outlined in orange.


Scenes
Shading
Groups
Add Item
Select
Filter
Name
Untitled (2)*
Fusion Item
Strips
Corners
h_0
h_1
h_2
h_3
h_4
h_5
h_6
h_7
Q-mediumCube
Mesh

Once you have a corner selected, the **Fusion Skirt** menu will is displayed in the **Properties** window.


Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Edit Selected Strip Corners
Mask Selected Strip Corners
Set Masks
Unmask
Unmask All
Side by Side Skirt parameters
Fusion Skirt
Display
Assembly
User Channels
Tags

Skirt Width 100 um

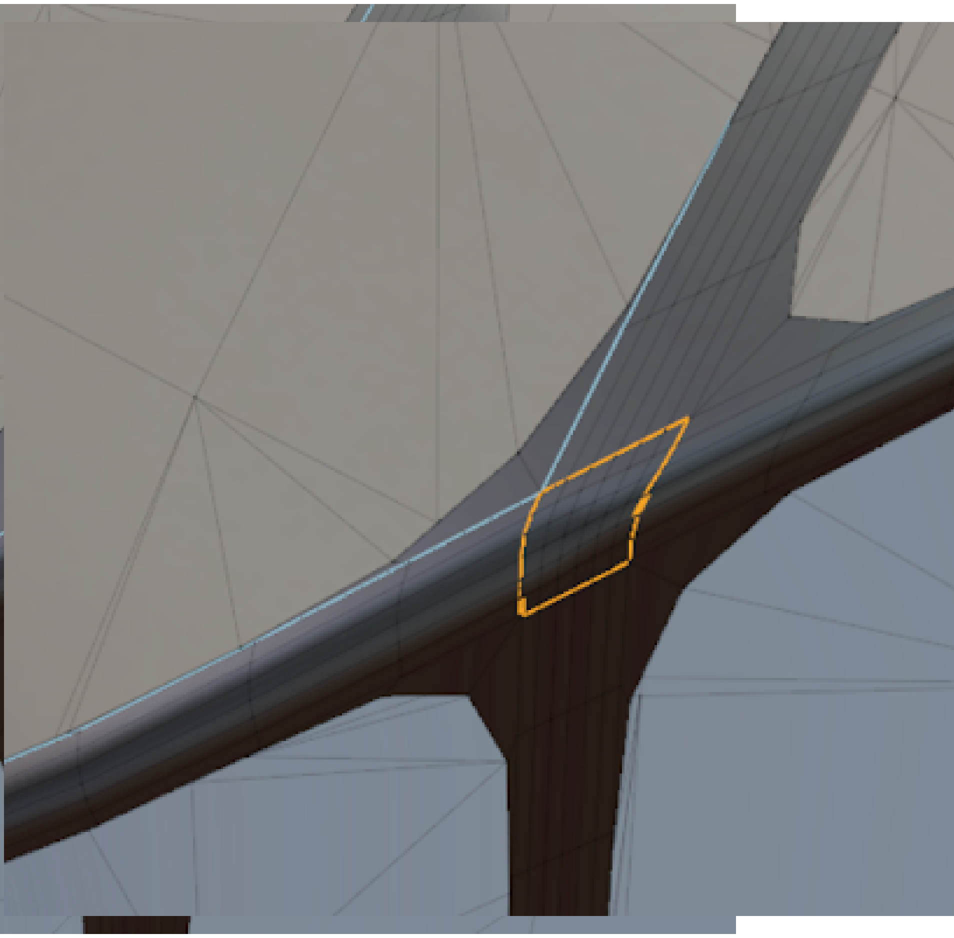
Skirt Outer Width 0 m

Smoothing 0.0

Density 1.0 %

Skirt Width will increase the widths of any **Surface Strips** that are connected to the corner, while smoothing will soften or sharpen the edges where two or more strips intersect.

However, corner smoothing must be used in conjunction with skirts and **Skirt Rows** must be added for it to work.

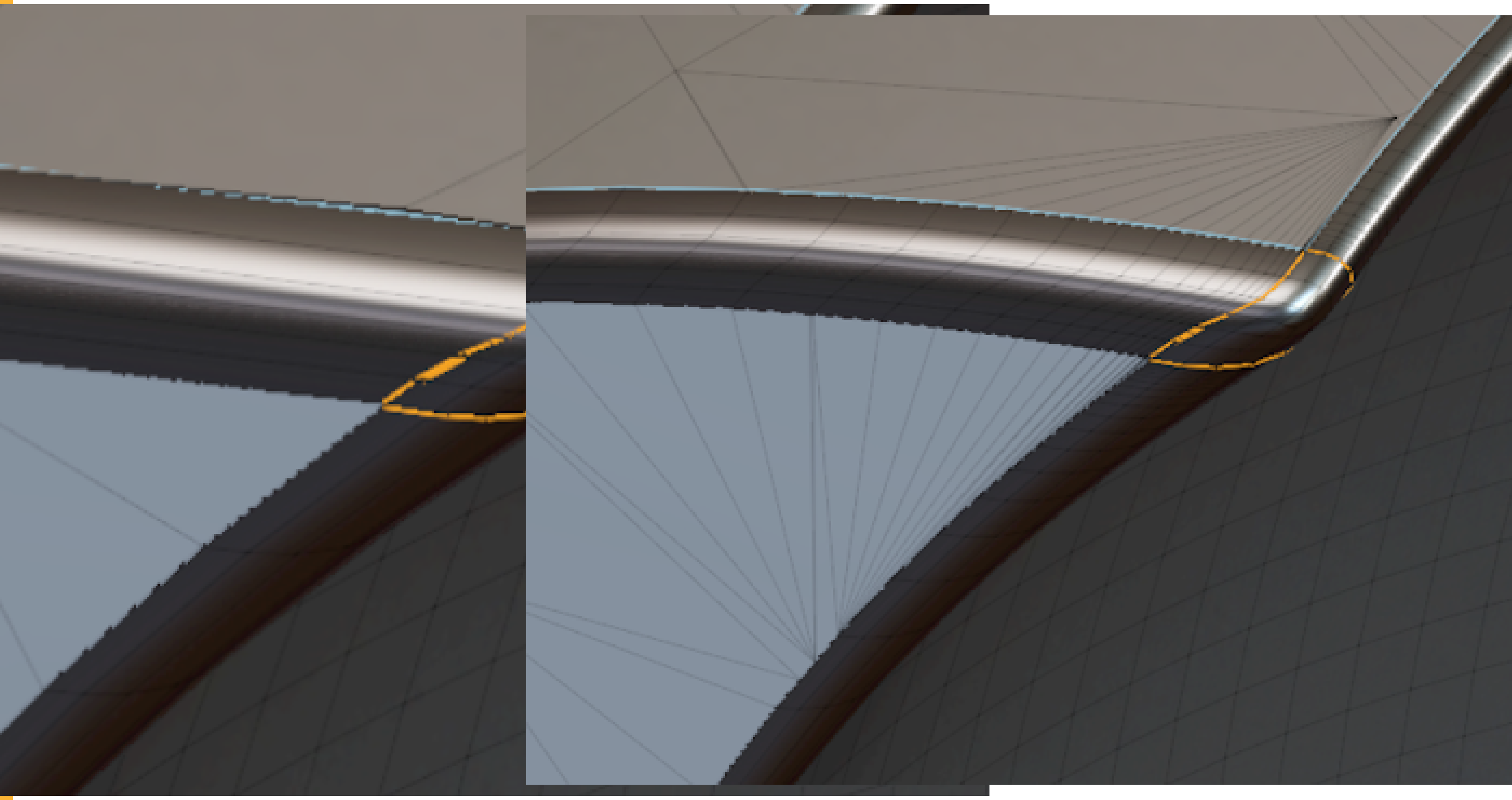


No **Smoothing**.

Smoothing with no **Density**.

Smoothing with **Density**.

Density changes how many polygons surround the outside of the corner. This tool is helpful for creating cleaner topology in a post-fusion model and is also useful for creating tighter corners when increasing **Smoothness**.



Density set to 100%.

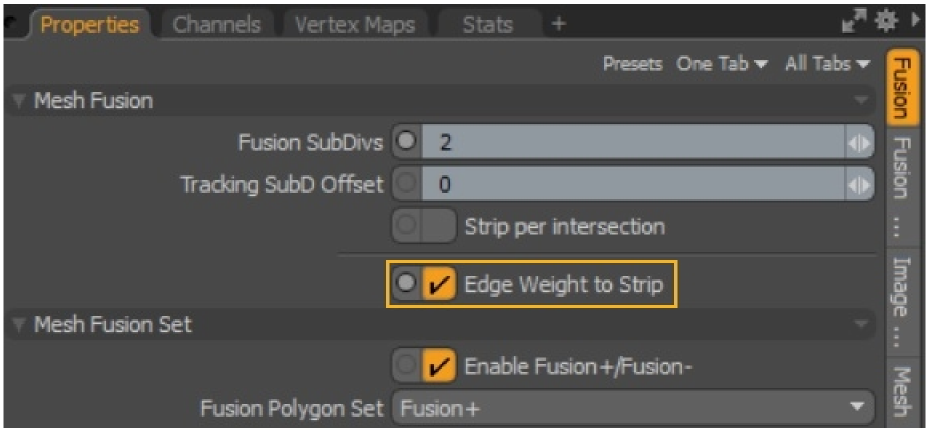
Density set to 1%.

Converting Edge Weights to Strips

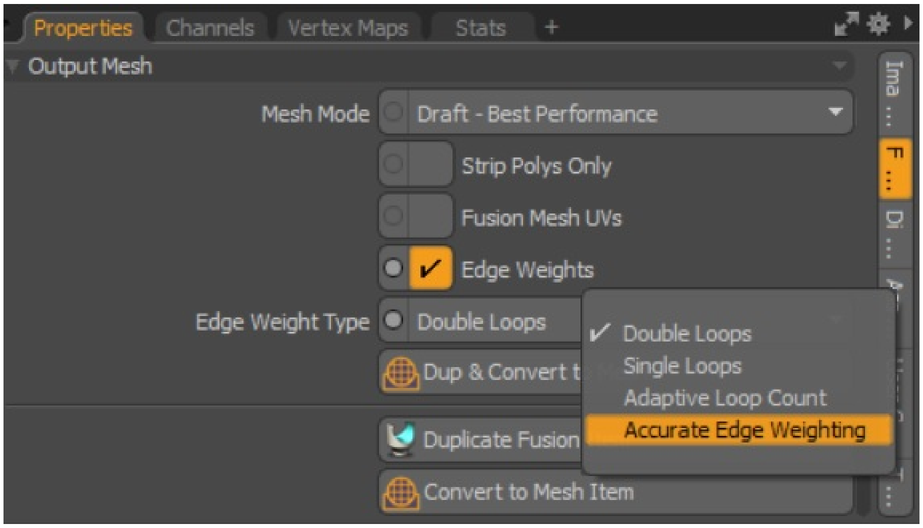
In addition to surface strips, Modo can convert edges with 100% weight values to strips automatically. You can use the edge weights on sharp creases to achieve the same strip width and profile as the strips at intersections between different source meshes. There are a number of restrictions on edges you can convert to strips:

- The selected edges must continue all the way to the end of the mesh, they cannot be open-ended.
- The edge weighting must be set to 100%, any other value is not converted to strips.
- The edges can not cross each other and the edges must have a noticeable crease, either a ridge or a gully.
- The edge ends must be “capped” by another fusion source item.

1. Select the edge or edges you want to convert to strips in the viewport.
2. Assign the edges a weight value of 100%.
   You can quickly assign a weight value by pressing **Shift**+**W** and entering 100% in the **Vertex Map Weight** viewport **Weight** control.
3. Enable **Edge Weight to Strip** in the source item **Fusion** tab.


Properties
Channels
Vertex Maps
Stats
+
Presets
One Tab
All Tabs
Mesh Fusion
Fusion SubDivs
2
Tracking SubD Offset
0
Strip per intersection
Edge Weight to Strip
Mesh Fusion Set
Enable Fusion+/Fusion-
Fusion Polygon Set
Fusion+
Fusion
Fusion ...
Image ...
Mesh

4. In the **Output Mesh** properties, enable **Edge Weights** and set the **Edge Weight Type** to **Accurate Edge Weighting**.


Properties
Channels
Vertex Maps
Stats
+
Output Mesh
Mesh Mode
Draft - Best Performance
Strip Polys Only
Fusion Mesh UVs
Edge Weights
Edge Weight Type
Double Loops
Dup & Convert t
Duplicate Fusion
Convert to Mesh Item
✓ Double Loops
Single Loops
Adaptive Loop Count
Accurate Edge Weighting
Ima ...
F ...
Di ...

The weighted edges are converted to strips and you can control them as you would any other Fusion strip.

Adjusting the Start and End Width of Strips with Variable Strip Widths

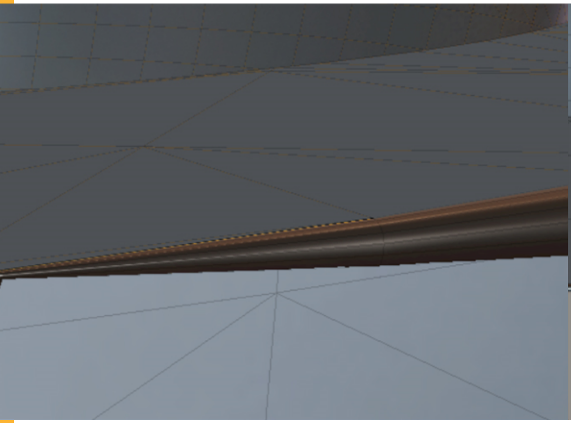
You can also adjust the start and end width of your Fusion Strips. This allows you to get final control over how sharp or beveled your edges are, with a smaller start or end value causing a sharper edge and a larger start or end width creating a rounded, or beveled, edge.

To use Variable Strip Widths:

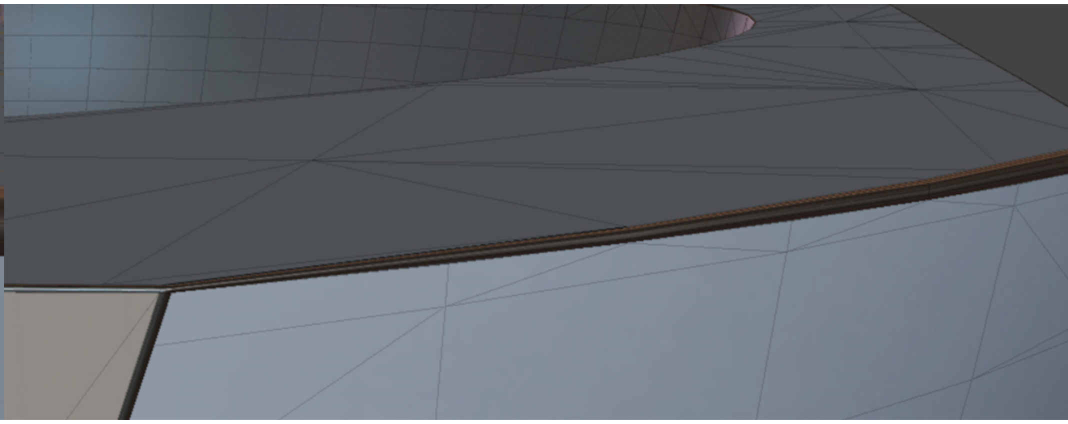
1. Select the Strip you want to change the Variable Width of
2. In the **Fusion Strip** tab in the **Properties** panel, check the **Tweak Variable Strip Widths** option

   Once enabled, **Start Strip Width** and **Strip End Width** parameters become available
3. Adjust the **Start Strip Width** and **Strip End Width** parameters as needed

When adjusting these parameters, keep in mind that the smaller the width of the strip is, the sharper you edge will be while the wider the strip is, the more rounded it will be. This allows you to create a gradated edge on models.



Strip with **Start Strip Width** set to 16 mm while **End Strip Width** is set to 1 mm



Strip with **Start Strip Width** set to 5 mm while **End Strip Width** is set to 1 mm

Embossing Surface Strips with MeshFusion

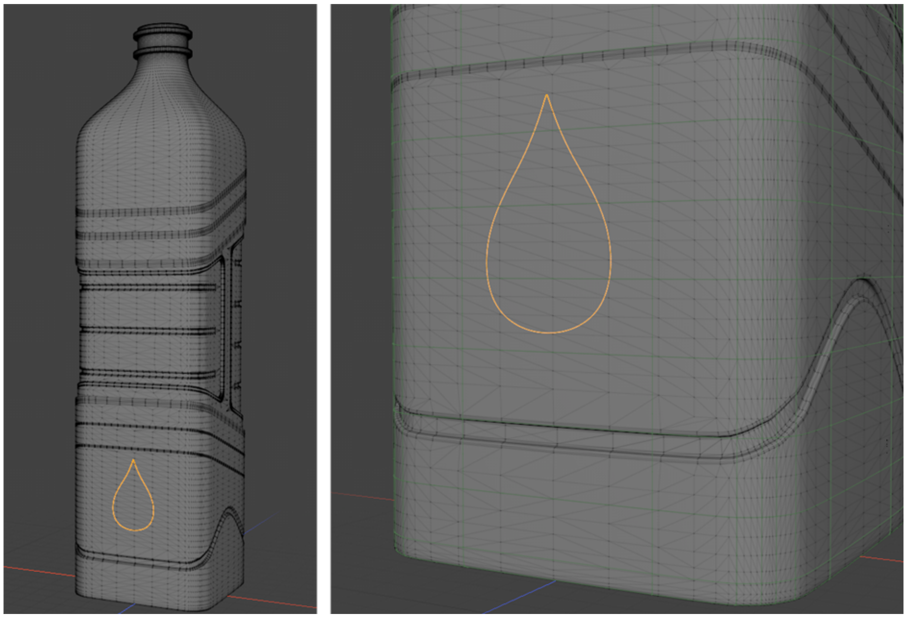
You can create embossed strips on Fusion Items using Curves, Beziers, B-Splines, or text.

Creating Embossed Strips with Flat Curves

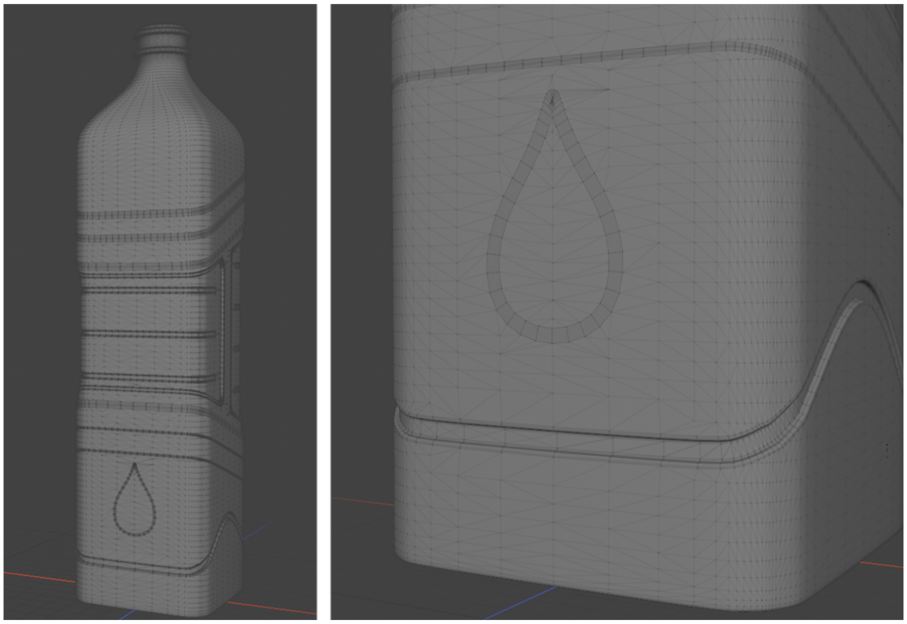
To create embossed strips using curves, you need to create surface strips on a Primary mesh using a Trim mesh.

Note: To learn more about mesh roles in MeshFusion, see Mesh Roles and Relationships.

1. Add a mesh and create a new Fusion Item, as described in Creating a New Fusion Item.
   Your mesh item becomes a primary and it's marked with green in the Item List and the 3D viewport. A Fusion Item is also created.
2. To produce a flat curve in an ideal position for creating surface strips, place the cursor in the center of your primary mesh and press **Alt**+**O** on the keyboard. This moves the Work Plane to that location, and your curves will be drawn in that position.
3. In a new mesh layer, create the geometry you want to use as the trim for the surface strip.
   The following example uses a shape drawn with a B-Spline, but any of the standard Curve tools can be used (Curve, Bezier).



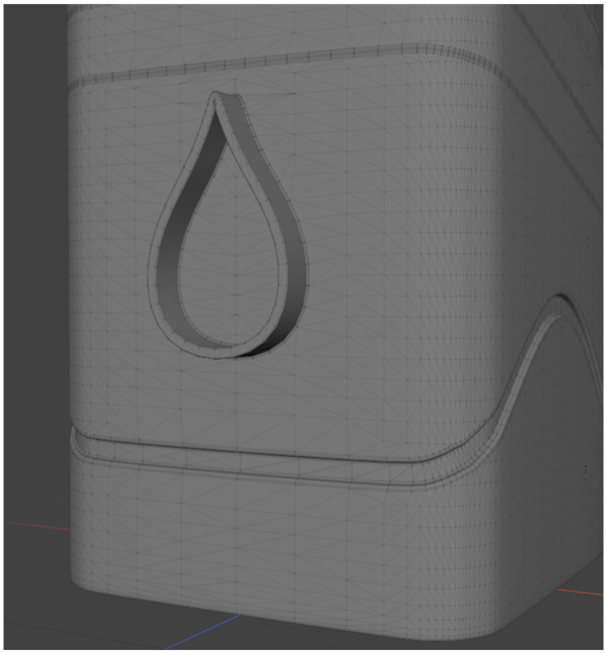
4. In the Item List, select your Curve mesh and Primary mesh. Then in the modeling toolbar's **Fusion** sub-tab on the left, under **Apply Surface Strip**, click the Full Strip with Grid Crossing button.
   Surface strips are created on your Primary mesh, and your Trim mesh is marked with pink in the Item List.



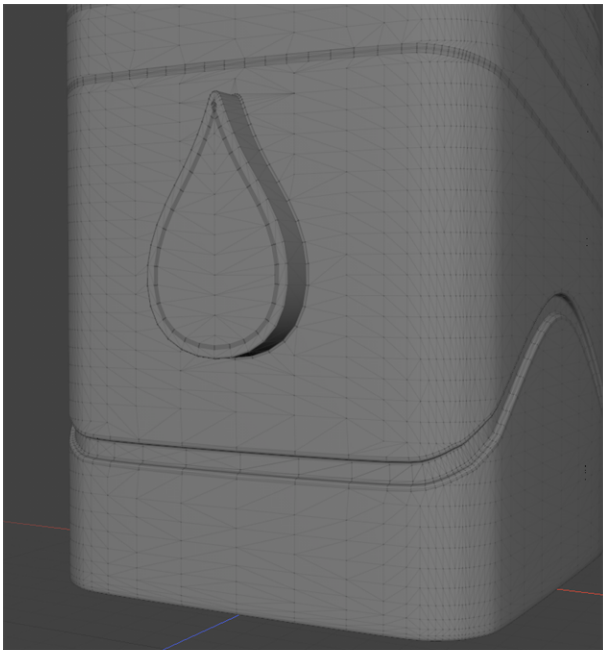
Note: If you haven't edited the **Strip Width** when creating the Fusion Item, the default 1mm is used. To increase the width: select your Trim mesh in the Item List to open its **Properties**, then adjust the **Mesh Fusion** > **Surface Strip Absolute Width** value. In the example above, strip width is 2mm.

If your Fusion Item is set to use relative strip width, this property has no effect on the strip width, instead, it is controlled by the Fusion Item property **Default Strip Settings** > **Strip Width**.

5. In the Items list, select your Fusion Item to open its **Properties** panel, and under **Fusion** properties, enable **Enable Embossing**.
6. Select the Trim mesh in the Item List to display its **Properties** panel.
7. Under **Mesh Fusion Embossing** properties, enable **Enable Embossing** and edit the **Outward**, **Inward**, and **Middle Offset** values. The embossing is now visible in the 3D viewport.

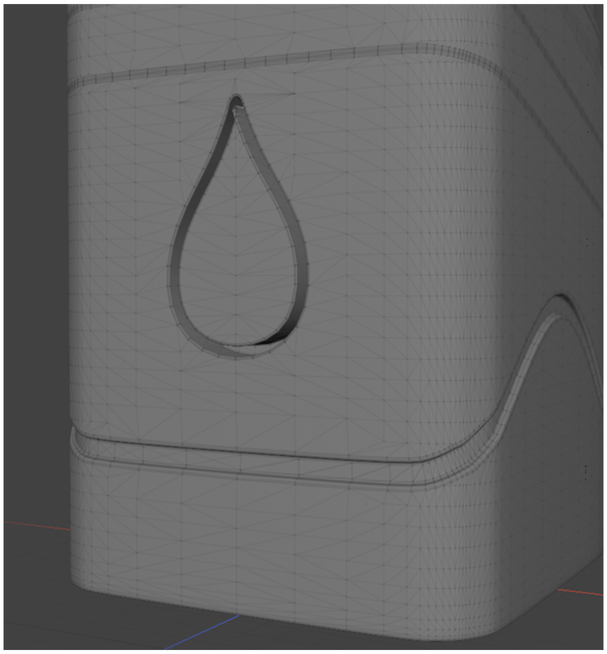


Middle Offset: 2mm

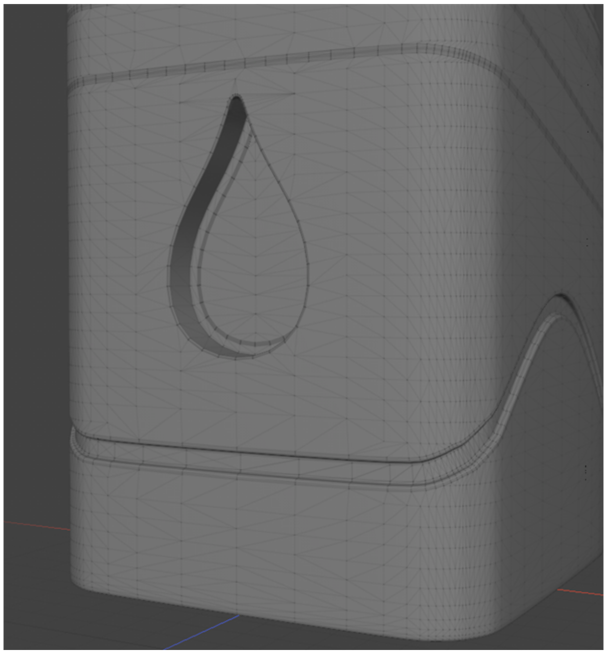


Inward and **Middle Offset**: 2mm

A negative offset will emboss in the other direction.



Middle Offset: -2mm



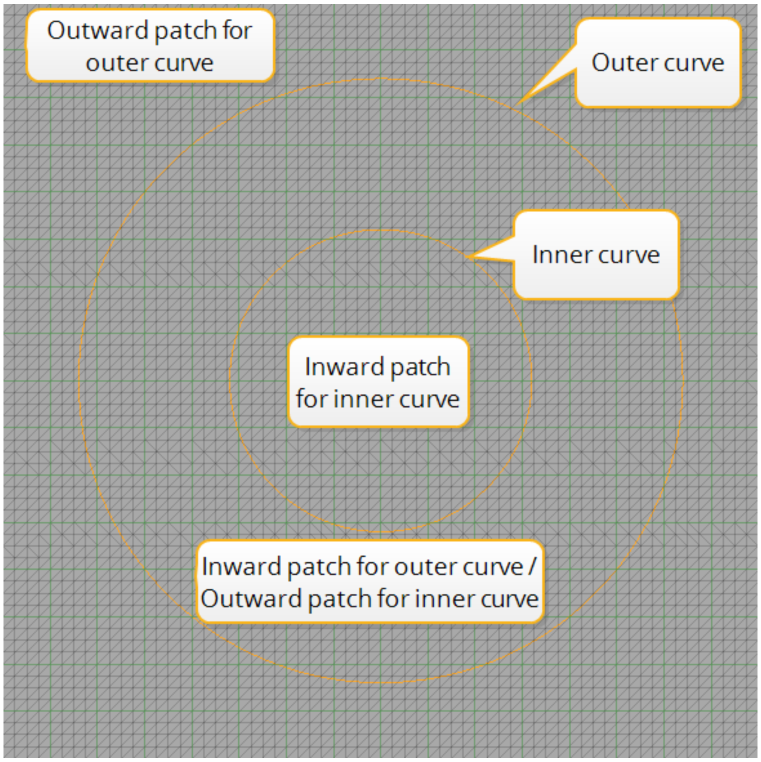
Inward and **Middle Offset**: -2mm

Tip: When working with Bezier curves, you might have sharp corners on your curve. In such cases, enable **Sharp Bezier Corners** to produce clean strips for embossing.

Note: If the outward patch interacts with another regular strip as well as the embossed one, you must leave **Outward Offset** at 0, as offsetting a regular strip is not currently supported and would produce unexpected results.
Intersections between embossed surface strips and regular surface strips are not currently supported either.

Working with Curves within Curves

When creating embossing using one loop within another, the direction of the curve is important in specifying the outward and inward patches. When both curves are drawn in the same direction, the inward area for the outer curve is the outward area for the inner curve. This will produce unexpected results for embossing.


Outward patch for outer curve
Outer curve
Inner curve
Inward patch for inner curve
Inward patch for outer curve / Outward patch for inner curve

Draw your outer curve clockwise, and your inner curve counter-clockwise. Alternatively, you can flip your curves after the fact. To do this, in **Polygons** selection mode, select your curve, and press **F** on the keyboard. This flips the direction of the curve.

Note: When drawing loops side by side, make sure they are drawn in the same direction, so their outward and inward patches match.

Creating Embossed Strips with Non-Flat Curves

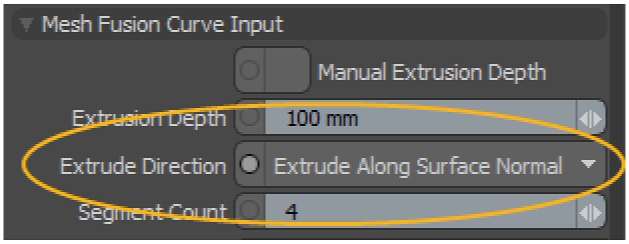
Instead of drawing your curve on the Work Plane, you might want to draw them on the primary surface. When you do this, curves are not flat, but follow the surface of the object, going around it.

Note: Make sure you don't have any intersecting curves.

1. Add a mesh and create a new Fusion Item, as described in Creating a New Fusion Item.
   A Fusion Item is added to the Items list and your primary mesh is marked with green.
2. In an empty mesh layer, enable **Mesh Constraint**.
   When enabled, the **Tool Properties** appear on the lower left.
3. In the **Tool Properties**, enable **Handle Constraint** and set **Geometry Constraint** to **Point**.
4. Select your empty mesh layer and draw a curve that you will use as the trim for the surface strip.

   Note: Your Primary mesh must be visible in the viewer for the constraint to work correctly.

5. In the Item List, select your Curve mesh and Primary mesh. Then in the modeling toolbar's **Fusion** sub-tab on the left, under **Apply Surface Strip**, click the Full Strip with Grid Crossing button.
   Surface strips are created on your Primary mesh, and your Trim mesh is marked with pink in the Item List.
6. Select the Trim mesh in the Item List to display its **Properties** panel, and set the following:
   - Set the **Extrude Direction**. For non-flat curves, the default **Extrude Perpendicular to Curve** option doesn't produce correct results. Select your curve mesh, and in the **Properties** panel, under **Mesh Fusion Curve Input**, set **Extrude Direction** to **Extrude Along Surface Normal**.



Mesh Fusion Curve Input

☐ Manual Extrusion Depth

Extrusion Depth: 100 mm

Extrude Direction: Extrude Along Surface Normal

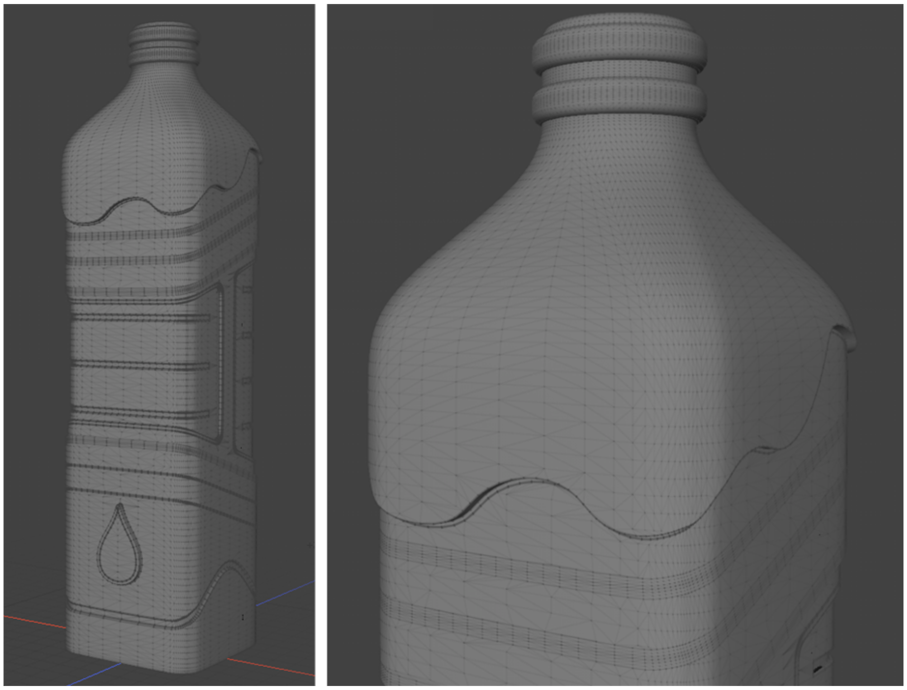
Segment Count: 4

   - Under **Mesh Fusion Embossing** properties, enable **Enable Embossing** and edit the **Outward**, **Inward**, and **Middle Offset** values.

Note: If the outward patch interacts with another regular strip as well as the embossed one, you must leave **Outward Offset** at 0, as offsetting a regular strip is not currently supported and would produce unexpected results.
Intersections between embossed surface strips and regular surface strips are not currently supported either.

- When working with Bezier curves, you might have sharp corners on your curve. In such cases, enable **Sharp Bezier Corners** to produce clean strips for embossing.

Setting **Inward Offset** to 2mm gives you the following result:



Using the Fusion Deformer

Fusion Deformer

The Fusion Deformer workflow allows you to deform flat curves to follow curved surfaces.

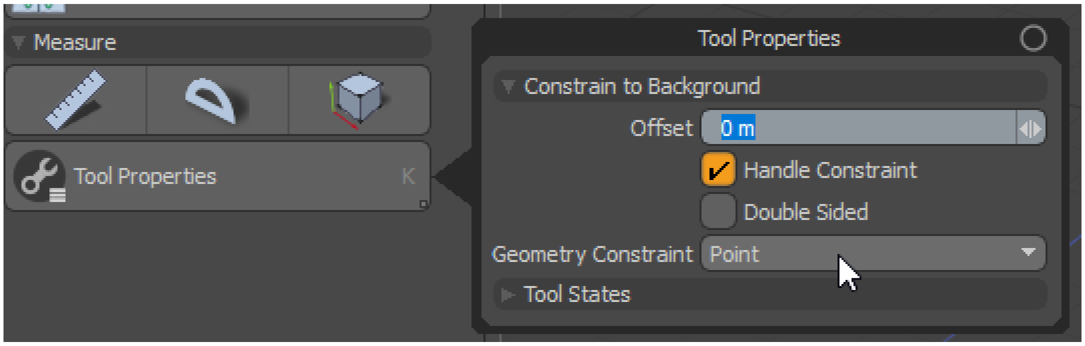
First, a flat 2D curve is deformed to become a 3D curve embedded into a curved surface. Then the 3D curve is used to create embossing on that surface, the same way as when the 3D curve is drawn on the surface manually, as described in Creating Embossed Strips with Non-Flat Curves.

This method is particularly suitable for slapping text strings onto curved surfaces, or deforming 2D drawings.

Creating the Strip

First, create a regular surface strip setup:

1. Add a mesh and create a new Fusion Item, as described in Creating a New Fusion Item.
   Your mesh item becomes a primary and it's marked with green in the Item List and the 3D viewport. A Fusion Item is also created.
2. Hide the Fusion Item by clicking its Visibility 👁 button in the Item List, to make it easier to see the curve you're going to draw.
3. In an empty mesh layer, enable **Mesh Constraint**, then in the **Tool Properties** on the lower left, enable **Handle Constraint** and set **Geometry Constraint** to **Point**. This is to make sure that the points of the curve don't fall too far away from the mesh for strip embossing to work.



Tool Properties

Constrain to Background

Offset 0 m

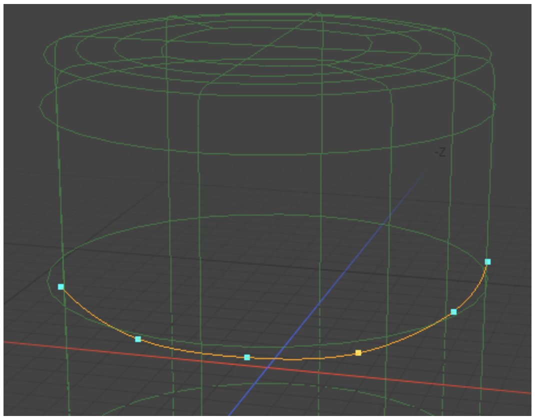
☑ Handle Constraint

☐ Double Sided

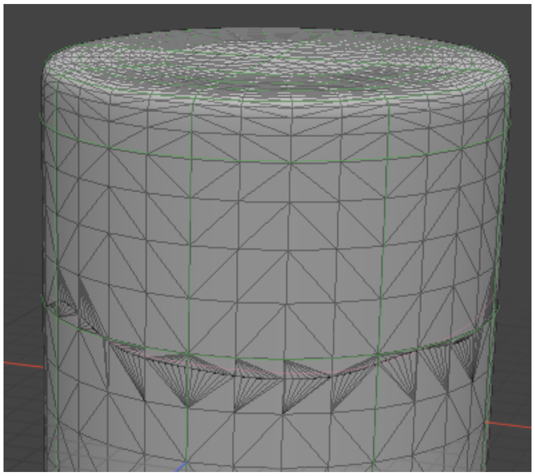
Geometry Constraint Point

Tool States

4. Draw an open curve in the empty mesh layer that you will use as the trim for the surface strip. You can use any of the standard Curve tools (Curve, Bezier, or B-Spline). For more information, see Adding Curves.



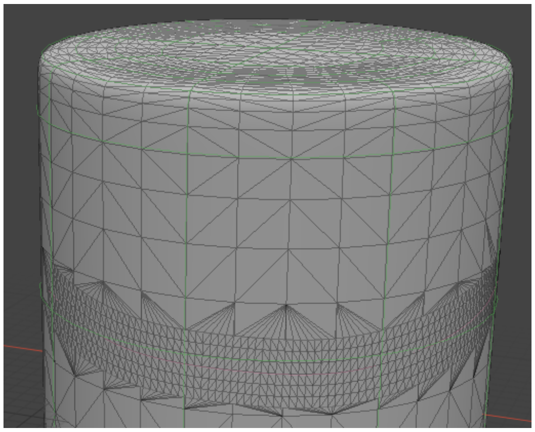
5. Drop the curve tool, disable **Mesh Constraint**, and make the Fusion Item visible again.
6. In the Item List, select your curve mesh and Primary (green) mesh, then under **Apply Surface Strip**, click the Full Strip with Grid Crossing button.
   Surface strips are created on your Primary mesh, and your Trim mesh is marked with pink in the Item List.



7. Now you have to make some modifications to how the strip is created. Select your curve mesh to open its **Properties**, and under **Mesh Fusion Curve Input** > **Extrude Direction**, select **Extrude Along Surface Normal**.

> **Note:** The default **Extrude Perpendicular to Curve** option produces incorrect results with non-flat curves.

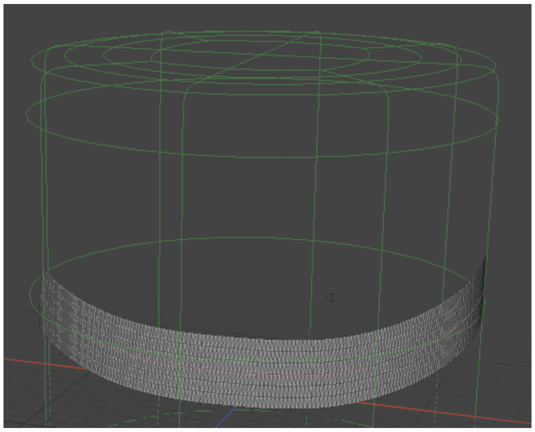
8. Set the width of the surface strip. Still in the curve mesh **Properties**, under **MeshFusion** > **Mesh Properties**, set the **Surface Strip Absolute Width**.
   Make sure that the strip is wide enough to fit the text string or drawing you are going to emboss. You can always change the width later. The example below uses a strip width of 180mm.



9. The strip mesh must be dense and uniform along the strip. To achieve this, select your Fusion Item to open its **Properties**, and under **Global Strip Settings**, enable **Abs Strip Quad Length**.
   The default **Strip Length Abs** of 10 mm is fine for models of about 1 meter size. Generally, about 1/100 of the model size works well.

> **Note:** Wider strips tend to develop kinks where the guiding curve makes a turn. If the strip mesh self-overlaps, this may create problems at later stages. To smooth out kinks, increase the value in **Default Strip Settings** > **Strip Smoothing** in the Fusion Item **Properties**.

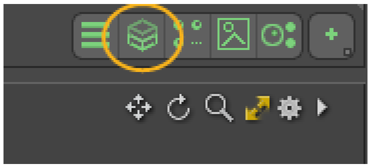
> **Tip:** You can declutter your scene by enabling **Fusion Mesh** > **Strip Polys Only**, since here we only care about the surface strip mesh, not the surrounding primary surface mesh.



Adding Text

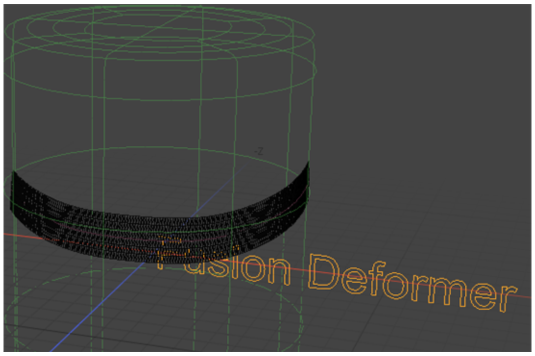
Once the strip is ready, it's time to create the flat 2D curves that will be deformed by the strip. This example uses Bezier curves created by the Text mesh operation.

To use the Text mesh operation, open the **Mesh Operations** list. In the Modo layout, you can do this using the top viewport controls. Click on the bar above the 3D viewport to reveal the controls, then click the Mesh Operations viewport button:

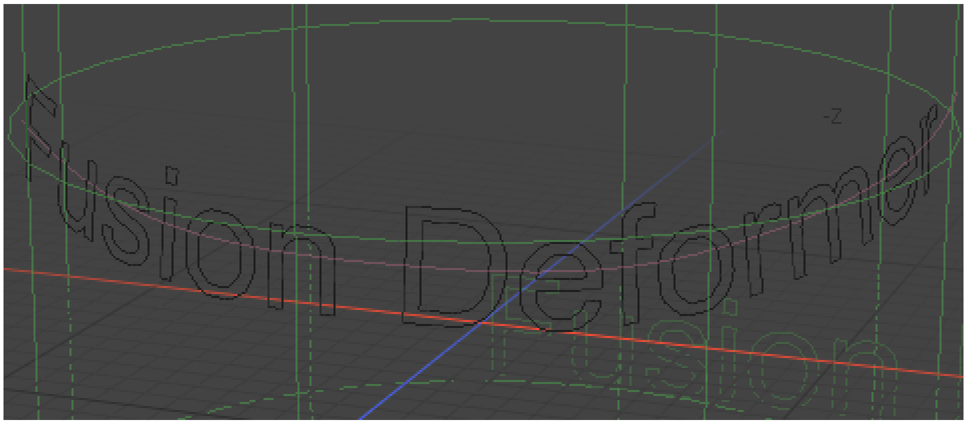


1. In an empty mesh, create a Text mesh operation.
   When the mesh operation is added, its properties are displayed in the **Properties** tab on the right.
2. In the **Properties**, edit the **Text**, **Font**, and **Size**. Set **Output Type** to **Bezier**, **Location** to **Middle**, and **Justification** to **Left**.

> **Note:** With left justification, the text is positioned to the left end of the text string in the world coordinate system origin. That point corresponds to the start of the strip, and the text string flows from the start of the strip to the end.
> Vertically, Y = 0 corresponds to the middle of the strip, which is why you would typically set **Location** to **Middle**.



3. Add the text item to the Fusion as a Primary: select the text mesh and your Primary (green) mesh, and in the **Fusion** sub-tab on the left, under **Set Mesh Role & Apply**, click the Apply Primary button.
   The text mesh is marked as a Primary mesh in the Item List, but nothing happens in the 3D viewport yet.
4. Select the Fusion Item and in its **Properties**, enable **Fusion Deformer**.
   The strip mesh is replaced by the deformed text string.



5. If you need to adjust the text, you can use the handles in the 3D viewport. To display them, select the Text mesh operation in the Mesh Operations list. If you need to edit the strip length, make sure to disable **Fusion Deformer** so you can see the strip you're editing.

So far, we've created the deformed text curve that follows our primary surface. To use it for embossing that primary surface, we need to create another instance of MeshFusion which will use the deformed curve output from our first instance of Fusion as input.

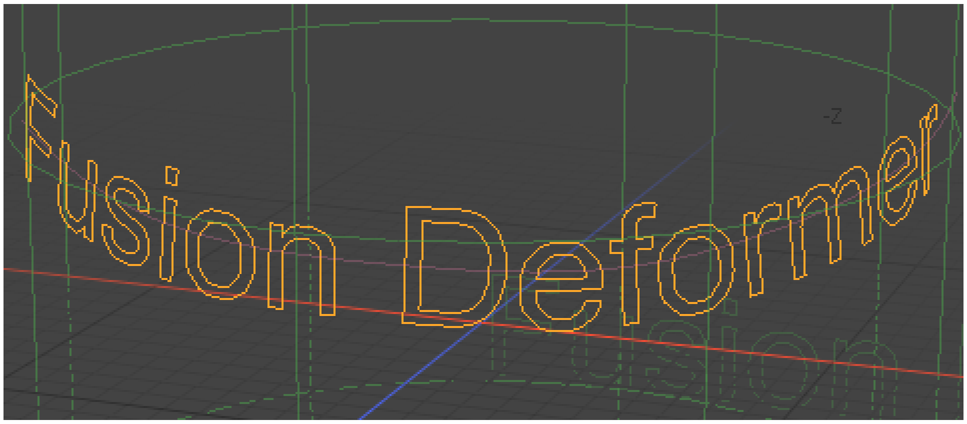
Creating the Embossing

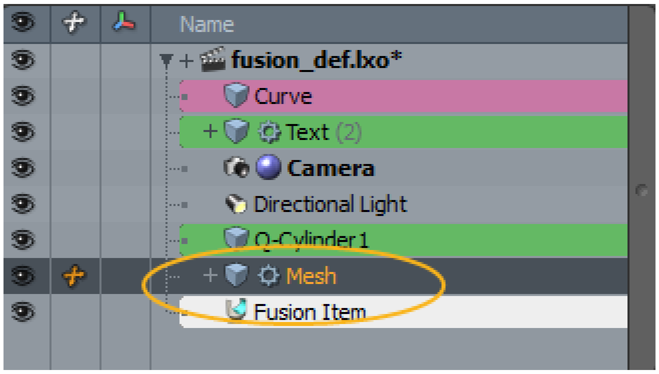
To emboss our text, we need to create another Fusion Item.

1. In a new mesh layer, add a **Merge Meshes** mesh operation.

> **Note:** For more information on how to do this, see Merge Meshes.

2. Expand the mesh operation in the list, and under **Sources**, click **(Add Sources)** to add the Fusion Item. Now our deformed curve appears as a separate item in the Item List.





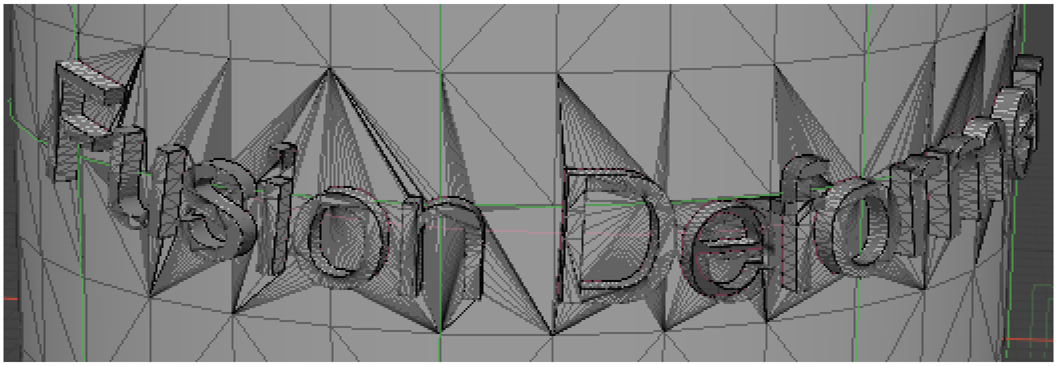
3. Select the Primary surface on which embossing is to appear, and in the **Fusion** sub-tab, under **Duplication**, click the Duplicate as Non-Fusion Mesh button.
   This duplicates your Primary mesh as a non-Fusion item.
4. Click New Fusion to create a second Fusion Item from the primary mesh.
   A second Fusion Item is created.
5. Select the duplicate Primary and the newly merged mesh that contains your deformed text, and in the **Fusion** sub-tab, under **Apply Surface Strip**, click the Full Strip with Grid Crossing button.
   Surface strips are generated on the Primary surface, and the curve mesh is marked with pink to indicate that it's a Trim.
6. Select only the deformed curve mesh and in its **Properties**, under **MeshFusion Curve Input**, set the following:
   - **Extrude Direction** to **Extrude Along Surface Normal**.

Note: To determine the outward and inward patches for embossing, MeshFusion expects the outer loops to be drawn clockwise and the inner loop counter-clockwise. For certain fonts, this is the other way around, and as a result, **Extrude Along Surface Normal** produces incorrect results. To correct this, in the **Text** mesh operation's properties, enable **Flip**.

- Under **MeshFusion**, change **Fusion SubDivs** from default **2** to **0**.
- Under **MeshFusion Embossing**, check **Enable Embossing**, and set **Middle Offset** and **Outward Offset** to the desired values.

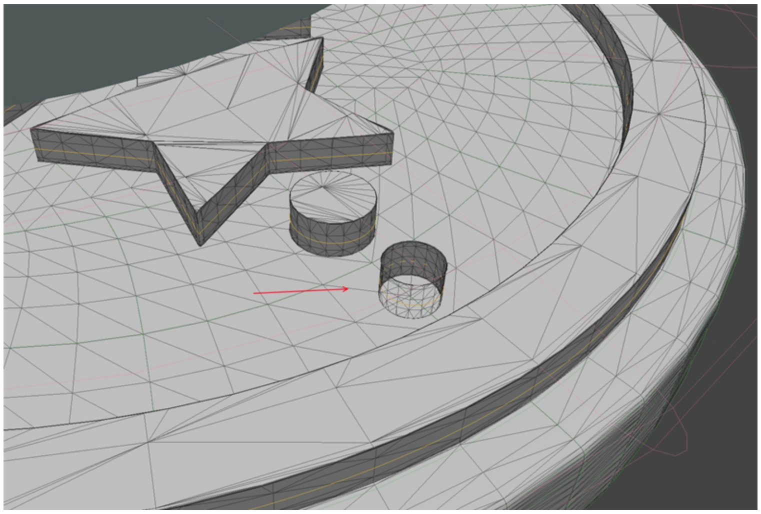
Note: Surface strips created from Bezier curves with sharp turns don't typically work well, as they self-overlap on the turns. You can fix this by enabling **MeshFusion Embossing** > **Sharp Bezier Corners** in the section of the deformed curve's **Properties**.

7. In the second Fusion Item's **Properties**, check **Enable Embossing**.
   Embossing appears in the 3D viewport.

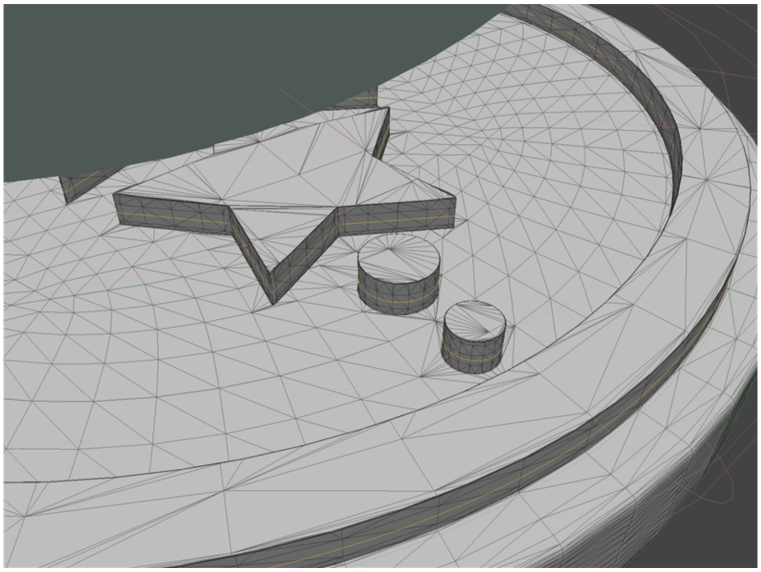


Case Study: Fixing Missing Surfaces in Fusion Models

Lets look at a Mesh Fusion scene which has issues with Fusion surfaces and curve meshes disappearing and walk you through how these issues can be fixed. In the following image, something is wrong.

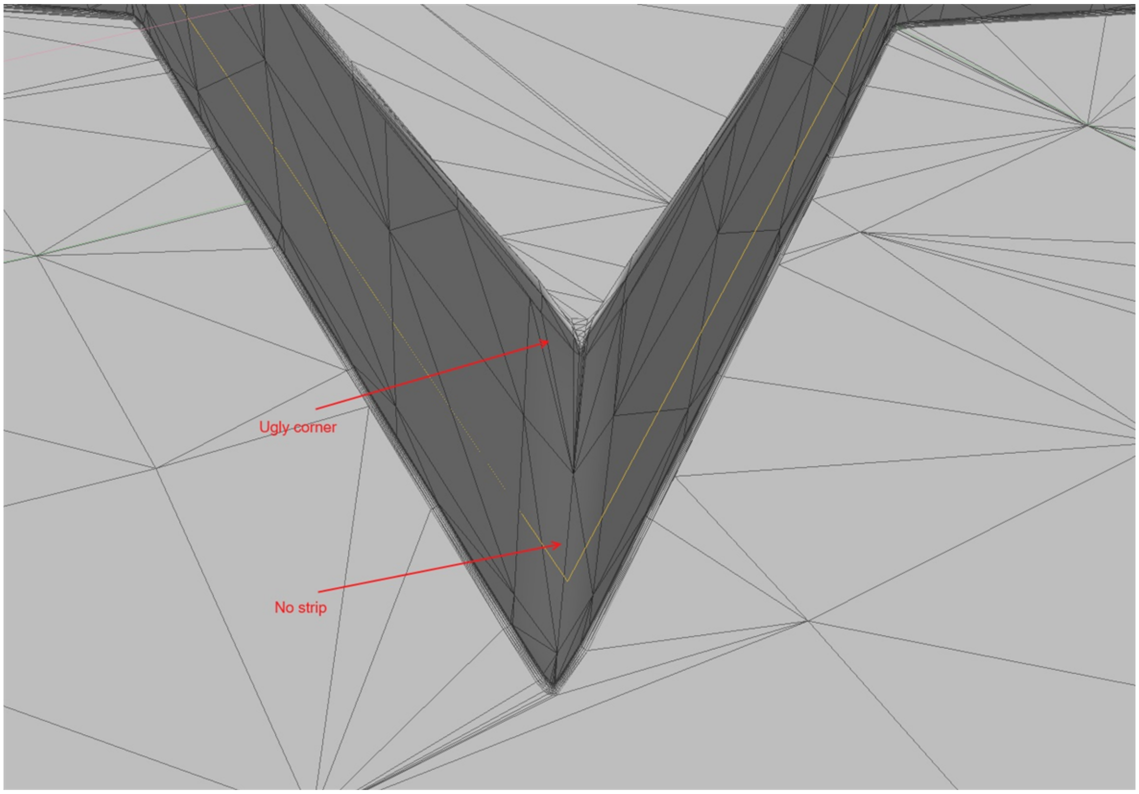


The sidewall of the small cylinder has inverted normals, and there is no cap on the cylinder. This is because the cylinder Bezier curve was drawn in the clockwise direction instead of counterclockwise. With the curve item selected in the Items list and Polygon selection mode active, we click on that curve loop and press **F** to flip the loop. In the following image, the result is correct.

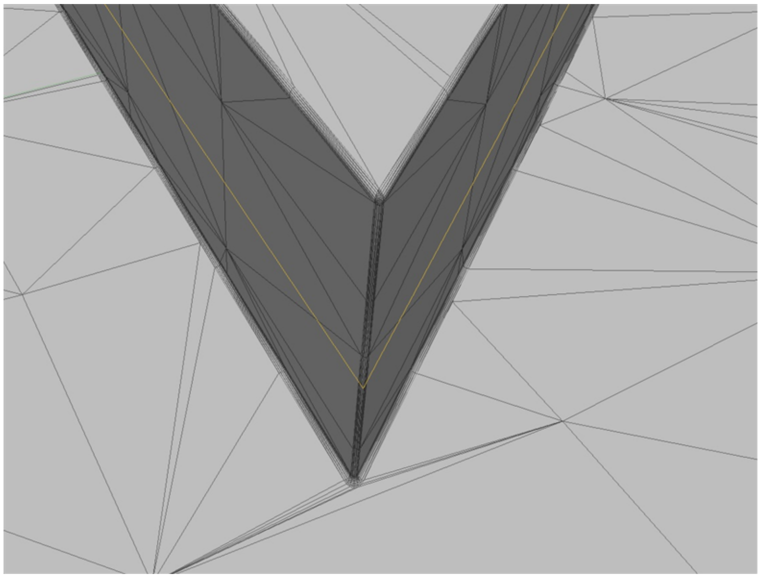


Tip: The rule is that the outward normal is pointing to the right as you draw the bezier curve. If you want a positive island like this example with the normal facing outward, you need to draw the curve counterclockwise. If you want a hole with the normal facing inward, draw the curve clockwise.

The following image shows the next problem with sharp corners.



An issue like this indicates that **Sharp Bezier Corners** is not enabled. With the curve selected in the Items list, enable **Sharp Bezier Corners** in the **Properties**. Once enabled, the corner is clean.



Another issue occurs when we select the mesh we want to use to cut into the main mesh, then click **Apply Subtractive Trim** in the **Fusion** tab, then select the cutting mesh and the rest of the curves in our scene and click **Trim**. The output mesh disappears from our scene.

We have two primary meshes in the scene. The main primary mesh (the base disk seen in the image) and the mesh extruded from curves. The extruded mesh grows upward from the disk.

The extruded mesh is unusual because it has open edges. You cannot make a closed surface by extrusion. When using open edges in a system like Mesh Fusion, which creates models by intersecting surfaces, you need to be careful not run into open edges when intersecting surfaces.

The reason why our Mesh Fusion failed in this scenario and the mesh disappeared is because the main primary mesh has two trims applied - a curve and the cutting mesh. However, the main primary mesh has only one trim applied, which is the cutting mesh.

Because of this, Mesh Fusion cannot perform a union between the two primaries first, so it applies the trims separately to the mesh extruded from curves and the main primary mesh. Because the mesh extruded from curves is a mesh with open edges, and our cutting mesh reaches deep enough to run into these open edges, the operation fails and the output mesh does not appear.

The fix is to make the main primary mesh use both trims, just like the curves mesh. All we need to do is select the main primary mesh and the curve and click **Trim** in the **Trim/Untrim** section of the **Fusion** tab.

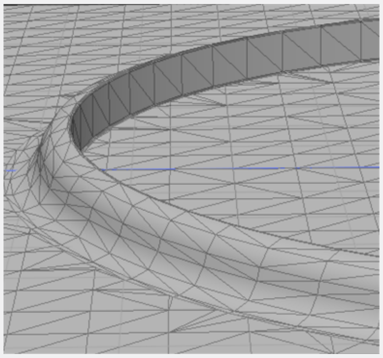
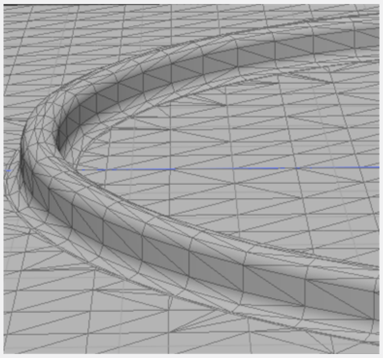
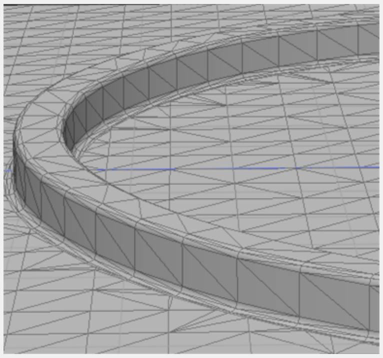
Tip: A general rule to follow is whatever trim you apply to your curve meshes, apply the same trims to your primary mesh which serves as the base for the curve mesh. This allows Mesh Fusion to weld the extruded mesh and the base primary together before applying any trims, thus avoiding the open edge problems.

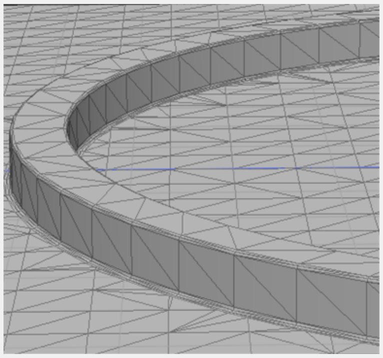
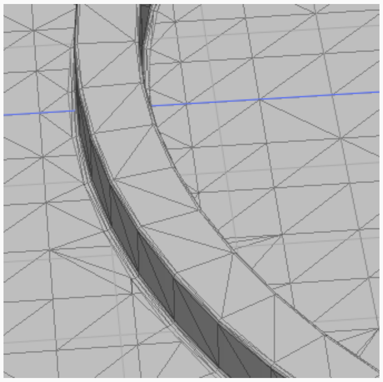
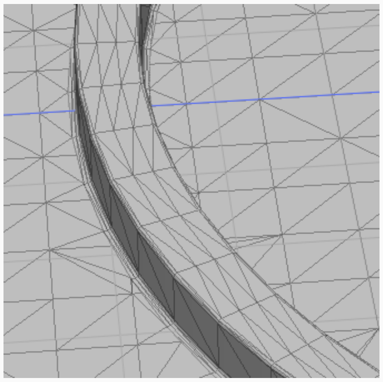
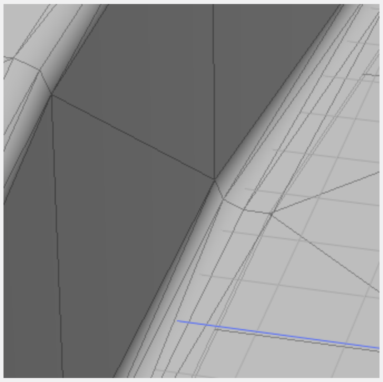
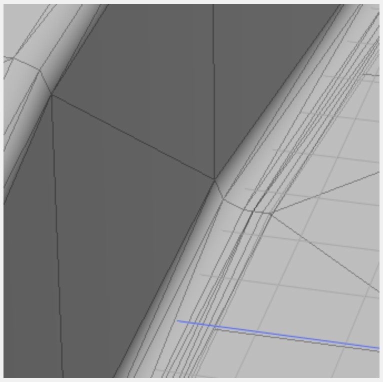
Embossing Properties

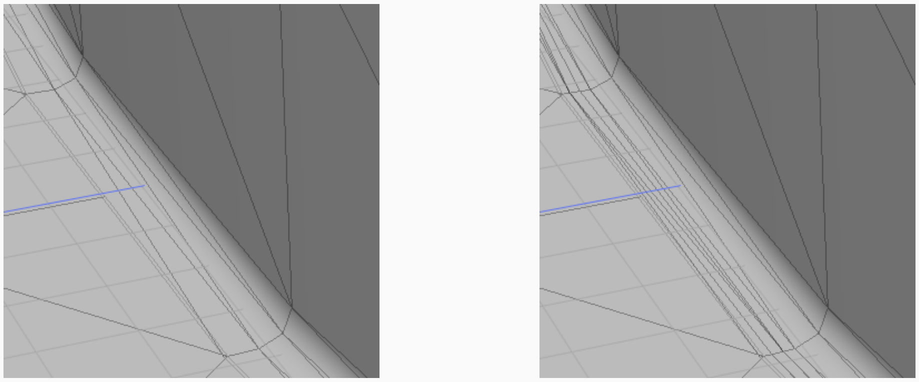
Mesh Fusion Embossing

Enable Embossing	When enabled, surface strips are embossed, using the Outward, Inward, and Middle Offset values.
Sharp Bezier Corners	Sharp corners on Bezier curves can result in incorrect strip geometry. When enabled, strips on sharp Bezier corners end and a new one begins.
Outward Offset	The amount by which the area outside the curve is offset.
Inward Offset	The amount by which the area inside the curve is offset.
Middle Offset	The amount by which the area within the curve is offset.

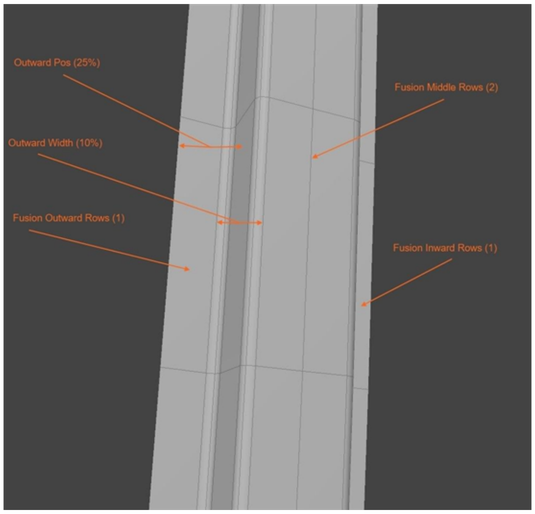
Mesh Fusion Advanced Embossing

<table>
<tr><td rowspan="4">Apply Embossing Preset</td><td colspan="3">Instead of directly editing the values, embosses the curve using a preset. The following options are available:</td></tr>
<tr><td>OneSided</td><td>Inward Position/Inward Width: 5%
Outward Position: 50%
Outward Width: 80%</td><td></td></tr>
<tr><td>TwoSidedSoft</td><td>Inward/Outward Position: 25%
Inward/Outward Width: 40%</td><td></td></tr>
<tr><td>TwoSidedMedium</td><td>Inward/Outward Position: 15%
Inward/Outward Width: 20%</td><td></td></tr>
</table>

	TwoSidedSharp **Inward/Outward Position**: 10% **Inward/Outward Width**: 10% 
Fusion Middle Rows	Sets the number of polygonal rows that make up the middle section of the strip. The default and minimum value is 1.  **Fusion Middle Rows**: 1  **Fusion Middle Rows**: 3
Fusion Inward Rows	Sets the number of polygonal rows on the inner shoulder. The default and minimum value is 1.  **Fusion Inward Rows**: 1  **Fusion Inward Rows**: 3
Fusion Outward	Sets the number of polygonal rows on the outer shoulder. The default and minimum

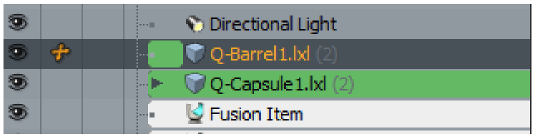
Rows	value is 1.  **Fusion Outward Rows**: 1 **Fusion Outward Rows**: 3
Inward Position	Adjusts the width of the strip from the inward patch to the middle of the strip. a setting of 0 uses the full length of the **Surface Strip Absolute Width**. Increasing this value decreases the width of the strip from the inside.
Outward Position	Adjusts the width of the strip from the outward patch to the middle of the strip. a setting of 0 uses the full length of the **Surface Strip Absolute Width**. Increasing this value decreases the width of the strip from the outside.
Inward Width	The width of the entire rounding area, from where the rise starts on the shoulder on the inside to where it ends at the beginning of the flat top, in the horizontal direction.
Outward Width	The width of the entire rounding area, from where the rise starts on the shoulder on the outside to where it ends at the beginning of the flat top, in the horizontal direction.

The image below shows you the parts of an embossed Fusion strip:


Outward Pos (25%)
Fusion Middle Rows (2)
Outward Width (10%)
Fusion Outward Rows (1)
Fusion Inward Rows (1)

Fusion Item

A **Fusion Item** is added to the **Item List** each time a new Fusion model is created. It generates the final Fusion model and offers a range of properties for editing the result.


Directional Light
Q-Barrel1.lxl (2)
Q-Capsule1.lxl (2)
Fusion Item

Fusion Item Properties

A **Fusion Item** includes the following properties:

Fusion Item	
Fusion On	Enables/disables the evaluation and display of the Fusion Item
Deferred Update	When enabled, the mesh is only updated on mouse up. This is useful when you're editing complex meshes and continuous updates are not smooth.

Fusion Item	
	Note: Deferred updating doesn't work smoothly while Preview is open and active. Pausing Preview while editing in the 3D viewport allows deferred updating to perform normally.
Optimize Performance	Allows you to optimize MeshFusion performance by caching Boolean operations, or caching meshes (strips and surface patches). Both forms of optimization limit the reevaluation of the Fusion Item to those elements affected by the current editing operation. **Note:** Mesh caching is only applied if the Mesh Mode is set to **Draft** The following options are available: • **Off** - optimization is disabled. • **Boolean** - caches Boolean operation only. • **Boolean + Mesh** - both Boolean and mesh caching is enabled. This is the default and recommended setting. **Note:** MeshFusion only takes advantage of mesh optimizations when the Fusion Item’s **Mesh Mode** is set to **Draft**. Best performance is achieved when optimization is used in conjunction with **Deferred Update**.
Draft Union	Improves the performance while working on a model by automatically and adaptively splitting the **Fusion Item** into *active* and *inactive* parts. The *active* part contains all **Primaries** and **Trims** necessary for the current editing, based on the selected Source Mesh. No Fusion strip geometry is generated or displayed in the 3D viewport for the active part. Instead, normal z-buffering

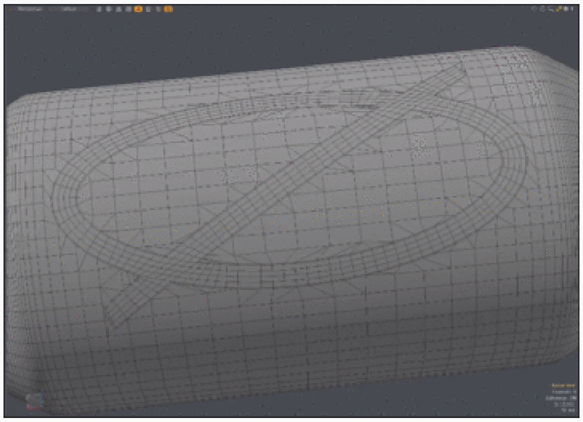
Fusion Item	
	is used to increase rendering efficiency. For information on Primaries and Trims, see Mesh Roles and Relationships. The *inactive* part contains all of the Fusion objects remaining (not selected) from the Source Meshes. Fusion strip geometry is generated and displayed in the 3D viewport for the inactive parts. The following options are available: • **Off** - the entire Fusion model is reincorporated. All Unions are evaluated and all Fusion strips are restored and displayed in the 3D viewport. • **Dynamic Separation** - When activated, the model is split into two parts; the *active* part that you are editing and the other *inactive* parts you are not editing. The Fusion strip geometry at the junction of the union is not displayed for the *active* selected source mesh. All of the Fusion strip geometry at the junction of the unions are displayed for non-selected items (*inactive* parts). **Note:** There is a delay at the start of editing if the selection has changed (the length of delay is related to complexity of the Fusion Item). • **Full Separation** - When activated, the entire model is split into as many separate parts as possible to get the best possible performance. There is no selection-based reconfiguration and therefore no delay when editing begins. Fusion strip geometry at the junction of the unions are not displayed for *active* or *inactive* parts.
Design Mode	An internal override of various Fusion Item properties. When enabled, some Fusion Item and strip properties are internally overridden (but retained). Strips become extremely narrow and strip corners become sharp. The overridden properties are the following:

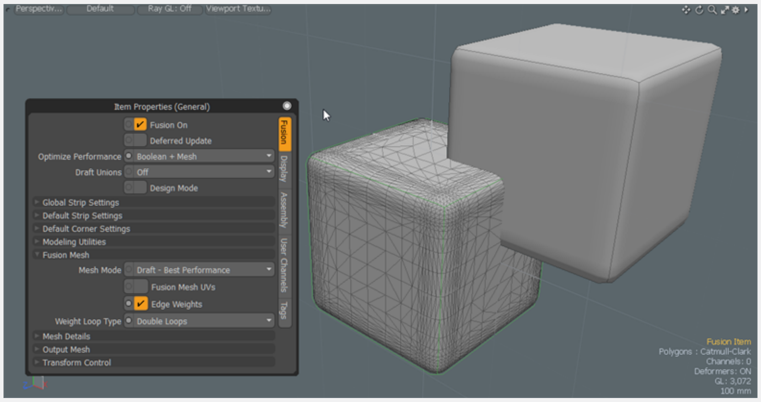
Fusion Item	
	• Strip Width • Strip Density • Corner Density • Corner Rounding Properties are restored when **Design Mode** is disabled. This mode is useful for improving performance, avoiding transient errors, and diagnosing meshing errors.
Global Strip Settings	
Strip Rows	The amount of rows of polygons that form the strip.
Strip Skirt Rows	Additional rows on the surface of the source mesh that provide better topology for post-MeshFusion modeling, such as corner rounding, or procedural modeling of a Fusion item.
Skirt Outer Rows	The amount of **Strip Skirt Rows** that run parallel to one another at a constant width along their entire length.
Corner Smoothing	The smoothing of Strip Corner quads. **0** is maximum sharpness.
Use Absolute Strip Widths	Set strip widths by absolute distance (in millimeters) or relative distance (in percentage). Toggled by **Switch Width Mode**.
Switch Width Mode	Toggles **Use Absolute Strip Widths**.
Abs Strip Quad Length	Sets strip quad length to an absolute distance. This is a global setting which only applies to all of the Fusion Item's strips.
Default Strip Settings	
Strip Length Abs	Sets the default strip quad length as the absolute distance. This may be overridden by individual Strip Item settings.
Base Quad Length	Sets the length of quads along a strip relative to adjacent quad sizes.

Fusion Item	
Adaptive Quad Length	This option increases strip density based on curvature.
Strip Width Abs	Sets the strip width as an absolute distance. **Note:** Editing the individual Strip Item settings overrides the defaults.
Strip Width	Strip width as a percentage of the original mesh quad size. Enforces the constant width regardless of edge angle. **Note:** Editing the individual Strip Item settings overrides the defaults.
Skirt Width	Skirt width as a percentage of the original mesh quad size. **Note:** Editing the individual Strip Item settings overrides the defaults.
Skirt Width Abs	Sets the skirt width as an absolute distance. **Note:** Editing the individual Strip Item settings overrides the defaults.
Skirt Outer Width	Sets the skirt outer row width as an absolute distance. This is subtracted from the the **Skirt Width Abs** value, and should always be less. **Note:** Editing the individual Strip Item settings overrides the defaults.
Strip Profile	Determines the bevel amount. **0%** is a flat bevel, **100%** is fully rounded.
Strip Smoothing	Specifies the amount of corrective smoothing at sharp turns

Fusion Item	
	along the strip.
Perpendicular Recovery	Forces the strip crossing edges to quickly recover to perpendicular position. The distance of the recovery is specified in the Recovery Length property. **Note:** Some strip intersections require crossing edges to be non-perpandiular. • **Off** - Cross edges return to perpendicular gradually over the entire strip segment length. • **Linear** - Cross edges return to perpendicular linearly over specified Recovery Length. • **Smooth** - Cross edges return to perpendicular smoothly (ease in and out) over specified Recovery Length. • **Ease Out** - Cross edges return to perpendicular easing out over specified Recover Length.
Recovery Length	Sets the distance of the crossing edges to return to be perpendicular.
Default Corner Settings	
Corner Density	Strips density ration near corners.
Corner Rounding	Strips edge rounding near corners.
Modeling Utilities	
Source Mesh Visibility	Toggles the visibility of source meshes.
Select Source Meshes	Selects all source meshes.
Select Entire Schematic	Select all items.
Set Output SubDs to Current	Sets the preferred output subdivision levels of all source meshes of the Fusion Item to their current subdivision levels.
Use Output SubD Levels	Sets the subdivision levels of all source meshes of the Fusion

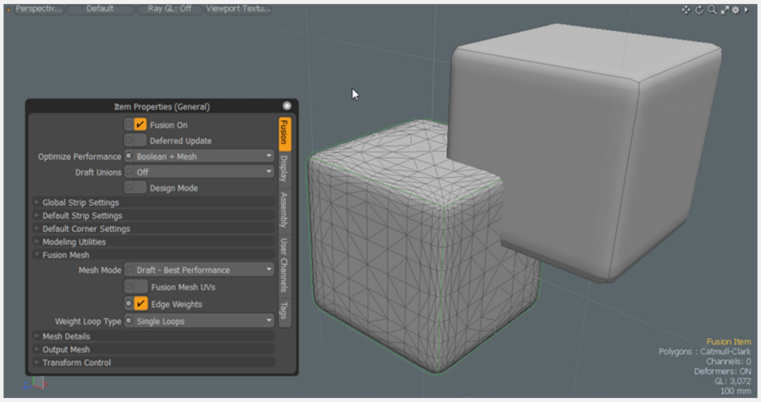
Fusion Item	
	Item to their preferred output subdivision levels.
Use Working SubD Levels	Sets the subdivision levels of all source meshes of the Fusion Item to one less than their preferred output subdivision levels.
Convert Fusion Schematic	Converts pre-10.2 Fusion Items to the new workflow, updates Fusion Items edited in the Schematic to the new workflow. **Note:** Fusion models manually created or edited in a Schematic viewport are not compatible with the new workflow. However, it is possible to work with an entirely manual Schematic Fusion model. For more information, see Legacy Fusion Items.
Delete Fusion Item	Removes the Fusion Item, keeping the source meshes.
Fusion Error Check	Checks for known internal errors.
Fusion Mesh	
Mesh Mode	Specifies the mesh type for the 3D viewport display and output. The following options are available: • **Draft - Best Performance** - Recommended for full performance optimization. This option is not recommended for Output (Strips and Surfaces are not unified). • **Airtight Draft** - Sets to the minimum airtight properties. Strips and surfaces are unified but no further refinement or optimization of the mesh is performed. This option may be slightly faster than other airtight mesh modes. • **Airtight Final** - Uses properties to produce the best topology output. Strips and surfaces are unified. Closely spaced vertices are merged. Pairs of tri-polygons are converted to quads where possible. No Selection Sets, Parts, or Material groups are created.

Fusion Item	
	• **Airtight Final w/Parts** - Uses the same mesh topology properties as **Airtight Final** and creates polygon Selection Sets, Parts, and Material groups. These are helpful when making selections for material assignment or further modeling of the output mesh. When set, it creates individual materials for each seam-enclosed part. Each part is an individual patch. A patch is a group of polygons surrounded by a loop of seams or strips. • **Airtight Final Item Parts** - Creates the same mesh topology as the **Airtight Final** mode. Like **Airtight Final w/Parts**, this option creates polygon Selection Sets, Parts, and Material Groups. However, those polygon sets are based on the source meshes rather than individual patches (groups of polygons surrounded by a loop of strips). Complex models containing many intersecting surfaces may generate dozens or hundreds of parts when using **Airtight Final w/Parts**. With this option, one Part/Material/Selection Set per source mesh is created.
Strip Polys Only	Creates only the strip polygons. The surface mesh is not generated. This is useful for procedural and direct modeling based on strips. For example, feeding this fusion item into a procedural using a Merge Meshes operation. This makes it easier to select this geometry to modify. 

Fusion Item	
Fusion Mesh UVs	Creates Mesh UVs for display and output. **Note:** The source mesh must have UVs. For more information, see Create UV Tool.
Edge Weights	Creates simulated CC edge weighting by internally adding loops to the source meshes.
Weight Loop Type	MeshFusion simulates Catmull-Clark edge-weighting by internally adding edge loops to the source meshes (original source meshes are unchanged). The following options are available: **Double Loops** - Adds two edge loops on either side of each weighted edge. Provides best fidelity to the CC surface, but produces denser Fusion meshes that may slow performance.  Item Properties (General) ☑ Fusion On ☐ Deferred Update Optimize Performance: Boolean + Mesh Draft Unions: Off ☐ Design Mode Mesh Mode: Draft - Best Performance ☐ Fusion Mesh UVs ☑ Edge Weights Weight Loop Type: Double Loops Fusion Item Polygons : Catmull-Clark Channels : 0 Deformers : ON GL : 3,072 100 mm Perspectiv... Default Ray GL: Off Viewport Textu... Global Strip Settings Default Strip Settings Default Corner Settings Modeling Utilities Fusion Mesh Mesh Details Output Mesh Transform Control Fusion Display Assembly User Channels Tags **Single Loops** - Adds a one edge loop on either side of each weighted edge. This option produces a surface that is often indistinguishable from that created with Double Loops. It produces cleaner, less dense, Fusion mesh items.

Fusion Item

Note: With certain source mesh topologies, fidelity to actual CC surface may be lower and greater Fusion SubDiv levels may be automatically enforced (effectively canceling the lower density benefits).

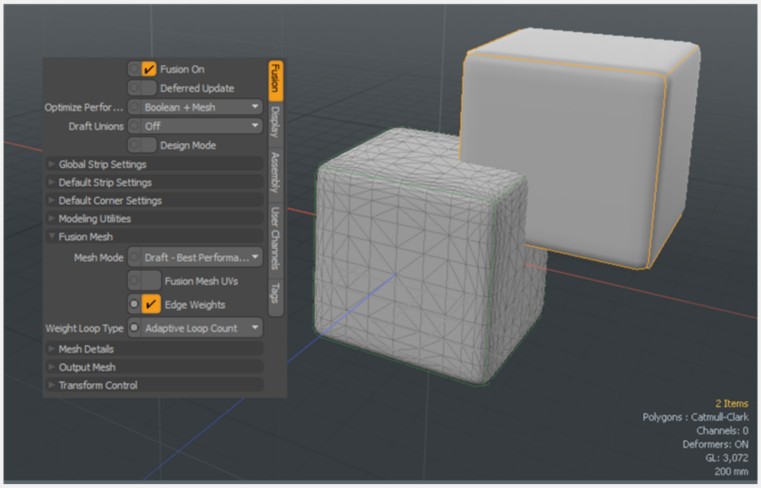


Item Properties (General)

☑ Fusion On
☐ Deferred Update
Optimize Performance: Boolean + Mesh
Draft Unions: Off
☐ Design Mode
Global Strip Settings
Default Strip Settings
Default Corner Settings
Modeling Utilities
Fusion Mesh
Mesh Mode: Draft - Best Performance
☐ Fusion Mesh UVs
☑ Edge Weights
Weight Loop Type: Single Loops
Mesh Details
Output Mesh
Transform Control

Perspectiv...
Default
Ray GL: Off
Viewport Textu...
Fusion
Display
Assembly
User Channels
Tags
Fusion Item
Polygons : Catmull-Clark
Channels: 0
Deformers: ON
GL: 3,072
100 mm

Adaptive Loop Count - Determines the best loop count (single or double loops) based on the nature of source mesh topology. Double loops are only created on those edges where single looping have fidelity and density requirement disadvantages.



☑ Fusion On
☐ Deferred Update
Optimize Perfor...: Boolean + Mesh
Draft Unions: Off
☐ Design Mode
Global Strip Settings
Default Strip Settings
Default Corner Settings
Modeling Utilities
Fusion Mesh
Mesh Mode: Draft - Best Performa...
☐ Fusion Mesh UVs
☑ Edge Weights
Weight Loop Type: Adaptive Loop Count
Mesh Details
Output Mesh
Transform Control

Fusion
Display
Assembly
User Channels
Tags
2 Items
Polygons : Catmull-Clark
Channels: 0
Deformers: ON
GL: 3,072
200 mm

Mesh Details

Fusion Item	
Relax Mesh Topo	Relaxes the mesh topology to produce more uniform polygons. **Note:** This option is usually not needed or recommended. It slows performance (especially with high iteration values), and may cause surface irregularities as described below. You should only use this option if you have problems with further modeling of the output mesh. MeshFusion may produce small irregular. For example, very long and narrow quads and tris, especially when small source meshes intersect with large quads on another source mesh. Relaxing the mesh makes those irregular quads more uniform. This may help when further modeling is performed on the output mesh, therefore preventing problems that can occur when certain types of distortions are applied to irregular meshes. However, relaxation can cause perturbations of the mesh (lumpiness) in areas where mesh curvature is significant as compared to the size of the polygons being relaxed. **Note:** Applies to **Final** Mesh Mode types only.
Relax Iterations	Determines how much the mesh relaxes and how far away from the strips the relaxation spreads. Specify the number of mesh topology relaxation iterations. The default value is 32. **Note:** Applies to **Final** Mesh Mode types only.
Relax Range	Determines how much the mesh relax range is. The Relax

Fusion Item	
	Range is measured in quads from the strip edges. The default value is 4. **Note:** Applies to **Final** Mesh Mode types only.
Remove Close Verts	Merges very closely spaced vertices along strip boundaries. Generally, this is helpful to produce lighter and cleaner meshes. However, when outputting very low density meshes surface irregularities can be introduced, creating denting and lumpiness around strip corners. The percentage value determines how aggressive Fusion merges the vertices. Lower values raise tolerances based on distance and curvature.
Cage Vertices	Uses cage vertices of the source meshes. This is useful for creating very low density Fusion meshes from simple, angular source meshes. Normally, source meshes are interpreted as Catmull-Clark surfaces, which means their vertices move significantly from there original ‘cage’ locations. When creating Fusion meshes based on simple, angular meshes and low Fusion SubDiv levels (0 or 1), this option maintains that angular quality by not allowing that CC shift. The vertices of the Fusion mesh remain identical to the source mesh. **Note:** This is not recommended for curved surfaces.
Extra Strip Row	Adds an extra row of edges along strip boundaries lying on the adjacent surface. This effectively ‘insulates’ the irregular polygons found at the surface/strip boundary, allowing those parts of the mesh to be edited (for example, moved relative to each other) without the surface anomalies that typically occur when disturbing those irregular polygons.

Fusion Item

This is useful for some specific post-Fusion modeling methods.

> **Note:** This is ignored when the Mesh Mode type is set to **Draft**.

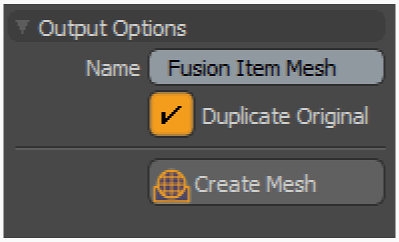
Output Mesh

Dup & Convert to Mesh

Creates a *frozen* copy of the Fusion mesh.

Options vary based on the Mesh Mode used.

If the Mesh Mode is set to **Airtight Final**, the following dialog displays:

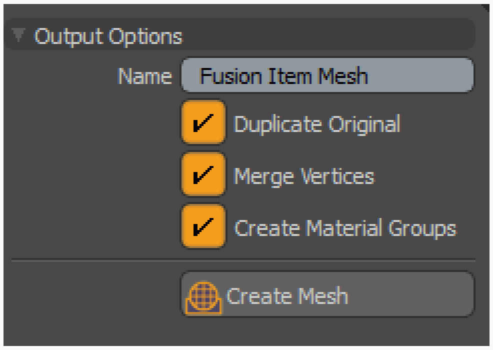


Output Options

Name: Fusion Item Mesh

☑ Duplicate Original

If the Mesh ModeMesh Modes set to **Airtight Final w/Parts** or **Airtight Final Item Part**, the following dialog displays:



Output Options

Name: Fusion Item Mesh

☑ Duplicate Original

☑ Merge Vertices

☑ Create Material Groups

> **Tip:** **Alt** + **Click** skips the display of the **Output Options** dialog and converts the mesh immediately, applying the last-used settings set.

Fusion Item	
	The following options are available: • **Name** - Allows you to provide a unique name for your output. • **Duplicate Original** - Converts the original Fusion Item. Recommended: Your original Fusion Item may be more reliable than a copy. We recommend to always enable this option. • **Merge Vertices** - Ensures that you get a single unified mesh; an airtight mesh. **Note:** We recommend to enable this option. The only exception is when you want to have a disconnected mesh. For example, where the surface and strip portions are not contiguous. • **Create Materials Groups** - Creates a container to hold material polygon groups. To view a list of materials created, open the **Mesh Ops Lists** tab and **Statistics** sub-tab and then expand **Polygons** > **Material**. Individual materials are listed for each source mesh used in your Fusion Item, using the source mesh name from which they came from. The Materials are also available in the **Shader Tree**. From the **Shader tree** you can select a material group and modify their properties. For more information, see Material Group. **Note:** You won't be able to use materials that require UV's unless the **Fusion Mesh UVs** option is enabled and the source meshes contain UVs. • **Create Mesh** - Allows you to provide a unique name for the new Mesh Item.

Fusion Item	
	Tip: When you have enabled either the **Airtight Final w/Parts** or **Airtight Final Item Parts** Mesh Mode, you can view a list of your parts by opening the **Lists** tab and expand **Polygons** > **Part**. Select the <Fusion Item> **Strips** item to view all of the polygons that belong to all of the strips used in your model. This is handy for advanced users who use scripting to manage their models.
Duplicate Fusion Item	Creates a duplicate Fusion Item.
Convert to Mesh Item	Creates a Mesh Item from current Fusion Item.
Transform Controls	
Position	Uses the X, Y, and Z input fields to apply specific offset distance values (calculated from the position the object was in when the tool was activated). When adjusting an object interactively in the viewport, the Position input fields display the current offset distance applied, providing useful user feedback.
Rotate	Uses the X, Y, and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, the Rotation input fields display the current offset distance applied, providing useful user feedback.
Scale	Uses the X, Y, and Z input fields to apply specific scaling values. When adjusting an object interactively in the viewport, the Scale input fields display the current scale factor applied, providing useful user feedback.
Reset	Resets the selected transform values to (0,0,0) returning the items to their default state.
Zero	Resets the chosen transform property values to 0, leaving the center position and item position intact.
Add	Adds transform items. Transform items are the channel

Fusion Item	
	groups associated with an item that store its transform values, controlling its position, rotation and scale. By default, new items do not have any transform items associated with them, even though they are visible within the Properties panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. You can simply transform the target item with one of the various transform tools, or by editing the value input fields. This action causes the particular transform item to be added automatically to the Channels viewport list. The **Add** function here can also be used to add the selected set of transforms to the channel list while keeping the default 0,0,0 values. A necessary step for referencing, in order to override the channels, they must first exist.
Realign Source Meshes	Re-aligns your source meshes to your Fusion Item. When enabled the source meshes are moved back into the place relative to the moved Fusion Item. The Position X, Y, and Z are transformed to zero and the Fusion Item is no longer offset from the source meshes.

Transforming Fusion Items

You can use the traditional Modo transform tools to move, rotate, and scale your Fusion Item and its source meshes. For more information on the transform tools, see Transforming Items.

By default, the Fusion Item always exists where its source meshes are located. However, you can transform your Fusion Item separately, detaching it from its source meshes.

To do this:

1. Select the Fusion Item in the **Item List**.
2. Activate the relevant transform tool and use the tool handles to move, rotate, or scale the item.

   Alternatively, in the Fusion Item's **Properties** tab, go to **Transform Control**, and edit the appropriate fields.

Note: The transform values indicated under **Transform Control** are not the Fusion Item's position, rotation, or scale values in world space, but the values relative to its source meshes.

You can re-align your source meshes to your Fusion Item by clicking the **Realign Source Meshes** button under **Transform Control**.

Legacy Fusion Items

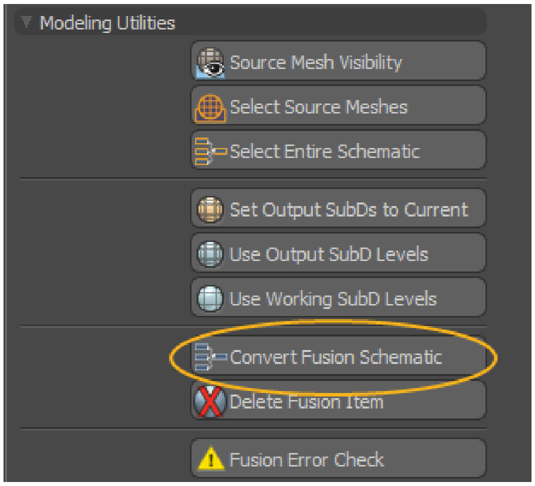
Fusion models manually created or edited in a Schematic viewport are not compatible with the new workflow. However, it is possible to work with an entirely manual Schematic Fusion model. At any time you can convert a manually edited schematic to work with the new workflow using the Convert Fusion Schematic function. Once done, they can be modified with the latest Fusion tools.

The same is true for models you manually edit in the **Schematic Fusion** layout. Schematic Fusion is not compatible with the new workflow, so after manually adjusting your model, you need to update it.

To convert a Fusion Item to the new workflow, follow these steps:

1. In the **Item List**, select the Fusion Item.

   The item's properties display in the **Properties** panel in the lower-right.
2. Under **Modeling Utilities**, click the **Convert Fusion Schematic** button.


Modeling Utilities
Source Mesh Visibility
Select Source Meshes
Select Entire Schematic
Set Output SubDs to Current
Use Output SubD Levels
Use Working SubD Levels
Convert Fusion Schematic
Delete Fusion Item
Fusion Error Check

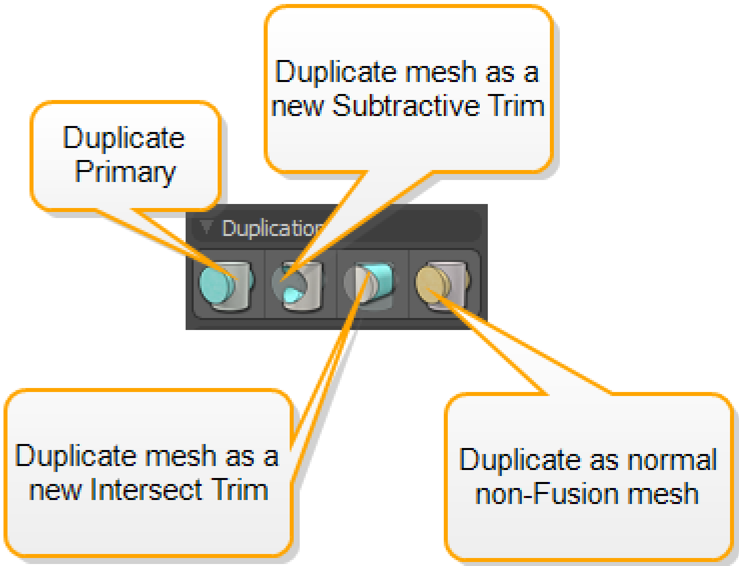
Your Fusion Item is converted to the new workflow and you can start editing it.

Duplicating Fusion Items

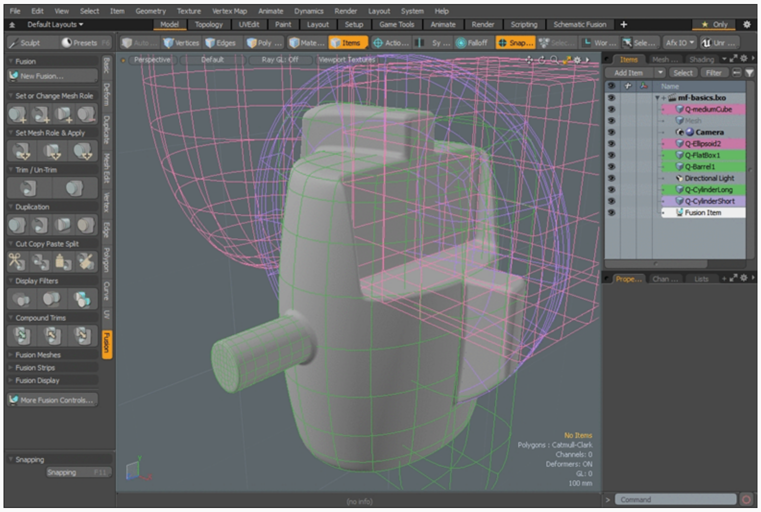
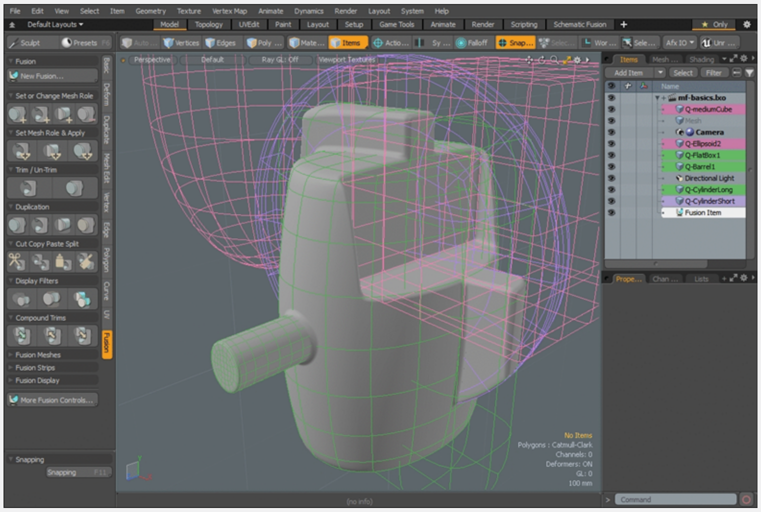
You can create duplicates and instances of Fusion Items, and use the traditional Modo transform tools to move, rotate, and scale them. For more information on the transform tools, see Transforming Items.

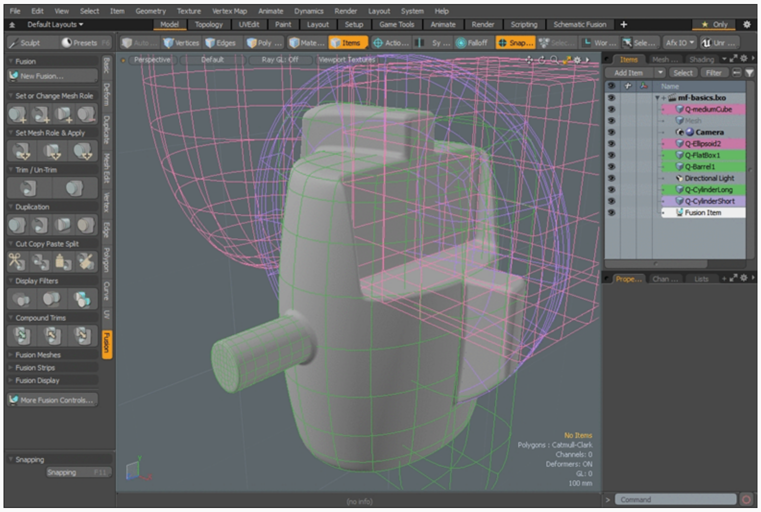
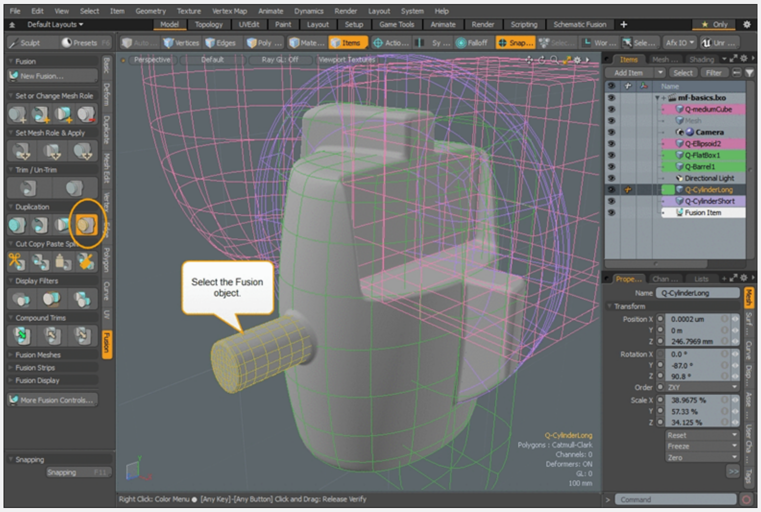
Creating Duplicates

To create a duplicate of a Fusion Item, select the Fusion Item in your scene and click one of the Duplicate Fusion tools:


Duplicate Primary
Duplicate mesh as a new Subtractive Trim
Duplication
Duplicate mesh as a new Intersect Trim
Duplicate as normal non-Fusion mesh

Tool	Description
Duplicate Primary	Duplicates the selected object, keep it as part of the existing **MeshFusion** object, and make it as a primary as the role. Click on the image below to view an animation.

	
Duplicate mesh as a new Subtractive Trim	Duplicates the selected object and creates a new subtractive trim. The object is displayed as a magenta wireframe. The item is also color-coded in the **Item List** as well. You must then define the relationship between the subtractive trim and another object, and apply **Trim** to see the results. Click on the image below to view an animation. 
Duplicate Mesh as a new Intersect Trim	Duplicates the selected object and creates a new intersect trim. The object is displayed as a orange wireframe. The item is also color-coded in the **Item List** as well. You must then define the relationship between the intersect trim and another object, and apply **Trim** to see the results.

	Click on the image below to view an animation. 
Duplicate as a normal non-Fusion mesh	Duplicates the selected object and creates a non-Fusion object. A mesh item is created and displayed in the 3D Viewport and **Item List**. Click on the image below to view an animation.  Select the Fusion object.

Creating Duplicates in the Item List

Alternatively, right-click the Fusion Item in the **Item List** and select **Duplicate** > **Duplicate**.

A copy of your **Fusion Item** is created in the **Item List** and the 3D viewport.

Creating Instances in the Item List

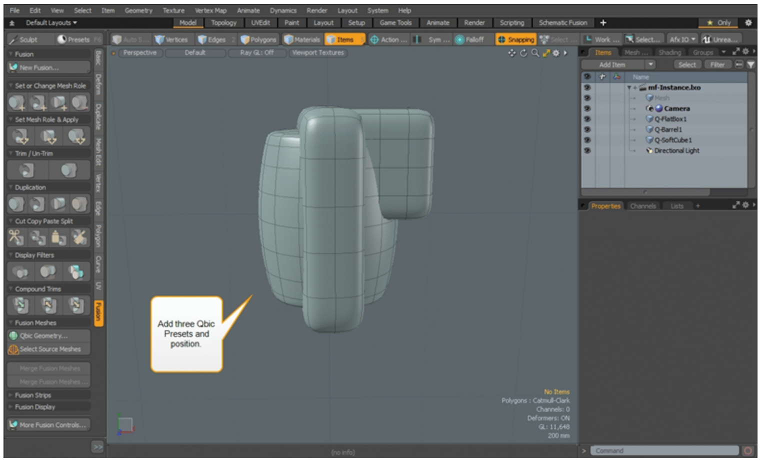
You can create an Instance of your Fusion Item similarly to creating duplicates. Right-click the Fusion Item and select **Duplicate** > **Instance**. A Fusion Instance is indicated in the **Item List** with italics. Any edits made to the original item is reflected in the instance mesh.

Note: For more information on Instances in Modo, see Instances.

Editing and transforming the source meshes, and changes to the Fusion properties get propagated to the Instance.

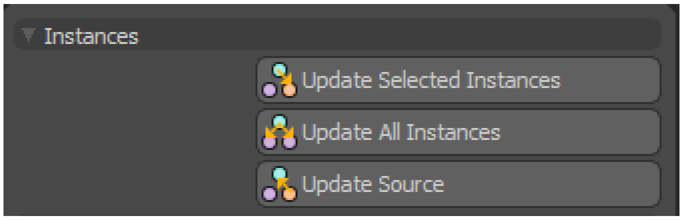
However, If you add or remove source meshes from the original Fusion Item, the changes don't affect the Instance already created. This is because when you create the Instance, the Schematic is also duplicated, which doesn't include the new meshes. This allows you to create Instance variants.

The following animation demonstrates the workflow.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
Schematic Fusion
Perspective
Default
Items
Properties
Channels
mf-Instance.lxo
Camera
Q-FlatBox1
Q-Barrel1
Q-SoftCube1
Directional Light

Add three Qbic Presets and position.

In cases when you want to apply the changes of the source Fusion Item to instances, you can synchronize them. Select the Fusion Item or instance to update, then in the Fusion **Properties** panel, under **Instances**, click the control you need:


Instances
Update Selected Instances
Update All Instances
Update Source

Udate Selected Instances synchronizes the selected instances with the current state of the source Fusion Item. **Update All Instances** updates all Fusion instances in your scene to match the current state of the source, and **Update Source** updates your source Fusion Item to match the state of the selected instance.

Tip: Use **Shift**+**Q** to hide a Fusion Source Mesh in cases where you need to select an item which is obscured by a currently selected item.

Fusion Mesh Conversion

When you've finished building a Fusion model, you can convert it into a regular mesh for further editing, rendering, or 3D printing.

For more information **Mesh Modes**, see Creating a New Fusion Item.

To convert a Fusion model to a regular mesh, follow these steps:

1. Select the **Fusion Item** in the **Item List**.
2. In the **Properties** panel (lower-right), ensure the **Fusion** tab is selected.
3. In the **Fusion Mesh** section, set the **Mesh Mode** to **Airtight Final**.

Note: To reveal the **Fusion Mesh** section, click the double arrow button on the bottom of the **Properties** panel.

4. In the **Output Mesh** section, click **Convert to Mesh Item**.

   Alternatively, click **Dup to Convert to Mesh** to create a duplicate fusion item mesh The original fusion Item is hidden.
5. Click **OK** to replace the fusion item with a regular mesh.
6. Click **OK** on the next dialog.

Sculpting

Sculpting in Modo is designed to be simple to use, while providing you with a powerful and intuitive toolset that empowers you to realize your vision with the least amount of strain. The Sculpting Techniques and Adding Images topics give you an overview of how Modo's brushing engine works. The subsequent Using Sculpting Brush Tools topic details the individual brushes and the Using Sculpting Utility Commands topic describes related tools for texturing and detailing your creation.

Sculpting Techniques

Modo provides you with the ability to sculpt your models in a variety of ways and then texture them easily in one integrated environment and work non-linearly. This flexibility means you can model, sculpt, apply a basic texture, and then go back and model some more. As with all the tools in Modo, the functions are deeply integrated and take advantage of the core functionality to provide an intuitive and powerful set of options.

Combining a natural blend of 2D- and 3D-painting techniques, you can select a brush and begin painting in any viewport. Modo tries to determine the best options automatically when painting or sculpting, but there are a couple requirements necessary to get the best results. Modo's default interface provides a **Paint** layout tab, which is dedicated to painting, sculpting, and Vertex Map-editing tools, and a toolbox with the related tools.

You may benefit greatly from using a pressure-sensitive tablet when painting or sculpting in Modo, but this is not a strict requirement. (Most users find tablets to be a more natural way to work with the pen-based input instead of using a mouse.) The additional control from the pressure-sensitive stylus can provide a greater degree of precision and subtlety when modeling and texturing creations.

Sculpting Types

Modo has three types of sculpting: **Mesh**, **Image-Based**, and **Multi-Res**. Mesh sculpting uses the raw geometry. It is great for building out rough forms, but it is not as good at producing fine details. Image-based sculpting uses image maps for producing fine details. You can use either grayscale displacement images (where shades of gray push vertices in or out a specific distance based on their shade) or 32-bit vector images that allow not only in-and-out movement but side-to-side movement as well for cavities and undercuts. With image-based sculpting, you can generate fine details for models, but it does have some limitations (such as the need for fixed-resolution bitmap images and the related UV map). Finally, Modo offers multiresolution (**Multi-Res**) sculpting, which requires no images or UV mapping, for you to sculpt on multiresolution meshes. This method provides the flexibility and immediacy of mesh sculpting and the detail of image-based sculpting. The only drawback is the greater amount of resources necessary to calculate the multiresolution meshes.

Each type of sculpting has its own set of requirements. If you understand the methods, you can determine how to best apply them for their intended purposes.

Mesh Sculpting

Mesh sculpting is rather simple in Modo. To do mesh sculpting, all that you need is some geometry and a brush. You can activate any of the sculpting tools available in the **Paint** layout and start sculpting. For more information in sculpting tools, see Using Sculpting Brush Tools. Modo doesn't care if the model is straight polygons or a subdivision surface. It pushes and pulls the vertices as you wish. To create complex sculptured meshes easily, you can apply the **Remesh** tool to distribute the polygons. Once applied, more details are added to your mesh. For more information, see Automatic Retopology Tool.

Note: When sculpting raw polygons, you can quite easily generate grossly non-planar polygons, which can cause rendering errors. You can generally resolve these errors by triangulating the model or by applying subdivision surfaces to the mesh.

Benefits: Easy setup, can work across multiple surfaces, not limited by material assignment, CPU-friendly, low memory requirements
Disadvantages: No high-frequency detail, only works on vertices

Image-Based Sculpting

Modo's displacement-mapping methods require subdivision surface models: regular models or Pixar's Catmull-Clark versions. These models also need an image map applied to all surfaces that you wish to sculpt. For best results with displacement maps, use 16-bit images. For vector displacement-based images, use 32-bit formats (such as EXR). The images also need to be applied to the surface with the Shader Tree by using a UV map. You can specify this in the Texture Locator. Defining a UV map can be a tedious process, but Modo provides tools, such as UV Unwrap and UV Relax, that make it easy for you to produce quality results in a short amount of time.

Once you meet these requirements, you can begin sculpting as you would for straight mesh sculpting. Activate a brush and a tip from the toolbox and then start sculpting a model.

Benefits: Good high-frequency detail, low polygon requirements until render time
Disadvantages: Setup required with UVs, video card display limitation for openGL viewports

Multi-Res Sculpting

Multiresolution sculpting is simpler to set up than image-based sculpting, but you do need to do it properly. First, this method requires the Pixar Catmull-Clark Subdivision Surfaces geometry. If your model has regular SubDs, then you must convert them to be Catmull-Clark subdivision surfaces. To do this:

1. Select all the SubD surfaces, press **Tab** to disable SubD mode, and then press **Shift+Tab** to apply the Catmull-Clark SubDs.
2. Next, select the **Multi-Res** checkbox in the Mesh Item's properties.

   Additionally, you may introduce some additional subdivision levels with the **Subdivision Level** option.

After you meet these requirements, you can activate a sculpting tool and begin to sculpt. With this method, you can add or remove subdivision levels by pressing the **+** and **-** keys on the numeric keypad and can go through the subdivision levels using the **Ctrl**+**Plus** and **Ctrl**+**Minus** keys.

With multiresolution sculpting, you can also use sculpting layers, which allow you to have multiple sets of displacement vectors in a mesh. For more information, see Creating Sculpting Layers

Benefits: Easy setup, no images or UV maps, powerful editing methodology
Disadvantages: Requires more RAM and processing power than other methods, result must be frozen to export to other apps

Customizing Brushes

Like the rest of the Modo toolset, the paint and sculpting tools are modular and composed through the Tool Pipe. If you are not familiar with the **Tool Pipe**, you don't have to use it. You can do 3D painting and sculpting with the provided tools. The standard tools fulfill most, if not all, of your modeling and texturing requirements.

The **Tool Pipe** brings flexibility and power to the painting system. It has components that you can add together to create custom tools. These components include the paint tool, a brush tip, custom inks, nozzles, and even falloffs. For basic painting and sculpting, you only need a tool and a brush tip. You can add or remove the inks, nozzles, and falloffs depending on the intended effect. With the **Tool Pipe** you can combine a variety of components to create customized tools specific to your creative needs.

Using Sculpting Brush Tools

Sculpting is a fast and intuitive method for building and refining 3D models. The brushes for sculpting can be found in the **Paint** layout interface, where Modo has a collection of tools and useful commands that are dedicated to painting and sculpting. You can also access the tools from the **Texture > Sculpt Tools** menu. Invoking any of the tools displays the associated properties in the appropriate viewport panel.

To sculpt, you need to select both a tool (a brush) and a tip (designates how the tool affects the geometry). When working with a mouse, drag over the surface to apply the stroke. You can also use a pressure-sensitive stylus and tablet. To scale the size of the brush tip, right-click and drag in the viewport.

Sculpting Tools		
	Push	Moves vertices away from the surface along the calculated normal direction of the polygon.
	Inflate	Pushes all vertices out along their normals with the push strength modulated by the brush.
	Fold	Creates a "hill-and-valley" effect by pushing vertices at the center of the brush in and pulling vertices at the edge of the brush outward.
	Clay	Raises the surface along the normal of the underlying polygons up to a desired level.
	Smooth	Evens out bumpy and irregular surfaces.
	Carve	Pushes the surface inward to create a valley.
	Scar	A pinch operation combined with a brush tip that pushes and pulls at the geometry, leaving a specific mark that looks like a scar.
	Flatten	Massages the mesh to a common flat plane.
	Smudge	Pulls vertices along with the brush briefly and then leaves them to rest as the brush passes on.
	Move	Grabs the mesh inside the brush and drags the surface in the direction of the brush.

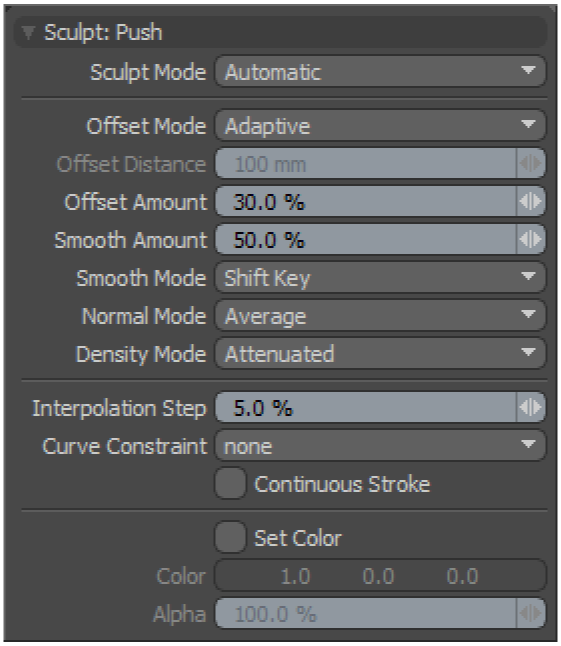
Sculpting Tools		
	Tangent Pinch	Pulls the geometry in towards the axis perpendicular to the brush stroke.
	Center Pinch	Pulls geometry in towards the center of the brush.
	Spin	Creates a vortex of vertices around the center of the brush.
	Emboss	Moves the surface of the mesh that is within the brush at the time of the click in or out along the local normals.
	Attenuate	Image-based sculpting only. Diminishes the strength of the sculpted data stored in the current displacement or vector displacement image texture.

Sculpting Masks		
	Mask: Paint	Adds masking to the target surface.
	Mask: Smooth	Softens or blurs an existing mask.
	Mask: Scale Down	Decreases the strength of the target mask.
	Mask: Erase	Removes masking from the target surface.
	Mask: Set Value	Applies the defined value to the mask to overwrite all existing values.

Push

The Push tool moves vertices away from the surface along the calculated normal direction of the polygon. By default, Modo calculates the normal from the center of the brush stroke. The effect is that of pulling the surface of the model outward as if the sculpt tool were moving vertices outward according to the profile of the brush. Pressing **Ctrl** while using the tool sends the surface downwards to create a valley rather than a hill. Pressing **Shift** while using the tool enables smoothing.

You can find the Push tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Customizing Paint Tools as well as the Customizing Inks and the Tablet Nozzles to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.



Sculpt: Push

Sculpt Mode: Automatic

Offset Mode: Adaptive

Offset Distance: 100 mm

Offset Amount: 30.0 %

Smooth Amount: 50.0 %

Smooth Mode: Shift Key

Normal Mode: Average

Density Mode: Attenuated

Interpolation Step: 5.0 %

Curve Constraint: none

☐ Continuous Stroke

☐ Set Color

Color: 1.0 0.0 0.0

Alpha: 100.0 %

Sculpt: Push

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

Normal Mode: Controls the method that Modo uses to displace the sculpted vertices. The options include:

- **Brush Center**: Uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.
- **Vertex**: Displaces each vertex along its own normal.
- **Average**: Calculates an average normal for all the vertices inside the brush.
- **Screen**: Displaces the vertices directly towards the active viewport.
- **Mouse Down**: Uses the normal of the mesh surface directly under the pointer when clicked. Modo applies all displacement according to that vector.
- **X,Y,Z**: Designates a World Axis along which to displace the vertices.

Density Mode: Determines how the sculpting stroke affects areas of the mesh that it has already covered. For example, brushing back and forth causes the tool to run back over itself and can have a number of different effects. This setting controls the sculpting tool during a single stroke. Once you lift the stylus off the tablet or release the mouse button, Modo resets the density control.

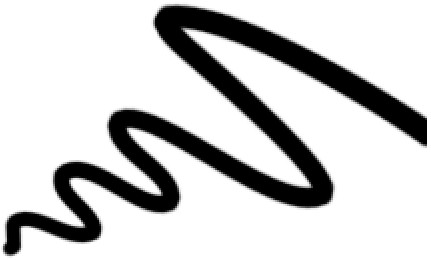
- **Additive**: Continues to add to the displacement each time the brush crosses over its own path with no maximum to limit the sculpting.

- **Attenuate**: Continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the **Offset Distance** or **Offset Amount** value.
- **Max Stroke**: Enforces a maximum offset based on the largest displacement value of the current stroke.
- **Max Map**: Limits the sculpting effect to the maximum amount in the current displacement image.

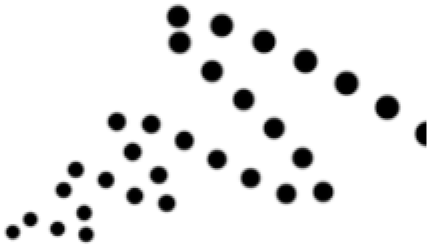
Interpolation Step: Defines the distance between each dot as a percentage of the brush size. Generally, paint strokes are not continuous lines but a series of dots (or dabs) in the image. By default, this value is set to **8%**. So, for a 64-pixel brush, Modo places dots approximately every 7 pixels. You can increase or reduce this value for a variety of effects. Increasing it to **100%** produces individual dots instead of a stroked line.

Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected). When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

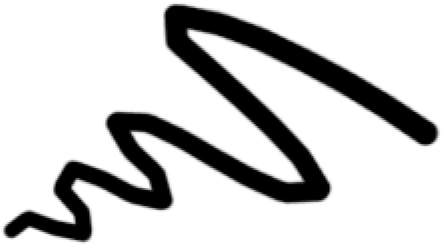
Continuous Stroke: When enabled, Modo draws a continuous stroke (such as a line) instead of as individual dots. Modo draws the stroke as linear segments between each interpolation step. Increasing the scale of the interpolation steps reveals the linear nature of the continuous stroke.



Default 10% **Interpolation Step** produces smooth stroke



Interpolation Step set to 200% reveals individual dots



200% **Interpolation Step** with **Continuous Stroke** enabled

Set Color: Indicates whether you can specify a color for painting onto an image map or not. When enabled, Modo applies the color with the sculpting stroke. This applies only to image-based sculpting and painting when both image maps are simultaneously selected in the Shader Tree.

Color: Specifies the RGB color applied to the image map when sculpting.

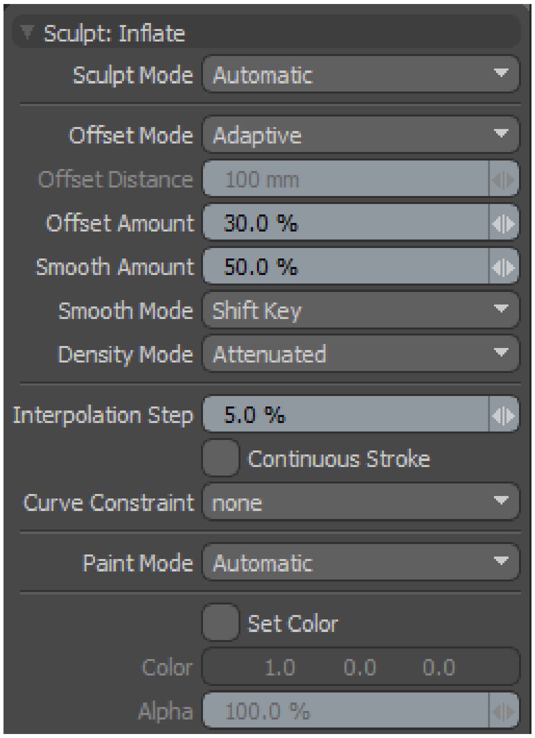
Alpha: Determines the opacity of the stroke applied to the image map.

Inflate

The Inflate tool works by pushing all vertices out along their normals with the push strength modulated by the brush. On a flat surface this tool is similar to the Push tool; however, on surfaces with curvature, the effect is very different and gives the appearance that the model is inflated from the inside (like a balloon). This is a good tool for increasing proportions quickly. Pressing **Ctrl** while using the tool sends the surface downward for a deflating effect. Pressing Shift while using the tool enables smoothing.

You can find the Inflate tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Inflate



Sculpt: Inflate

Sculpt Mode	Automatic
Offset Mode	Adaptive
Offset Distance	100 mm
Offset Amount	30.0 %
Smooth Amount	50.0 %
Smooth Mode	Shift Key
Density Mode	Attenuated
Interpolation Step	5.0 %
	☐ Continuous Stroke
Curve Constraint	none
Paint Mode	Automatic
	☐ Set Color
Color	1.0 0.0 0.0
Alpha	100.0 %

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

Density Mode: Determines how the sculpting stroke affects areas of the mesh that it has already covered. For example, brushing back and forth causes the tool to run back over itself and can have a number of different effects. This setting controls the sculpting tool during a single stroke. Once you lift the stylus off the tablet or release the mouse button, Modo resets the density control.

- **Additive**: Continues to add to the displacement each time the brush crosses over its own path with no maximum to limit the sculpting.

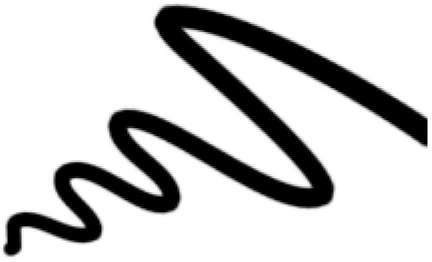
- **Attenuate**: Continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the **Offset Distance** or **Offset Amount** value.
- **Max Stroke**: Enforces a maximum offset based on the largest displacement value of the current stroke.
- **Max Map**: Limits the sculpting effect to the maximum amount in the current displacement image.

Interpolation Step: Defines the distance between each dot as a percentage of the brush size. Generally, paint strokes are not continuous lines but a series of dots (or dabs) in the image. By default, this value is set to **8%**. So, for a 64-pixel brush, Modo places dots approximately every 7 pixels. You can increase or reduce this value for a variety of effects. Increasing it to **100%** produces individual dots instead of a stroked line.

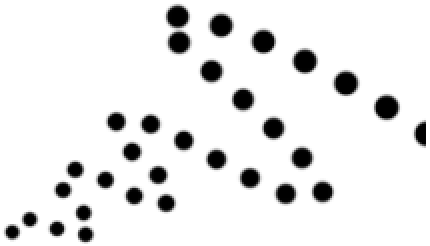
Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected). When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

Paint Mode: Define how Modo applies simultaneous image-based sculpting and image-based painting.

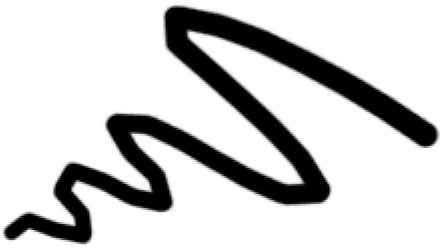
Continuous Stroke: When enabled, Modo draws a continuous stroke (such as a line) instead of as individual dots. Modo draws the stroke as linear segments between each interpolation step. Increasing the scale of the interpolation steps reveals the linear nature of the continuous stroke.



Default 10% **Interpolation Step** produces smooth stroke



Interpolation Step set to 200% reveals individual dots



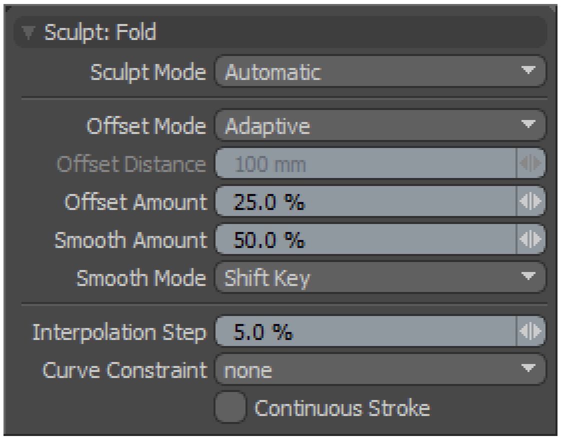
200% **Interpolation Step** with **Continuous Stroke** enabled

Fold

The Fold tool creates a "hill-and-valley" effect by pushing vertices at the center of the brush in and pulling vertices at the edge of the brush outward. As with all sculpting tools, the profile of the brush is very important in the final look of the surface. Pressing **Ctrl** while using the tool inverts the effect forcing the center outward and the edges inward. Pressing **Shift** while using the tool enables smoothing.

You can find the Fold tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Fold



Sculpt: Fold

Sculpt Mode: Automatic

Offset Mode: Adaptive

Offset Distance: 100 mm

Offset Amount: 25.0 %

Smooth Amount: 50.0 %

Smooth Mode: Shift Key

Interpolation Step: 5.0 %

Curve Constraint: none

☐ Continuous Stroke

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

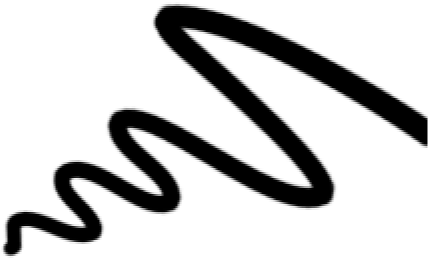
Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

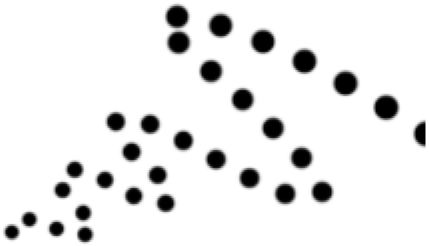
Interpolation Step: Defines the distance between each dot as a percentage of the brush size. Generally, paint strokes are not continuous lines but a series of dots (or dabs) in the image. By default, this value is set to **8%**. So, for a 64-pixel brush, Modo places dots approximately every 7 pixels. You can increase or reduce this value for a variety of effects. Increasing it to **100%** produces individual dots instead of a stroked line.

Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected). When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

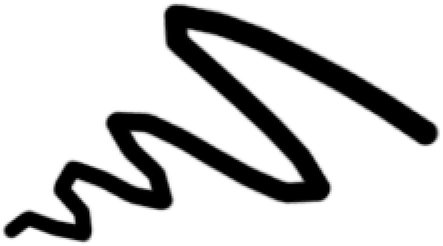
Continuous Stroke: When enabled, Modo draws a continuous stroke (such as a line) instead of as individual dots. Modo draws the stroke as linear segments between each interpolation step. Increasing the scale of the interpolation steps reveals the linear nature of the continuous stroke.



Default 10% **Interpolation Step** produces smooth stroke



Interpolation Step set to 200% reveals individual dots



200% **Interpolation Step** with **Continuous Stroke** enabled

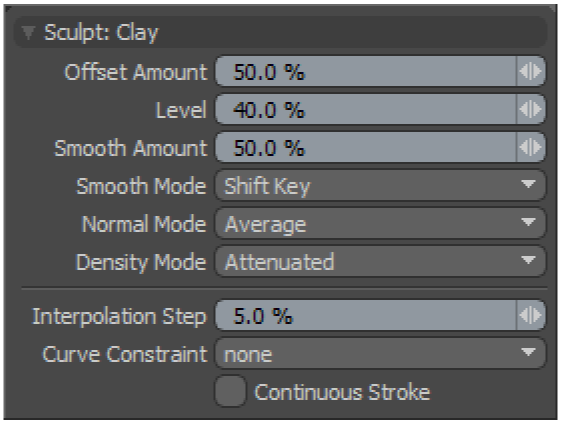
Clay

The Clay tool raises the surface along the normal of the underlying polygons up to a desired level (as indicated by the **Level** value). Once that level is reached, the surface remains at that fixed height. By default, Modo calculates the normal from the center of the brush. As long as the center, itself, doesn't overlap any previous strokes, the offset amount remains the same.

You adjust the amount of surface offset with the **Level** value. For example, with a level set to **0%**, nothing happens because all the points lie on the brush plane. If you change the level to **50%**, Modo raises the plane by 50% of the brush radius, and the effect is now apparent. Increasing the level even more increases the offset effect.

Note: The plane's orientation depends on the specified **Normal Mode**. You can achieve different effects by selecting a different **Normal Mode**. Generally, to get the best results specify a fine (small) **Interpolation Step**.

The Clay tool is in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.



Sculpt: Clay

Offset Amount 50.0 %

Level 40.0 %

Smooth Amount 50.0 %

Smooth Mode Shift Key

Normal Mode Average

Density Mode Attenuated

Interpolation Step 5.0 %

Curve Constraint none

☐ Continuous Stroke

Sculpt: Clay

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Level: Defines the maximum amount of offset any stroke can reach as percentage of the brush's radius. Increasing the brush size also affects the offset amount.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

Normal Mode: Controls the method that Modo uses to displace the sculpted vertices. The options include:

- **Brush Center**: Uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.
- **Vertex**: Displaces each vertex along its own normal.
- **Average**: Calculates an average normal for all the vertices inside the brush.

- **Screen**: Displaces the vertices directly towards the active viewport.
- **Mouse Down**: Uses the normal of the mesh surface directly under the pointer when clicked. Modo applies all displacement according to that vector.
- **X,Y,Z**: Designates a World Axis along which to displace the vertices.

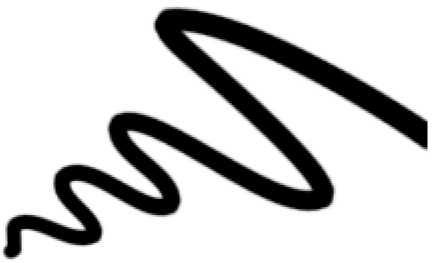
Density Mode: Determines how the sculpting stroke affects areas of the mesh that it has already covered. For example, brushing back and forth causes the tool to run back over itself and can have a number of different effects. This setting controls the sculpting tool during a single stroke. Once you lift the stylus off the tablet or release the mouse button, Modo resets the density control.

- **Additive**: Continues to add to the displacement each time the brush crosses over its own path with no maximum to limit the sculpting.
- **Attenuate**: Continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the **Offset Distance** or **Offset Amount** value.
- **Max Stroke**: Enforces a maximum offset based on the largest displacement value of the current stroke.
- **Max Map**: Limits the sculpting effect to the maximum amount in the current displacement image.

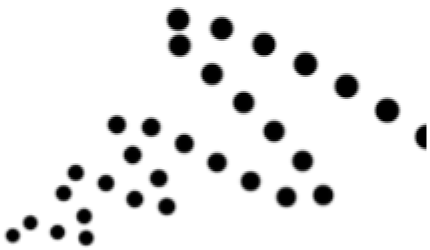
Interpolation Step: Defines the distance between each dot as a percentage of the brush size. Generally, paint strokes are not continuous lines but a series of dots (or dabs) in the image. By default, this value is set to **8%**. So, for a 64-pixel brush, Modo places dots approximately every 7 pixels. You can increase or reduce this value for a variety of effects. Increasing it to **100%** produces individual dots instead of a stroked line.

Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected). When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

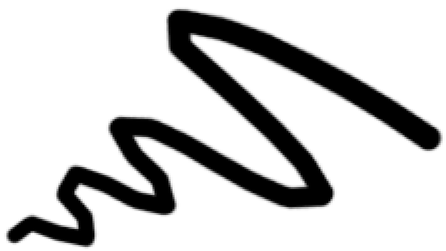
Continuous Stroke: When enabled, Modo draws a continuous stroke (such as a line) instead of as individual dots. Modo draws the stroke as linear segments between each interpolation step. Increasing the scale of the interpolation steps reveals the linear nature of the continuous stroke.



Default 10% **Interpolation Step** produces smooth stroke



Interpolation Step set to 200% reveals individual dots



200% **Interpolation Step** with **Continuous Stroke** enabled

Smooth

The **Smooth** tool evens out bumpy and irregular surfaces by averaging the positions of vertices under the brush tip. This makes surface inconsistencies and details disappear to produce a result similar to melting (like wax). The **Smooth** tool is essential to the sculpting workflow, which is why it is always available when using any of the other sculpting tools.

Tip: The **Smooth** tool allows you to maintain border edges and corner vertex of border polygons by using the Lock Corners and Lock Borders options.

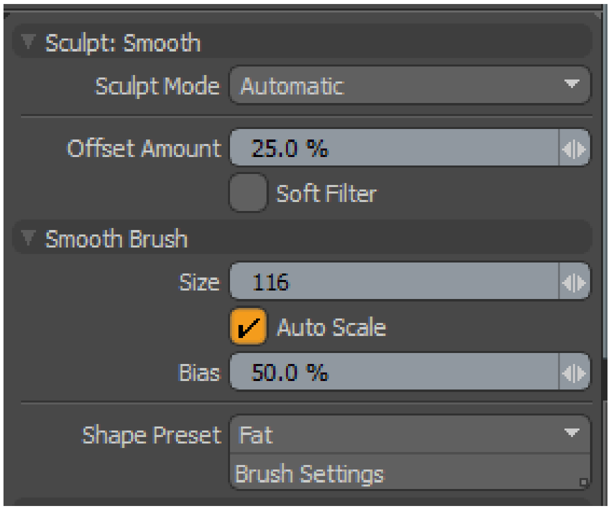
The **Smooth** tool is in the **Paint** layout in the **Sculpt Tools** sub-tab in the toolbox. You can also access it from the **Texture** > **Sculpt Tools** menu.

By pressing **Shift** when sculpting, you can access the **Smooth** tool whenever any other tool is active. Press and hold **Ctrl/Cmd** to apply a more gentle sculpting technique that attempts to preserve the volume of the mesh. This has the same affect as the **Preserve Volume** option in the Deform Smooth tool.

You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface.

Sculpt: Smooth Properties

The following options are available:



Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Soft Filter: Softens the image map by filtering out small details to produce a different result from the standard smoothing. This only applies to image-based sculpting.

For information about the Smooth Brush options, see Customizing Paint Tools.

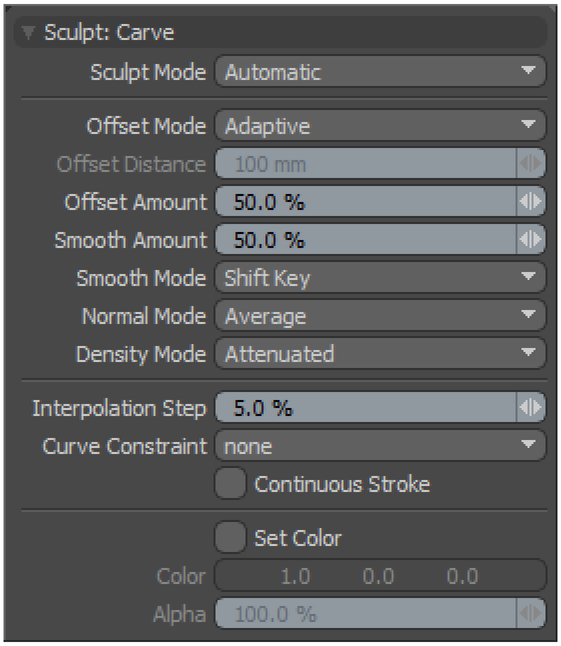
For information about locking borders and corners when using this tool, see Locking Borders and Corners.

Carve

The Carve tool works by pushing the surface inward to create a valley determined by a fixed maximum amount per stroke. This simulates carving away from the surface and is useful for scoring or scratching a surface. To easily create wrinkles, use a **Sharp** brush tip. Pressing **Ctrl** while using the tool sends the surface upwards to create a hill (rather than a valley). Pressing **Shift** while using the tool enables smoothing.

You can find the Carve tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Carve



Sculpt: Carve

Sculpt Mode: Automatic

Offset Mode: Adaptive

Offset Distance: 100 mm

Offset Amount: 50.0 %

Smooth Amount: 50.0 %

Smooth Mode: Shift Key

Normal Mode: Average

Density Mode: Attenuated

Interpolation Step: 5.0 %

Curve Constraint: none

☐ Continuous Stroke

☐ Set Color

Color: 1.0 0.0 0.0

Alpha: 100.0 %

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

Normal Mode: Controls the method that Modo uses to displace the sculpted vertices. The options include:

- **Brush Center**: Uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.
- **Vertex**: Displaces each vertex along its own normal.
- **Average**: Calculates an average normal for all the vertices inside the brush.
- **Screen**: Displaces the vertices directly towards the active viewport.
- **Mouse Down**: Uses the normal of the mesh surface directly under the pointer when clicked. Modo applies all displacement according to that vector.
- **X,Y,Z**: Designates a World Axis along which to displace the vertices.

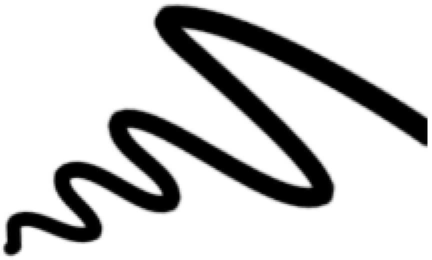
Density Mode: Determines how the sculpting stroke affects areas of the mesh that it has already covered. For example, brushing back and forth causes the tool to run back over itself and can have a number of different effects. This setting controls the sculpting tool during a single stroke. Once you lift the stylus off the tablet or release the mouse button, Modo resets the density control.

- **Additive**: Continues to add to the displacement each time the brush crosses over its own path with no maximum to limit the sculpting.
- **Attenuate**: Continues to add to the displacement as the brush crosses over its own stroke until the displacement reaches the **Offset Distance** or **Offset Amount** value.
- **Max Stroke**: Enforces a maximum offset based on the largest displacement value of the current stroke.
- **Max Map**: Limits the sculpting effect to the maximum amount in the current displacement image.

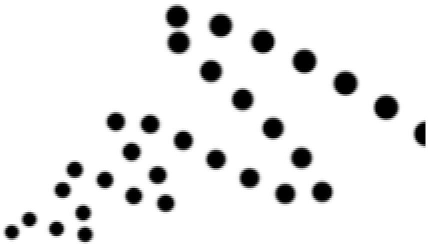
Interpolation Step: Defines the distance between each dot as a percentage of the brush size. Generally, paint strokes are not continuous lines but a series of dots (or dabs) in the image. By default, this value is set to **8%**. So, for a 64-pixel brush, Modo places dots approximately every 7 pixels. You can increase or reduce this value for a variety of effects. Increasing it to **100%** produces individual dots instead of a stroked line.

Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected). When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

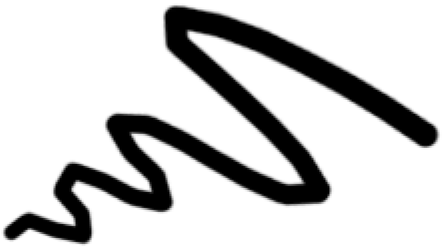
Continuous Stroke: When enabled, Modo draws a continuous stroke (such as a line) instead of as individual dots. Modo draws the stroke as linear segments between each interpolation step. Increasing the scale of the interpolation steps reveals the linear nature of the continuous stroke.



Default 10% **Interpolation Step** produces smooth stroke



Interpolation Step set to 200% reveals individual dots



200% **Interpolation Step** with **Continuous Stroke** enabled

Set Color: Indicates whether you can specify a color for painting onto an image map or not. When enabled, Modo applies the color with the sculpting stroke. This applies only to image-based sculpting and painting when both image maps are simultaneously selected in the Shader Tree.

Color: Specifies the RGB color applied to the image map when sculpting.

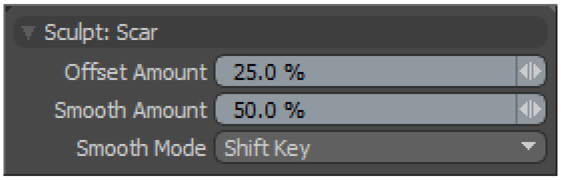
Alpha: Determines the opacity of the stroke applied to the image map.

Scar

The Scar tool does a simultaneous pinch operation combined with a unique brush tip that pushes and pulls at the geometry leaving a specific mark that looks like a gash or scar. You can use this tool to make wrinkles — especially when combined with smoothing. Pressing **Ctrl** while using the tool sends the surface outwards creating a bump rather than a depressed scar. Pressing **Shift** while using the tool enables smoothing.

You can find the Scar tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Scar



Sculpt: Scar

Offset Amount 25.0 %

Smooth Amount 50.0 %

Smooth Mode Shift Key

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

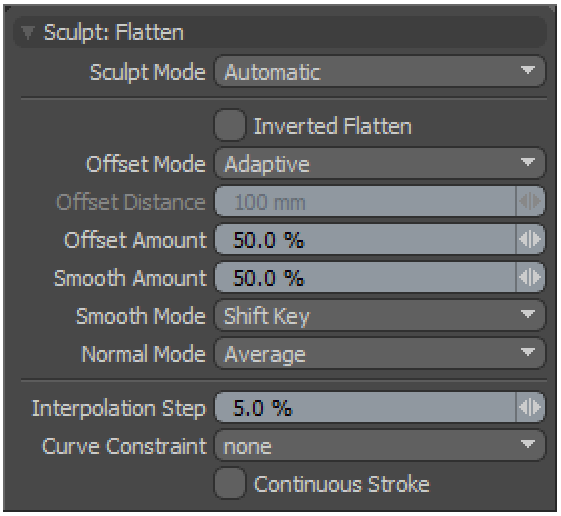
Flatten

The Flatten tool massages the mesh to a common flat plane. By default, Modo applies changes based on the center of the brush (in the default **Normal Mode**). This creates an ever-adapting plane that changes orientation as the brush moves over the surface of the mesh. In this mode, if you use deliberate brush strokes or multiple strokes over the same area, you can create consistent surfaces.

To create a plateau based on an exact position, set the **Normal Mode** to **Mouse Down**. With this setting, Modo reads the orientation of the surface when you click and uses that as a reference when moving the surrounding geometry. Pressing **Ctrl** while using the tool sends the surface upwards instead of flattening it. Pressing **Shift** while using the tool enables smoothing.

The Flatten tool is in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Flatten



Sculpt: Flatten

Sculpt Mode: Automatic

☐ Inverted Flatten

Offset Mode: Adaptive

Offset Distance: 100 mm

Offset Amount: 50.0 %

Smooth Amount: 50.0 %

Smooth Mode: Shift Key

Normal Mode: Average

Interpolation Step: 5.0 %

Curve Constraint: none

☐ Continuous Stroke

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Inverted Flatten: When enabled, somewhat inverts the function of the Flatten tool by raising what is under the brush, instead of lowering what is above it.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

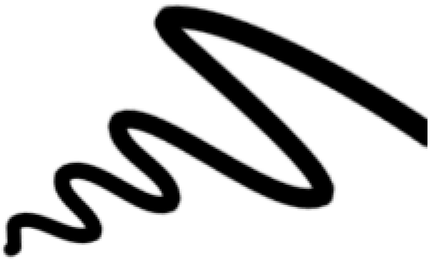
Normal Mode: Controls the method that Modo uses to displace the sculpted vertices. The options include:

- **Brush Center**: Uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.
- **Vertex**: Displaces each vertex along its own normal.
- **Average**: Calculates an average normal for all the vertices inside the brush.
- **Screen**: Displaces the vertices directly towards the active viewport.
- **Mouse Down**: Uses the normal of the mesh surface directly under the pointer when clicked. Modo applies all displacement according to that vector.
- **X,Y,Z**: Designates a World Axis along which to displace the vertices.

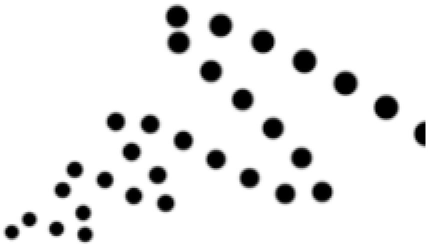
Interpolation Step: Defines the distance between each dot as a percentage of the brush size. Generally, paint strokes are not continuous lines but a series of dots (or dabs) in the image. By default, this value is set to **8%**. So, for a 64-pixel brush, Modo places dots approximately every 7 pixels. You can increase or reduce this value for a variety of effects. Increasing it to **100%** produces individual dots instead of a stroked line.

Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected). When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

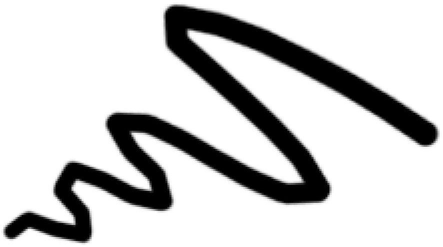
Continuous Stroke: When enabled, Modo draws a continuous stroke (such as a line) instead of as individual dots. Modo draws the stroke as linear segments between each interpolation step. Increasing the scale of the interpolation steps reveals the linear nature of the continuous stroke.



Default 10% **Interpolation Step** produces smooth stroke



Interpolation Step set to 200% reveals individual dots



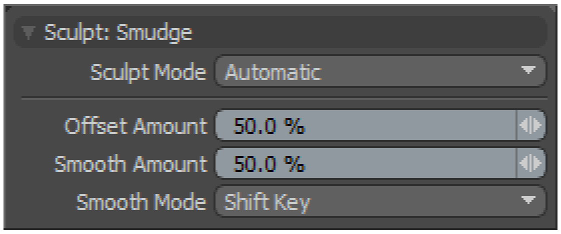
200% **Interpolation Step** with **Continuous Stroke** enabled

Smudge

The Smudge tool pulls vertices along with the brush briefly and then leaves them to rest as the brush passes on. This creates a smudging effect and is useful for adding organic imperfections in a surface. Pressing **Ctrl** while using the tool inverts the effect by sliding the surface toward the opposite direction. Pressing **Shift** while using the tool enables smoothing.

You can find the Smudge tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Smudge



Sculpt: Smudge

Sculpt Mode: Automatic

Offset Amount: 50.0 %

Smooth Amount: 50.0 %

Smooth Mode: Shift Key

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing **Shift** while using the sculpting tool. You change the default behavior with the **Smooth Mode** option. This is a convenience to enhance the sculpting workflow.

Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

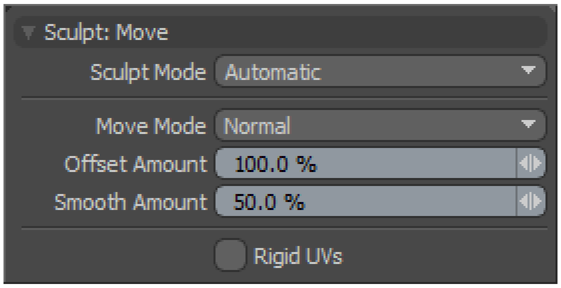
- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

Move

The Move sculpting tool grabs the mesh inside the brush and drags the surface in the direction of the brush. The tool uses the screen orientation to derive the perpendicular axis for the move. For example, if you have rotated the 3D viewport so that you are looking directly downward at the surface, the resulting move translates the mesh on the same XZ plane. If your intended result is to lift the surface, rotate the 3D viewport so that the tool is working at more of a perpendicular (or glancing) angle to the surface of the mesh. Pressing **Ctrl** while using the tool moves the surface in the opposite direction. Pressing **Shift** while using the tool enables smoothing.

You can find the Move tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Move



Sculpt: Move

Sculpt Mode: Automatic

Move Mode: Normal

Offset Amount: 100.0 %

Smooth Amount: 50.0 %

☐ Rigid UVs

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing the **Shift** key while using the sculpting tool. This is a convenience to enhance the sculpting workflow.

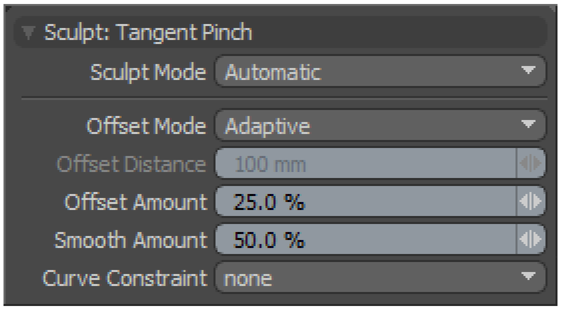
Rigid UVs: When enabled and using the Move tool to edit a UV map, Modo attenuates the effect based on the tip size and shape. By enabling this option, you can move entire UV islands as solid rigid pieces by moving the brush tip over an area and dragging.

Tangent Pinch

The Tangent Pinch tool pulls the geometry in towards the axis perpendicular to the brush stroke. If the brush stroke is moving up or down, the geometry pulls inward from the right and left. With this tool, you can quickly create or accentuate grooves or creases. Pressing **Ctrl** while using the tool inverts the effect by forcing the center area outward. Pressing **Shift** while using the tool enables smoothing.

You can find the Tangent Pinch tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Tangent Pinch



Sculpt: Tangent Pinch

Sculpt Mode: Automatic

Offset Mode: Adaptive

Offset Distance: 100 mm

Offset Amount: 25.0 %

Smooth Amount: 50.0 %

Curve Constraint: none

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing the **Shift** key while using the sculpting tool. This is a convenience to enhance the sculpting workflow.

Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected).

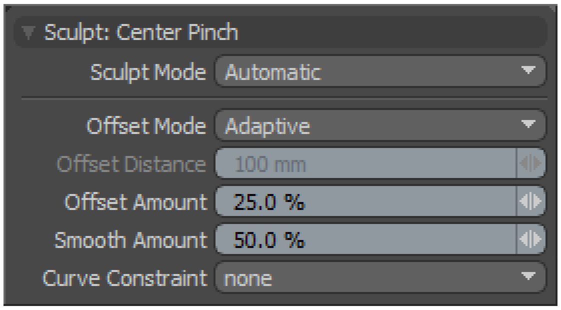
When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

Center Pinch

The Center Pinch tool pulls geometry in towards the center of the brush. As the components gravitate towards the center, they stay on the original surface to produce a sliding effect. Pressing **Ctrl** while using the tool inverts the effect to force the center area outward. Pressing **Shift** while using the tool enables smoothing.

You can find the Center Pinch tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Center Pinch—



Sculpt: Center Pinch

Sculpt Mode: Automatic

Offset Mode: Adaptive

Offset Distance: 100 mm

Offset Amount: 25.0 %

Smooth Amount: 50.0 %

Curve Constraint: none

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing the **Shift** key while using the sculpting tool. This is a convenience to enhance the sculpting workflow.

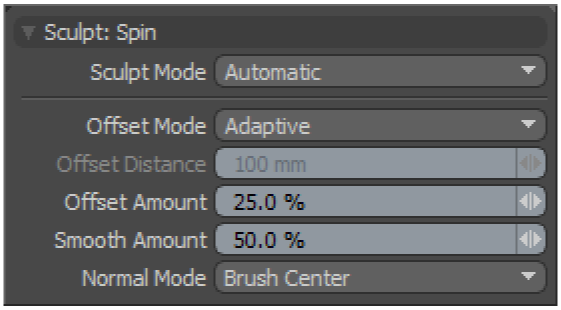
Curve Constraint: Determines whether you can use pre-drawn curves (either a Spline or Bezier curve) and then constrain the brush strokes to the curve when painting to produce a very smooth, controlled, and repeatable result. This works by having the curve element in the background layer (visible, but not selected). When you set the **Curve Constraint** option to either **Background 2D** or **Background 3D**, Modo constrains any paint strokes into the canvas near the curve to the nearest curve position. Setting this to **none** disables the constraining. **Background 2D** projects the curve from any orientation onto the surface through the viewport. **Background 3D** constrains the stroke to the curve's position in 3D space and only affects surfaces that are within close proximity (as defined by the brush's size).

Spin

The Spin tool creates a vortex of vertices around the center of the brush. The **Normal Mode** option determines the axis around which the vortex occurs. Use this tool for creating swirls in a surface or for adding some organic perturbation. The default **Normal Mode** is **Brush Center**, which causes all vertices within the brush to rotate around that center point using the normal of the geometry at that point as the axis of rotation. To control the axis of rotation by rotating the view, change **Normal Mode** to **Screen**. With the default method, **Brush Center**, Modo moves the vertices mostly along the existing surface; with **Screen**, Modo can create overhangs and more intense swirls. Pressing **Ctrl** while using the tool swirls the surface the opposite direction. Pressing **Shift** while using the tool enables smoothing.

You can find the Spin tool in the **Paint** tab in the **Sculpt** submenu. You can also access it from the **Texture > Sculpt Tools** menu. You can combine this tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Spin



Sculpt: Spin
Sculpt Mode: Automatic
Offset Mode: Adaptive
Offset Distance: 100 mm
Offset Amount: 25.0 %
Smooth Amount: 50.0 %
Normal Mode: Brush Center

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Mode: Specifies the method to calculate how the tool moves the surface when brushing.

- **Adaptive**: Adapts to the surface that is being brushed on. The adaptation is based on the setting for the **Offset Amount** and the size of the brush.
- **Absolute**: Conforms to the **Offset Distance** amount regardless of brush scale.

Offset Distance: Disabled unless you set **Offset Mode** to **Absolute**. When active, the value determines how far to displace the surface.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth Amount: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing the **Shift** key while using the sculpting tool. This is a convenience to enhance the sculpting workflow.

Normal Mode: Controls the method that Modo uses to displace the sculpted vertices. The options include:

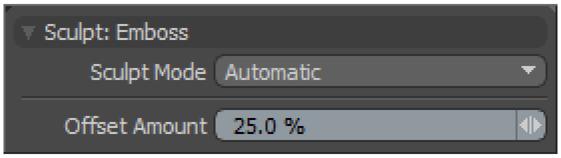
- **Brush Center**: Uses the geometry at the very center of the brush to calculate the direction along which to displace the mesh.
- **Vertex**: Displaces each vertex along its own normal.
- **Average**: Calculates an average normal for all the vertices inside the brush.
- **Screen**: Displaces the vertices directly towards the active viewport.
- **Mouse Down**: Uses the normal of the mesh surface directly under the pointer when clicked. Modo applies all displacement according to that vector.
- **X,Y,Z**: Designates a World Axis along which to displace the vertices.

Emboss

Similar to the Move tool, the Emboss tool is not a stroke-based deformer. With this tool, when you click, Modo moves the surface of the mesh that is within the brush at the time of the click in or out along the local normals. Moving the brush to the right, moves the mesh out from the surface; moving to the left, pushes the mesh inward. The value of the brush modulates the embossing. With a Smooth Brush, the Emboss tool creates single dimples or bumps on the surface. The Emboss tool is most effective when combined with certain brushes, such as the image-based Preset brush. By using the Geometry to Brush option, you can create interesting brushes, such as ears and other parts. By pressing **Ctrl** when applying an image, Modo inverts the image and reverses the displacement direction.

Additionally, you can combine the tool with the various Customizing Paint Tools as well as the Customizing Inks and the Tablet Nozzles to customize how the brush affects the surface. When activated, Modo displays the following attributes in the **Properties** panel.

Sculpt: Emboss



Sculpt: Emboss

Sculpt Mode Automatic

Offset Amount 25.0 %

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Offset Amount: Sets the Offset Amount. This option is disabled by default. To enable, set **Sculpt Mode** to **Adaptive**. This percentage value modulates the extent of the brushstroke. The total sculpting deformation is equal to the brush size times the percentage specified as the **Offset Amount**.

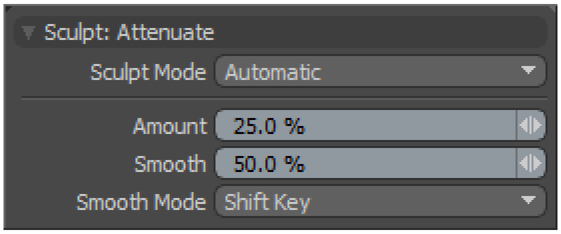
Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Attenuate

The Attenuate tool works with image-based and multi-resolution sculpting. This tool works by diminishing the strength of the sculpted data stored in the current displacement or vector displacement image texture. The effect is similar to melting away the sculpted detail and can be useful for fixing areas of your model. Pressing **Shift** as you use this tool enables smoothing.

You find the Attenuate tool in the **Paint** tab for the interface in the **Sculpt** submenu. You can also access this tool by choosing **Texture > Sculpt Tools**. You can combine the Attenuate tool with the various Brush Tips as well as the Inks and the Tablet Nozzle to customize the way each affects the surface. After you activate the Attenuate tool, you see the following attributes in the **Properties** panel.

Sculpt: Attenuate


Sculpt: Attenuate

Sculpt Mode: Automatic

Amount: 25.0 %

Smooth: 50.0 %

Smooth Mode: Shift Key

Sculpt Mode: Specifies the sculpting method used. By default, the setting is **Automatic**, which performs mesh-based sculpting when there are no displacement images, and automatically switches to image-based sculpting when an image is available. You may want to force Modo to use mesh-based sculpting even with an image applied for displacement. Doing this can be helpful for blocking out shapes and forms because mesh sculpting only affects the cage vertices of the base geometry.

- **Automatic**: Switches between mesh and image sculpting depending on availability of a displacement image map texture.
- **Mesh**: Forces mesh sculpting only.
- **Image Map (Details)**: Forces image sculpting only.

Amount: Modulates the strength of the attenuation.

Note: This is an important value for controlling sculpted detail. As such, you can adjust this value by middle-clicking and dragging to the left or right.

Smooth: Controls the amount of smoothing applied when using a specific tool. This is available with almost every sculpting tool in Modo and can be applied by pressing the **Shift** key while using the sculpting tool. This is a convenience implemented specifically to enhance the sculpting workflow.

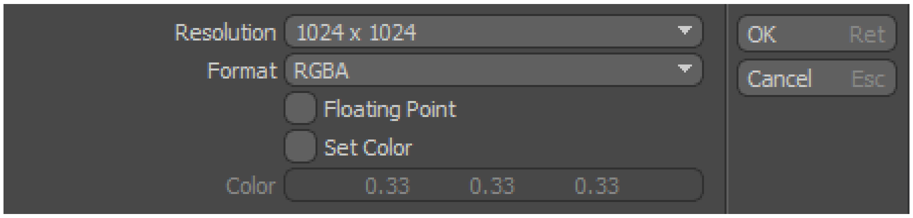
Smooth Mode: Provides alternate methods for applying smoothing while sculpting.

- **Shift Key**: Enables smoothing when pressing the **Shift** key.
- **Low Pressure**: Enables smoothing with light sculpting strokes on a pressure-sensitive tablet. Once the pressure threshold is exceeded, the smoothing effect attenuates toward the full pressure effect. At full pressure, the brush works normally as determined by the tool; therefore, this option establishes a dual function for the brush.
- **Uniform**: Enables smoothing at all times while sculpting; therefore, the tools make softer strokes.

Adding Images

Bitmap images are an essential component of texturing, painting, and image-based sculpting. So that you don't need to use an image-editing application, with Modo you can create blank images directly easily and quickly with the **New Image** button or the **File > New Image** command. You can save these image files to use in a variety of applications. Once you create an image, you must place it properly in the **Shader Tree** so that Modo can apply it to any surface you want to affect. You may also need to set the image's layer effect properly (such as specifying a **Displacement** for an image-based displacement). For more information on working with layers in the **Shader Tree** and assigning effects, see the Shader Tree topic. Modo also has utilities for you to add specific types of new images to the **Shader Tree**. For more information about adding textures, see Adding Texture Commands below.

When you create a new image, you first specify the file's name, location, and format in an OS-specific dialog. Once you create the image file, Modo opens a panel where you designate the specifics about the image, itself.



Resolution: 1024 x 1024

Format: RGBA

☐ Floating Point

☐ Set Color

Color: 0.33 0.33 0.33

Resolution: Indicates the image's resolution. When painting in 3D, Modo displays the images with OpenGL. To obtain the best performance for rendering textures in GL, work with square images. The available choices (based on standard GL texture sizes) are: **64 x 64**, **128 x 128**, **256 x 256**, **512 x 512**, **1024 x 1024**, **2048 x 2048**, **4096 x 4096**, and **8192 x 8192**.

Format: Specifies the color format for the image. The available choices are: **gray**, **RGB** color, or **RGBA** color (RGB with a transparency channel). Gray images are typically smaller in file size and are very useful with textures for displacement or bump maps. RGB images are opaque. Usually, the best choice for color textures is RGBA to implement different levels of opacity. With transparency (the alpha channel), the new texture may not completely obscure the materials and textures layered beneath it.

Floating Point: Specifies whether to use floating-point values or not for the image. Floating-point images (also known as HDRI, High Dynamic Range Images) give a broader range of potential color values. By default, this checkbox is clear, and Modo creates an integer image that defines colors in 8 bits per channel (0 to 255). A floating-point image can use values many times and can be useful for light sources or for detailed displacement maps.

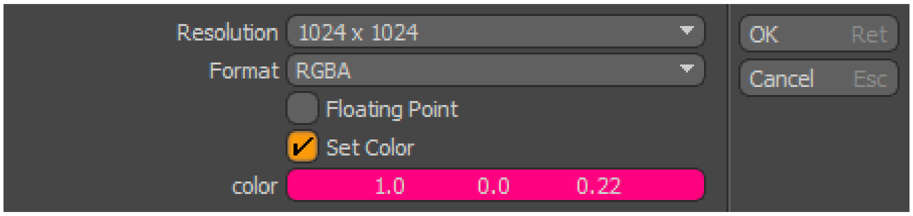
Set Color: Specifies whether or not the image has a base color or base transparency to start. By default, this checkbox is clear so that new images are completely blank (with no color or alpha pixels). If you select this checkbox, each RGBA image has an opaque alpha (100% so all pixels are visible) and a base color as specified in the **Color** option.

Color: Specifies the base color if you selected the **Set Color** checkbox. Modo uses this RGB color value to fill the new image.

Tip: To save an image as a 16-bit image, you must select the **Floating Point** checkbox. Otherwise, all images default to being 8-bit images regardless of the specified image format.

Adding Image Sequences

You can add image sequences as easily as adding still images. Image sequences are a series of still frames stored as consecutively numbered image files. You can use these for baking out animated textures or for painting animated sequences. When applied to a surface, the sequence frame number equals the rendered frame number. If frames are missing, Modo repeats the last frame until it finds another frame. To create a new image sequence, use the **File > New Image Sequence** command or the **New Sequence** button in the Clips Browser. In the OS-specific dialog, define the name, location, and format of the image files. After you click **OK**, in the **New Sequence Clip** dialog specify the resolution and other options for the frames.


Resolution 1024 x 1024
Format RGBA
☐ Floating Point
☑ Set Color
color 1.0 0.0 0.22

Resolution: Indicates the resolution of the images in the image sequence. When painting in 3D, Modo displays the images with OpenGL. To obtain the best performance for rendering textures in GL, work with square images. The available choices (based on standard GL texture sizes) are: **64 x 64**, **128 x 128**, **256 x 256**, **512 x 512**, **1024 x 1024**, **2048 x 2048**, **4096 x 4096**, and **8192 x 8192**.

Format: Specifies the color format for the images. The available choices are: **gray**, **RGB** color, or **RGBA** color (RGB with a transparency channel). Gray images are typically smaller in file size and are very useful with textures for displacement or bump maps. RGB images are opaque. Usually, the best choice for color textures is RGBA to implement different levels of opacity. With transparency (the alpha channel), the new texture may not completely obscure the materials and textures layered beneath it.

Floating Point: Specifies whether to use floating-point values or not for the images. Floating-point images (also known as HDRI, High Dynamic Range Images) give a broader range of potential color values. By default, this checkbox is clear, and Modo creates integer images that define colors in 8 bits per channel (0 to 255). Floating-point images can use values many times larger and are useful for light sources or for detailed displacement maps.

Set Color: Specifies whether or not the images have a base color or base transparency to start. By default, this checkbox is clear so that new images are completely blank (with no color or alpha pixels). If you select this checkbox, each RGBA image has an opaque alpha (100% so all pixels are visible) and a base color as specified in the **Color** option.

Color: Specifies the base color if you selected the **Set Color** checkbox. Modo uses this RGB color value to fill the new image.

Saving Images

Once you have modified your images or sequences in Modo in any way (such as through baking, sculpting, or painting), you need to save the images. Modo provides several ways to save images:

- You can select an image in the **Images** viewport, right-click, and then choose **Save**.
- You can select the image to save and use the **File > Save Image** or **File > Save Sequence** command.

- To save all images, use the **File > Save All Images** command.
- When you close a session, Modo prompts you to save any unsaved images. If prompted, click **Yes** to save the images.

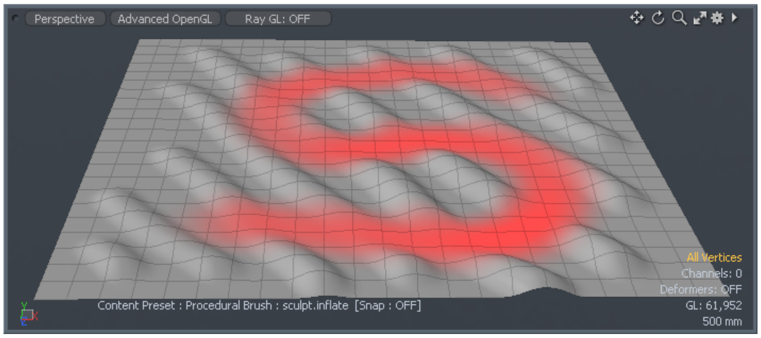
Adding Texture Commands

The **Add Texture** commands appear in the main **Paint** interface tab in the **Utilities** sub-tab with some additional options available in the **Hair Tools** sub-tab. You can use these as shortcuts to quickly add an appropriate image to the **Shader Tree** for the most common **Image Mapping** options.

Icon	Command	Description
	Add Color Texture	Creates a new, blank **Image Map** texture layer in the **Shader Tree** with **Diffuse Color** set as the **Effect**. (You must set the **Effect** as **Diffuse Color** to paint a surface with the paint tools.) Modo adds the image to the currently selected Material Group in the **Shader Tree** independent of the current foreground item selections. To ensure proper **Shader Tree** placement, first select the **Material Group Item** where the texture layer should reside. You may have to adjust the projection method in the Texture Locator. The image format defaults to the **Default Image Format** specified in the **Preference** dialog, but you can change the format in the **New Image** dialog when you create the image.
	Add Displacement Texture	Creates a new, blank **Image Map** texture layer (in the OpenEXR format) in the **Shader Tree** with **Vector Displacement** as the **Effect**. This format is the most flexible one for image-based sculpting with the sculpting tools. Modo adds the image to the currently selected Material group in the **Shader Tree** independent of the current foreground item selections. To ensure proper **Shader Tree** placement, first select the **Material Group Item** where you want the texture layer to reside. You may need to adjust

Icon	Command	Description
		the projection method in the Texture Locator. For best results, accept the default values of the **New Still** dialog and only adjust the resolution if necessary.
	Add Mask Texture	This command automatically creates a new, blank **Image Map** texture layer in the **Shader Tree** with **Layer Mask** as the **Effect**. Mask textures are useful in limiting the effect another texture layer has on a shaded surface. Modo adds the image to the currently selected Material group in the **Shader Tree** independent of the current foreground item selections. To ensure proper **Shader Tree** placement, first select the **Material Group Item** where you want the texture layer to reside. This layer's position in the **Shader Tree** is important because a **Layer Mask** only masks the texture layer directly above it in the tree. You may need to adjust the projection method in the Texture Locator**.** The image format defaults to the **Default Image Format** specified in the **Preference** dialog, but you can change the format in the **New Image** dialog.
	Add Fur Texture	Modo provides three **Add Fur** commands that automatically create new, blank **Image Map** texture layers (in the OpenEXR format) in the **Shader Tree**. • **Add Fur Texture**, once added to the **Shader Tree**, has **Fur Vector** as the **Effect**. • **Fur Direction** has **Fur Direction** specified as its **Effect**. • **Fur Length** has **Fur Length** specified as its **Effect**. These provide the most flexible method of styling **Fur** material layers with the styling tools. Modo adds the images to the currently selected Material group in the **Shader Tree** independent of the current foreground item selections. To ensure proper **Shader Tree** placement, first select the **Material Group Item** where you want the texture layer to reside (such as the group with the **Fur Material** item). You may need to adjust the projection method in the Texture Locator**.** For best results, accept the default values in the **New Still** dialog and only adjust the resolution if necessary.

Using Sculpting Masks



Ray GL: OFF
Channels: 0
Deformers: OFF
GL: 61,952
500 mm
Content Preset : Procedural Brush : sculpt.inflate [Snap : OFF]

You can use masking in Modo to limit sculpting to specific areas of a surface. Think of masks as sculpting shields. You define these areas by using the same brush-based tools used for other Modo painting and sculpting functions. When brushing over the surface with a mask tool, the masked areas change to a bright red color to denote their masked state. You cannot sculpt these areas until you erase or clear the mask. Masks are not simply on or off, but can fade across the surface to have a soft border between the masked and sculpted areas. Modo provides specific tools to control how soft- or hard-edged the masks are when you create them.

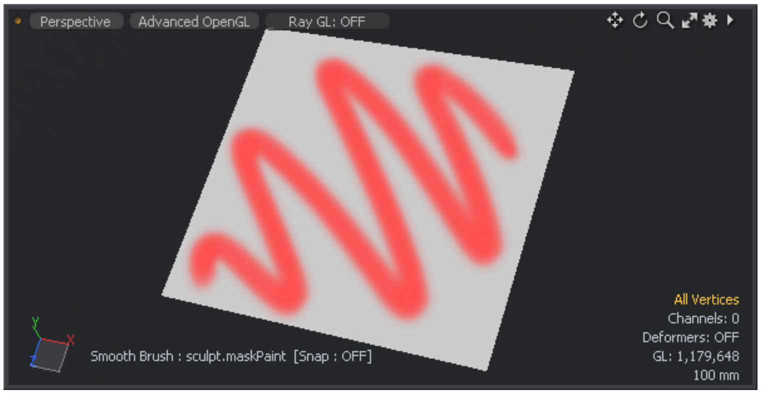
Usage

You apply masking with the mask brushes in the **Sculpt Tools** sub-tab of the toolbox in the **Paint** and **Sculpt** interface. Modo supports masking only with multiresolution sculpting. Therefore, the mesh must be in Catmull Clark subdivision mode and have the **Multiresolution** option enabled. For more information on how to set up multiresolution meshes for sculpting, see Multi-Res Sculpting. You can apply masking at any time while sculpting or prior to sculpting to keep certain areas from being affected at all. You can mask sculpted areas to shield them from, for example, having high-frequency surface details added.

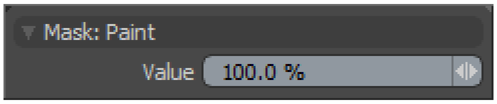
To create a mask, locate the masking tool icons in the toolbox. Click the **Mask Paint** tool to activate it. Right-click and drag directly in the 3D viewport to set the brush size. Then drag over the target surface to create the mask, which shows as a bright red color on the surface. The red coloring temporarily overwrites any texture applied to the surface, but the texture returns when you remove the mask. Mask tools also respond to pressure sensitivity if you apply the mask with a tablet and stylus. Once you create the mask, press the **spacebar** to deactivate the current tool. Now, you can select and apply any of the sculpting tools to the target surface. Masked areas remain unaffected by sculpting until you modify, disable, or clear the mask, itself. To clear a mask, click the **Clear Mask** button under the toolbox. You can also temporarily disable, invert, or adjust the mask's opacity. You can control the overall visibility of masks per viewport by clicking the gear icon in the upper-right corner to adjust the viewport options. To show or hide masks, select or clear the **Psub Mask** checkbox in the **Active Mesh** sub-tab.

Sculpting masks are convenient and easy to apply, but there are some details you should know. Modo determines the resolution of the mask by the subdivision level of the target surface. Also, Modo does not save sculpting masks with the mesh. Only one mask is available at any time and you cannot save masks separately. To create a mask that can be saved, use a falloff. (Sculpting tools also adhere to the application of falloffs.) You also cannot save falloffs directly, but Modo derives Vertex Map- or image-based falloffs from sources that can be saved.

Mask: Paint



The Mask Paint tool adds masking to the target surface. Modo determines how to add the mask by the brush tip. You can combine the Mask Paint tool with the Customizing Inks and Tablet Nozzles to customize brush strokes. While the Mask Paint tool is active, you can right-click and drag in the 3D viewport to set the size of the brush. Dragging over the target surface paints the mask into the surface additively for each stroke. Pressing **Ctrl** while dragging over the surface inverts the brush's effect to act like an eraser.

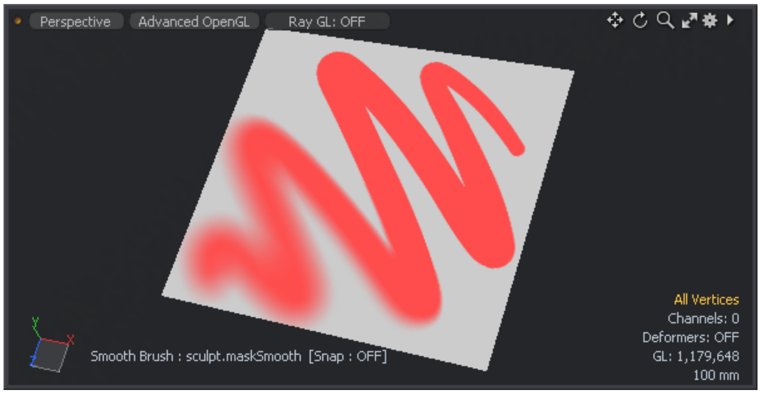


Mask: Paint

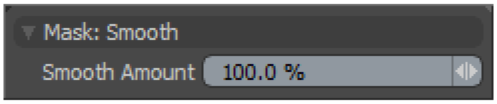
Value 100.0 %

Value: Determines the strength of the mask applied. If you use a pressure-sensitive stylus and tablet, this option specifies the maximum strength.

Mask: Smooth


Perspective
Advanced OpenGL
Ray GL: OFF
Smooth Brush : sculpt.maskSmooth [Snap : OFF]
All Vertices
Channels: 0
Deformers: OFF
GL: 1,179,648
100 mm

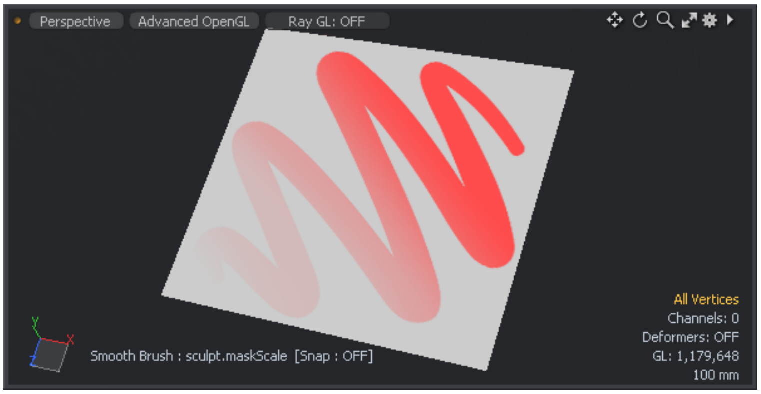
The Mask Smooth tool softens or blurs an existing mask. Modo blurs the mask based on the brush tip. You can also combine this tool with the Customizing Inks and Tablet Nozzles options to customize the brush strokes. While the tool is active, you can right-click and drag in the 3D viewport to set the size of the brush. Dragging over the target surface softens the mask, which gets softer with each additional stroke.


Mask: Smooth

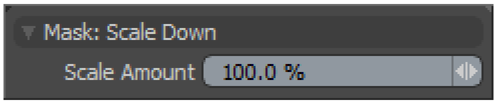
Smooth Amount 100.0 %

Smooth Amount: Determines the strength of the blurring effect on the mask.

Mask: Scale Down


Perspective
Advanced OpenGL
Ray GL: OFF
Smooth Brush : sculpt.maskScale [Snap : OFF]
All Vertices
Channels: 0
Deformers: OFF
GL: 1,179,648
100 mm

The Mask Scale Down tool decreases the strength of the target mask. How Modo scales the mask is determined by the brush tip. You can also combine this tool with the Customizing Inks and Tablet Nozzles options to customize the brush strokes. While the tool is active, you can right-click and drag in the 3D viewport to set the size of the brush. Dragging over the target surface decreases the strength of the mask, which gets weaker with each additional stroke. Pressing **Ctrl** while stroking over the surface inverts the brush's effect to strengthen (or increase) the mask's effect.

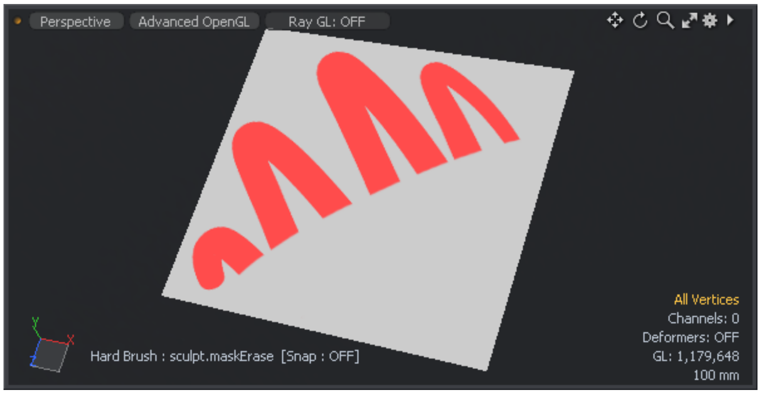


Mask: Scale Down

Scale Amount 100.0 %

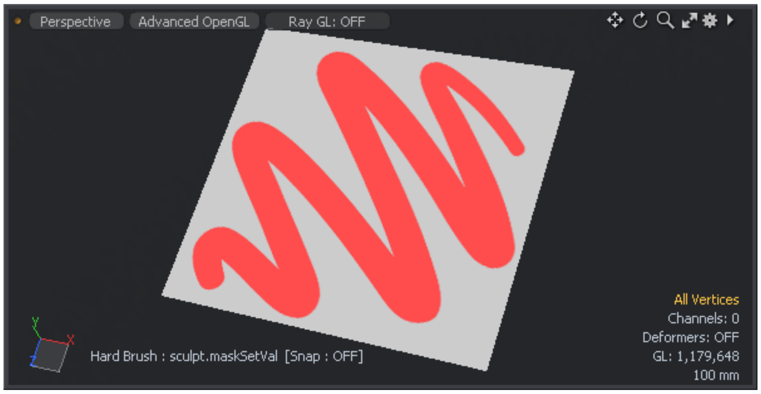
Scale Amount: Determines how quickly the values are scaled to adjust the overall strength of the scaling effect.

Mask: Erase

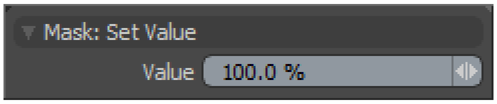

Perspective
Advanced OpenGL
Ray GL: OFF
Hard Brush : sculpt.maskErase [Snap : OFF]
All Vertices
Channels: 0
Deformers: OFF
GL: 1,179,648
100 mm

The Mask Erase tool completely removes masking from the target surface. While the tool is active, you can right-click and drag in the 3D viewport to set the size of the brush. Dragging over the target surface erases the mask from the surface. Pressing **Ctrl** while stroking over the surface inverts the brush's effect to act like the Mask Set Value tool. (The Mask Erase tool has no options panel.)

Mask: Set Value


Perspective
Advanced OpenGL
Ray GL: OFF
Hard Brush : sculpt.maskSetVal [Snap : OFF]
All Vertices
Channels: 0
Deformers: OFF
GL: 1,179,648
100 mm

The Mask Set Value tool applies the defined value to the mask to overwrite all existing values. This tool creates hard-edged, solid strokes. You can combine the Mask Set Value tool with the Tablet Nozzles option to customize the brush strokes. While the tool is active, you can right-click and drag in the 3D viewport to set the size of the brush. Dragging over the target surface applies the mask value to the target surface. Pressing **Ctrl** while stroking over the surface removes mask values to act like the Mask Erase tool.



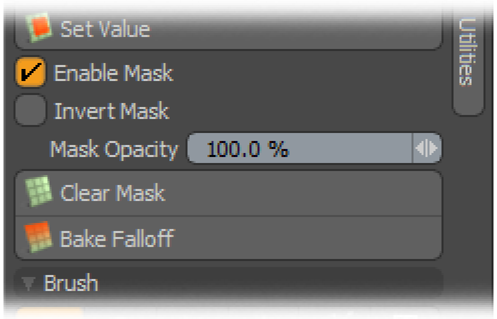
Mask: Set Value

Value 100.0 %

Value: Determines the strength of the mask applied to the surface.

Masking Controls

Directly under the masking tools in the Sculpting toolbox, Modo provides are several options related to sculpting masks.



☑ Enable Mask

☐ Invert Mask

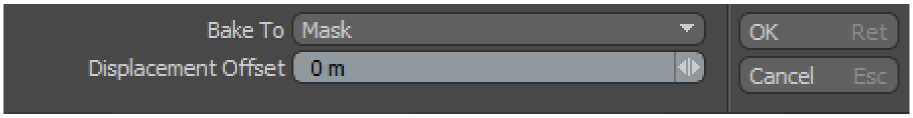
Mask Opacity 100.0 %

Enable Mask: By clearing this checkbox, you can temporarily disable a mask without having to clear or remove the mask. You can then use the mask at a later time during the same sculpting session. (Modo does not save masks with the model or scene.) When selected, the mask is active; when clear, the mask is inactive, but it is still visible.

Invert Mask: When enabled, inverts the effect of the mask (but not its viewport representation).

Mask Opacity: Determines the strength of the visible representation in the 3D viewport. This doesn't affect the mask strength, itself, but only changes how Modo displays it.

Clear Mask: Clears a mask from a surface quickly. Clicking this button completely removes all masking values from the surface.
Bake Falloff: Converts any active falloff into a sculpting mask. With any falloff active, clicking **Bake Falloff** opens the **Bake Falloff** dialog.



Bake To Mask

Displacement Offset 0 m

Bake To: Select either **Mask** to merge the current falloff into the current sculpting mask or **Displacement** to use the falloff as a means to displace (offset) the current surface.

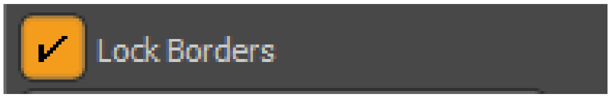
Displacement Offset: Controls the maximum amount of displacement at **100%** for the falloff when you set **Bake To** to **Displacement**.

Locking Borders and Corners

The Sculpt Tools provide locking options to allow you to lock borders and corners.

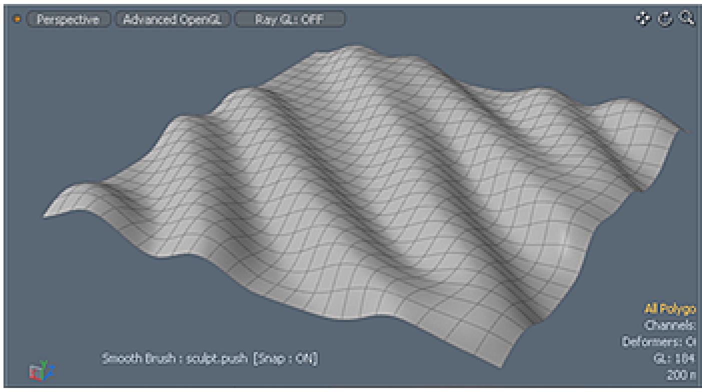
Lock Borders

You can find the **Lock Borders** checkbox in the **Paint** layout at the bottom of the Sculpting toolbox on the left panel. When enabled, **Lock Borders** locks the position of any open edges to make them immovable by sculpting tools. Setting this option has a variety of uses, but it is particularly useful for sculpting on a mesh where the edges need to seamlessly meet up with another mesh.

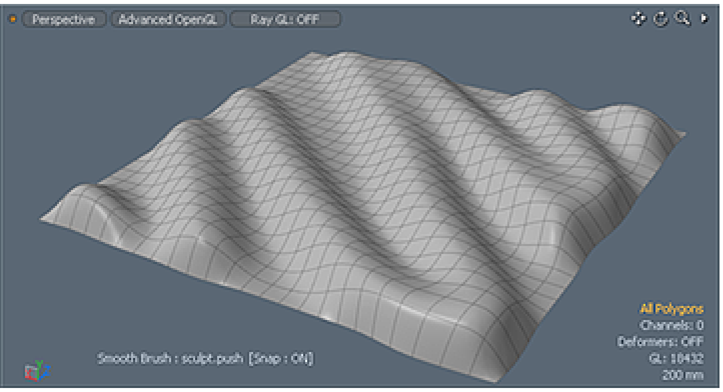


☑ Lock Borders

You can distinguish open edges by selecting all the geometry in a layer in **Polygons** mode by pressing **Alt**+**Shift**+**A**. Then press **Ctrl/Cmd**, which changes the **Edges** button to the **Boundary** button. When you click **Boundary**, Modo indicates all the open edges in the selection. If there are no edges indicated, then the geometry has no open edges and is considered water-tight.


Perspective
Advanced OpenGL
Ray GL: OFF
Smooth Brush : sculpt.push [Snap : ON]
All Polygo
Channels:
Deformers: O
GL: 184
200 n

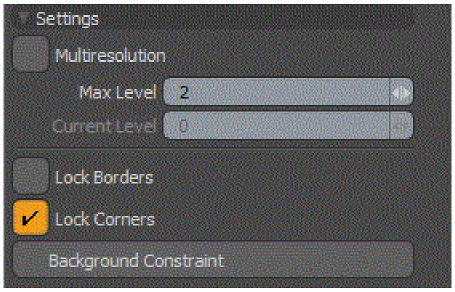
Lock Borders disabled during sculpting


Perspective
Advanced OpenGL
Ray GL: OFF
Smooth Brush : sculpt.push [Snap : ON]
All Polygons
Channels: 0
Deformers: OFF
GL: 18432
200 mm

Lock Borders enabled during sculpting

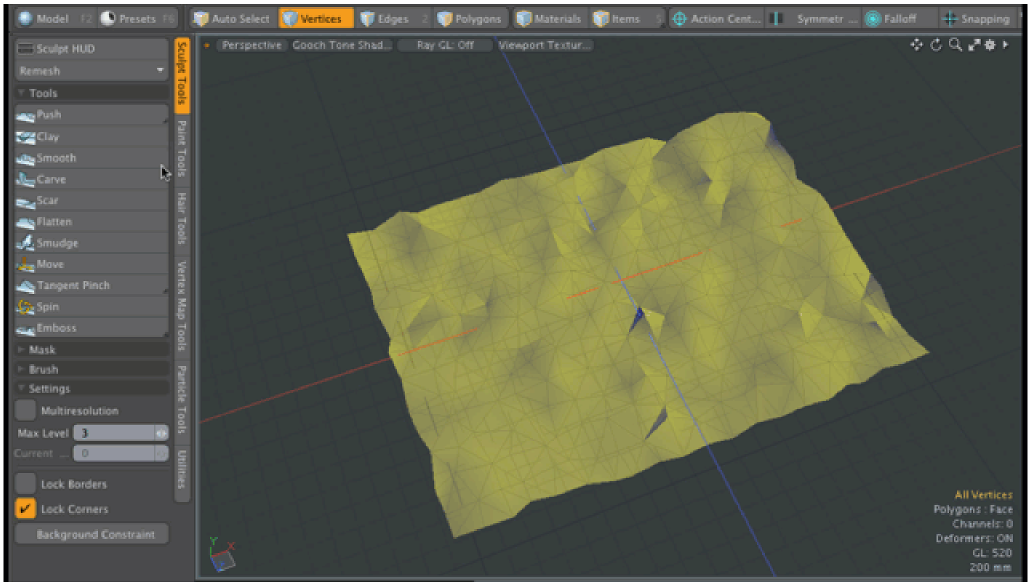
Lock Corners

You can find the **Lock Corners** checkbox in the **Paint** layout at the bottom of the Sculpting toolbox on the left panel. When enabled, **Lock Corners** locks the vertex positions at corner vertices with a valence of 2 and makes the corner vertex immovable by the sculpting tools. The corner vertex is shared by single surface polygon and it is on the Boundary.



Settings
☐ Multiresolution
Max Level 2
Current Level 0
☐ Lock Borders
☑ Lock Corners
Background Constraint

The following is an example of applying the **Smooth** tool and enabling the **Lock Corners** option. Corners are smoothed and remain sharp after smoothing.


Model
Presets
Auto Select
Vertices
Edges
Polygons
Materials
Items
Action Cent...
Symmetr ...
Falloff
Snapping
Sculpt HUD
Remesh
Tools
Push
Clay
Smooth
Carve
Scar
Flatten
Smudge
Move
Tangent Pinch
Spin
Emboss
Mask
Brush
Settings
Multiresolution
Max Level 3
Current ... 0
Lock Borders
Lock Corners
Background Constraint
Sculpt Tools
Paint Tools
Hair Tools
Vertex Map Tools
Particle Tools
Utilities
Perspective
Gooch Tone Shad...
Ray GL: Off
Viewport Textur...
All Vertices
Polygons : Face
Channels: 0
Deformers: ON
GL: 520
200 mm

Creating Sculpting Layers

Modo's multiresolution sculpting supports the layering of sculpting data, so you can separate and control sculpting in multiple passes. For example, you can adjust broad forms of a model on one sculpting layer, and then do smaller details on another layer independent of the first, and edit the opacity afterwards to further control the final look.

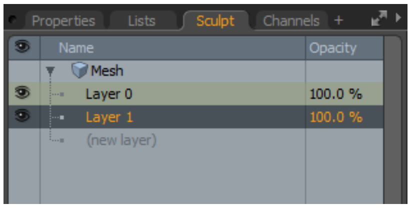
Layers also allow you to experiment non-destructively, as new layers can be removed without affecting other layers. Each new layer you add to an item affects all other layers, regardless of layer position, so one layer can add shape to a model, and the next layer can be used to remove the added forms. When combined with a sculpting modifier, you can control the strength of individual layers relative to values generated by another item, so corrective sculpts can be controlled by transform values, or any other criteria.

Sculpting layers are specific to Catmull-Clark multiresolution models, which you can enable for your selected Mesh Item in the **Painting/Sculpting** layout or directly in the item's **Properties** panel.

Sculpting layers can be found in the **Painting** interface, in the **Sculpt** viewport on the bottom right of the screen.

Note: If the viewport is not visible, you can add it manually by clicking on the **+** icon and selecting **Data Lists** > **Sculpting**.

The viewport shows selected meshes that have multiresolution enabled and their layers. When sculpting, the viewport always shows the blended result of all layers at their current opacity level.



Name	Opacity
Mesh	
Layer 0	100.0 %
Layer 1	100.0 %
(new layer)	

The **Name** column in the middle displays the layer's name. You can edit it by clicking the name and entering a new one. The eye icon on the left indicates the state of the layer. If the icon is turned off, the displacement vectors of the layer are not rendered. **Opacity**, on the right, specifies the strength of a layer's sculpting information as it applies its displacement to a mesh. Edit it by clicking on the field and entering a new value. You can enter both positive and negative values, and even drive values higher than 100% for interesting effects.

Adding a New Layer

To create a new layer, in the **Sculpt** viewport, click **(new layer)** below the existing layers. A new layer is added called **Layer** with a number appended to the end.

To edit a layer, right-click its name. In the pop-up menu the following commands are available.

Command	Description
Rename	Changes the name of the layer. The layer name must be unique in the mesh.
Duplicate	Duplicates the current layer.
Apply Opacity	Applies the opacity value to displacement vectors and resets the opacity to 1.0.
Merge Layers	Merges all layers in a mesh into the current layer.
Add Modifier	Adds a sculpting modifier to the current sculpt layer. This drives the **Opacity** value through the item channel.
Clear	Clears the displacement vectors in the current layer.
Remove	Removes the current layer.
Import	Imports the displacement vectors to the current layer from an external **XML** file.

Command	Description
Layer	
Export Layer	Exports the displacement vectors in the current layer to an external **XML** file.

Sculpt Modifiers

A Sculpt Modifier adds a channel item related to a specific sculpting layer that you can place in the **Schematic** viewport. With this, you can change the transparency of a particular sculpting layer automatically, when linked to another channel.

For example, if you bend a leg object with two joints, the knee area might suffer from the deformation, so a sculpting layer can be added to fix the shape. However, when the leg is straight, you no longer need the fix, so you can automatically drive the opacity of that layer depending on the rotation of the joints, using a modifier linked to the rotation of the lower leg joint.

Adding a Sculpt Modifier

The Sculpt Modifier can be added in one of two ways:

- Right-click on a layer in the **Sculpt** viewport, in the pop-up menu, click **Add Modifer**.
- Using the Preset Browser.

Single sculpt modifiers drive one layer's opacity. To control the opacities of multiple layers at once, add a modifier to each layer. The modifier identifies the layer with the modifier item name. If you rename a layer with the **Rename** command from the context menu, the modifier name linked to the sculpting layer is also renamed automatically.

The sculpting modifier is a kind of deformer, but it does not deform vertex positions.

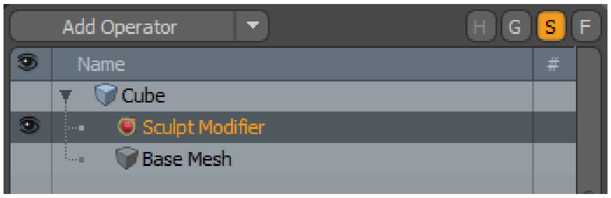
Sculpt Modifier Procedural

Sculpt Modifier is also available as a procedural mesh operation. You can find the tool in the **Mesh Ops** tab. In the **Model** layout, the tab is on the right side of the interface, next to the **Item List**.

To add the Sculpt Modifier mesh operation:

1. Click the **Add Operator** button, and navigate to **Mesh Operations** > **Other** > **Sculpt Modifier**, or enter its name in the search bar.
2. Double-click it to add it to the **Mesh Operations** list.

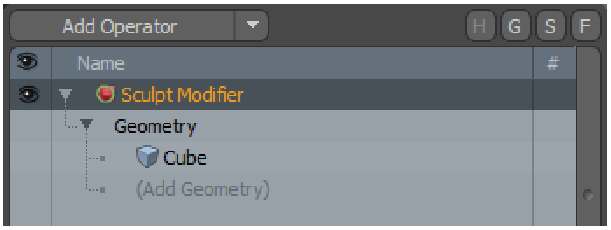
   Sculpt Modifier is added to the list.



3. To connect the Sculpt Modifier to a Sculpting Layer, rename it to match the name of the layer.

When the Mesh Operations list is not in Solo mode, you can expand the Sculpt Modifier item by clicking the small arrow in front of it. This reveals the sub-tools the operation uses.

Note: For more information on the visibility of the **Mesh Operations** list, see Solo Mode.



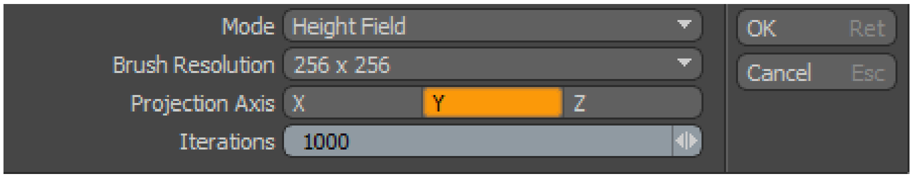
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

Using Sculpting Utility Commands

Modo provides several commands to help you use the sculpting tools. These commands are in the **Utilities** sub-tab of the default **Painting** menu.

Geometry to Brush

With this feature, you can convert any selected geometry into a brush for painting or sculpting. Modo can create height-field (grayscale image) or vector displacement (32-bit color image) brushes from the currently selected geometry. To have Modo convert the geometry properly, the geometry must be surrounded by open edges. Enclosed objects (referred to as water-tight objects), such as a sphere or torus, produce no results.



Mode Height Field
Brush Resolution 256 x 256
Projection Axis Y
Iterations 1000

Mode: Determines what type of brush Modo creates. You can specify one of two options: **Height Field** or **Vector Displacement**. **Height Field** creates a grayscale image that is useful for displacing geometry. It only takes into account the actual height of the geometry and ignores insets and undercuts. The **Vector Displacement** option is more accurate in capturing the contours of the geometry and accommodates insets and undercuts in the geometry.

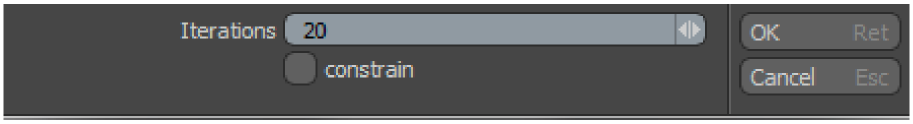
Brush Resolution: Defines the final resolution of the generated image map. Higher resolutions may be necessary to capture finer detail.

Projection Axis: Defines the facing direction of the geometry.

Iterations: Determines how Modo captures the image. Higher iterations produce more accurate results but take longer to calculate. The default setting of **1000** should be fine in most cases. If you feel the shape of the base geometry isn't being captured properly, try increasing the **Iterations** value.

Relax Mesh

This command uses the same mesh-smoothing algorithm as the **Geometry to Brush** command. With the **Relax Mesh** command Modo applies a number of iterations, which affect the strength of the smoothing. The default number of iterations is **5**. You use this command to convert a mesh with modeled detail into a sculpt-ready mesh. You can also use it to create UV maps for very detailed objects when applied to a morph map. To create a UV map, first create a morph Vertex Map, apply the **Relax Mesh** command to the new morph, and then UV map the simplified mesh. Once finished, you delete the additional morph, and Modo retains the UV map for the original mesh.

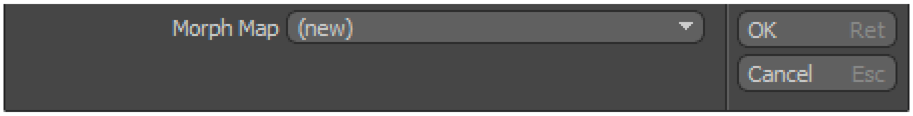


Iterations: Defines how many times Modo applies the relax operation to the model. Higher values produce increasingly smoother geometric surfaces.

constrain: When enabled, Modo takes into account additional, unselected geometry in the Mesh Item layer for the relaxed mesh. This gives you an additional degree of control over smoothing. You can place geometry inside a mesh, select the mesh to be smoothed, select the **constrain** checkbox, and click **OK**. In this case, the relaxed mesh conforms around the unselected geometry. This option is similar to using the **Background Constrain** function, but it requires the additional geometry to be in the same mesh layer.

Copy to Morph

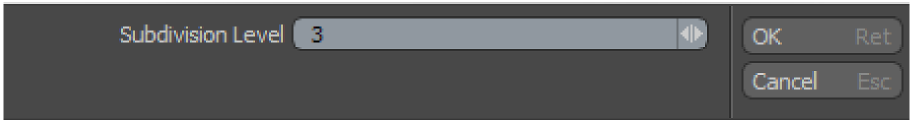
Use to copy the current state of your mesh into an absolute morph Vertex Map quickly. With this command, you can store the current sculpt state of your mesh to restore at any time with the Morph tool. This is only relevant to the changes made directly to the mesh and is not useful when working with image-based or multiresolution sculpting.



Morph Map: Creates a new morph Vertex Map or stores the sculpt information in an existing morph map. To create a new map, select **(new)**. To store to an existing morph map, select it from the list.

Convert to Multiresolution

With this command, Modo converts any image-based displacement information into multiresolution sculpting information. Modo can derive the displacement from either a sculpting or a standard image-based displacement (or both). For a procedural displacement, you need to bake it down to an image map first. To convert a mesh, select the target mesh and then apply the command, which is in the **Utilities** sub-tab of the Sculpting and Painting interface. With the dialog open, select the desired **Subdivision Level** and click **OK**.



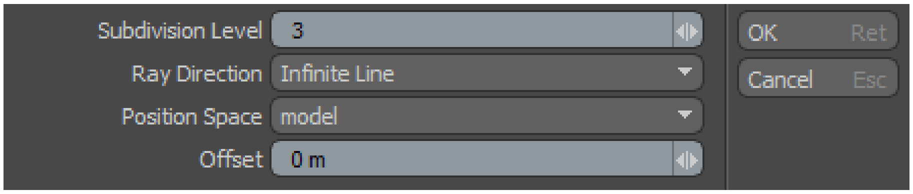
Subdivision Level 3

Subdivision Level: Specifies the maximum multiresolution **Subdivision Level**. Make sure to set the value high enough to capture all the details of the original displacement, but not so high that the mesh becomes unwieldy.

Background to Multiresolution Mesh

This command allows you to generate displacement vectors for multiresolution meshes. It raycasts background surface items from each position on the active multiresolution mesh along the normal vector. The delta vectors are stored as displacement vectors of the multiresolution mesh. You can use this command to quickly set up rough sculpture meshes from high-resolution template meshes.

To apply the command, select the target mesh in the **Paint** layout and click the **Utilities** sub-tab on the left, then select **Background to Multiresolution Mesh**. The following dialog displays.



Subdivision Level 3
Ray Direction Infinite Line
Position Space model
Offset 0 m

Subdivision Level - Specifies the maximum multiresolution subdivision level, with which Modo converts the active mesh to multiresolution Catmull-Clark polygons.

Ray Direction - Specifies how Modo computes the hit position from a point on the active mesh to the background.

- **Infinite Line** - Finds the closest position on the background surface from the point on the active mesh along an infinite normal line.
- **Normal Vector** - Uses half ray from the point on the active mesh if the background is larger than the active mesh.
- **Inverse Normal** - Uses inverse half ray from the point on the active mesh if a large active mesh contains background surfaces.

Position Space - Specifies the item space of the background item.

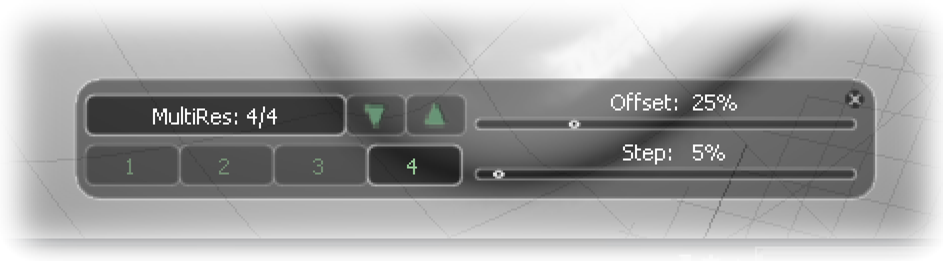
- **model** - Modo transforms the background item to the model space of the target item first, then it computes the hit position on the target item space. This is useful when the background item has different item transforms from the target item, but both meshes are identical on each item space.

- **world** - Modo computes the hit position from the target mesh on the background mesh in world space and then back from the World Position to the target Mesh Item space to store the results as multiresolution vectors.

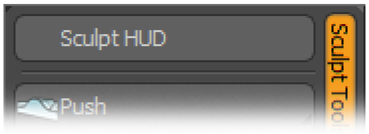
Offset: Offsets the hit positions along a normal vector at the hit position on the background surface.

Using the Multi Resolution Palette

A multiresolution mesh is a special mode of the Pixar Subdivision Surfaces geometry type meant specifically for sculpting, which allows you to interactively step up and down through multiple subdivision levels. You can easily sculpt a form at a lower subdivision level, step up the level to add fine details, and then return to a lower subdivision level to adjust the overall shape without affecting the fine details developed at the higher level.



When working with a multiresolution mesh for sculpting, Modo shows the **Multires** palette in the 3D viewport to give you a visual reference to the number of subdivisions applied, information about the current subdivision level, and the controls to step up and down through the levels. Additionally, you can adjust brush settings with the provided sliders. When enabled, this palette appears when any sculpting tools are active. You can show or hide the palette by clicking the **Sculpt HUD** button at the top of the **Sculpting** toolbox.



To properly set up a mesh as a multiresolution mesh, you need to make sure that the mesh is in Pixar's Catmull-Clark Subdivision mode. Catmull-Clark SubDs are differentiated by their rollover color. Regular SubDs are a pale lavender color on rollover, but Catmull-Clark Subds are a pale olive color. You may also use the Statistics viewport panel. Under **Polygon > By Type > Catmull-Clark**, click the **+** and **-** buttons to select and de-select while in **Polygons** mode to see which surfaces are designated as such. If you find a surface is a regular Subdivision surface, you can convert it by using the **Statistics** panel. Select the SubD geometry **Polygon > By Type > SubDivs**, click **+** to select, press **Tab** to disable, and then press **Shift**+**Tab** to convert the selection to Catmull-Clark. Then select the **Multiresolution** option in the Mesh Item's properties or directly in the sculpting interface to properly prepare the surface for multiresolution sculpting.

The palette, itself, has several buttons and sliders. With the up and down arrow buttons, you can sequentially step up or down through the multiresolution levels or you can go directly to a specific level with the numeric buttons at the bottom. Use the **Offset** slider to designate the currently active brush's strength, and adjust the brush **Step** slider to define how often a brush dot is dropped to the drawing surface. Lower **Step** values produce smoother strokes but can slow interactivity; high **Step** values may show the individual dots although that may be the desirable behavior. You can dismiss the palette by clicking the **x** button. To show it again, click the **Sculpt HUD** button.

UVing

UV mapping is an essential part of precision texture mapping. It can also be useful for controlling the application of procedural texture and gradient layers, as well as Anisotropic shading. Many functions and tools of Modo generate automatic UV maps, but you may find you need to modify or replace those maps. The following topics describe working with vertex and UV maps in Modo, and give an overview of the UDIM workflow.

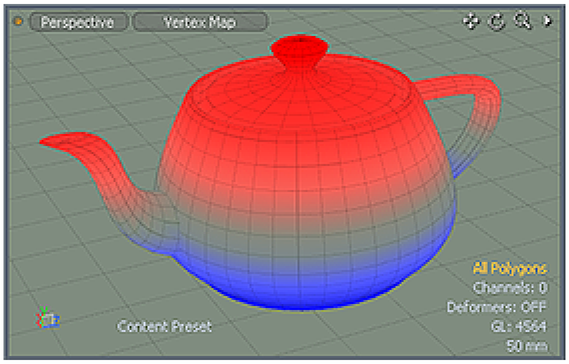
Working with Vertex Maps

Vertex mapping does not directly relate to texture mapping, but it can play an important role when texturing models for rendering. Vertex mapping is a means of storing vertex-specific information within a model for a variety of uses.

Vertex maps may act as instructions to tell a certain function how to act upon a model (as with Weight Maps). Some Vertex Maps store positions' offset information (as in morph maps). They can also store 2D coordinate information to define how textures apply to a surface by way of a UV map. Regardless of the map type, Vertex Maps are essential to working in 3D. As with modeling, there are a variety of options for creating, displaying, and working with Vertex Maps in Modo.

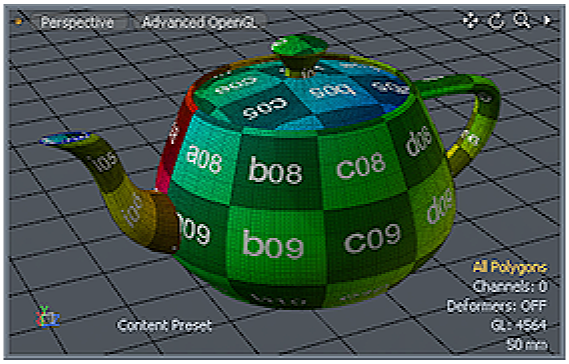
Modo offers the following vertex map types:

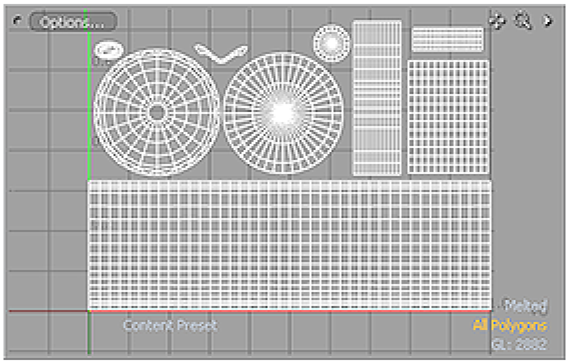
Weight Maps



Weight Maps are discontinuous, one-dimensional Vertex Maps holding weight percentages. You can use Weight Maps for a variety of purposes. For example, a Weight Map can modulate the strength of a tool's effect when combined with a weight falloff operator. As another example, you can control the smoothing properties of a Subdivision Surfaces model by adjusting the subdivision Weight Map - essentially controlling how round or sharp a corner crease becomes when converted to Subdivision Surfaces. B-spline curves also use Weight Maps to store vertex weight information. The **Shader Tree** can use Weight Maps for modulating surface attributes with masks and gradients. You can also export Weight Maps to other applications that support them for use in texture or deformation modulation. When you set a 3D viewport to **Vertex Map** shading mode, Modo represents a weight value of **0** with a neutral green shade; a negative weight value shows with a blue, “cool” shade; and a positive value appears as a red, “warm” shade.

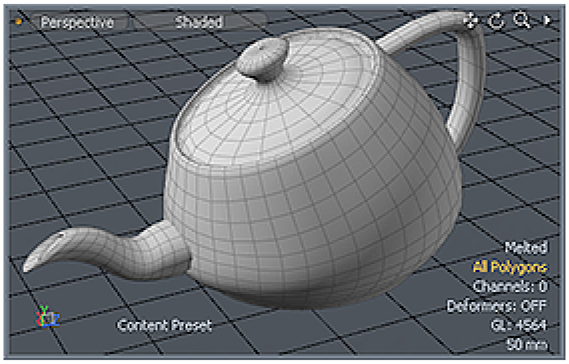
UV Maps


Perspective
Advanced OpenGL
a08
b08
c08
b09
c09
All Polygons
Channels: 0
Deformers: OFF
GL: 4564
50 mm
Content Preset


Options...
Content Preset
Melted
All Polygons
GL: 2882

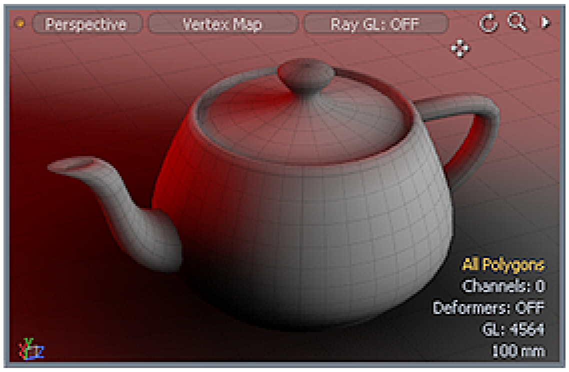
You use UV maps to explicitly link vertices in the 3D mesh to locations on a 2D image for high-accuracy texture mapping. Modo stores UV maps as discontinuous, 2D Vertex Maps with values for UV coordinate space data. The UV map can be though of as a method to unfold any 3D mesh so that it lays flat and allows the 2D image to be applied to the mesh more accurately.

Morph Maps


Perspective
Shaded
Melted
All Polygons
Channels: 0
Deformers: OFF
GL: 4564
50 mm
Content Preset

Morph maps offer a means of saving alternate, non-destructive deformations for any model. You can animate individual morph map deformations over time. For example, this provides the ability to animate a facial expression. To animate any morph map, you must first apply a morph deformer to the model. Modo supports two varieties of morph maps: relative and absolute. Relative morph maps offset a particular vertex relative to its current position. An absolute morph map positions a vertex to a specific absolute position. Relative Vertex Maps are the more common variety. The main benefit to working with morph maps in Modo is that if you make any topology changes to a specific morph map, Modo applies those changes to all the associated maps.

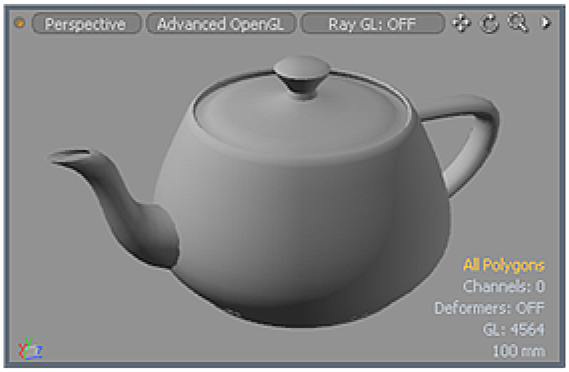
Color Maps



Modo provides support for RGB and RGBA vertex color maps. Many use vertex color maps in games to provide additional color variations in combination with texture maps to simulate a fully-shaded environment with little (to no) CPU shading overhead. You can paint vertex color maps with the **Vertex Map > Color Tool** combined with an airbrush-type falloff.

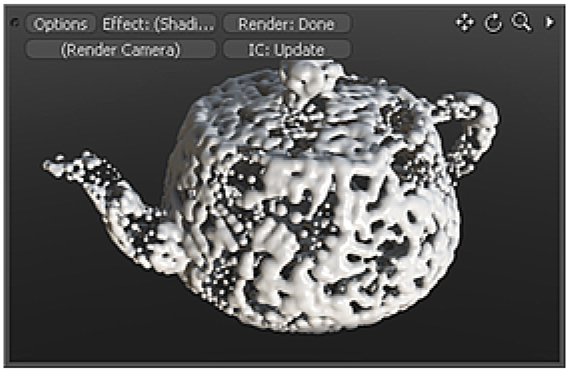
Note: For more information on using a vertex map for rendering, see Vertex Map Texture.

Vertex Normal Maps


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 4564
100 mm

Vertex normal maps work in tandem with the smoothing function to shade polygons as either faceted or smooth. A vertex normal map is a map with the values of the **Smoothing** setting baked into the vertices. Most often, you generate vertex normal maps in an external CAD package when converting a NURBS-based model to polygons. The normal values are computed from the smooth NURBS surface to eliminate smoothing errors of the highly triangulated polygonal surface when rendered elsewhere.

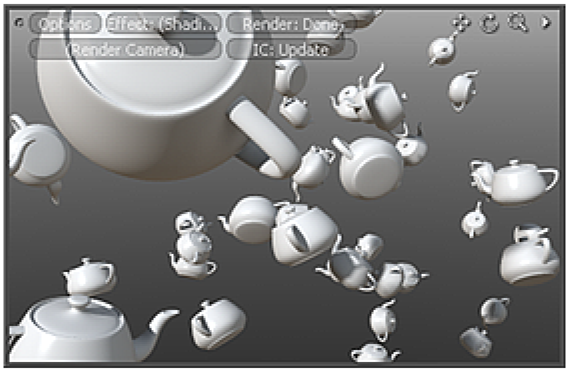
Particle Maps


Options
Effect: (Shadi...
Render: Done
(Render Camera)
IC: Update

Particle maps are useful to give you control over particle- and volume-related items (such as when using point clouds for Replicators and Texture Replicators) and for use with blobs, volumes, and sprites. (Point clouds are just masses of particles.) Particle maps include dissolve maps, which control the opacity of individual particles, and size maps, which control the scale of individual particles.

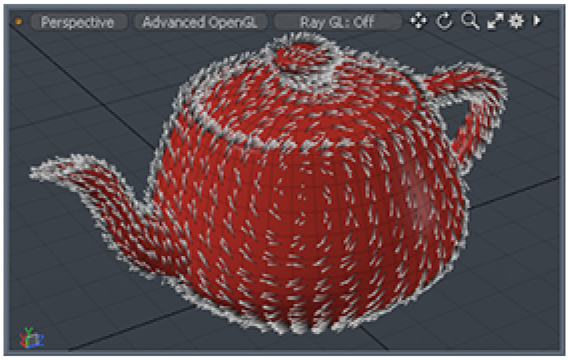
Note: Particle maps cannot be applied to procedural particle items, such as particle clouds and the Particle Generator.

Transform Maps



Vertex transform maps are useful as they give you individual control over Replicators. You can apply a transform map and then select a target vertex (with a replicated element applied). With the standard transform tools, you can adjust rotations, scales and positions on an individual particle basis.

Vector Maps

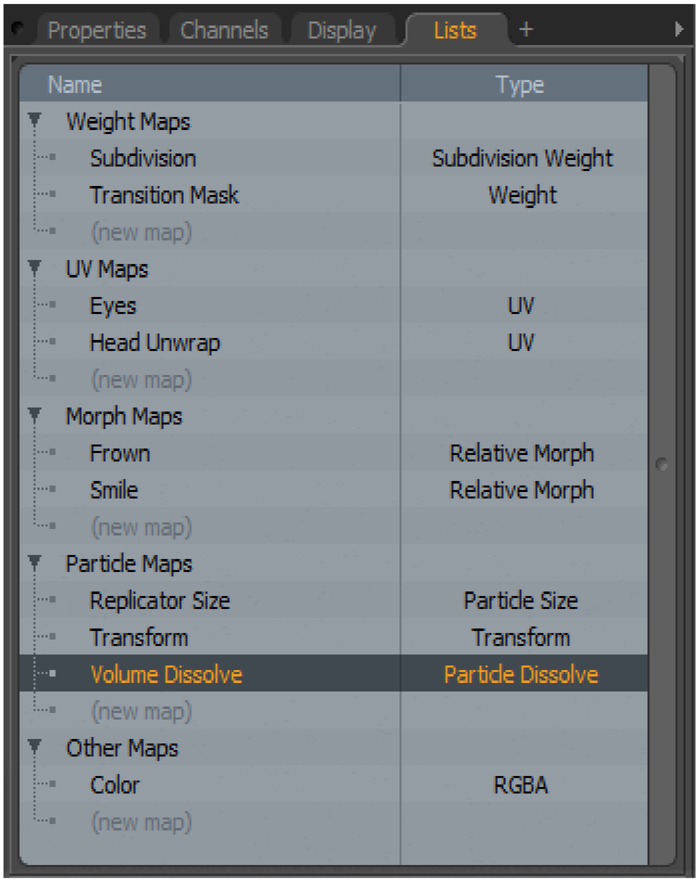


With vertex vector maps you can comb and (to a degree) style fur by using an embedded Vertex Map instead of resorting to more complex image maps or guides. Vertex vector maps also work in a similar fashion to control the scale and orientation when applied to Replicators when you use a Surface Generator as a point source. This avoids creating unnecessary UV maps and keeps objects self-contained in that they don't require any external assets to function.

Other Maps

Modo also supports various other maps types, such as pick maps. Having other maps available is mainly for storing of selection sets and retaining compatibility with other applications. See Managing Vertex Maps

Lists Viewport

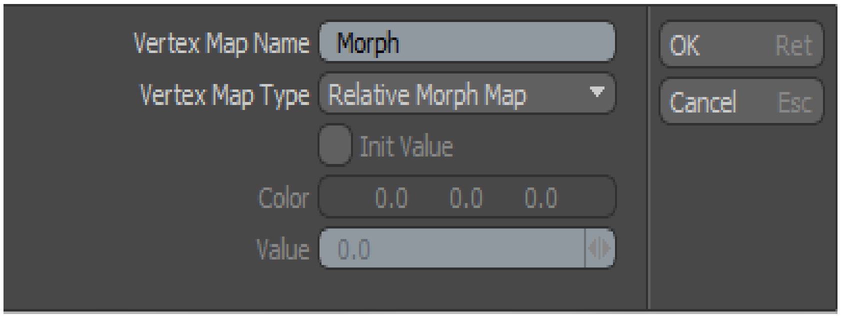


Name	Type
Weight Maps	
Subdivision	Subdivision Weight
Transition Mask	Weight
(new map)	
UV Maps	
Eyes	UV
Head Unwrap	UV
(new map)	
Morph Maps	
Frown	Relative Morph
Smile	Relative Morph
(new map)	
Particle Maps	
Replicator Size	Particle Size
Transform	Transform
Volume Dissolve	Particle Dissolve
(new map)	
Other Maps	
Color	RGBA
(new map)	

Modo stores map-specific information within the model, itself, at the vertex level, but Modo organizes Vertex Maps, like item layers, in the **Lists** viewport panel. Within the viewport, you can select maps to make them active for editing. You can also create, delete, and rename maps from within the **Lists** view. You access the various options by right-clicking a Vertex Map's name and choosing one of the available commands. In general, you may not work with the **List** view as much as the Item List or Shader Tree viewports, but the functions it provides are essential to working with Vertex Maps. All the attributes of the viewport are covered in the **Vertex Map List** viewport topic.

Tip: Vertex maps are item-layer specific with each Mesh Item having its own dedicated maps. The only maps visible within the **Lists** viewport are for layers that are visible (controlled by the **Item List**). Different layers can share map names (and usually do if you use the default names that Modo generates), but this can cause collision problems and produce incorrect results when baking out texture maps for rendering or export.

Managing Vertex Maps


Vertex Map Name: Morph
Vertex Map Type: Relative Morph Map
☐ Init Value
Color: 0.0 0.0 0.0
Value: 0.0

Before adding a map to a mesh, first determine the type of map required.

Map Type	Description
Weight Map	Weight strength values stored for falloff and texturing purposes.
SubD Weight Map	Weight strength value influences edge creasing in SubDivision Surface geometry.
UV Map	UV maps translate 3D vertex positions to flat 2D coordinates.
Relative Morph Map	Vertex position offset is relative to the base vertex position.
Absolute Morph Map	Vertex position is offset to a specific absolute position in 3D space.
RGB Map	Vertex color map is defined by R, G, and B color values.
RGBA Map	Vertex color map defined by R, G, and B color values and an additional A (Alpha) value.
Pick Map	Like a selection set, defines groupings of vertices.
Vertex Normal Map	Surface normal direction (smoothing) values stored as fixed value.
Edge Pick Map	Like a selection set, defines groupings of edges.
Particle Size Map	Determines scale values for individual particles.

Map Type	Description
Particle Dissolve Map	Determines transparency values for indvidual particles.
Transform Map	Determines transform amounts for individual vertices or particles.
Vector Map	Determines length (or scale) and orientation over the target elements.
Tangent Basis	Surface normals in Mikk tangent space.
B-Spline Weight Map	B-spline vertex weight values stored for vertices that define a curve.
Particle Item Map	Defines the input mesh item the replicator creates at a given vertex.

Creating Vertex Maps

To create a new map:

1. Select **Vertex Map > Create**.

   This opens the **Create New Vertex Map** dialog.

Tip: You can also select **(new map)** in the **Lists** viewport to open the dialog. Modo pre-selects a type based on the viewport's map type.

2. Select the map type and assign a name to the map.
3. With some map types you can set a base value. Click **Init Value** and adjust the value accordingly.

   The type of the initial value varies according to the map type. For example, an **RGB Map** defines color, whereas a **Weight Map** contains weight percentages.
4. Click **OK** to create the map.

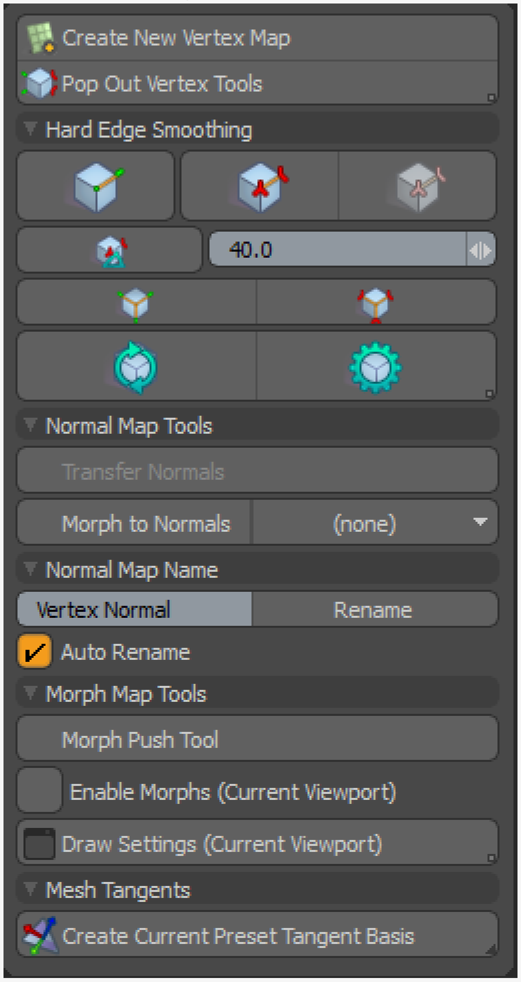
To rename a map, right-click it and select **Rename**, or highlight the map and select **Vertex Map > Name**. Both actions open a dialog and you can type the new name. Make sure to select the proper map type if necessary. Also, you may use the click, pause, click method to rename the layer inline.

To remove a map, select the target map and choose the **Vertex Maps > Delete** command, or select the layer and press the **Delete** key.

Viewing and Editing Vertex Maps

How you display and edit Vertex Maps depends on the map type. For most types of maps, do this directly in the 3D Viewports, which provides real-time feedback when you change values.

Tip: To quickly open a panel of useful UV Map tools. click **Vertex Map** > **Vertex Normal Tools**.


Create New Vertex Map
Pop Out Vertex Tools
Hard Edge Smoothing
40.0
Normal Map Tools
Transfer Normals
Morph to Normals
(none)
Normal Map Name
Vertex Normal
Rename
Auto Rename
Morph Map Tools
Morph Push Tool
Enable Morphs (Current Viewport)
Draw Settings (Current Viewport)
Mesh Tangents
Create Current Preset Tangent Basis

Weight Maps

Weight Maps display directly in any 3D viewport when you specify the viewport display option is set as type **Vertex Map**. You can edit Weight Maps by selecting components and applying values or by using the Vertex Map Weight Tool function. If you combine the Weight tool with an Airbrush Falloff falloff, you can interactively paint weight values.

Particle Item Maps

When you have a Replicator with a **Point Source** mesh item and a **Prototype** mesh item, a particle item map allows you to define which input prototype mesh item the Replicator creates at a given vertex.

Create a Replicator using any mesh as the **Point Source**, and multiple mesh items as the **Prototype**. You can add a group of mesh items as the **Prototype**.

Note: See Groups Viewport for more information on creating groups of meshes.

Once a Particle Item Map is created (see Creating Vertex Maps), all replicated items show the same prototype. To edit the prototype at a specific vertex, you need to work out the appropriate vertex map value to set.

Particle item values always range from 0.0 to 1.0, with values for each potential prototype at even distribution through that range. In other words, a replicator with three prototype inputs has three different ranges of values, (0.0-0.33), (0.33-0.66), and (0.66-1.0) :

- If the value is from 0.0 to 0.33, the first prototype in the group is used.
- If the value is between 0.33 and 0.66, the second prototype in the group is used.
- If the value is larger than 0.66 then the third prototype in the group is used.

To update the values, make sure you have a particle item map selected in the **Vertex Map List**. Then you can either use **Set Value** in the **Vertex Map** menu, or paint values using the **Set Value** tools in the **Vertex Map Tools** tab.

UV Maps

UV maps have their own 2D viewport, which is specific to editing maps of these types. 3D Viewports only display the result of the UV map in the **Texture**, **Texture Shaded**, or **Advanced OpenGL** modes. Modo has a dedicated UV-editing layout, by default, with additional windows available under the **Layout > Windows** sub-menu.

Morph Maps

When you select a morph map in the **Lists** viewport, it is visible within any of the 3D Viewports in all of the various modes. When animating morph maps and applying them as deformers, you see the morph, itself, only if you select the **Enable Deformers** checkbox within the viewport options. (You can access these options by pressing **O** when moveing the pointer over the target viewport.) **Enable Deformers** is enabled (as the default option) for the **Layout**, **Animation**, and **Render** layouts, but it is disabled in the **Modeling**, **Painting**, and **UV** view Layouts.

Color Maps

When you select a vertex color map in the **Lists** viewport, Modo displays it directly in any 3D viewport if the viewport display option is set as type **Vertex Map**. You can edit color maps by selecting components and applying values or by using the Vertex Color Tool function. If you combine the Color tool with an Airbrush falloff, you can interactively paint vertex color values.

Vertex Normal Maps

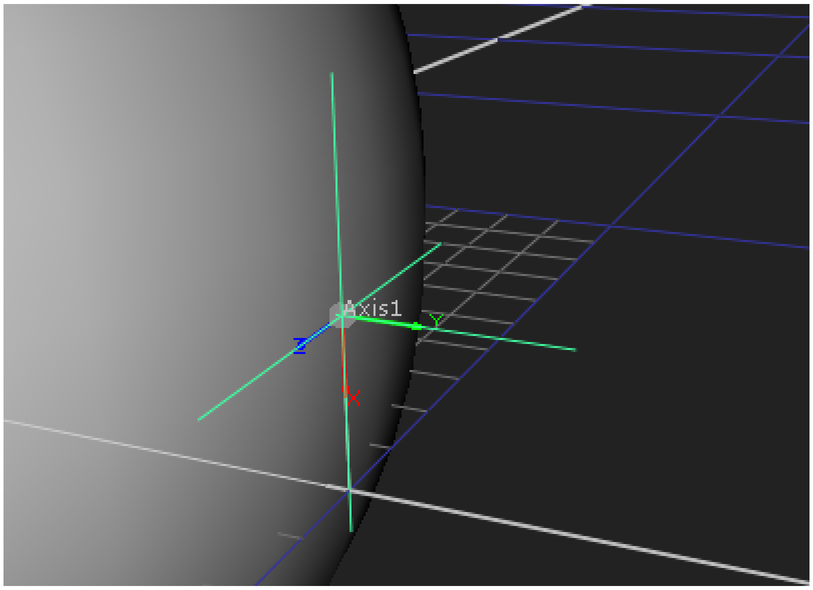
Vertex normals, when present, override any smoothing settings (as defined by material item settings in the Shader Tree). You usually generate vertex normal maps in an external application and use them when rendering highly-triangulated CAD in Modo. However, with Modo you can bake the material item's smoothing results into a vertex normal map with the Set Vertex Normals command.

Vertex Vector Maps

A vertex vector map, when active, appears as simple lines that project outward in the normal direction from each vertex in a model. When you select a vertex vector map in the Lists viewport, you can edit the map directly with the fur styling tools and modify their orientation. However, vectors are simple direction and size controls, so complex effects (such as **Curl** or **Root Puff**) won't have the same effect as they would when styling with guides. Therefore, with a vertex vector map, work with simple, short styles. You can also use vertex vector maps to control the scale and orientation of Replicators when combined with Surface Generators.

About Tangent Space

Tangent space provides a way of locating normals on a surface that differs from world space and object space. A tangent space originates from a point on a surface and has one axis that is at a tangent to the surface. The other two axes are perpendicular to the tangent, with one as the surface normal at the origin.



The image above shows a tangent space at a point on a sphere. The Y-axis is coincident with the surface normal.

Note: A Tangent Basis Vertex Map only stores its data relative to the surface. This means that the tangent space is unaffected by mirroring, rotation, scaling or translation.

Modo creates tangent space normal maps when using the **Bake from Object to Texture** option with an **Image Map** of type **Normal**. For more information on texture baking, see Baking.

Note: Because tangent space normal maps are automatically aligned to the underlying surface, a tangent basis texture can be reused when the geometry is deformed. Object space maps are less flexible as the object space changes under deformation.

Tangent space normal maps also provide certain efficiency savings over object space normal maps as they can be encoded with less data.

Mikk Tangent Space

In 2008, Morten Mikkelsen, a student of the University of Copenhagen, wrote his masters thesis: "Simulation of Wrinkled Surfaces Revisited". In it he proposed a method for calculating surface normals in tangent space. His method solves certain problems regarding hard edges, and portability of tangent spaces between 3D graphics engines. Subsequently, Mikk tangent spaces are now supported by a number of popular engines.

Modo lets you apply a Mikk Tangent Basis to vertex maps. This provides greater compatibility with the Unreal engine and other engines that support Mikk tangent space basis.

Source Tangent Basis Maps

Modo supports the generation of tangent basis maps that are compatible with Valve's Source engine.

Legacy Unity Tangent Basis Maps

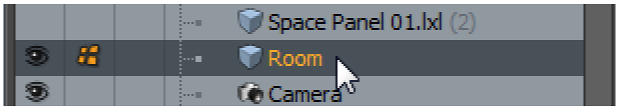
As of version 5.3, the Unity engine switched to using Mikk tangent basis maps as their default tangent basis standard. However, Modo lets you generate tangent basis maps using Unity's legacy standard, for which Unity maintains support.

Adding a Tangent Basis Map

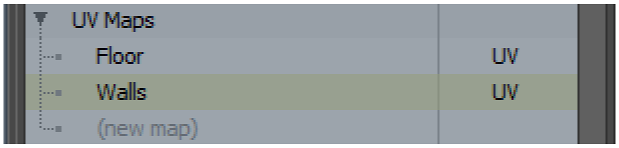
You can add a Tangent Basis to UV Maps that are active on selected meshes. The Tangent Basis must be added before baking.

Note: If you edit the mesh, you must reselect **Create ... Tangent Basis** prior to the next bake to update the Tangent Basis Normal Map.

5. Select the Mesh Item in the **Item List**.



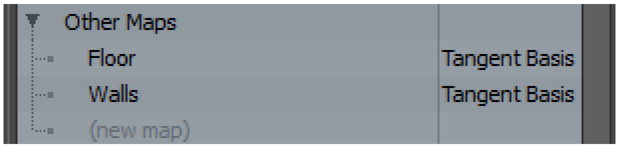
6. If you have more than one UV Map and want to apply a Tangent Basis to certain maps, open the **Lists** tab and select the UV Maps. Otherwise, leave the map(s) unselected to create Tangent Bases for all UV maps.


UV Maps
Floor UV
Walls UV
(new map)

Tip: You can **Ctrl**+click to select multiple maps.

7. From the menu, select **Vertex Map** > **Create ... Tangent Basis**, selecting the required tangent basis type.

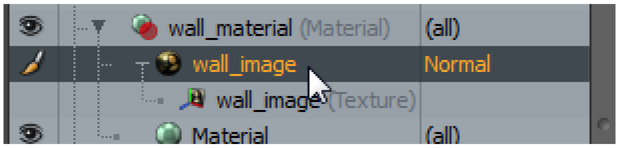
   A new **Tangent Basis** map is populated for each UV map, and added to the **Vertex Map List**.


Other Maps
Floor Tangent Basis
Walls Tangent Basis
(new map)

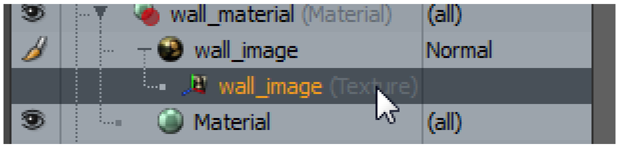
Baking for Unreal Engine 4.7+

For versions of the Unreal Engine at 4.7 or above, you should make the following changes prior to baking:

1. Select the Normal Image Map in the **Shader Tree**.


wall_material (Material) (all)
wall_image Normal
wall_image (Texture)
Material (all)

2. In the **Properties**, open the **Texture Layers** tab. Check **Invert Green**.
3. Select the Image Map's texture in the **Shader Tree**.


wall_material (Material) (all)
wall_image Normal
wall_image (Texture)
Material (all)

4. Open the **Texture Locator** tab and set the **Tangent Vector Type** to **dPdu, Cross Product**.

Note: For improved compatibility, low poly meshes should be converted into triangles before baking and exporting to FBX.

About Empty Tangent Basis Vertex Maps

Modo lets you add a Tangent Basis Vertex Map in the Vertex Map List using the **(new map)** option. Though, this creates an empty map, which is not directly usable by real-time engines. Also, there are no tools to edit these maps directly. The option has been added for scripting purposes. Modo artists can create empty placeholder Tangent Basis maps, which can be populated with custom maps by bespoke scripts to ensure compatibility with in-house graphics engines.

Vertex Mapping

Vertex maps provide functions for modeling, shading, and rendering. The tools and commands that Modo provides are useful for generating a variety of maps and for editing the results to fit your specific needs. For information about Vertex Maps, see Working with Vertex Maps. For information about a specific function, click the function below to view its topic.

Vertex Map Commands and Tools

Vertex Normal Tools

Create New Vertex Map

Clamp

Apply Morph

Edge Weight Tool

Set Vertex Map Name

Quantize

Background to Morph

Set Vertex Normals

Copy

Jitter

Vertex Map Math

Transform Vertex Normals

Paste

Unify

Vertex Map Math Tool

Morph Tool

Clear

Cull

Vertex Shader Baking

Background to Morph Tool

Delete

Mirror

Vertex Curvature Baking

Show Vertex Normals

Set Value

Transfer Vertex Map

Vertex Color Tool

Set Vertex

Smooth Vertex Map

Vertex Map Weight Tool

Vertex Normal Tools

Modo provides a set of tools for working with vertex normals. Use these tools to control the shading on the mesh using a system of hard and soft edges, as defined by vertex normals. Vertex normals provide an efficient way to smooth shade a polygon mesh, giving it the appearance of a much higher poly mesh. Vertex normal smoothing provides much more control over edge hardness than material smoothing. You can use these tools to gain more control of vertex normals and edge smoothing while creating art used for game engines.

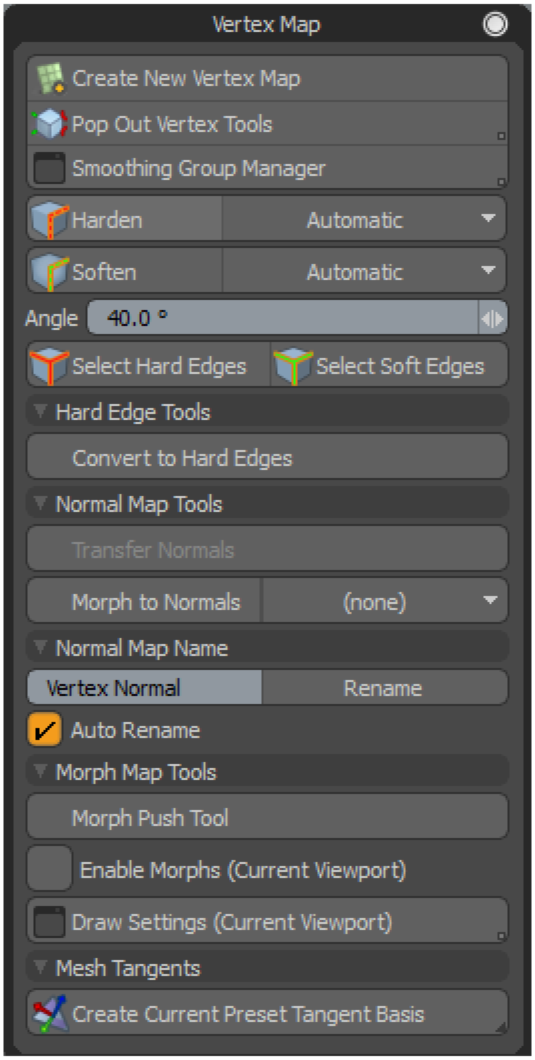
Often when you are working on a low resolution model, you may end up with shading errors on hard angles creating artifacts. You can set options to calculate the normals, and use smoothing groups for blending to resolve such problems.

You can use the **Vertex Map** tools to quickly select hard or soft surfaces and edit an existing vertex map. If none exists, one is automatically added that holds the normals for the entire mesh. You can then select edges and edit the map as required. The vertex map is automatically updated if you alter the mesh and the map is recalculated for the updated normals on your mesh.

Opening the Vertex Normal Tools

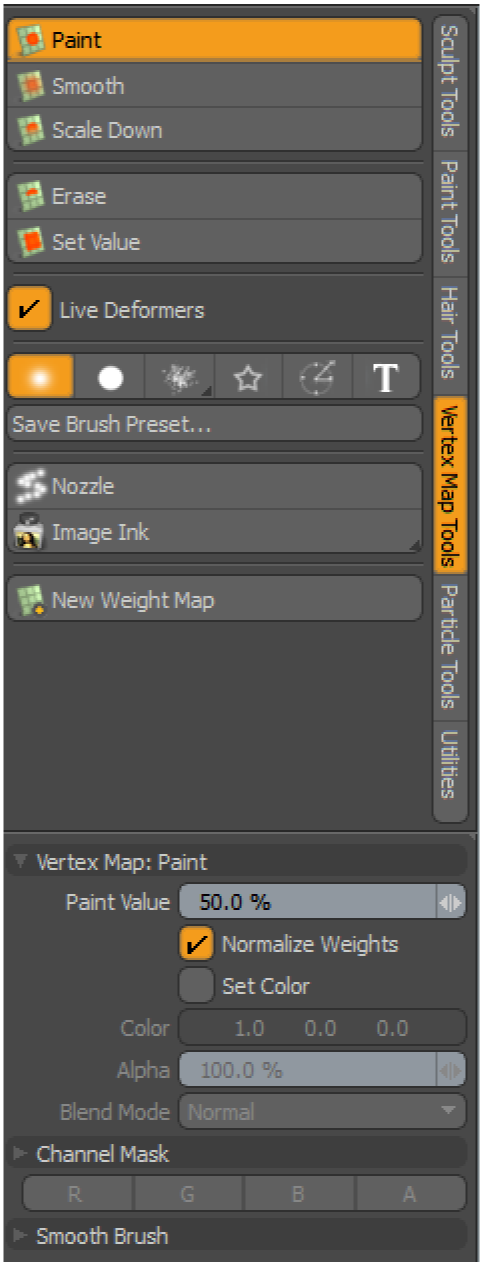
The Vertex Normal tools can be found in the following locations:

- From the menu bar, click **Vertex Map** > **Vertex Normal Tools...**.


Vertex Map
Create New Vertex Map
Pop Out Vertex Tools
Smoothing Group Manager
Harden
Automatic
Soften
Automatic
Angle
40.0 °
Select Hard Edges
Select Soft Edges
Hard Edge Tools
Convert to Hard Edges
Normal Map Tools
Transfer Normals
Morph to Normals
(none)
Normal Map Name
Vertex Normal
Rename
Auto Rename
Morph Map Tools
Morph Push Tool
Enable Morphs (Current Viewport)
Draw Settings (Current Viewport)
Mesh Tangents
Create Current Preset Tangent Basis

Tip: You can work with a smaller sub-set of these tools. For more information, see Using the Pop Out Vertex Tools.

- Alternatively, open the **Game** layout and open the **Vertex Map** tab on the left panel.
- If you want to adjust vertex color maps and vertex weight maps, open the **Paint** layout. On the left panel, open the **Vertex Map Tools** sub-tab, and click **Paint**. For more information, see Vertex Map Paint.


Paint
Smooth
Scale Down
Erase
Set Value
Live Deformers
Save Brush Preset...
Nozzle
Image Ink
New Weight Map
Sculpt Tools
Paint Tools
Hair Tools
Vertex Map Tools
Particle Tools
Utilities
Vertex Map: Paint
Channel Mask
R
G
B
A
Smooth Brush

☑ Live Deformers

Paint Value 50.0 %

☑ Normalize Weights

☐ Set Color

Color 1.0 0.0 0.0

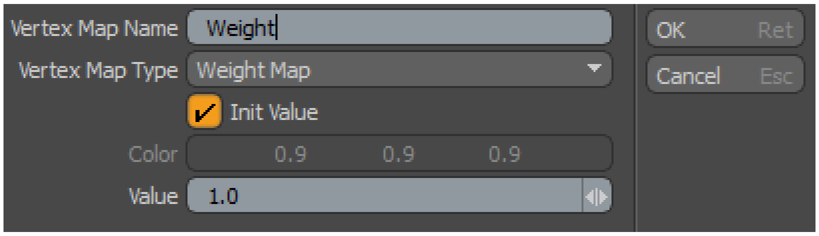
Alpha 100.0 %

Blend Mode Normal

Creating a New Vertex Map

Vertex maps provide functions for modeling, shading, and rendering. These maps work in tandem with the smoothing and hardening functions to shade polygons. You can create a new vertex map, provide a unique name for the map, and set an initial value. Once created, the named map immediately appears within the appropriate section of the **Lists** viewport.

1. On the Vertex Map dialog, click **Create New Vertex Map**.



Vertex Map Name: Weight
Vertex Map Type: Weight Map
☑ Init Value
Color: 0.9 0.9 0.9
Value: 1.0

2. Edit the fields and click **OK**.

   For more detailed information about these options, see Create New Vertex Map.

> **Note:** You can apply smoothing options to an existing Material. Vertex normals are calculated as part of Modo's native smoothing and they update as geometry is edited and when applied deformers change the mesh item.

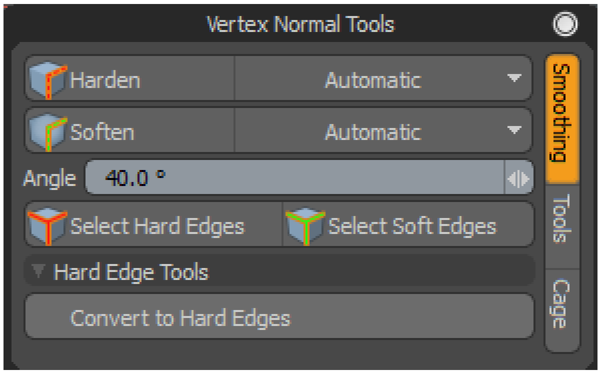
Using the Pop Out Vertex Tools

You can use a small palette of vertex normal tools, which provides easy access to common tools used for smoothing, normal maps, and cage options. From this palette you can select several different types of geometry to apply changes.

1. On the Vertex Map dialog, click **Pop Out Vertex Tools**.

   Alternatively, press **Alt** and click **Pin Vertex Tools**.

   This opens a small tool palette.



Harden: Automatic
Soften: Automatic
Angle: 40.0 °

Using the Smoothing Group Manager

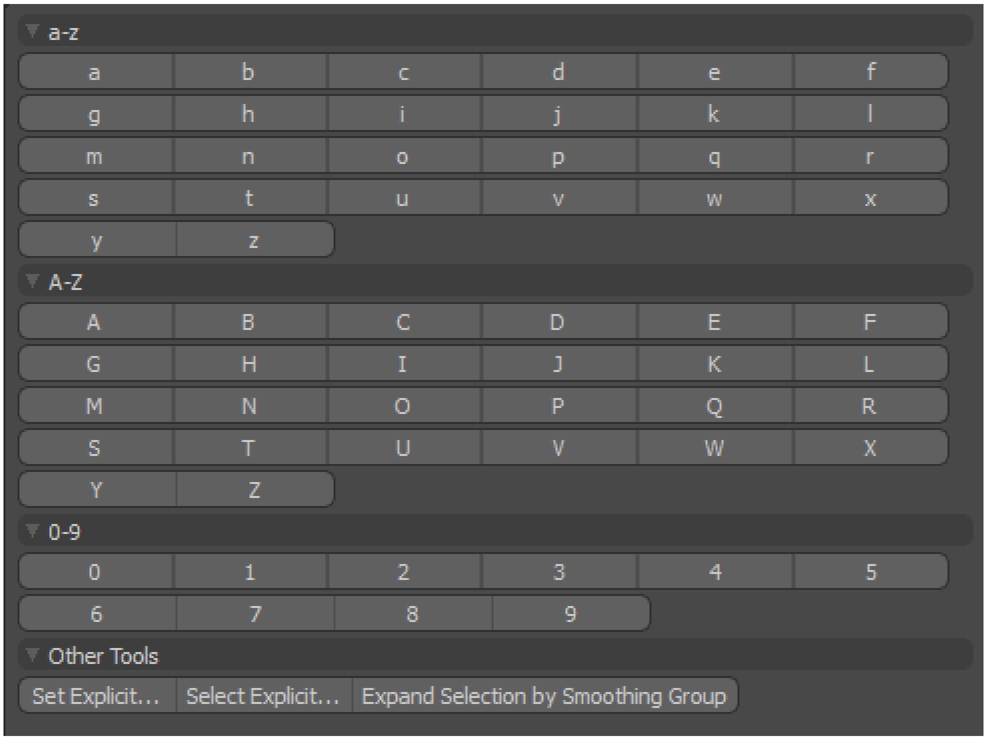
Modo allows you to group geometry and create polygon set smoothing groups to target smoothing areas. This tool is designed to ease your workflow when working on complex models. This is especially true in the case of lower resolution game models where you might prefer to set smoothing based on groups.

Smoothing groups are assigned alphanumeric characters (A-B, 0-9) representing **sets**. Imported models using smoothing groups display highlighted letters and numbers in the Smoothing Group palette for specified smoothing groups.

Launching the Smoothing Group Manager

On the menu bar, click **Vertex Map** > **Vertex Normal Tools...** and then click **Smoothing Group Manager**.

Alternatively, on the menu bar, click **Geometry** > **Polygon** > **Open Smoothing Group UI...**.

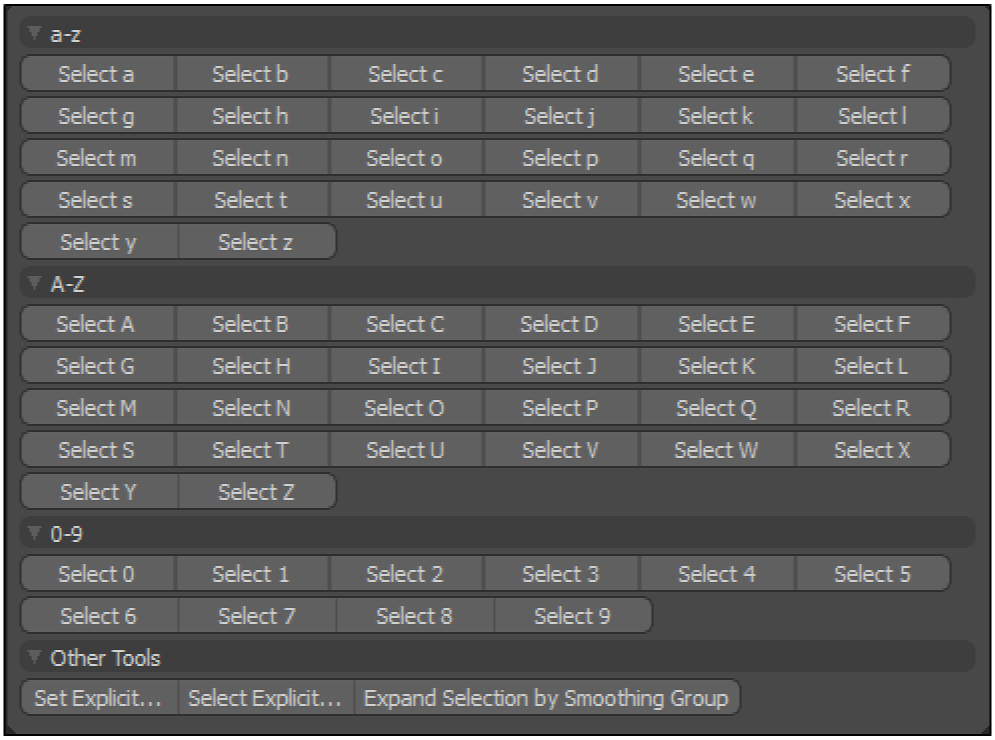

a-z
a b c d e f
g h i j k l
m n o p q r
s t u v w x
y z
A-Z
A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z
0-9
0 1 2 3 4 5
6 7 8 9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

Creating Smoothing Groups

Select the desired polygons and click a letter(s) and number(s) to assign them to a Smoothing Group. The letter/number displays in orange to indicate that the smoothing group was created. All of the selected polygons are smoothed together.

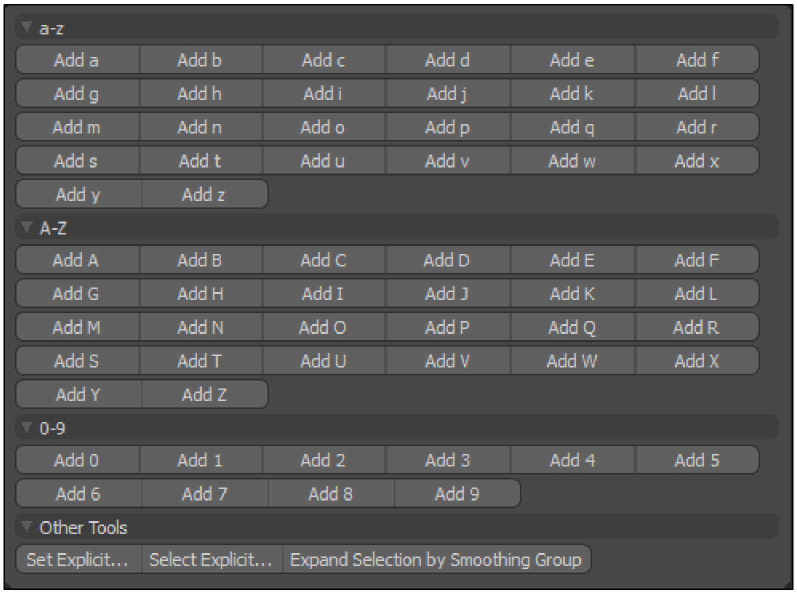
Selecting Polygons Assigned to a Smoothing Group

Press **Alt** and click an assigned group. The assigned polygons are selected on your mesh item.

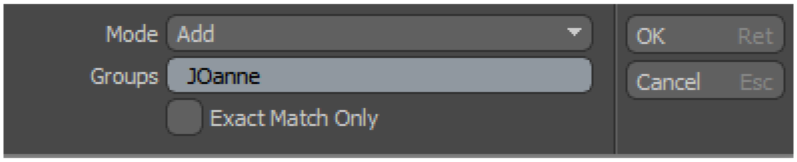

a-z
Select a
Select b
Select c
Select d
Select e
Select f
Select g
Select h
Select i
Select j
Select k
Select l
Select m
Select n
Select o
Select p
Select q
Select r
Select s
Select t
Select u
Select v
Select w
Select x
Select y
Select z
A-Z
Select A
Select B
Select C
Select D
Select E
Select F
Select G
Select H
Select I
Select J
Select K
Select L
Select M
Select N
Select O
Select P
Select Q
Select R
Select S
Select T
Select U
Select V
Select W
Select X
Select Y
Select Z
0-9
Select 0
Select 1
Select 2
Select 3
Select 4
Select 5
Select 6
Select 7
Select 8
Select 9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

Assigning Polygons to an Existing Smoothing Group

Select additional polygons on your mesh item, hold **Shift**, and click an assigned smoothing group. All of the polygons are assigned to this smoothing group..


a-z
Add a
Add b
Add c
Add d
Add e
Add f
Add g
Add h
Add i
Add j
Add k
Add l
Add m
Add n
Add o
Add p
Add q
Add r
Add s
Add t
Add u
Add v
Add w
Add x
Add y
Add z
A-Z
Add A
Add B
Add C
Add D
Add E
Add F
Add G
Add H
Add I
Add J
Add K
Add L
Add M
Add N
Add O
Add P
Add Q
Add R
Add S
Add T
Add U
Add V
Add W
Add X
Add Y
Add Z
0-9
Add 0
Add 1
Add 2
Add 3
Add 4
Add 5
Add 6
Add 7
Add 8
Add 9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

Alternatively, if you have a unique smoothing group name, click **Select Explicit...**, set **Mode** to **Add**, and enter the unique smoothing group name in the **Group** field. The selected polygons are added to that group.



Mode Add

Groups JOanne

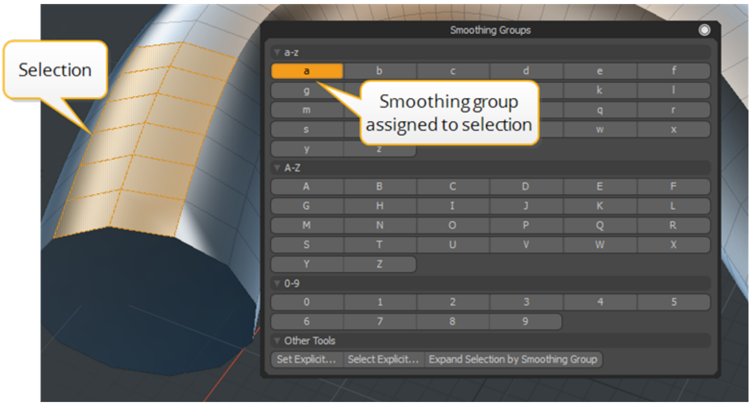
☐ Exact Match Only

OK Ret
Cancel Esc

Removing Polygons From a Smoothing Group

1. Select the polygons you want to remove from the smoothing group.

   In the **Smoothing Groups** manager, the active smoothing group(s) for the selection are highlighted in orange.



Selection
Smoothing group assigned to selection
Smoothing Groups
a-z
A-Z
0-9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

2. Click the letter/number of the smoothing group you want to remove.

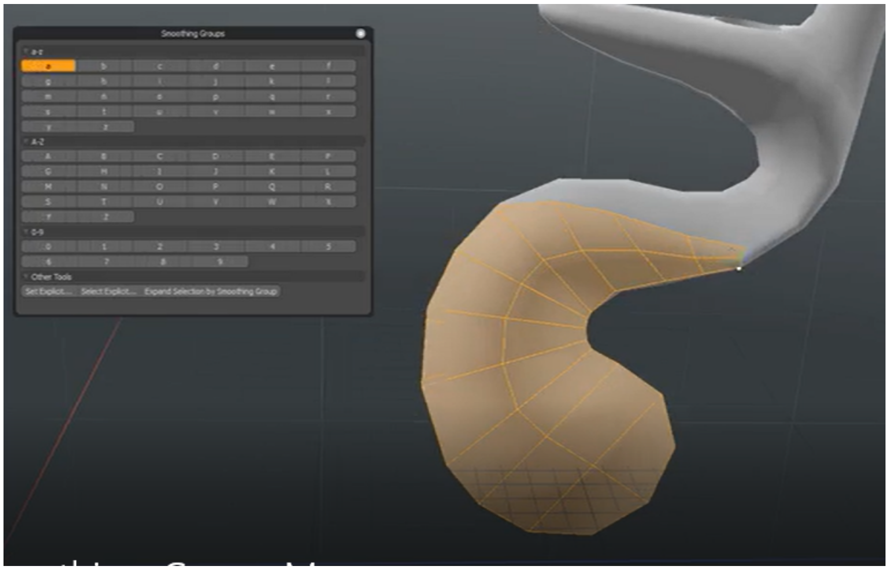
   The smoothing group is no longer highlighted and smoothing is removed from the selected polygons.

Blending Smoothing Groups

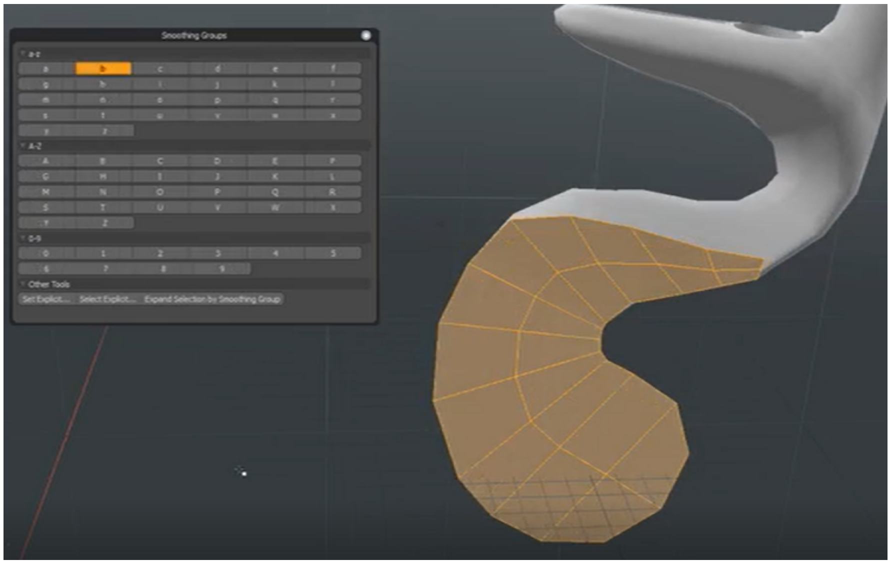
You may want to blend a selection of polygons in a smoothing group with another smoothing group to create a soft edge where the smoothing groups meet. In order to do that, you need to create more than one smoothing group.

1. Make a polygon selection, click the letter(s)/number(s) for the smoothing group it is assigned to. In this example, I have clicked **a**.

   The selection of polygons are removed from the assign smoothing group and hard edges are created.

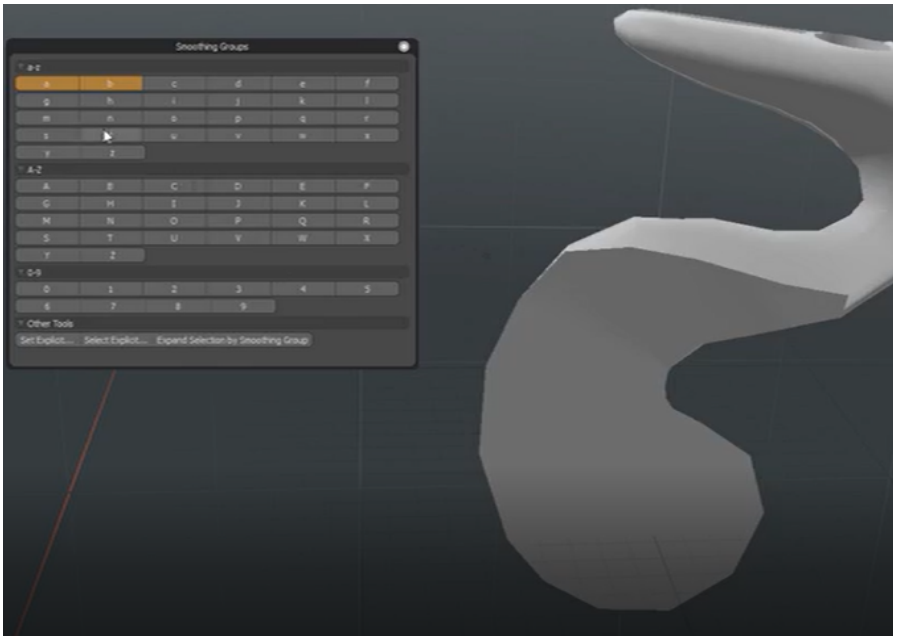

Smoothing Groups
a-z
A-Z
0-9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

2. Assign a new smoothing group by clicking on another unused letter(s)/number(s). In this example I have assigned these selected polygons to the smoothing group **b**.

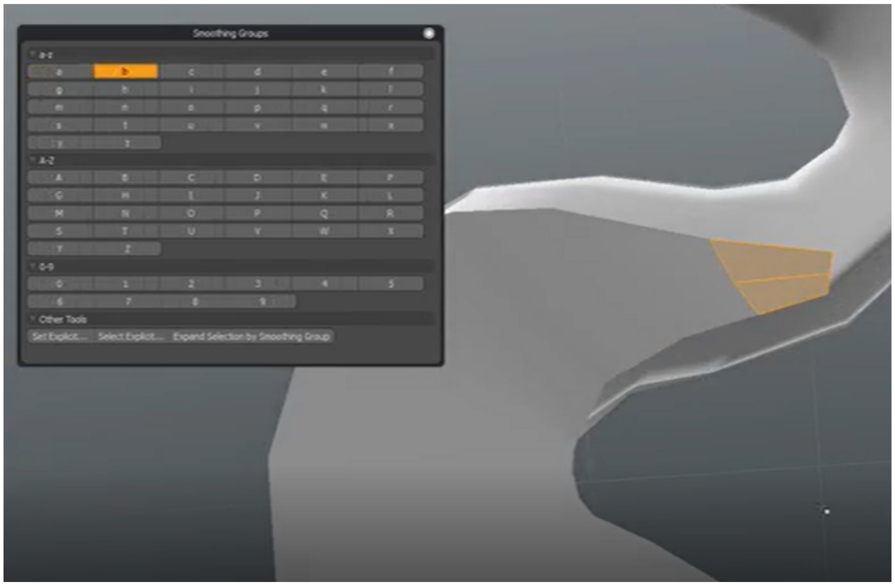

Smoothing Groups
a-z
A-Z
0-9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

3. Click in the 3D viewport to deselect those polygons.

   The letter buttons **a** and **b** fades out to tell you that there are multiple smoothing groups used by this selection.

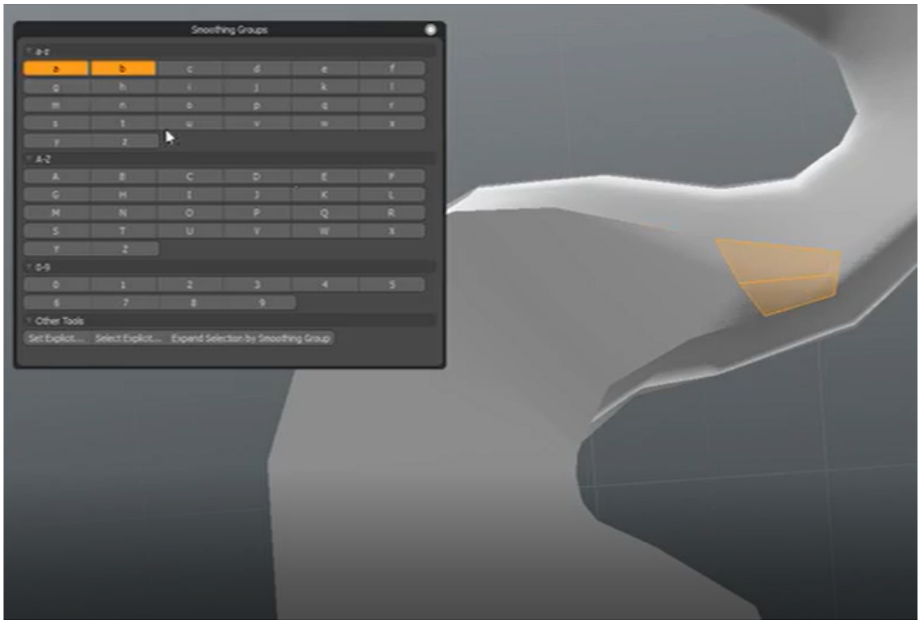

Smoothing Groups
a-z
A-Z
0-9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

4. Next, select an area of polygons in the newly created smoothing group to blend. In this example, I have selected the top polygons in the smoothing group **b**.


Smoothing Groups
a-z
A-Z
0-9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

5. Select the smoothing group letter(s)/number(s) you want to blend your selection with. In this example, I have selected the smoothing group **a**.

   The polygons are blended into the selected smoothing group. Both **a** and **b** smoothing groups display in orange in the Smoothing Groups window.

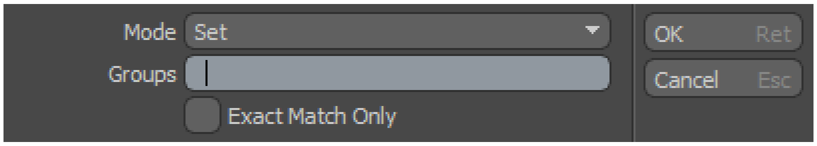

Smoothing Groups
a-z
A-Z
0-9
Other Tools
Set Explicit...
Select Explicit...
Expand Selection by Smoothing Group

Working with Explicit Smoothing Groups

You can add, remove, set, or expand a new selection of polygons to an explicit set. This is useful when you have imported a model containing pre-assigned smoothing groups using explicit names.

1. Select the polygons you want to explicitly apply to a smoothing group and click **Select Explicit...**.

   This opens the select by smoothing groups window.


Mode: Set
Groups
☐ Exact Match Only
OK Ret
Cancel Esc

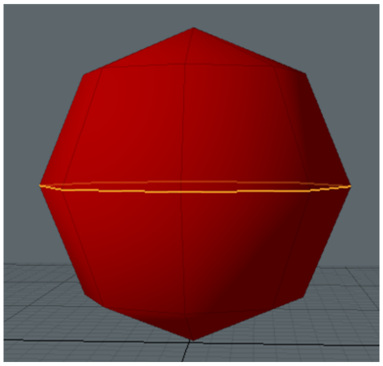
2. From the **Mode** dropdown menu, select one of the following:
   - **Set** - Sets the current selection of polygons to the specified group. Previously smoothing group polygons are overwritten.
   - **Add** - Adds the current selection of polygons to the specified group.
   - **Remove** - Removed the current selection of polygons from the specified group.
   - **Expand** - Selects all the other polygons on your mesh and deselects the current selection. You can then specify a smoothing group.
3. In the **Name** field, type the name of the unique smoothing group name.
4. **Optional** - Enable **Exact Match Only** to locate smoothing groups matching the capitalization you have specified.

Setting Hard and Soft Edge Smoothing Options

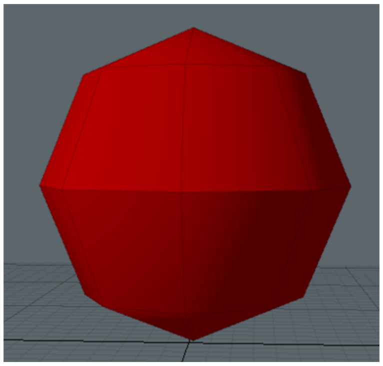
The Vertex Normal Tools allow you to harden or soften selected geometry. The vertex map is automatically updated if you alter the mesh and the map is recalculated for the updated normals on your mesh.

To apply softer or harder normals to a selection of edges:

1. Select edges of your mesh item and click **Harden** or **Soften**.
    - **Harden** - The hard edges split the smoothing of vertex normals across them and produce sharp edges.

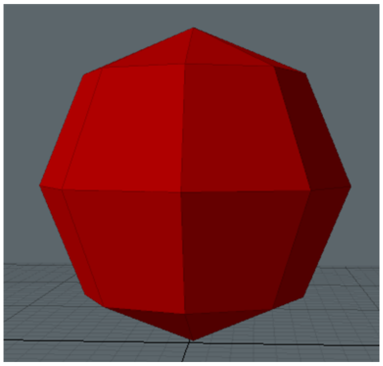


Softened edge mesh with central edges selected.

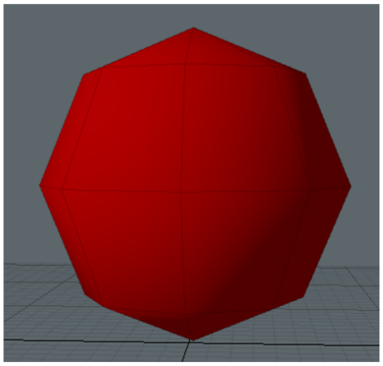


Selected edges have been hardened.

    - **Soften** - Soft edges are applied to the selection of vertex normals and produce a unified smooth surface.

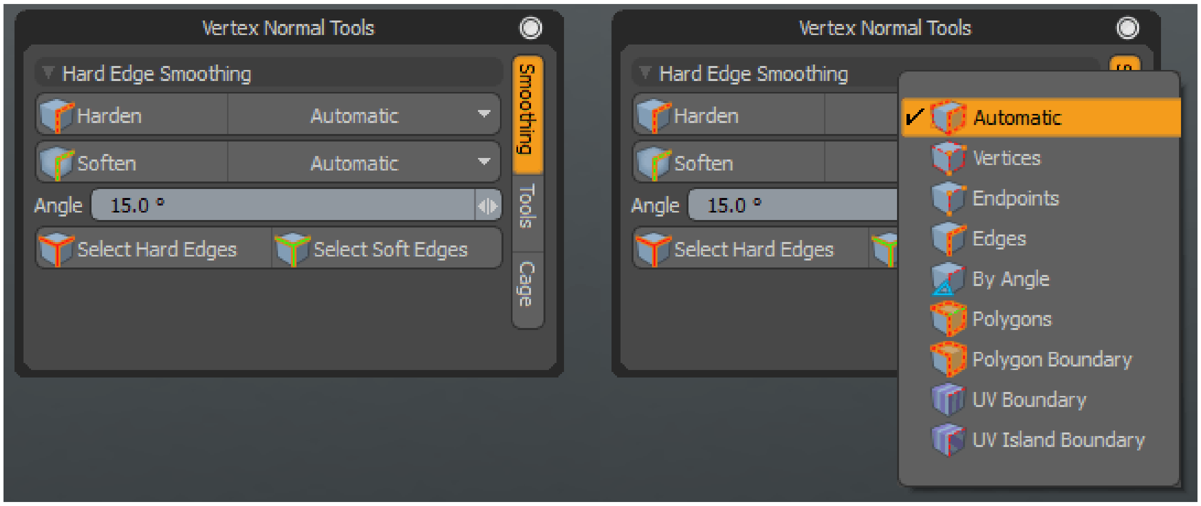


A mesh without smooth shading.



All edges have been softened.

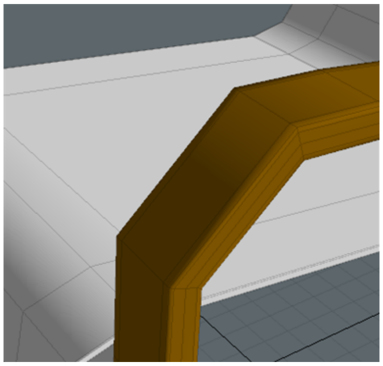
2. From the drop-down menu, select the type of geometry to apply. The following options are available:

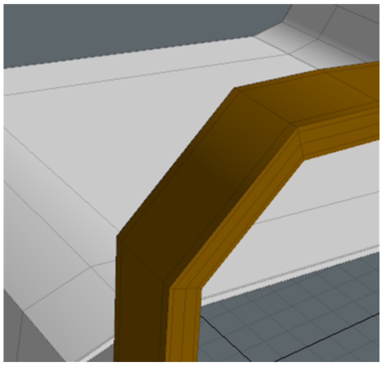

Vertex Normal Tools
Hard Edge Smoothing
Harden
Automatic
Soften
Automatic
Angle
15.0 °
Select Hard Edges
Select Soft Edges
Smoothing
Tools
Cage
Vertex Normal Tools
Hard Edge Smoothing
Harden
Soften
Angle
15.0 °
Select Hard Edges
✓ Automatic
Vertices
Endpoints
Edges
By Angle
Polygons
Polygon Boundary
UV Boundary
UV Island Boundary

- **Automatic** - Automatically uses vertices, edges, or polygons mode depending on the current selection mode.
- **Vertex** - Hardens or softens all edges connected to the selected vertices. **Shift**+click to harden or soften all edges connected to any of the selected vertices.
- **Endpoints** - Hardens or softens only the edges that have both end vertices selected.
- **Edges** - Hardens or softens the selected edges.

Note: Hard edges are stored in an edge pick map called **HardEdges**. If no vertex map is present, and one edge is added, all other edges are soft. If you soften any hard edges, the affected edges are removed from the **HardEdges** map.
If you soften edges and there is no **Vertex Map** active on the mesh, a new vertex map is calculated for all normals, effectively smoothing the entire mesh.

- **By Angle** - If hardening, hardens edges which are sharper than the given angle. If softening, soften edges are shallower than the given angle. This is useful for avoiding smoothing at hard edges, such as right angles.



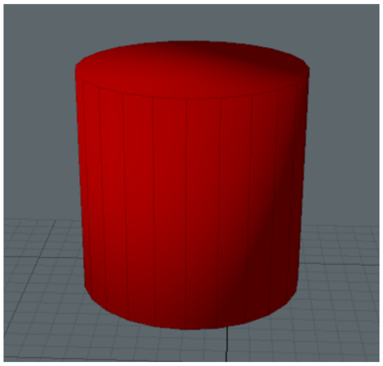


A chair mesh with soft edges.

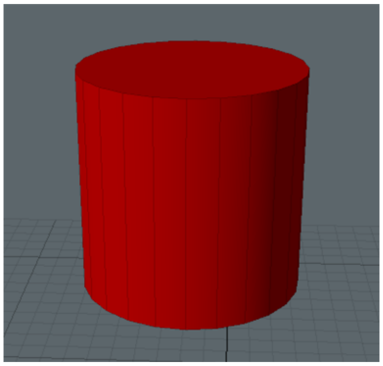
All edges at seams sharper than **60°** have been hardened.

Note: Harden the smoothing across all selected edges that are between polygons at a sharper angle than the specified value (the Smoothing Angle). If no edges are selected, all edges are evaluated. Edit the **Smoothing Angle** before applying the **By Angle** command. You can change the **Smoothing Angle** and click the button to add hard edges to the **HardEdges** selection set. However, modifying the **Smoothing Angle** and hardening, does not remove any hard edges that were added as a result of prior operations.

- **Polygons** - Hardens or softens the edges on the boundary of the current polygon selection, softening the internal edges.
- **Polygon Boundary** - Hardens or softens the edges on the boundary of the current polygon selection (leaving the internal edges untouched).
- **UV Boundary** - If hardening, harden edges which lie on the boundary of the current UV map. If softening, soften edges which lie on the interior of the current UV map islands.



Mesh with all soft edges.



Edges at UV boundaries have been hardened.

- **UV Island Boundary** - If hardening, hardens edges which belong to multiple UV islands. If softening, soften edges which only belong to one UV island. (A UV island is a connected group of polygons in a UV map.)

Procedural Hard Edges

The Hard Edges tool is also available as a mesh operation.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

To use the mesh operation:

1. Select the edges to modify.
2. In the **Mesh Ops** tab, click **Add Operator**.

Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **New Tab** > **Data Lists** > **Mesh Ops**.

3. Under **Mesh Operations** > **Edit**, double-click **Hard Edge**.

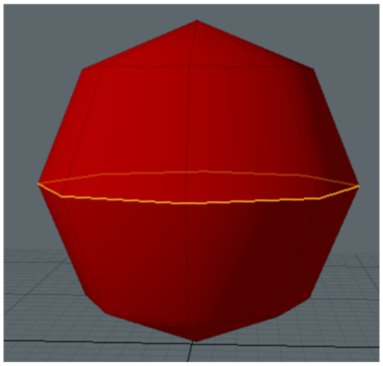
   The selected edges are hardened and the operation is added to the Mesh Operations list.

The Hard Edge mesh operation's properties are displayed in the lower right panel, in the **Properties** tab.

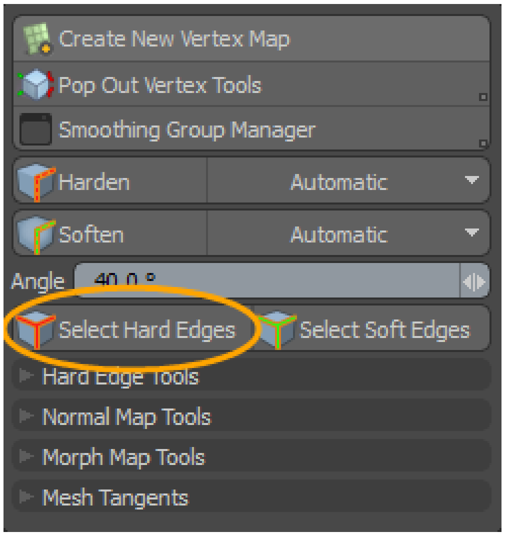
- You can change what **Selection Type** the **Automatic Mode** should use.
- To harden the edges, **Hardness** should be set to **Hard**. To soften, set it to **Soft**.
- The **Mode** options are the same as for the direct modeling equivalent.
- Enabling **Clear Other Hard Edges** removes any existing hard edges before running the operation.
- **Angle** specifies the angle value for **Angle Mode**, and **UV Map** specifies what map is used when **Mode** is set to **UV Boundary** or **UV Island Boundary**.

Selecting Hard Edges

The **Select Hard Edges** tool allows you to select all edges marked as hard. Hard edges are stored in the **HardEdges** map.



1. Click **Vertex Map** > **Vertex Map Tools...**
2. On the Vertex Map dialog, click **Select Hard Edges**.


Create New Vertex Map
Pop Out Vertex Tools
Smoothing Group Manager
Harden
Automatic
Soften
Automatic
Angle
40.0 °
Select Hard Edges
Select Soft Edges
Hard Edge Tools
Normal Map Tools
Morph Map Tools
Mesh Tangents

Alternatively, use the following keys to add or remove hard edges from your current selection.

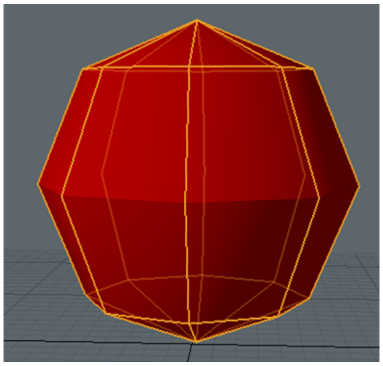
- **Shift**+click **Select Hard Edges** to add all hard edges to the current edge selection.
- **Ctrl**+click **Select Hard Edges** to remove all hard edges from the current edge selection.

Procedural Select Hard Edges

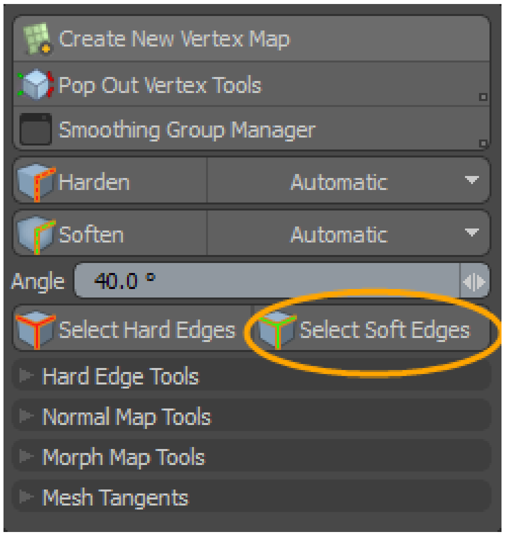
There is a procedural selection operation for selecting hard or soft edges. For more information, see Select by Pattern.

Selecting Soft Edges

The Select Soft Edges tool allows you to select all edges not marked as hard. This is all edges except those stored in the **HardEdges** map.



1. Click **Vertex Map** > **Vertex Map Tools...**
2. On the Vertex Map dialog, click **Select Soft Edges** using one of the following:


Create New Vertex Map
Pop Out Vertex Tools
Smoothing Group Manager
Harden
Automatic
Soften
Automatic
Angle
40.0 °
Select Hard Edges
Select Soft Edges
Hard Edge Tools
Normal Map Tools
Morph Map Tools
Mesh Tangents

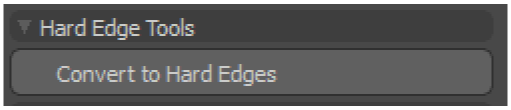
Alternatively, use the following keys to add or remove hard edges from your current selection.

- **Shift**+click **Select Soft Edges** to add all edges not marked as hard to the current edge selection.
- **Ctrl**+click **Select Soft Edges** to remove all edges not marked as hard from the current edge selection.

Converting to Hard Edges

Marks any edge which has its shading split at either end (or both ends) as a hard edge. Optionally, it removes all smoothing groups and vertex normal maps after marking the relevant edges as hard.

1. In the **Vertex Map** dialog, under **Hard Edge Tools**, click **Convert to Hard Edges**.


Hard Edge Tools
Convert to Hard Edges

2. In the **Convert to Hard Edges** dialog, select one of the following:
   - **Clear Smoothing Groups**
   - **Remove Normal Maps**

Procedural Hard Edge Convert

The Hard Edge Convert tool is also available as a mesh operation. This operation converts the selected edges that are split by smoothing groups or vertex normal maps to Modo's Hard Edges system. It can be useful when importing CAD scenes that include vertex normal maps.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

To use the mesh operation:

1. Select the edges to modify.
2. In the **Mesh Ops** tab, click **Add Operator**.

Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **New Tab** > **Data Lists** > **Mesh Ops**.

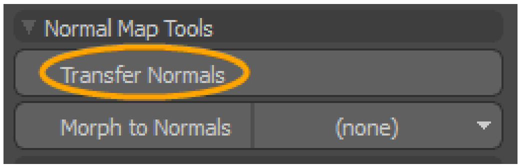
3. Under **Mesh Operations** > **Edit**, double-click **Hard Edge Convert**.

   The selected edges are hardened and the operation is added to the Mesh Operations list. The mesh operation's properties are displayed in the **Properties** tab in the lower right panel. After applying the operation, you can delete any existing normal maps or smoothing groups by enabling **Remove Normal Map** or **Remove Smoothing Group**.

Transferring Normals from another Mesh Shader

You can copy a vertex normal map from a background mesh using the **Transfer Normals** button. For example, you may want to use the shading from a higher resolution mesh on other mesh items. The normals are copied from the background mesh to the active foreground mesh.

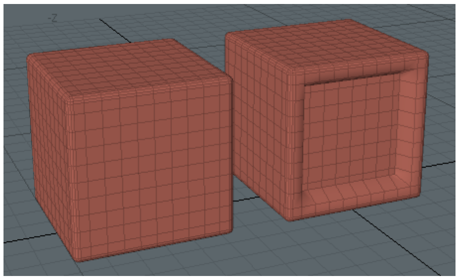
The transfer works by finding the closest (by distance) vertex-polygon pair on the background mesh to each vertex-polygon pair on the foreground mesh, and copies the normal value over. Once done, the normals themselves can be moved apart, rotated, and scaled in **Items** selection mode.



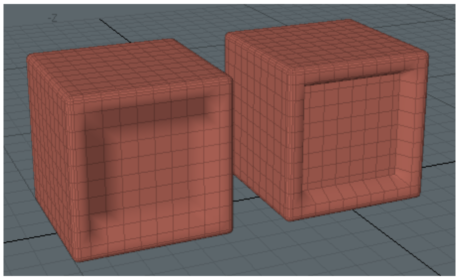
1. Select the foreground mesh.

Note: Make sure the background mesh (or the layer directly below the active layer in the **Items** list when there are multiple item layers) is marked as visible (👁) and unselected in the 3D viewport.

2. In the **Vertex Map** dialog, click **Transfer Normals**.



Two mesh items.



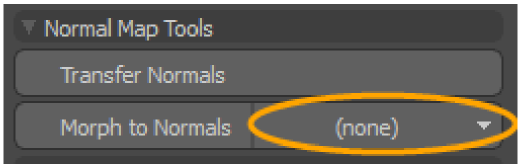
The Vertex Normals have been copied from the right mesh to the left mesh.

Applying Morph to Normals

For each selected vertex, **Morph to Normals** takes the direction of the offset defined in the currently selected Relative Morph Map, and sets the vertex normal to point in that direction. The vertex position offset is relative to the base vertex position. If the offset has zero length, or the value is unset in the morph map, the normal is unchanged.

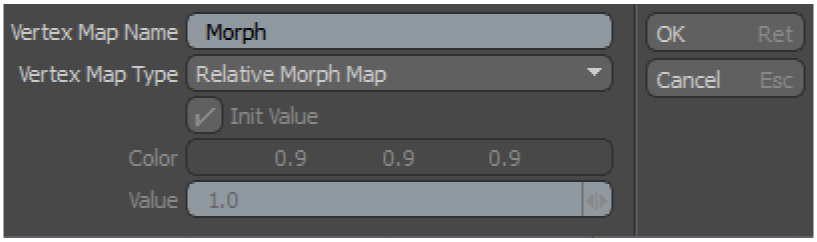
For more information, see Working with Vertex Maps and Set Vertex Normals.

1. In the **Vertex Map** dialog, click the down arrow and select a morph map from the dropdown list.



2. If you do not have a Morph Map, click **(new)** and specify the options.

   For more information, see Create New Vertex Map.



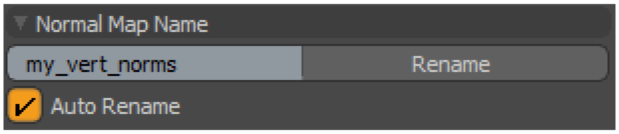
Vertex Map Name: Morph
Vertex Map Type: Relative Morph Map
☑ Init Value
Color: 0.9 0.9 0.9
Value: 1.0

3. In the Vertex Map dialog, click **Morph to Normals**.

Renaming the Normal Map

The default name for the calculated Vertex Normal Map is **Vertex Normal**.

1. To rename it, edit the name and click **Rename**.



Normal Map Name

my_vert_norms

☑ Auto Rename

2. Check **Auto Rename** to automatically assign the given name to the vertex normal map after updating smoothing on a mesh item.

Using the Morph Push Tool

When baking normals, you can use a morph map as a cage to determine the offset distance for the rays. The **Morph Push Tool** lets you alter the cage size to customize the results of a bake. It also provides a more flexible alternative to Modo's Push tool for deforming morph maps.

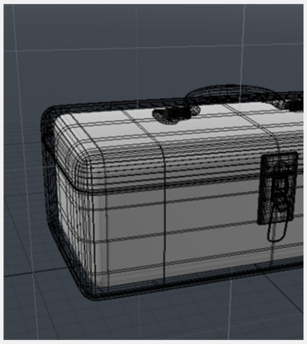
> **Note:** For more information on using **Cages**, see Bake from Object to Selected Texture

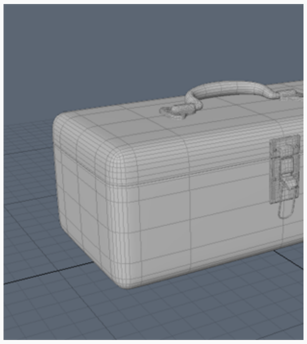
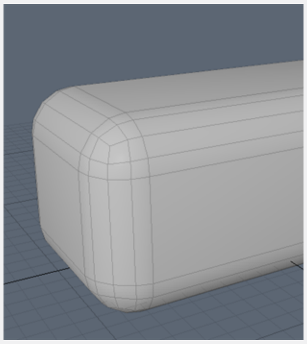
Compared to the **Push** tool, the **Morph Push Tool** has the following additional features:

- It always computes high quality normals to push along.
- It uses angle-weighted normals.

These features ensure a cleaner push, with no discrepancies from non-planar polygons or differently triangulated meshes.

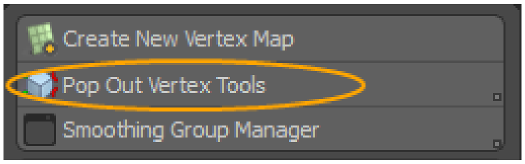
The images below shows the **Morph Push Tool** used to create a cage for a low-poly mesh prior to baking.

	A foreground high poly mesh of a toolbox above a low poly mesh (selected). The low poly mesh also has an empty **Morph Map**.

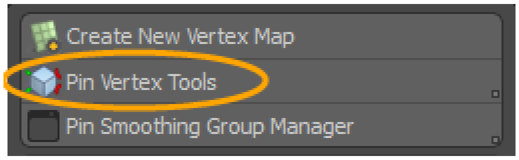
	The two meshes as viewed in **Game** mode.
	The **Morph Map** on the low poly mesh has been pushed using the **Morph Push Tool** to create a suitable **Cage**.

To apply the Morph Push Tool:

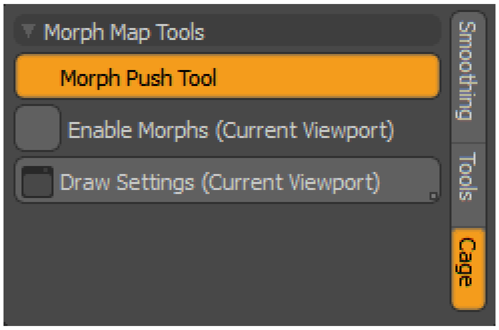
1. On the Vertex Normals dialog, click **Pop Out Vertex Tools**.


Create New Vertex Map
Pop Out Vertex Tools
Smoothing Group Manager

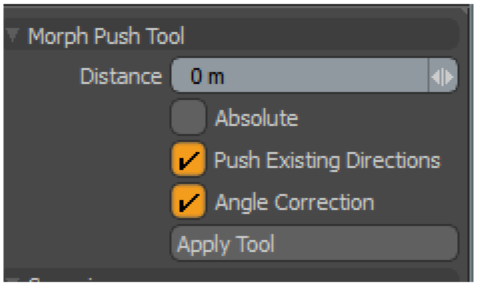
Alternatively, hold **Alt** and click **Pin Vertex Tools**


Create New Vertex Map
Pin Vertex Tools
Pin Smoothing Group Manager

2. In the Vertex Normal palette, open the **Cage** tab, and click **Morph Push Tool**.


Morph Map Tools
Morph Push Tool
Enable Morphs (Current Viewport)
Draw Settings (Current Viewport)
Smoothing
Tools
Cage

3. Click in the 3D viewport to apply the tool.
4. On the left panel, edit the parameters as required, and click **Apply Tool**.



Morph Push Tool

Distance 0 m

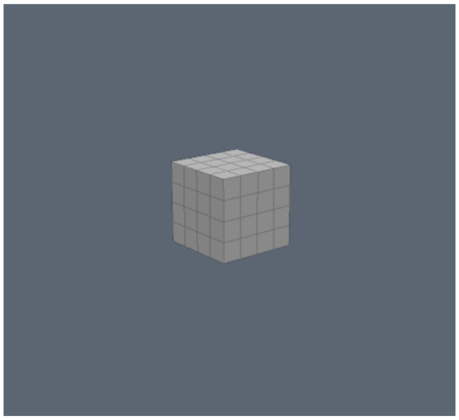
☐ Absolute

☑ Push Existing Directions

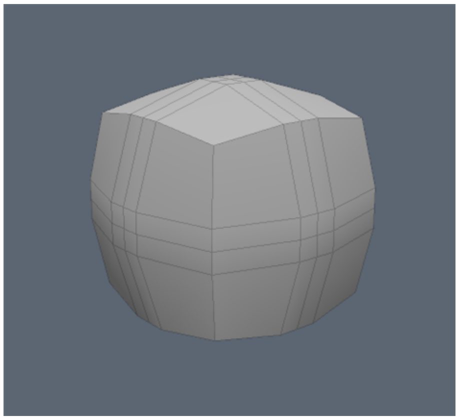
☑ Angle Correction

Apply Tool

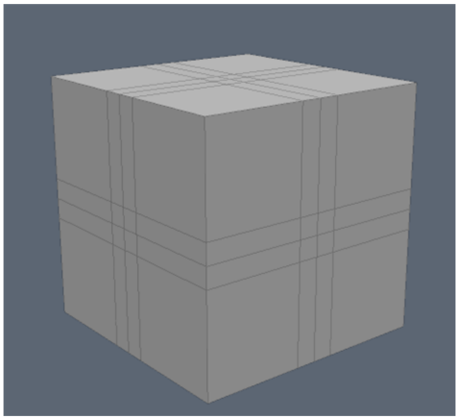
Distance	The distance to push the morphed position of each vertex. **Tip:** Click and drag in the 3D viewport to adjust the **Distance**.
Absolute	This makes the distance value absolute, rather than relative. Enable this to ensure that the distance each vertex is pushed is uniformly set to equal the **Distance** value, regardless of the distance that was already defined in the **Morph Map**.
Push Existing Directions	When checked, this ensures that the direction of the existing morph is used when pushing the vertices (instead of the default average normal).
Angle Correction	Enable this to retain the form of the mesh as best as possible when pushing. When enabled, the vertices on flat planes are pushed out by the **Distance** amount, while vertices on corners are pushed out more, to compensate for the averaged normal.



A **1m** cube with an empty **Morph Map**.

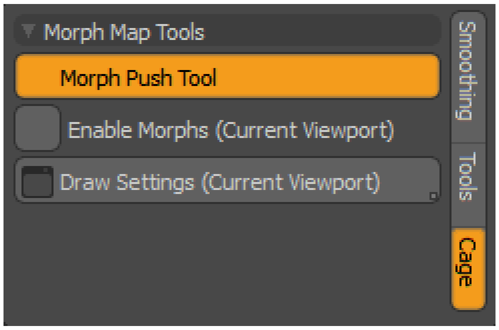


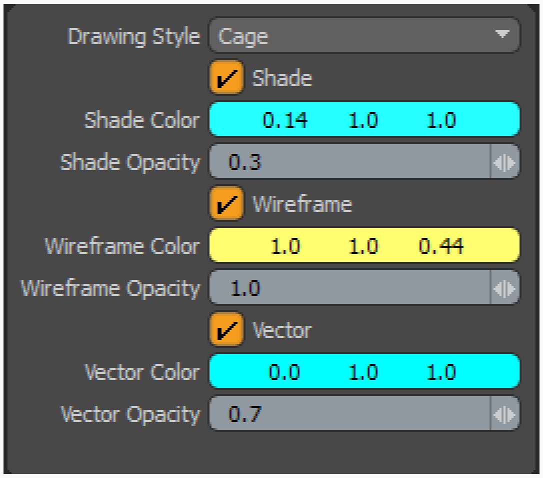
The cube has been pushed by **1m** without **Angle Correction**.



The same **1m** push but with **Angle Correction**.

Other Morph 3D Viewport Options


Morph Map Tools
Morph Push Tool
Enable Morphs (Current Viewport)
Draw Settings (Current Viewport)
Smoothing
Tools
Cage

Enable Morphs (Current Viewport)	Allows the 3D Viewport to display the results of animated morphs. Enabling this option disables deformers.
Draw Settings (Current Viewport)	Specifies how you would like to draw Morph Maps. The following options are available when enabled: **Drawing Style** - Specify how you would like to draw Morph Maps. The following options are available: • **Default** - Modo draws the traditional morphed mesh. • **Cage** - Modo draws the base mesh and the morphed mesh as a cage. When this option is selected, additional properties are revealed:  Drawing Style: Cage ☑ Shade Shade Color: 0.14 1.0 1.0 Shade Opacity: 0.3 ☑ Wireframe Wireframe Color: 1.0 1.0 0.44 Wireframe Opacity: 1.0 ☑ Vector Vector Color: 0.0 1.0 1.0 Vector Opacity: 0.7

Shade - When enable, draws a transparent morphed mesh as a cage.

Shade Color - Specifies the color of the morph cage.

> **Note:** Only displayed when a Shader, Wireframe, or Vector is enabled.

Shade Opacity- Specifies the opacity value of the morph cage.

Wireframe - Draws the cage in wireframe mode.

Wireframe Color - Specifies the color of the wireframe.

Wireframe Opacity - Specifies the wireframe opacity.

Vector - Draws line segments from the vertices of the base mesh to the corresponding vertices of the morph mesh.

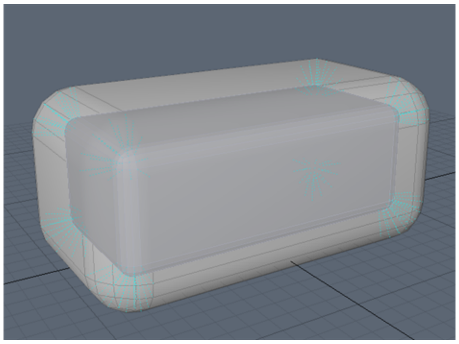
Vector Color - Sets the vector color.

Vector Opacity - Sets the vector opacity.

To display a morphed mesh, you need to select a morph map with **Enable Deformers** disabled.

Viewing the Morph Maps Cage View

You can quickly see the difference between an original mesh and the morphed version using the **Cage** view. This is useful for viewing morph maps that are used as cages for baking.



For more information on using **Cages**, see Bake from Object to Selected Texture

Enabling the Cage View

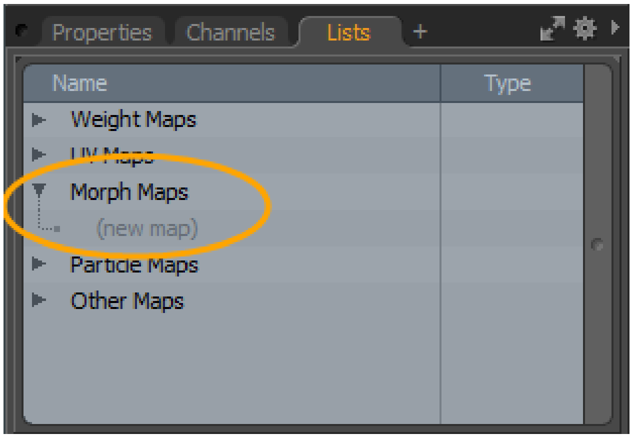
To enable **Cage View**, you must first configure the **3D viewport Properties**.

1. Hover the mouse over the **3D viewport**, press **O** to open the **3D viewport Properties**.
2. Open the **Drawing and Control** tab, under the **View and Shading** section, enable **Overlay Drawing**.
3. Under the **Animation** section, disable **Enable Deformers**.

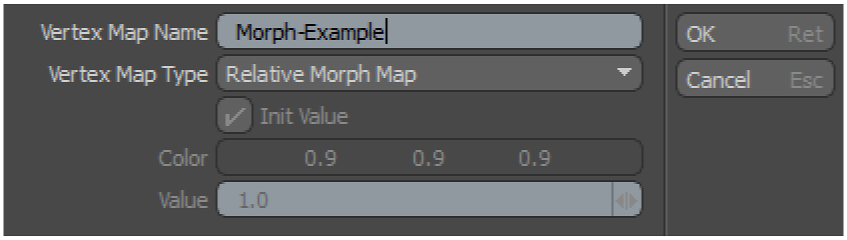
To view a cage:

The following is a simple workflow example viewing the cage for a morph map applied to a cube primitive.

1. On the left panel of the **Model** layout, press **Ctrl**/**Cmd** and click the **Cube** icon.
2. On the right panel, open the **Lists** tab, expand **Morph Maps** > **Morph**, and click **(new map)**.


Properties
Channels
Lists
+
Name
Type
Weight Maps
UV Maps
Morph Maps
(new map)
Particle Maps
Other Maps

3. In the **Create New Vertex Map** dialog, enter a new name, set **Vertex Map Type** to **Relative Morph Map**, and click **OK**.



Vertex Map Name: Morph-Example

Vertex Map Type: Relative Morph Map

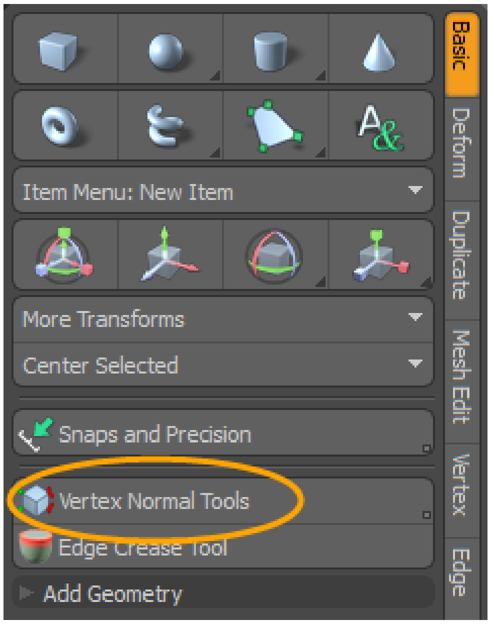
☑ Init Value

Color: 0.9 0.9 0.9

Value: 1.0

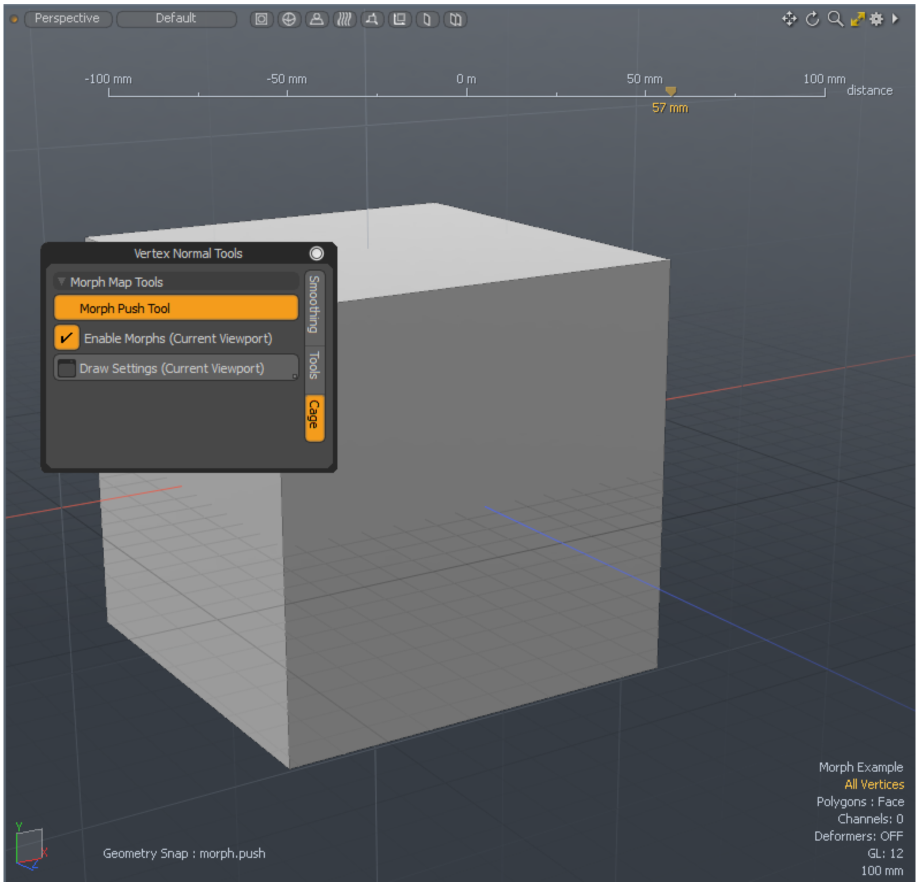
OK Ret
Cancel Esc

4. On the left panel, in the **Basic** tab, click **Vertex Normal Tools**.

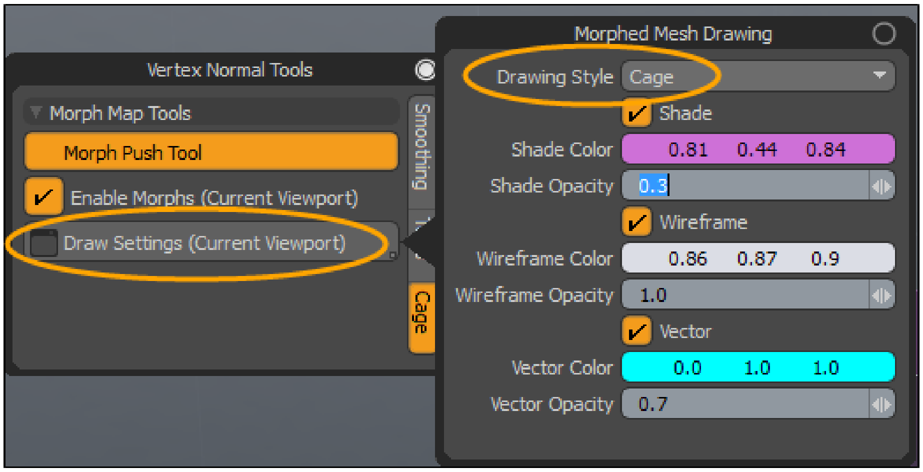


Basic
Deform
Duplicate
Mesh Edit
Vertex
Edge
Item Menu: New Item
More Transforms
Center Selected
Snaps and Precision
Vertex Normal Tools
Edge Crease Tool
Add Geometry

5. In the Vertex Normal Tools palette, click **Morph Push Tool**, and click-and-drag in the 3D viewport.

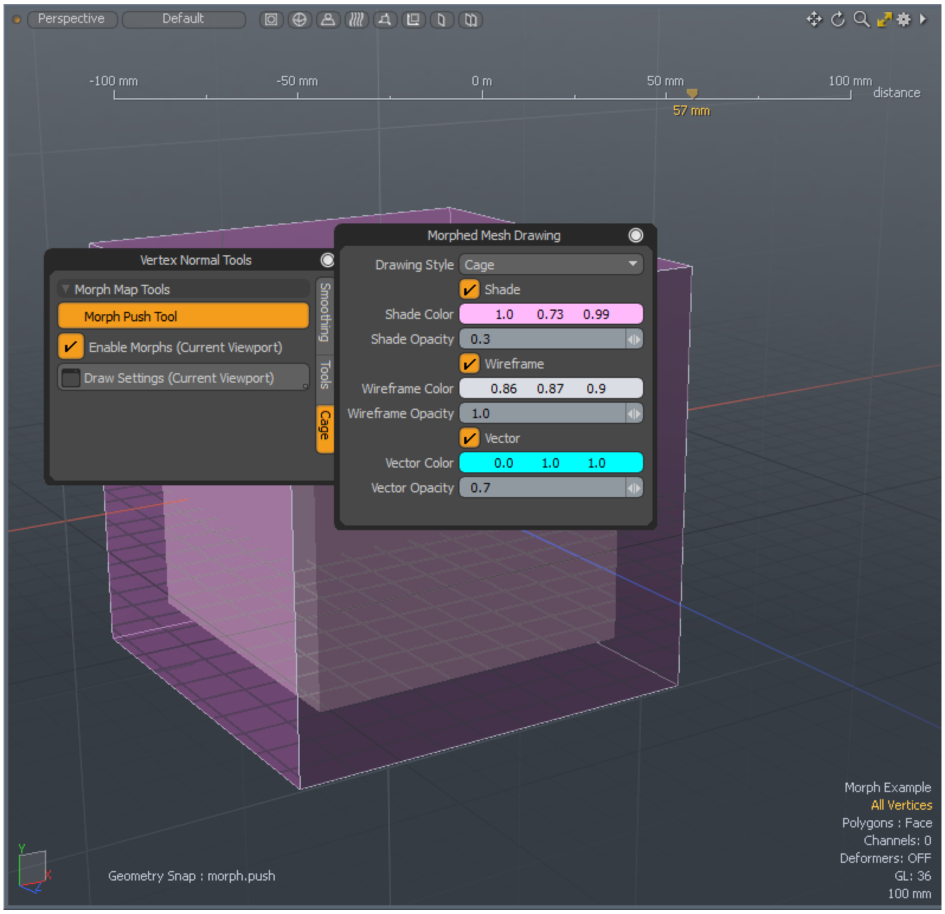

Perspective
Default
-100 mm
-50 mm
0 m
50 mm
100 mm
distance
57 mm
Vertex Normal Tools
Morph Map Tools
Morph Push Tool
Enable Morphs (Current Viewport)
Draw Settings (Current Viewport)
Smoothing
Tools
Cage
Morph Example
All Vertices
Polygons : Face
Channels: 0
Deformers: OFF
GL: 12
100 mm
Geometry Snap : morph.push

6. Click **Draw Setting (Current Viewport)** and set **Drawing Stye** to **Cage**. You can also set the other options if desired. For more information, see Other Morph 3D Viewport Options.


Vertex Normal Tools
Morph Map Tools
Morph Push Tool
Enable Morphs (Current Viewport)
Draw Settings (Current Viewport)
Smoothing
Cage
Morphed Mesh Drawing

Drawing Style: Cage
☑ Shade
Shade Color: 0.81 0.44 0.84
Shade Opacity: 0.3
☑ Wireframe
Wireframe Color: 0.86 0.87 0.9
Wireframe Opacity: 1.0
☑ Vector
Vector Color: 0.0 1.0 1.0
Vector Opacity: 0.7

The original, unmorphed mesh is overlaid on the current geometry, and dashed lines are drawn between the unmorphed mesh and the morphed mesh.


Perspective
Default
-100 mm
-50 mm
0 m
50 mm
100 mm
distance
57 mm
Vertex Normal Tools
Morph Map Tools
Morph Push Tool
Smoothing
Tools
Cage
Morphed Mesh Drawing
Geometry Snap : morph.push
Morph Example
All Vertices
Polygons : Face
Channels: 0
Deformers: OFF
GL: 36
100 mm

☑ Enable Morphs (Current Viewport)
☐ Draw Settings (Current Viewport)

Drawing Style: Cage
☑ Shade
Shade Color: 1.0 0.73 0.99
Shade Opacity: 0.3
☑ Wireframe
Wireframe Color: 0.86 0.87 0.9
Wireframe Opacity: 1.0
☑ Vector
Vector Color: 0.0 1.0 1.0
Vector Opacity: 0.7

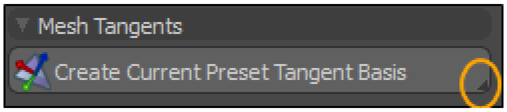
Creating Mesh Tangent Maps

You can create tangent basis maps for various renderers using the **Mesh Tangents** options.

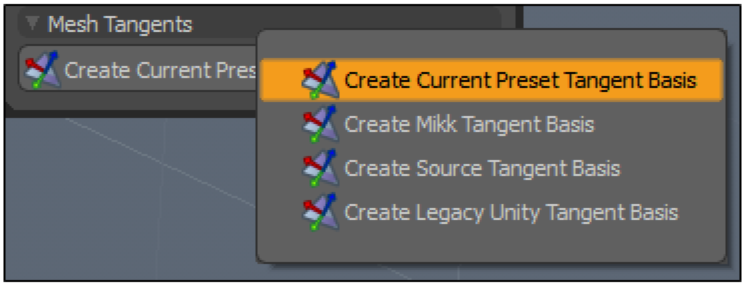
Tip: A number of Mesh Tangent options are also available from the **Vertex Map** > **Tangent Vectors** menu.

Note: If no UV maps are selected, the commands iterate over all active meshes, and tangent basis maps are generated for all UV sets on each active mesh.

1. On the **Vertex Normal** dialog, on the **Create Current Preset Tangent Basis** button, click the dropdown arrow.


Mesh Tangents
Create Current Preset Tangent Basis

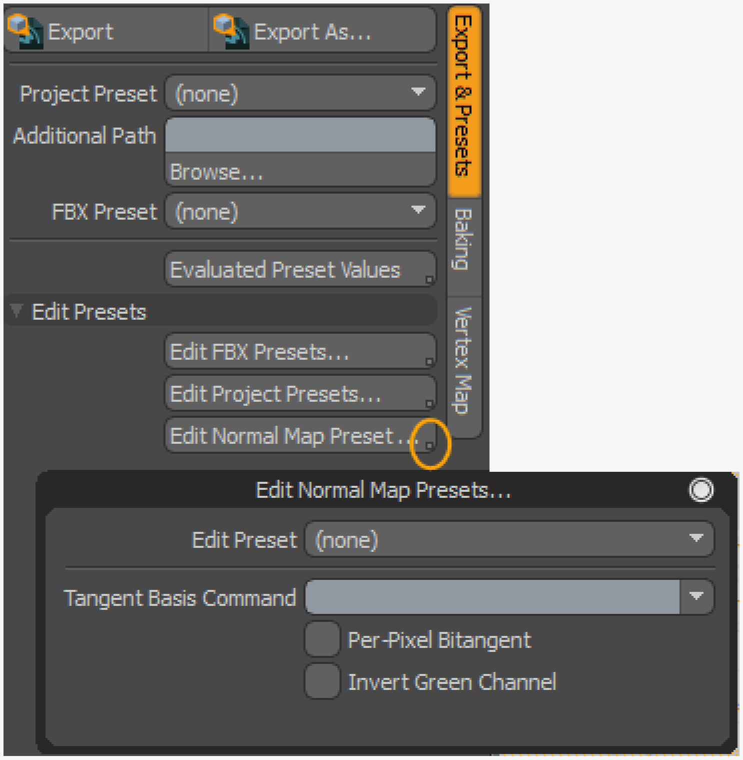
2. Select one of the following options.


Mesh Tangents
Create Current Pres
Create Current Preset Tangent Basis
Create Mikk Tangent Basis
Create Source Tangent Basis
Create Legacy Unity Tangent Basis

> **Note:** The default preset is called **None**, and the default label for the button is **Create Current Preset Tangent Basis**. Changing the normal map preset changes the label.

- **Create Current Preset Tangent Basis** - Create a tangent basis map using the tangent command that is defined in the current normal map preset used in the Game Asset Exporter. See Editing Presets for how to choose and edit normal map presets for a project.
- **Create Mikk Tangent Basis** - Creates a tangent basis map using Mikk Tangent Space.
- **Create Source Tangent Basis** - Creates a tangent basis map compatible with Valve's Source engine.
- **Create Legacy Unity Tangent Basis** - Creates a tangent basis map compatible with the Unity Engine's legacy tangent basis standard.

Tip: A full list of tangent basis map generation commands can be found in the **Game** layout, **Export & Presets** toolbox tab, under the **Edit Normal Map Preset...** option. For more information, see Game Asset Exporter.


Export
Export As...
Export & Presets
Baking
Vertex Map
Edit Presets
Edit FBX Presets...
Edit Project Presets...
Edit Normal Map Preset...
Edit Normal Map Presets...
Browse...
Evaluated Preset Values

Project Preset (none)

Additional Path

FBX Preset (none)

Edit Preset (none)

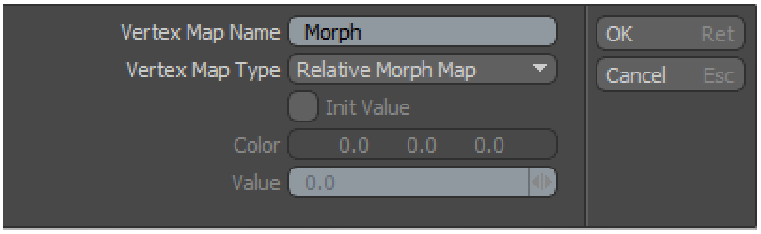
Tangent Basis Command

☐ Per-Pixel Bitangent

☐ Invert Green Channel

Create New Vertex Map

With the Create New Vertex Map command you can create a new Vertex Map, provide a unique name for the map, and set an initial value. Once created, the named map immediately appears within the appropriate section of the **Lists** viewport.


OK Ret
Cancel Esc

Vertex Map Name Morph

Vertex Map Type Relative Morph Map

☐ Init Value

Color 0.0 0.0 0.0

Value 0.0

Vertex Map Name: Type a unique identifying name for the Vertex Map.

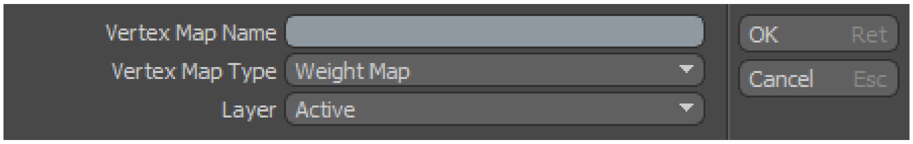
Vertex Map Type: Specify the type of map to create.

Map Type	Description
Weight Map	Stores weight strength values for falloffs and texturing.
SubD Weight Map	Stores weight strength values that influence the creasing values in SubDivision (SubD) Surface geometry.
UV Map	Translates 3D vertex positions to flat 2D coordinates.
Relative Morph Map	Offsets vertex positions relative to the base vertex position.
Absolute Morph Map	Offsets vertex position to a specific absolute position in 3D space.
RGB Map	Defines a vertex color map based on red, green, and blue color values.
RGBA Map	Defines a vertex color map based on red, green, and blue color values and an additional transparency (Alpha) value.
Pick Map	Defines groupings of vertices and is similar to a selection set.
Vertex Normal Map	Stores surface normal direction (smoothing) values as fixed values.
Edge Pick Map	Defines groupings of edges and is similar to a selection set.

Init Value: When enabled, defines an initial value for the newly created Vertex Map. For vertex color maps, this would be **Color** values: as RGB values or RGBA values. For Weight Maps, this would be a single numerical **Value**.

Set Vertex Map Name

You use this command to rename Vertex Maps. Select the Target Vertex Map and click **Name** . In the **Set Vertex Map Name** dialog, specify the options as needed.



Vertex Map Name

Vertex Map Type Weight Map

Layer Active

Vertex Map Name: Type a unique, identifying name for the Vertex Map in this box. This name replaces the current name for the selected Vertex Maps.

Vertex Map Type: Specifies the appropriate map type based on the map being renamed. When renaming a Vertex Map, Modo automatically selects the correct map type.

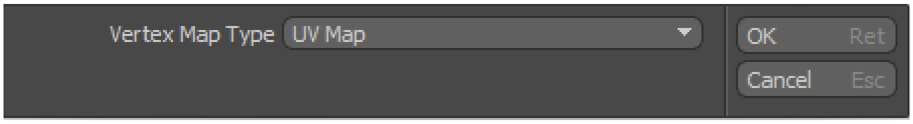
Layer: Determines whether Modo renames the foreground items, the background items, or both. If you select **Active**, Modo renames only the foreground item's Vertex Map. If you select **Inactive**, Modo renames the background items. If you select **Both**, Modo renames the current Vertex Map for all visible items. (**Both** is only available when layer items share identical map names.)

Vertex Map Commands

Copy

You can use the **Copy** command with the **Paste** command to take information from one Vertex Map and apply it to another.

You can access the **Copy** and **Paste** commands by choosing **Vertex Map > Copy** or **Vertex Map > Paste**. You can also select the map in the **Vertex Map** list and then right-click the map and choose **Copy**. Then, select a different map in the same list, right-click again and choose **Paste** to complete the operation.



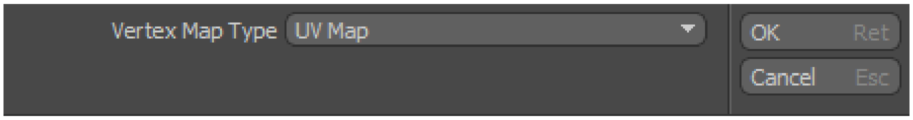
Vertex Map Type UV Map

Vertex Map Type: For proper results, make sure when copying values with the **Vertex Map > Copy** command that the proper Vertex Map type is defined for both maps.

Paste

You can use the **Paste** command with the **Copy** command to take information from one Vertex Map and apply it to another.

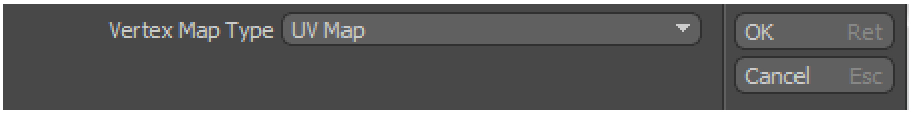
You can access the **Copy** and **Paste** commands by choosing **Vertex Map > Copy** or **Vertex Map > Paste**. You can also select the map in the **Vertex Map** list and then right-click the map and choose **Copy**. Then, select a different map in the same list, right-click again and choose **Paste** to complete the operation.


Vertex Map Type UV Map

Vertex Map Type: For proper results, make sure when copying values with the **Vertex Map > Copy** command that the proper Vertex Map type is defined for both maps.

Clear

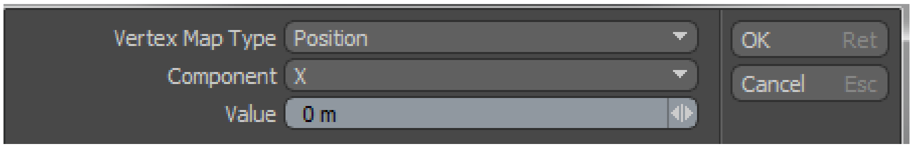
Choosing **Vertex Map > Clear** resets the values of any Vertex Map edits. For example, if you make changes to a Weight Map and then click **Clear** or use the menu command, those values are discarded and Modo returns the Weight Map to its initial state. You can also right-click the map's name in the **Vertex Map** list and choose **Clear**.


Vertex Map Type UV Map

Delete

To completely remove a Vertex Map from the **Vertex Map** list and the current Modo session, select the map in the list, and then use the **Delete** command. You can right-click the entry in the list and choose **Delete** or choose **Vertex Map > Delete**.

Set Value



Vertex Map Type: Position

Component: X

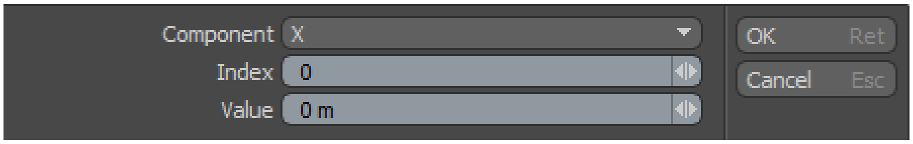
Value: 0 m

Use the Set Value tool to quickly and accurately set the value of a particular Vertex Map or a group of selected Vertex Maps. Because Modo stores much of the data in a 3D model as a Vertex Map, this is a powerful tool. Vertex maps are typically referred to when discussing things such as morph maps, Weight Maps, or UV data, but Vertex Maps are also used for storing the basic positional information for all vertices in a mesh. When using the Set Value tool, Modo changes the dialog to reflect current conditions. For example, if there are no selected RGBA maps, then there is no RGBA choice available. Also, the contents of the **Component** list changes to match the type of the selected Vertex Map.

To use the Set Value tool, select the desired Vertex Map class in the **Type** list. (The contents of this list change depending on what types of Vertex Maps are active in the currently selected data.) Now, in the **Component** list specify the setting to change. Finally, specify a **Value** for the Vertex Map.

With this tool you can manage many different aspects of your model's data quickly. For example, you could quickly set the UV coordinates to **U** = **0** for an entire set of selected vertices so that they snap to an edge. Alternatively, you could use the tool to quickly set the selected vertices' Weight Map to **100%**. You could even use this tool to set all the selected vertices to an absolute location such as **X** = **0**.

Set Vertex



Component: X

Index: 0

Value: 0 m

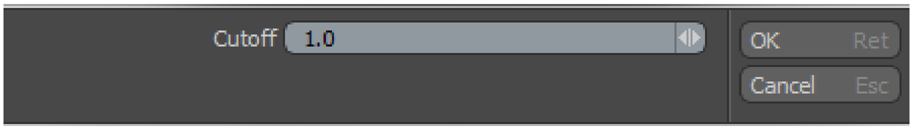
With the **Set Vertex** command you can define precise values for individual vertices in a Vertex Map. In the **Set Vertex Value** dialog you can specify these options:

Component: Specifies which component Modo sets. Some Vertex Maps have multiple components. For example, a UV Vertex Map has two components **U** and **V**; an RGB map has three components **R**, **G**, and **B**; and a Weight Map has only a single component: the weights value. Using the **Set Vertex** command on a UV map with the **Component** set to **0**, sets the value on the **U** component of the map. Using a value of **1** as the **Component**, sets the **V** value.

Index: Specifies which vertex is to be affected. If you turn on **Show Indices** under **Visibility Options**, you can see the vertex indices in the viewport.

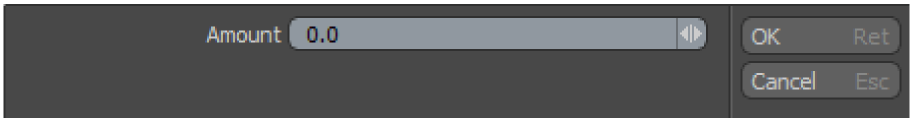
Value: Determines the numeric value for the vertex.

Clamp


Cutoff 1.0

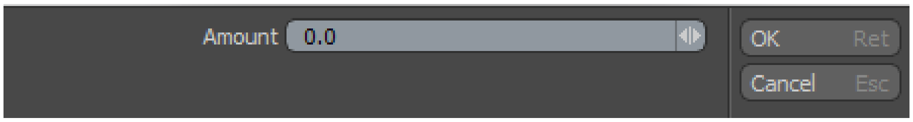
Vertex Map Clamp command limits the values in a Vertex Map based on the **Cutoff** value.

Quantize


Amount 0.0

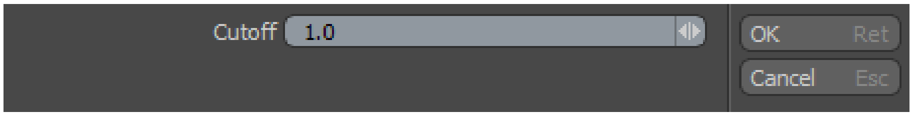
The Vertex Map Quantize tool reduces the Vertex Map selection values mathematically. Modo evaluates all the selected components and simplifies them based on the **Amount** value set in the **Quantize Vertex Map** dialog. Increasing the **Amount** results in fewer diverse values in the Vertex Map. To apply this command, choose **Vertex Map > Quantize**.

Jitter


Amount 0.0

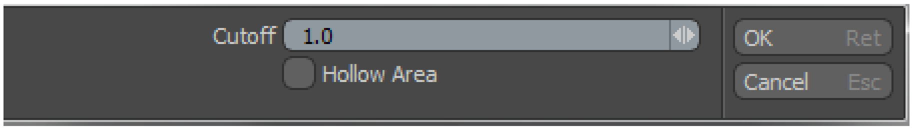
The Vertex Map Jitter tool procedurally affects selected Vertex Map values based on an **Amount** value. This tool makes the placement of the vertices in the Vertex Map random. To apply this command, choose **Vertex Map > Jitter**.

Unify


Cutoff 1.0

A discontinuous Vertex Map is one where a single vertex has multiple UV map values. The **Vertex Map Unify** command merges continuous and discontinuous Vertex Map values if their difference is below the threshold given as the **Cutoff** value. Modo computes the difference from the largest component value difference (as opposed to the difference of the magnitude of the entire vector). One way to visualize this operation is to make some discontinuous UVs that are very close to similar continuous UVs. Then use the command to close the seam.

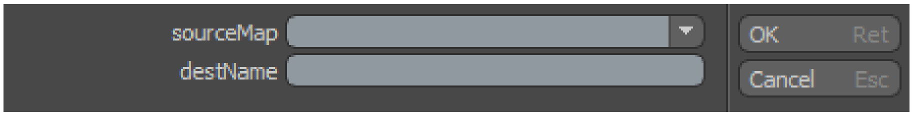
Cull


Cutoff 1.0

☐ Hollow Area

The **Cull** command removes extra Vertex Map values. It clears the values below the **Cutoff** setting. If you select the **Hollow Area** checkbox, then Modo removes only values that have non-culled neighbors and limits culling to isolated values or values that are below the threshold and are not connected to any vertices with values above the threshold.

Mirror



The **Mirror** command creates mirrored Vertex Map values across the user-defined **Symmetry** axis. The target geometry with the Vertex Map must be symmetrical, itself, and you must select the **Symmetry** checkbox before running the command. In the dialog, specify a source map and a Destination map. If the source and destination are the same map (have the same name), then Modo mirrors the values from one side to the opposite side. If the destination map is different from the Source map, then Modo creates a new map (or assigns the mirrored values to an existing map).

Source Map: Lists the active Vertex Maps on the current Mesh Item layer(s). Select the source map for the mirroring operation from the list.

Destination Map: Specifies the name of the target map for the mirrored vertex values. Type the name of the destination map. If the name is the same as the source map, then Modo mirrors the values across the designated symmetry axis in the same Vertex Map to create a symmetrical map (if it wasn't symmetrical to begin with). If the name is different from the source map, then Modo creates the resulting map as a mirrored version of the source map (not symmetrical).

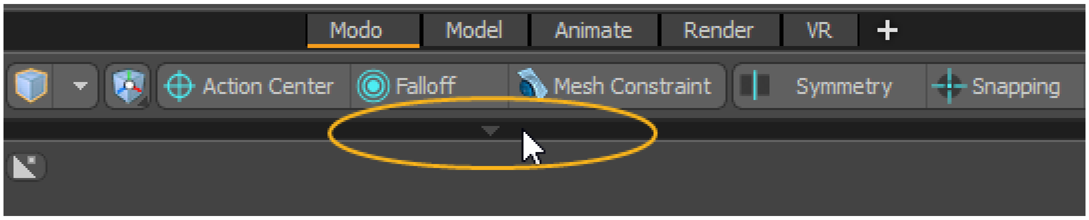
Working With Vertex Maps Procedurally

You can create Vertex maps and assign weight values, seams, and RGBA values to them non-destructively using mesh operations. The tools are accessible from the Mesh Operations List.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

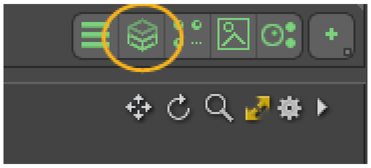
The easiest way to access the Mesh Operations list is using the **Modo** layout.

1. Click the thin bar above the 3D viewport, indicated by a small arrow in the middle.
   This reveals additional controls.



2. Click the Mesh Operations list button.

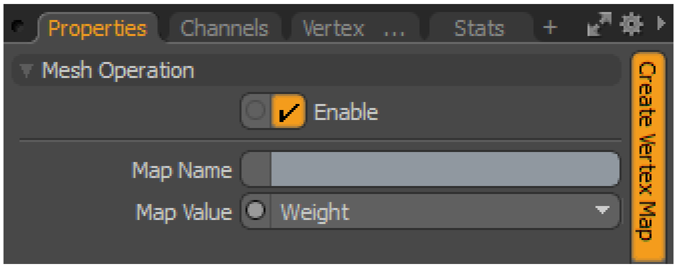
The Mesh Operations List opens on the right.



Creating a New Vertex Map Procedurally

The Create Vertex Map mesh operation allows you to create a new vertex map from the Mesh Operations list.

1. In the Mesh Operations list, click **Add Operator**, and under **Mesh Operations** > **Vertex Map**, double-click **Create Vertex Map**.
   The Create Vertex Map mesh operation is added to the list and its **Properties** are displayed on the right pane.



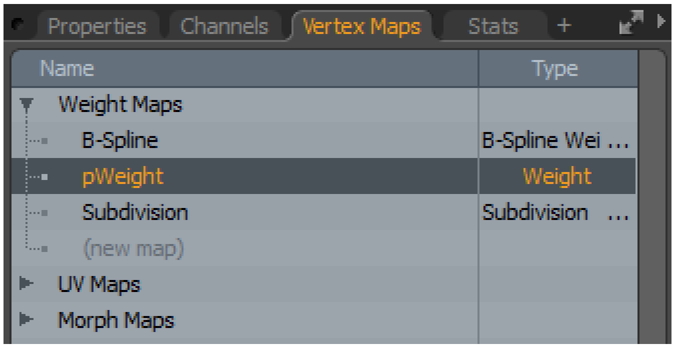
☑ Enable

Map Name

Map Value ○ Weight

2. In the Create Vertex Map **Properties**, enter a new **Map Name** and **Map Value**.
   You have the following **Map Value** options:
   - **Weight** - Stores weight strength values for falloffs and texturing.
   - **UV Seam** - Stores information on the seam edges from a UV map.
   - **RGB** - Defines a vertex color map based on red, green, and blue color values.
   - **RGBA** - Defines a vertex color map based on red, green, and blue color values and an additional transparency (Alpha) value.

   Your new Vertex Map is created and listed in the Vertex Map List.



Name	Type
Weight Maps	
B-Spline	B-Spline Wei ...
pWeight	Weight
Subdivision	Subdivision ...
(new map)	
UV Maps	
Morph Maps	

Tip: You can select the vertex map associated with the selected mesh operation by clicking **Select Map**, so you don't have to manually select vertex maps from the **Vertex Map List**.

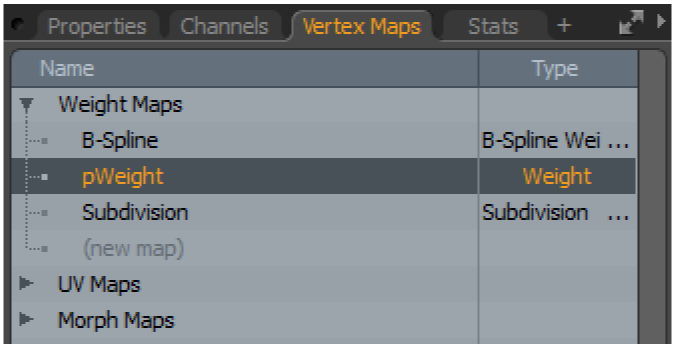
Setting Weight Values

You can set weight values to a specified Weight map using the Set Weight mesh operation.

Tip: Enable the Show Weightmaps ☒ button at the top of the 3D viewport to see the weight values in the viewport, or press **O** and ensure **Show Weightmaps** is selected in the **Active Meshes** tab.

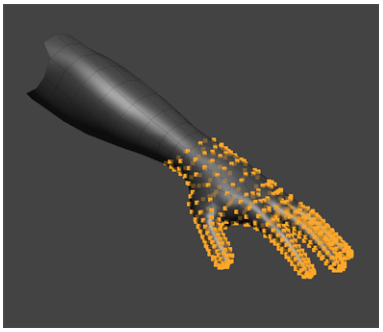
You need to create a Weight map for your mesh, as described in Creating a New Vertex Map Procedurally. Then set weights the following way:

1. In the **Vertex Maps** list on the lower right, under **Weight Maps**, select the weight map you created.



Name	Type
Weight Maps	
B-Spline	B-Spline Wei ...
pWeight	Weight
Subdivision	Subdivision ...
(new map)	
UV Maps	
Morph Maps	

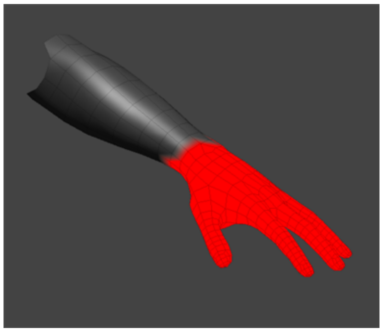
2. In the Item List, select your mesh item and in the 3D viewport, select the vertices, edges, or polygons for which you want to set weights.



3. In the Mesh Operations list, under **Mesh Operations** > **Vertex Map**, double-click **Set Weight**.
   This adds the **Set Weight** mesh operation to the list and opens its properties in the **Properties** panel on the right pane.
4. In the Set Weight **Properties**, select the **Map Type** you want to use. A **Weight Map** is embedded on the mesh, while a **Weight Container** is a separate item.

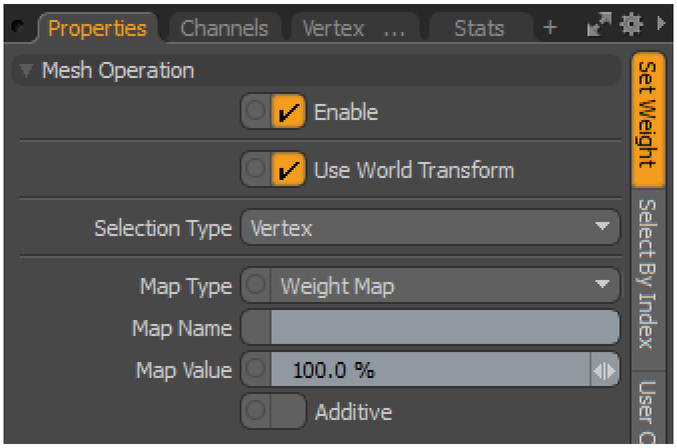
> **Note:** For more information on Weight Maps and Weight Containers, see Working with Weight Maps.

5. Set **Map Name** to the name of the Weight Map you want to use, in our example, 'pWeight'.
   If you have Show Weightmaps enabled in the **Default** view, the selected part of the mesh is colored red in the 3D viewport, indicating that the default 100% map value is assigned to it.



6. Adjust the **Map Value** to set the weight value.

Set Weight Properties



☑ Enable

☑ Use World Transform

Selection Type: Vertex

Map Type: Weight Map

Map Name:

Map Value: 100.0 %

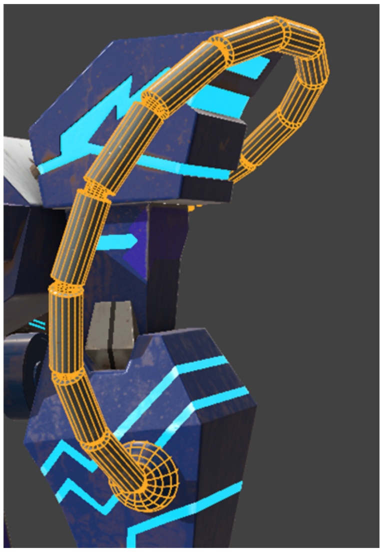
☐ Additive

Enable	Enables/disables the Set Weight mesh operation.
Use World Transform	Sets the coordinates from Local Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Selection Type	The element type selected. This mesh operation can take a **Vertex**, **Edge**, or **Polygon** selection.
Map Type	Set if you want to use a **Weight Map** or a **Weight Container**. For more information on the differences, see Working with Weight Maps. **Note:** If **Map Type** is set to **Weight Map** and the map can't be found, a new Weight Map is created.
Map Name	The name of the Weight Map to edit.
Map Value	The weight value to be applied to the selected elements.
Additive	When enabled, the **Map Value** is added to the current map value. When disabled, the current map value is set to the amount specified in **Map Value**.

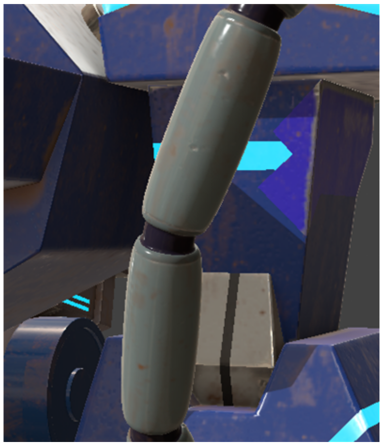
Setting Edge Weight Values

You can set edge weight values on a subdivision map using the Set Edge Weight mesh operation.

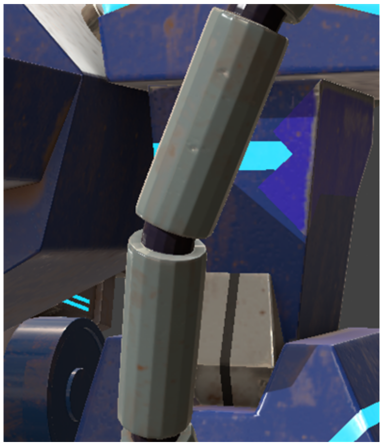
1. In the Item List, select your mesh item and in the 3D viewport, select the edges for which you want to set weights.



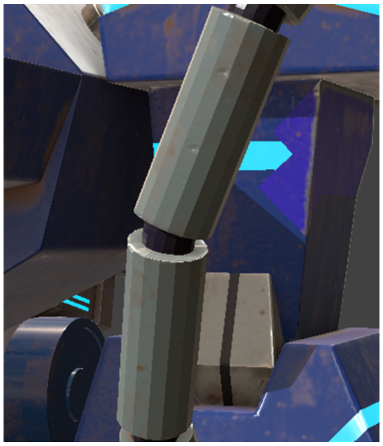
2. In the Mesh Operations list, under **Mesh Operations** > **Vertex Map**, double-click **Set Edge Weight**. This adds the Set Weight mesh operation to the list and opens its properties in the **Properties** panel on the right pane.
3. In the Set Edge Weight **Properties**, set the **Map Value** to the required percentage.
   The edges are modified in the viewport.



Map Value 0%



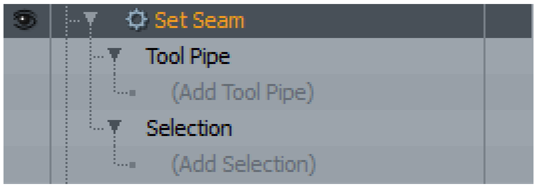
Map Value 25%



Map Value 50%

Mesh Operation Inputs

The procedural vertex map operations all have inputs you can edit in the Mesh Operations list. You can expand them in the list by clicking the arrow ▼ in front of them. This reveals the inputs the operation uses.



The Inputs of Set Seam

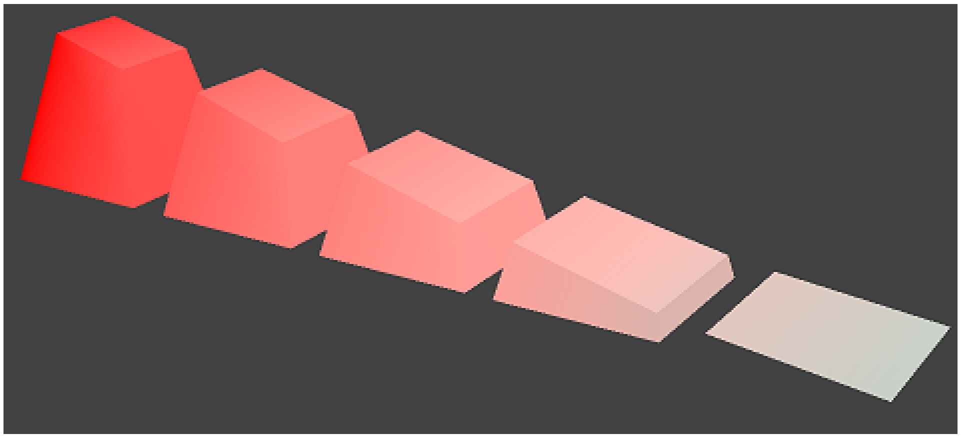
- **Tool Pipe** - Add falloffs and sub-tools to the operation. For example, instead of selecting vertices, edges, or polygons, you can use falloffs to create a selection.
- **Selection** - Select vertices/edges/polygons or modify your existing selection. For more information on procedural selection, see Procedural Selection.

The following procedural vertex map operations are available.

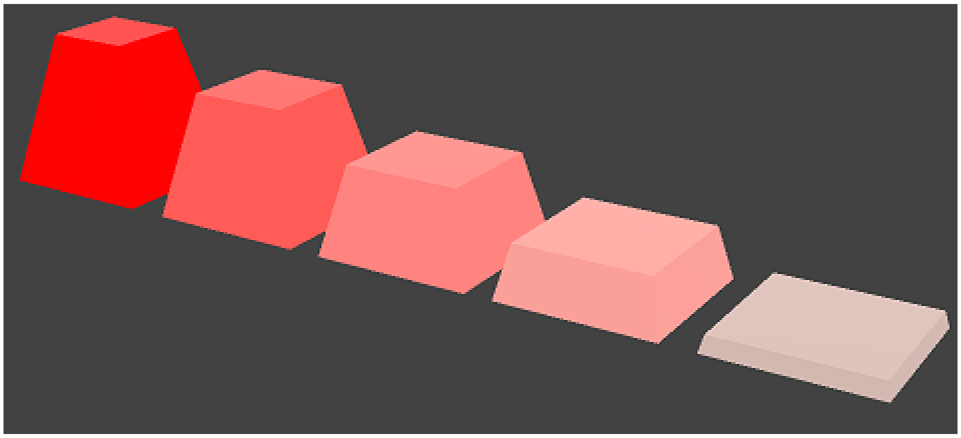
Flood Weight

The flood weight operation searches each island on vertex maps for the lowest and highest weight value, and then floods each island with the lowest or highest value found. This gives you the option to create flatter surfaces in situations where you might not be able to easily achieve rigid effects.

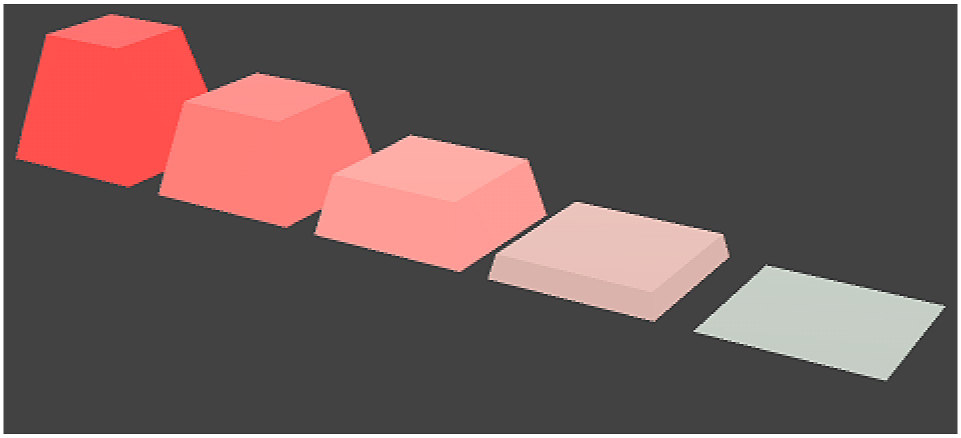
Note: An island in Modo is referring to a single piece of geometry in the vertex map.



The original mesh, with a weight map, a linear falloff, and a polygon bevel operation applied. There are five islands on this mesh.



A flood weight operation assigned to the mesh with **High/Low** set to **High**.



A flood weight operation assigned to the mesh with **High/Low** set to **Low**.

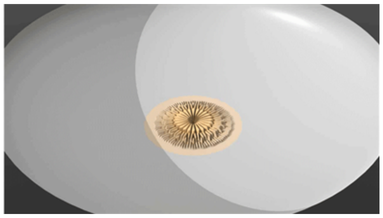
Flood Weight Properties

Enable	Enables the flood weight operation.
Map Type	Specify if the flood weight operation should be assigned to the following weight map types: • **Weight Map** - when the operation searches for the map, it only looks for standard weight maps on the parent mesh that use the **Map Name**. • **Weight Container** - when the operation searches for the map, it only looks for weight containers in the scene which use the **Map Name**.
Map Name	The name of the weight map or weight container where the flood weight operation should be assigned. This operation must be assigned to an existing map or container.
Select Map	Selects the map specified in **Map Name** so you can view it in the viewport and make edits if required.

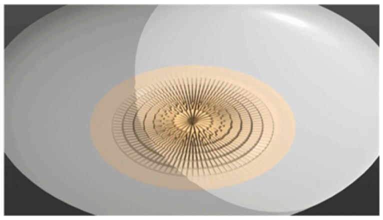
High/Low	Select **High** to flood the island with the highest weight found, or select **Low** to flood the island with the lowest weight found.

Grow Weight

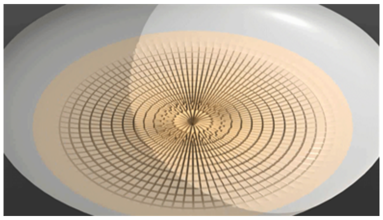
The grow weight mesh operation grows or shrinks a weight map based on a gradient value and an amount of steps. This operation searches the mesh for weight values, and grows the weight map from the vertex (or vertices) which use the highest weight value. In the following example, the vertices in the center of the mesh have the highest value of 1, so the grow weight operation grows the map out from there.



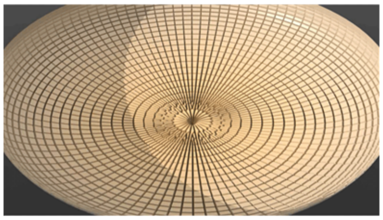
A **Steps** value of **2**.



A **Steps** value of **6**.



A **Steps** value of **15**.



A **Steps** value of **24**.

To add a grow weight operation to your weight map:

1. Ensure a weight map exists on your mesh. To learn how to add a weight map procedurally, see Creating a New Vertex Map Procedurally and Setting Weight Values.
2. In the **Model** layout, go to the **Mesh Ops** tab and click **Add Operator**.
3. Type **Grow Weight** into the search and double-click the **Grow Weight** operation.
4. In the Grow Weight **Properties**, select from the following in the **Map Type** dropdown:
    - **Weight Map** - the grow weight operation looks for weight maps on the selected mesh, instead of weight containers in the scene. Select this option if you are using a weight map on the selected mesh. See Weight Maps.
    - **Weight Container** - if you are using a weight container instead of a weight map, make sure to select **Weight Container**. See Creating a Weight Container.
5. Type the name of the weight map or weight container that you would like to grow in **Map Name**.

To see the list of available vertex maps in your scene, click **Vertex Map List** in the **Modo** layout, or the **Lists** tab in all other layouts.

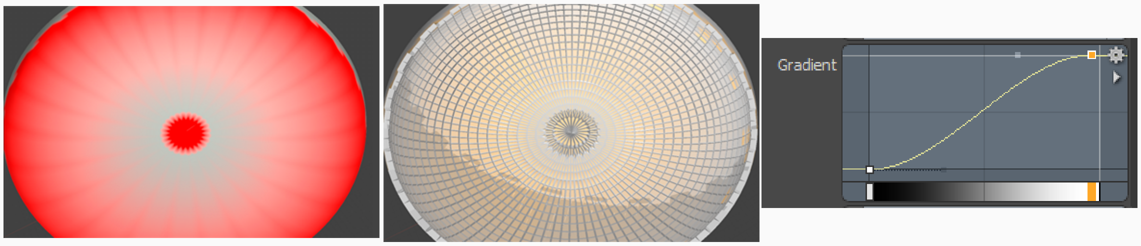
6. Increase the **Steps** value to grow the weight map based on the **Gradient** values.
7. Move the **Gradient** handles to watch how the different values in the gradient affect the grow weight result. Press **Ctrl**+**Alt** and click to add handles to the gradient.

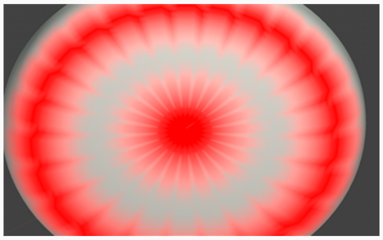
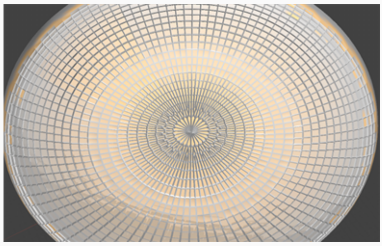
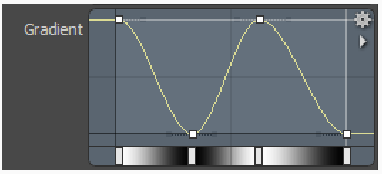
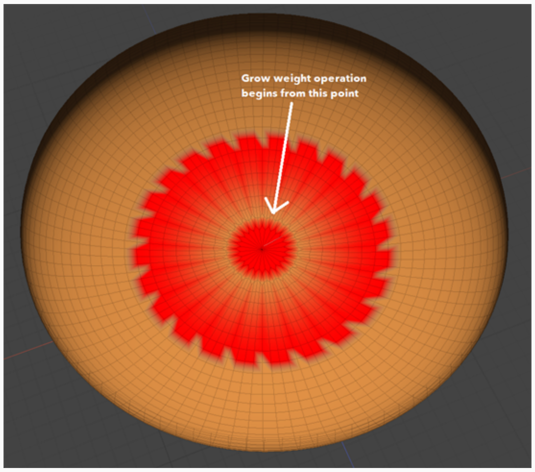
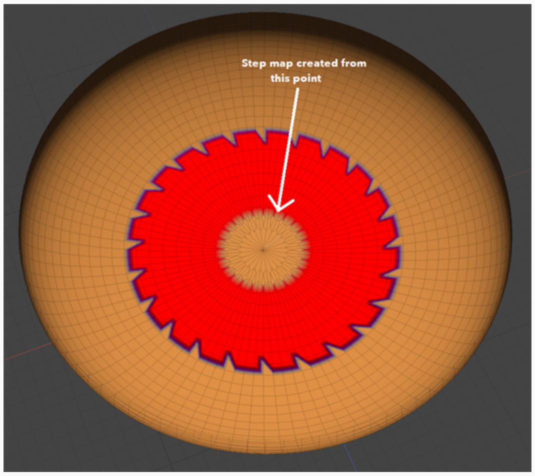
You can then assign other mesh operations to your mesh operation stack, such as a polygon bevel, and see how the new weight map affects the bevel.

> **Tip:** You can add a **Select By Index** operation in the **Selection** menu of the grow weight operation to select a vertex index number to start the grow operation from. For more information on procedural selection, see Procedural Selection.

Grow Weight Properties

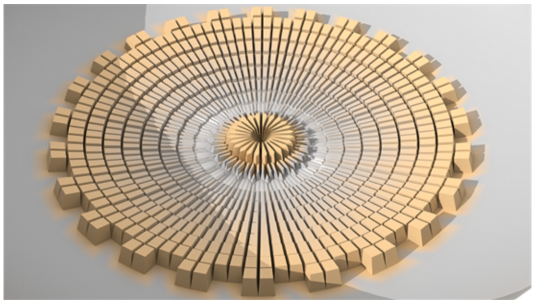
Enable	Enables and disables the operation.
Use World Transform	Sets the coordinates from Local Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Map Type	Specify if the grow weight operation is assigned to the following weight map types: • **Weight Map** - when the operation searches for the map, it only looks for standard weight maps on the parent mesh that use the **Map Name**. • **Weight Container** - when the operation searches for the map, it only looks for weight containers in the scene which use the **Map Name**.
Map Name	The name of the weight map or weight container the grow weight operation is being assigned to. This operation must be assigned to an existing map or container.
Select Map	Selects the map specified in **Map Name** so you can view it in the viewport and make edits if required.
Steps	The amount of steps to grow or shrink your weight map.
Grow/Shrink	Select whether you want the operation to grow or shrink the existing weight map: • **Grow** - grows the weight map and maps the vertices found in each **Step** to the **Gradient**. For example, if **Steps** is set to **10**, the operation attempts to grow connected vertices ten times, on each step only choosing vertices that haven't

	been hit by the operation previously. • **Shrink** - fades weights on a weight map away. This option requires a selection in the operation to work. You can use the **Select By Index** operation in the **Selection** menu for the grow weight operation to select a vertex to start shrinking from. For more information on procedural selection, see Procedural Selection.
Grow Connection Type	Select whether the grow weight operation grows to all vertices on a connected polygon, or connected vertices only: • **Polygon** - grows the weight map based on connected polygons and assigns the weight map value to all vertices on the found polygon. • **Vertex** - grows the weight map based on connected vertices. In **Vertex** mode, artifacts can occur if only half the vertices in a polygon are assigned a weight value. In this case, use **Polygon** instead.
Threshold	When the grow operation finds the initial weights to grow from, you can add a threshold value to also include weights which are a % of the highest weight value found. For example, if the highest weight on your mesh is 1.0, and the **Threshold** is 10%, the grow weight operation also includes vertices with a weight of 0.9 and grows from them too. If the highest weight on your mesh is 1.0, and the **Threshold** is 50%, the grow weight operation also includes vertices with a weight of 0.5 and grows from them.
Gradient	Define the gradient used by the grow weight operation to grow the values across the vertices on the mesh. Drag the gradient handles to move the gradient and press the middle mouse button on the gradient to add new handles. The following examples show the effect the grow weight gradient has on the weight map and mesh with a polygon bevel applied. 

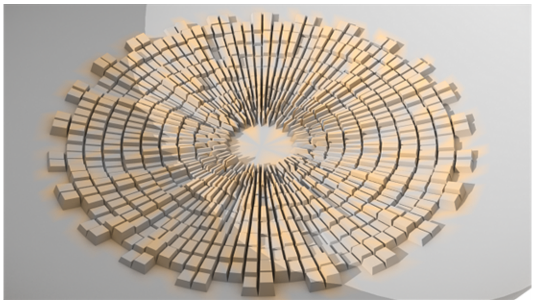
	  
Border Map Name	Allows you to input a weight map to be used as a border map for the grow weight operation. When the grow weight operation hits a value in the border map that is greater than the grow value, it stops growing.
Select Border Map	Selects the map specified in **Border Map Name** so you can view it in the viewport and make edits if required.
Flood Border	When enabled, once the grow weight operation hits the border defined in **Border Map Name**, vertices outside the border are flooded with the highest weight map value.
Use Falloff	Enable to use a falloff. The falloff can be selected in the **Tool Pipe** menu of the grow weight operation. The operation applies the grow, then applies the falloff weight after the grow.
Start Falloff	If the step where the vertex was found is equal or above the percentage defined here, the falloff is applied.
Save Step Map	When enabled, creates a new weight map called **StepMap** that shows where the first vertices were found to grow from for the grow weight operation. The step map stops assigning weights at the value defined in **Steps**.  The weight map with a grow weight operation applied.  The **StepMap**, which assigns all vertices after the first grow step the weight it found at the first grow step.

Jitter Weight

The jitter weight operation adds random number jitter to your vertex maps. This allows you to introduce more organic looking shapes and animations to your work, especially if you feel your work is looking too rigid. In the following example, a jitter is applied to a rigid and uniform looking mesh to add texture.



A polygon bevel using a weight map with no jitter operation assigned.



A polygon bevel using a weight map with a jitter weight operation assigned.

Jitter Weight Properties

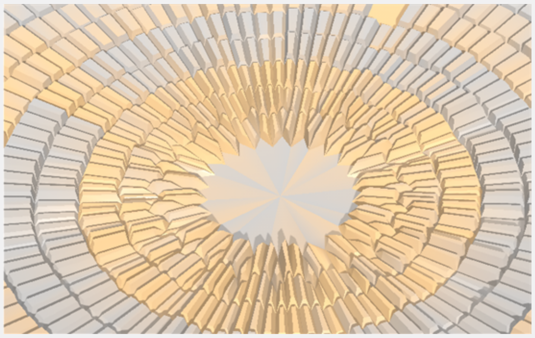
Enable	Enables or disables the operation.
Use World Transform	Sets the coordinates from Local Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Map Name	The name of the weight map or weight container where you want to apply the jitter weight operation. This operation must be assigned to an existing map or container.
Select Map	Selects the map specified in **Map Name** so you can view it in the viewport and make edits if required.
Map Type	Specify if the grow weight operation is assigned to the following options: • **Weight Map** - when the operation searches for the map, it only looks for standard weight maps on the parent mesh that use the **Map Name**. • **Weight Container** - when the operation searches for the map, it only looks for weight containers in the scene which use the **Map Name**.
Seed	The seed for the random number generator.
Minimum	You can set the minimum amount of jitter you would like on your weight map using the

<table>
<tr><td></td><td>

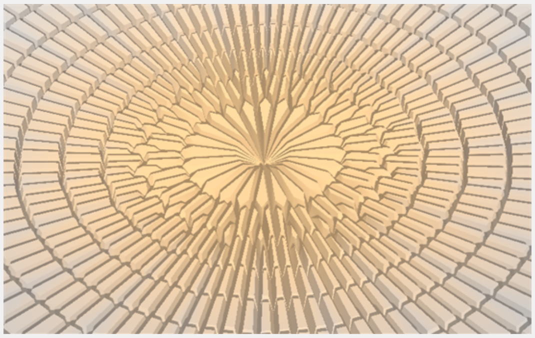
gradient editor. If you would like your weight map to have jitter everywhere, ensure the **Minimum** is set to a value above 0.

With the gradient editor, you can define the minimum jitter for current weights on the weight map. For example, at position 1 on the graph, you can define the minimum jitter applied to vertices with a weight of 1 applied.

Note: You can see the most amount of jitter when **Minimum** is 0, and **Maximum** is 1.



Minimum set to 0.



Minimum set to 0.3.

Note: See Gradient Editor for more information on using the gradient editor.

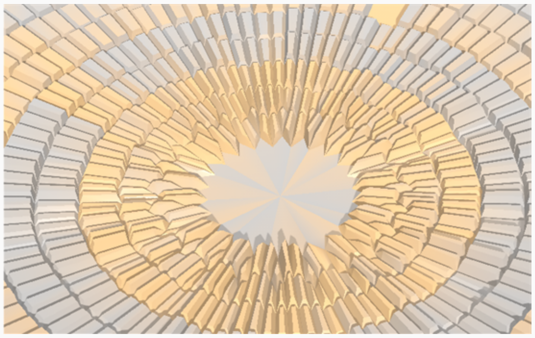
</td></tr>
<tr><td>

Maximum

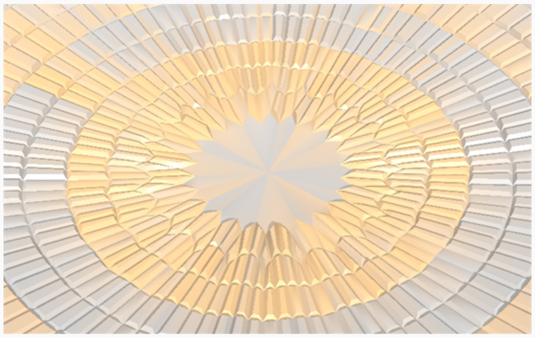
</td><td>

Set the maximum amount of jitter you would like on your weight map using the gradient editor. This is useful if you want to limit the amount of jitter across your weight map.

With the gradient editor, you can define the maximum jitter for current weights on the weight map. For example, at position 1 on the graph, you can define the maximum jitter applied to vertices with a weight of 1 applied.



Maximum set to 1.



Maximum set to 0.5.

</td></tr>
</table>

Remap Weight

The remap weight operation takes an existing weight map, and remaps the values based on a gradient input. If a target map is specified, it remaps the weights to a new weight map, otherwise it overwrites the existing weight map.

Enable	Enables or disables the operation.
Use World Transform	Sets the coordinates from Local Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Map Type	Specify if the remap weight operation is assigned to the following options: • **Weight Map** - when the operation searches for the map, it only looks for standard weight maps on the parent mesh that use the **Map Name**. • **Weight Container** - when the operation searches for the map, it only looks for weight containers in the scene which use the **Map Name**.
Map Name	The remap operation is assigned to the map name you provide here. It must be a map that already exists in the scene.
Select Map	Selects the map specified in **Map Name** so you can view it in the viewport and make edits if required.
Target Map Type	If a target map is provided, the remap is stored to the target map. If no target map is provided, the remap operation overwrites the map provided in **Map Name**. The options are: • **Weight Map** - when the operation searches for the map, it only looks for standard weight maps on the parent mesh that use the **Map Name**. • **Weight Container** - when the operation searches for the map, it only looks for weight containers in the scene which use the **Map Name**. • **Particle Item** - when the operation searches for the map, it only looks for particle item maps in the scene which use the **Map Name**. See Particle Item Maps.
Target Map Name	Enter the name of the target map. If a map with that name does not exist, a new map is created. If a target map is provided, the remap operation is stored to the target map. If no target map is provided, the remap operation overwrites the map provided in **Map Name**.
Select	Selects the map specified in **Target Map Name** so you can view it in the viewport and

Target Map	make edits if required.
Falloff Type	You can specify if falloffs applied to the remap operation are applied before or after the remap: • **None** - no falloffs are used. • **Before** - the falloff is applied before the remap operation is performed, so values that have had a falloff applied are remapped. • **After** - the falloff is applied after the remap operation is performed, so the remapped values have the falloff applied. **Note:** You can add falloffs to your remap weight operation by clicking **Add Tool Pipe** in the **Tool Pipe** under the **Remap Weight** operation in the **Mesh Ops** list.
Use Inputs	Choose how to apply the remap to the given weight map values: • **Gradient** - maps the existing weights to new weights using the gradient editor. • **Inputs** - maps the existing weights to new weights using the **Input Maximum**, **Input Minimum**, **Output Minimum**, and **Output Maximum** values. • **Both** - maps the existing weights to new weights using both the **Gradient** and **Inputs** options. The operation does the **Inputs** remapping first, using the input values, and then maps the value obtained from that using the gradient.
Input Minimum	The minimum input value of the existing weight map that should be remapped. For example, if your existing weight map has a lowest weight value of -1.0, you might want to specify that the **Input Minimum** is -1.0. Or, if you want to limit the remap range, you can specify a weight value higher than -1.0.
Input Maximum	The maximum input value of the existing weight map that should be remapped. For example, if your existing weight map has a highest value of 1.0, you might want to specify that the **Input Maximum** is 1.0. Or, if you want to limit the remap range, you can specify a weight value lower than 1.0.
Output Minimum	The minimum value of the remapped weights. The **Input Minimum** is remapped to the **Output Minimum**. For example, if your **Input Minimum** is **-1.0**, you can remap all weights with that value to 0.0 by entering **0.0** in **Output Minimum**.
Output	The maximum value of the remapped weights. The **Input Maximum** is remapped to the

Maximum	**Output Maximum**. For example, if your **Input Maximum** is **1.0**, you can remap all weights with that value to 0.5 by entering **0.5** into **Output Maximum**.
Automatic Input: Min	Automatically finds the minimum weight on the weight map specified in **Map Name**. For example, if the lowest weight on your given weight map is -1.0, it automatically remaps that weight to the **Output Minimum**.
Automatic Input: Max	Automatically finds the maximum weight on the weight map specified in **Map Name**. For example, if the highest weight on your given weight map is 1.0, it automatically remaps that weight to the **Output Maximum**.
Remap Value	When **Use Inputs** is set to **Gradient** or **Both**, you can edit the gradient using the gradient handles to remap the weight values. **Note:** See Gradient Editor for more information on using the gradient editor.

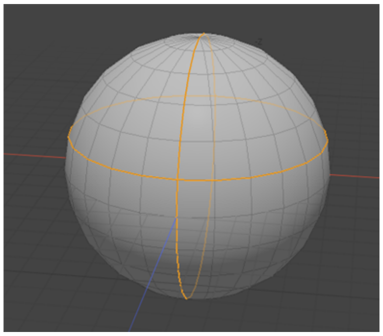
Set Seam

You can use the Set Seam mesh operation to set the values of a UV Seam (Cut) map. This is the procedural version of the direct modeling Cut Map tool.

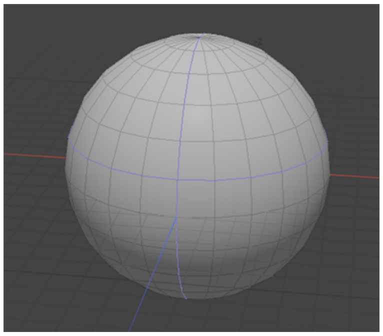
Note: For more information on the Cut Map tool, see Creating Seams.

You need to create a UV Seam map for your mesh, as described in Creating a New Vertex Map Procedurally. Then set seam values the following way:

1. In the **Vertex Maps** list on the lower right, under **Other Maps**, select the UV Seam map you created.
2. In the Item List, select your mesh item and in the 3D viewport, select the edges for which you want to set seam values.



3. In the Mesh Operations list, under **Mesh Operations** > **Vertex Map**, double-click **Set Seam**.
   This adds the Set Seam mesh operation to the list and opens its properties in the **Properties** panel on the right pane.
4. In the **Set Seam** Properties, set **Map Name** to the name of the UV Seam map you want to use.
   Seams are added along the selected edges. In the 3D viewport, the seams are highlighted in blue.



You can then use the UV Unwrap mesh operation to create UVs cut along the new seams. For more information, see Procedural UV Unwrap.

Set Seam Properties

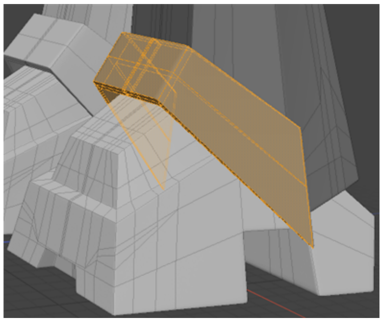
Enable	Enables/disables the Set Seam mesh operation.
Use World Transform	Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Map Name	The name of the UV Seam map to edit.

Set RGB

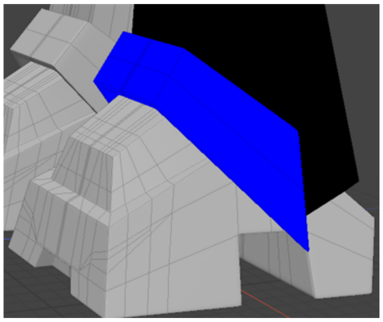
You can set values on RGB/A maps using the Set RGB mesh operation.

You need to create an RGB or RGBA map for your mesh, as described in Creating a New Vertex Map Procedurally. Then set the values the following way:

1. In the Item List, select your mesh item and in the 3D viewport, select the vertices, edges, or polygons for which you want to set the values.

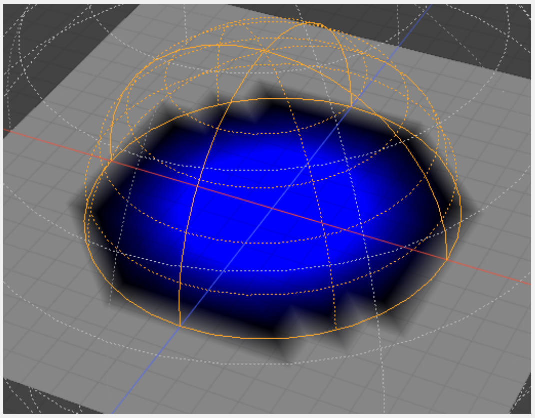
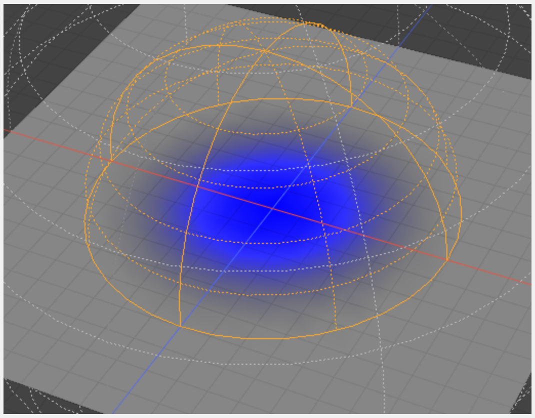


2. In the Mesh Operations list, under **Mesh Operations** > **Vertex Map**, double-click **Set RGB**.
   This adds the Set RGB mesh operation to the list and opens its properties in the **Properties** panel on the right pane.
3. In the Set RGB **Properties**, set the relevant **Map Type**, set **Map Name** to the name of the RGB or RGBA map you want to use, and edit the **Map Values**.
   The color of the selected vertices is updated in the 3D viewport.



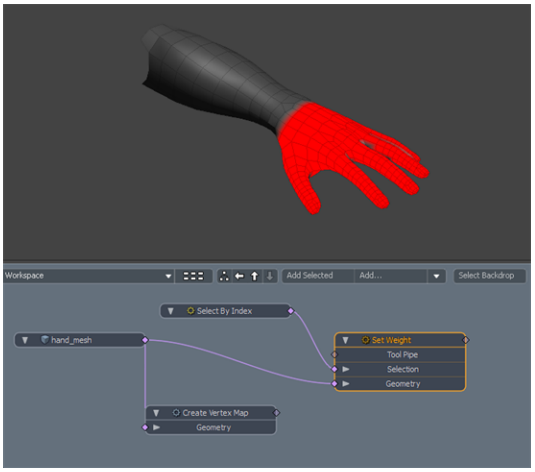
Set RGB Properties

Enable	Enables or disables the Set RGB mesh operation.
Use World Transform	Sets the coordinates from Model Space to World Space where vertices are defined relative to an origin common to all the objects in a scene.
Selection Type	The element type selected. This mesh operation can take a **Vertex**, **Edge**, or **Polygon** selection.
Map Type	Choose between **RGB** and **RGBA** maps. For more information on the differences, see

	Working with Weight Maps. **Note:** If a map of the set type can't be found, a new map is created.
Map Name	The name of the map to edit.
Map Value - R/G/B/A	The red, green, blue, and alpha values. **Note: Map Value - A** is only used if **Map Type** is set to **RGBA**.
Alpha Uses Falloff	If you're using a falloff with this option enabled, the falloff weight is applied to the Alpha value. When disabled, Alpha is a set value.  Alpha Uses Falloff disabled  Alpha Uses Falloff enabled
Additive	When enabled, the **Map Value** is added to the current map value. When disabled, the current map value is set to the amount specified in **Map Value**.
Weight Method	Determines how the red, green, blue, and alpha component is defined. • **Per-Attribute** - the red, green, blue, and alpha components are defined by the **Map Value R/G/B/A** attributes. • **Gradient Mapped** - allows you to use the **Gradient** to define the red, green, blue, and alpha components.
Gradient	When **Weight Method** is set to **Gradient Mapped**, use this gradient to define the RGBA values of the RGB map. **Note:** See Gradient Editor for more information on using the gradient editor.

Vertex Maps in the Schematic Viewport

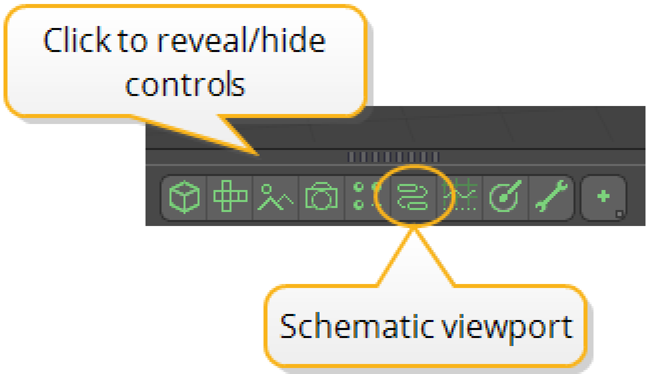
The Vertex Map mesh operations are also available in the Schematic viewport.


Workspace
Add Selected
Add...
Select Backdrop
Select By Index
hand_mesh
Set Weight
Tool Pipe
Selection
Geometry
Create Vertex Map
Geometry

Create Vertex Map and Set Weight in the Schematic

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.

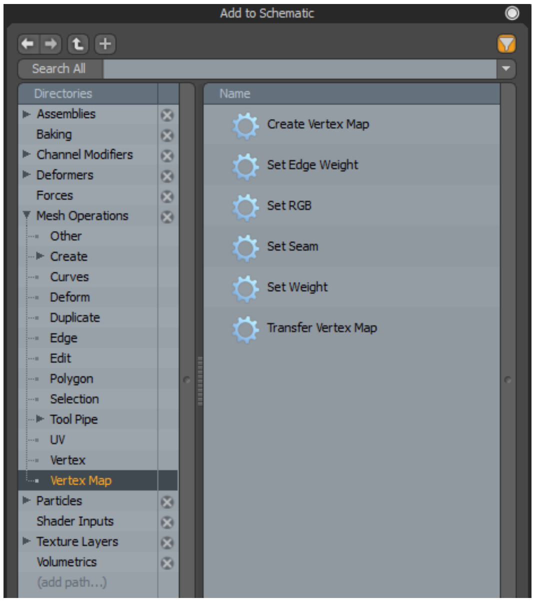

Click to reveal/hide controls
Schematic viewport

>

MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

To access the Vertex Map tools, click **Add...** . The Vertex Map tools can be found under **Mesh Operations** > **Vertex Map**.


Add to Schematic
Search All
Directories
Assemblies
Baking
Channel Modifiers
Deformers
Forces
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Edit
Polygon
Selection
Tool Pipe
UV
Vertex
Vertex Map
Particles
Shader Inputs
Texture Layers
Volumetrics
(add path...)
Name
Create Vertex Map
Set Edge Weight
Set RGB
Set Seam
Set Weight
Transfer Vertex Map

Topological Vertex Map Transfer

The **Topological Transfer** command can transfer UV, morph, color, vertex normal, and weight in selected vertex maps. The process can be applied to multiple surfaces at the same time.

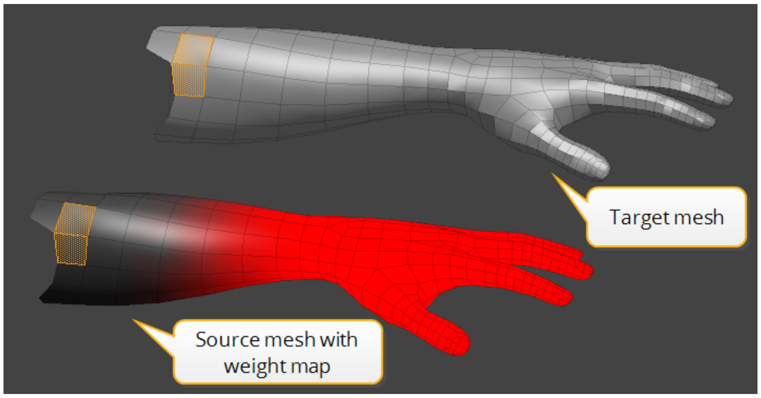
> **Note:** You can see your existing vertex maps and create new ones in the Vertex Map List. For information on using vertex maps, see Working with Vertex Maps.

To transfer a map:

1. Select any two adjoining polygons on the source surface.
2. Add the exact same polygons on the target surface in the same order to the selection by holding Shift and clicking the polygons.

   The source and the target can be in the same or in different mesh layers. The following example shows the transfer of a weight map from one mesh to another.

> **Tip:** To display weight maps in the 3D viewport, click the Show Weightmaps button at the top-left section of the viewport.



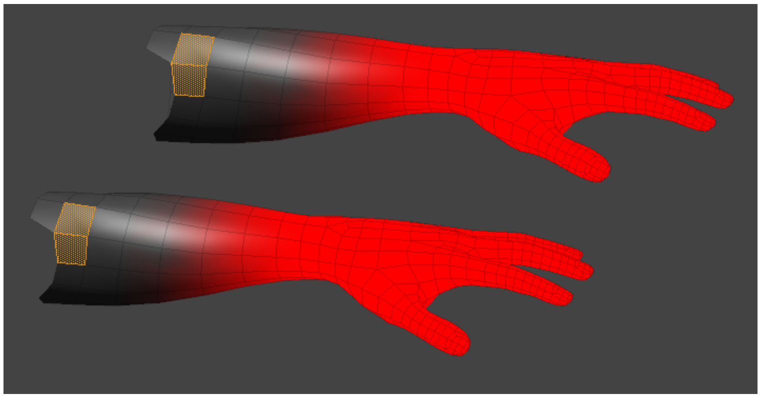
3. In the menu bar, click **Vertex Map** > **Topological Transfer...**

   The **Vertex Map Topological Transfer** dialog displays.

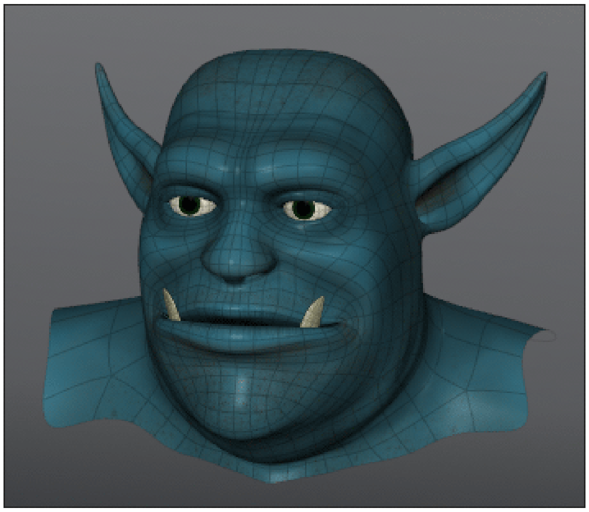
4. Set the **Type** of vertex map you want to transfer. If you want to transfer all maps of the selected type, enable **All Maps**. When disabled, Modo transfers the currently selected map only.

5. Click **OK**.

   Your selected vertex map is transferred to the target surface.

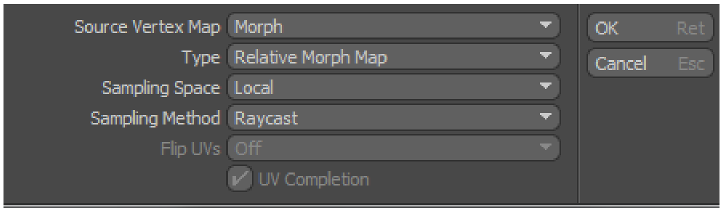


Transfer Vertex Map



With Modo, you can transfer any Vertex Map from one item to another arbitrary item. In a way, this is similar to baking a texture map, but working with vertex data instead. For example, you can take a series of morph maps for different facial expressions and transfer them to another similar object. For the best results, the objects need to be similar. In the examples, Modo's sculpting tools were used to have the old man's face loosely resemble the ogre's facial size and position of features — matching lips to lips, brow to brow, and so on. Doing this gives more predictable results as corresponding vertices are closer together.

To use the **Transfer Vertex Map** command, you must have base Vertex Maps of the proper type present in both items before using the transfer function. Keep the item you want to transfer from as a background item (visible, but not selected). In the forground, have the item you're transfering to with the appropriate Vertex Map selected in the **Vmap** list. Then choose **Vertex Map > Transfer Vertex Map**, and specify the settings in the **Transfer Vertex Map** dialog.



Source Vertex Map: Morph
Type: Relative Morph Map
Sampling Space: Local
Sampling Method: Raycast
Flip UVs: Off
☑ UV Completion

Source Vertex Map: Indicates the appropriate Vertex Map from the background item that you are transfering the map from. Modo only lists the types of maps that correspond to the specified **Type**. If there are no corresponding maps associated with the foreground object, you cannot select a map.

Type: Specifies the type of Vertex Map for the transfer operation. Both maps (the source and the target) must be identical in function. For example, you cannot transfer a color Vertex Map to a morph map.

Sampling Space: Determines the coordinate system for sampling the positions. If you select **Local**, Modo samples the position in the target item's space; otherwise, it samples in **World** space.

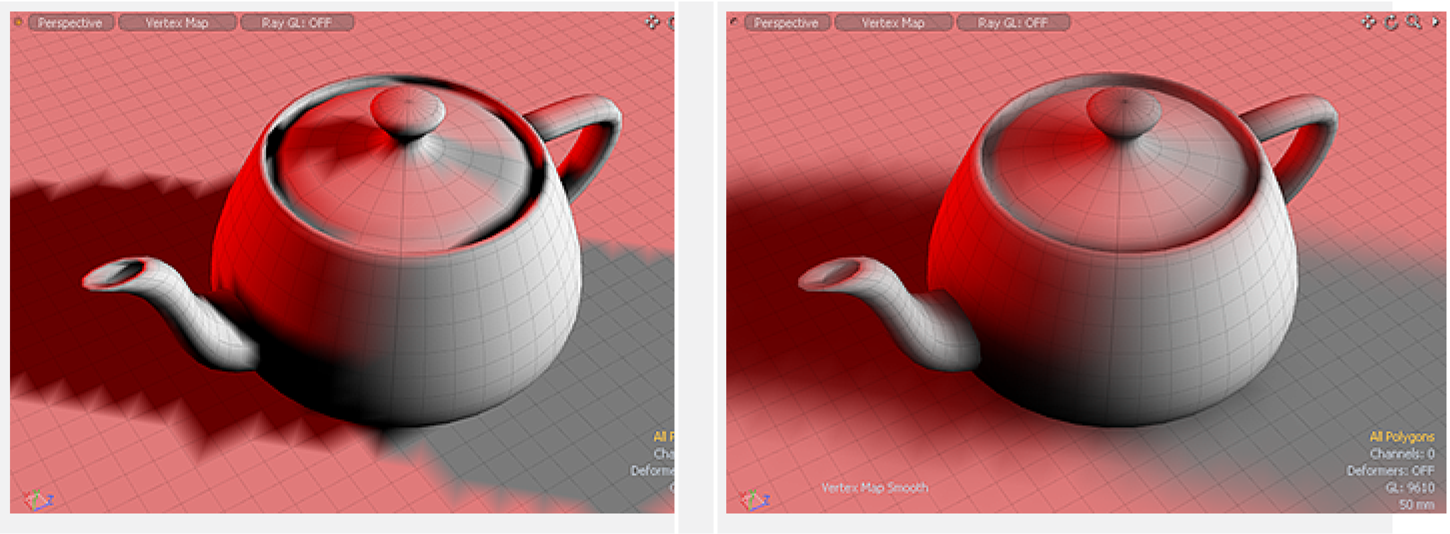
Sampling Method: Specifies the method to sample positions on the target Mesh Item from the source position. If you select **Raycast**, Modo computes the hit position along the normal vector at the source position on the source Mesh Item. With **Distance**, Modo finds the closest position on the target Mesh Item from the source position.

Flip UVs: Reverses the sampled UV values to horizontal or vertical. This is useful when making mirrored UV polygons.

UV Completion: When enabled, generates discontinuous UV polygons based on the sampling polygon information. Modo samples each position from the different polygons over UV seams. This should always be selected, but there are rare instances where you may want it disabled.

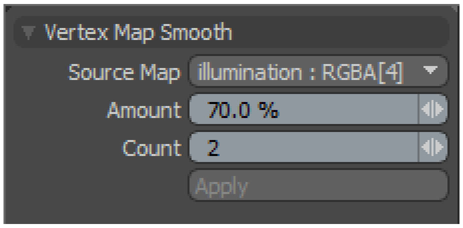
Smooth Vertex Map

Modo provides a Vertex Map smoothing tool that is useful for softening (or blurring) the values in a Vertex Map. You can also apply this tool to Weight Maps as well as Vertex Color maps. When applied to a Vertex Color map generated with illumination baking, Modo produces a look similar to global illumination but which is applicable to real-time uses (such as in games).



To apply vertex smoothing, you must first have a base Vertex Map of the proper type present: either a Weight Map or a Vertex Color map. The Smooth tool has no effect on UV or Morph maps. Next, make sure the viewport is set to the **Vertex Map** display mode with the map selected so that the effect of interactive editing shows. Then activate the tool by choosing **Vertex Map > Smooth Tool**. In the tool's **Properties** panel, select the map to edit in the **Source Map** list and adjust the size of the blur with the **Count** option.

value. To deactivate the tool and to accept the changes when finished, press the **spacebar**.



Source Map: illumination : RGBA[4]
Amount: 70.0 %
Count: 2

Vertex Map Smooth

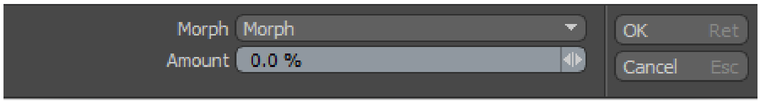
Source Map: Specifies the Vertex Map for editing.

Amount: Adjusts the overall strength of the blurring effect on the map.

Count: Defines the size of the blur across the number of adjacent vertices. A value of **1** averages one adjacent vertex with each vertex; a value of **2** averages two adjacent vertices with each vertex; and so on.

Apply Morph

With the **Apply Morph** command you can apply an existing morph map into a new one or directly onto the base mesh, which is usually necessary to export a mesh for using in an external application. To use the commmand, choose **Vertex Map** > **Apply Morph**. In the **Apply Morph** dialog, select a named **Morph** and specify an **Amount**. Modo moves the vertices of the currently active map based on the values of the selected morph.

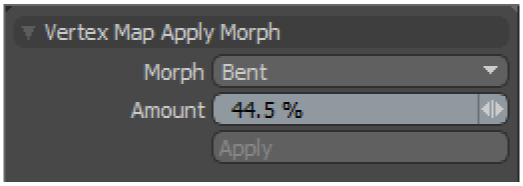


Morph: Morph
Amount: 0.0 %

Morph: Indicates the relevant morph map to apply. This list shows all morph maps currently available in the active mesh.

Amount: Determines the strength of the applied morph map as a percentage. Set an exact percentage for Modo to use when applying the currently selected morph map. You can achieve dramatic results by indicating a value greater than **100%** or by using a negative number.

Morph Tool

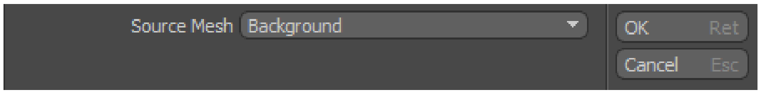


You can use the Morph tool to apply a morph Vertex Map to another morph map on the same mesh. This is a way to move a morph value from one map to another, to edit a morph map further, or to apply multiple maps to a single map for simplified deforming. You can apply this command multiple times to mix together different morph maps into the new map.

Before invoking the command, make sure that you select the Target Vertex Map in the **Lists** viewport. Then choose **Vertex Map** > **Apply Morph**. In the **Vertex Map Apply Morph** dialog, select the morph map to apply from the **Morph** list and then specify an **Amount**. Modo uses the amount to control the strength of the map's application. After you click **Apply**, Modo sets the values into a new map.

Background to Morph

With the **Background to Morph** dialog, you can take geometry in a background layer and apply its vertex positions to the foreground layer as a morph map. For the best results, both the Target and Source meshes should have the same number of vertices in the same winding order. You can also use this to combine a multi-mesh morph from an external application into Modo's native single mesh with a morph deformer. Modo uses the **Source Mesh** as the base for the morph. You need to create and select a morph map before opening this dialog.

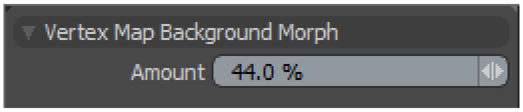


Source Mesh: Specifies the starting mesh to use when applying this function.

- **Background**: Uses the visible background layer (or the layer directly below the active layer when there are multiple background item layers) to define the vertex positions for the morph map.
- **Last Selected**: Takes the last selected item layer in the History stack to determine the source.

Background to Morph Tool

With the background layer as the Target shape, the Background Morph tool moves the vertices of the mesh layer toward those of the background geometry layer. The tool is very useful when combined with various falloff modifiers.



Vertex Map Background Morph

Amount 44.0 %

Amount: Determines (with a percentage value) how closely the foreground mesh should match the vertices in the background layer. You can set this to negative values or values greater than **100%** to produce dramatic effects.

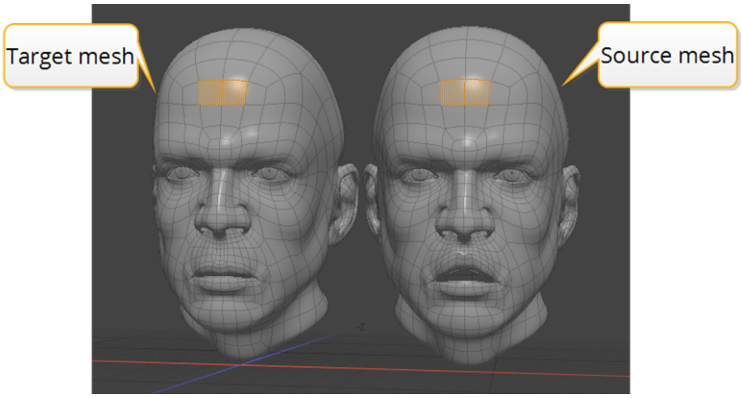
Topological Morph Creation

The **Topological Morph Creation** tool allows you to create morph maps on multiple selected meshes or parts, based on the geometry of a source mesh or part. For example, if you generate your morph shapes as separate meshes instead of morph maps, you can merge them all into morph maps using this tool.

To use the tool:

1. Specify the source by selecting any two adjoining polygons on it.
2. Add the corresponding two polygons in the same order on the target to the selection by holding **Shift** and clicking on the polygons.

   The source and the target can be in the same or in different mesh layers.



3. In the menu bar, click **Vertex Map** > **Topological Morph Creation...**

   This opens the **Topological Morph Creation** dialog.

4. Enter a name for the new morph map in the **Morph Name** field, and if you want to create an absolute morph map instead of a relative one, enable **Absolute**.

> **Note:** A relative morph map offsets vertices relative to their current position. An absolute morph map positions vertices to a specific absolute position.

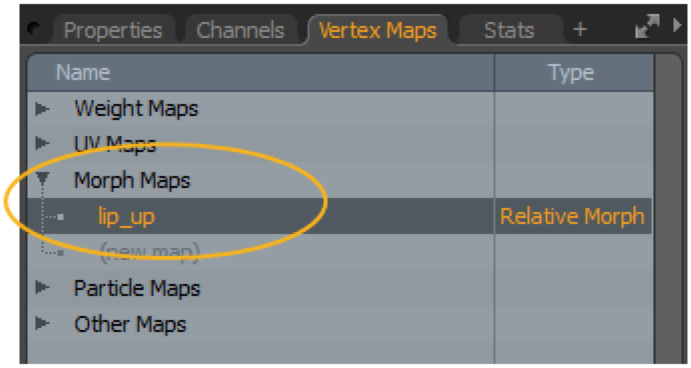
5. Click **OK**.

   A new morph map is created based on the geometry of the source, and is added to the source and the target.

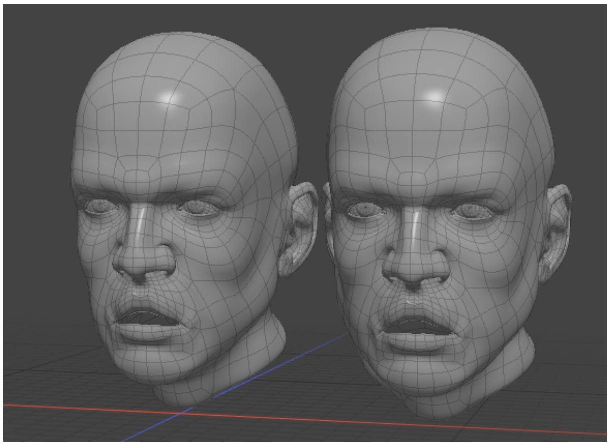
6. Select your target and open the **Vertex Maps** list on the lower right panel.

> **Note:** If the **Vertex Maps** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Vertex Map List**.

7. Expand **Morph Maps**, and click the newly added morph map.


Properties
Channels
Vertex Maps
Stats
+
Name
Type
Weight Maps
UV Maps
Morph Maps
lip_up
Relative Morph
(new map)
Particle Maps
Other Maps

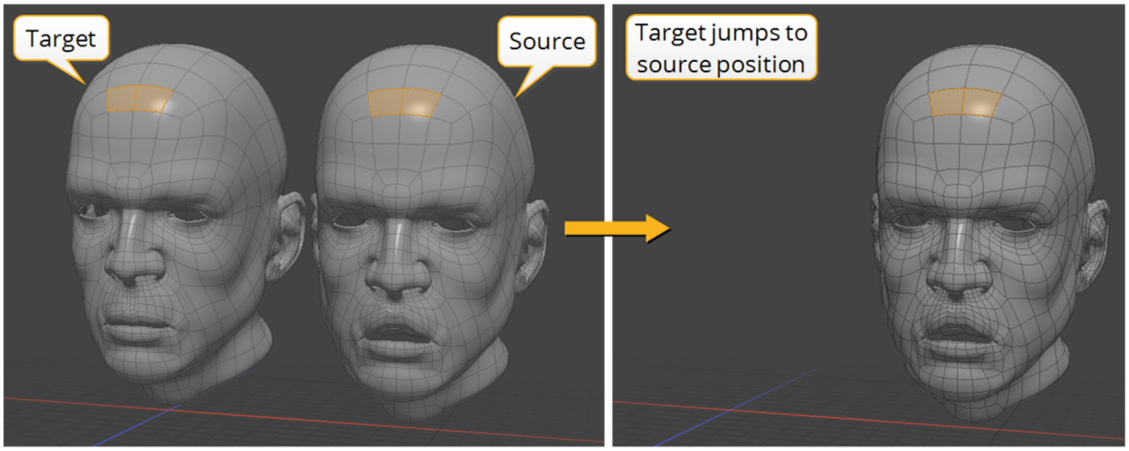
The map is displayed on your target mesh in the 3D viewport.



Morph Creation in the Same Mesh

When using Topological Morph Creation on parts of the same mesh item, applying the command transforms the target to the source part's location. This is because each polygon island is relative to the origin, so when Modo creates the morph, it moves the polygons to match them.

See the image below for an example:



You can avoid this by creating a morph map that sets the position of the source to the position of the target, and activate the morph map before running the Topological Morph Creation command.

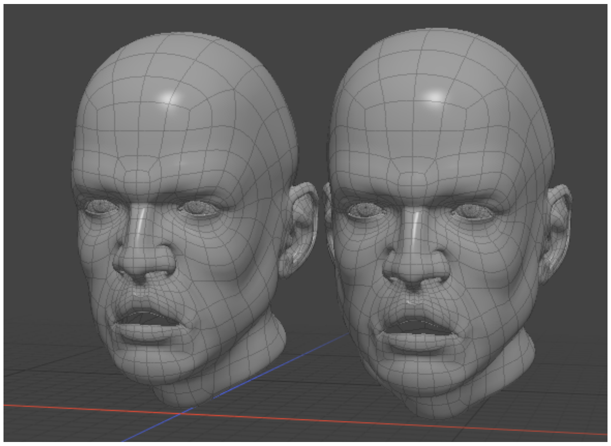
1. In **Polygons** selection mode, double-click the source part to select it, and in the **Vertex Maps** list, under **Morph Maps,** click **(new map)**.

   This opens the **Create New Vertex Map** dialog.
2. Enter a **Vertex Map Name** and select **Relative Morph Map** as the **Vertex Map Type**, then click **OK**.

   Your new morph map appears in the **Vertex Maps** list.
3. With the new morph map selected in the list, move your source on top of the target part.
4. Select two adjoining polygons on the source and the target.
5. Make sure that the morph map is selected in the **Vertex Maps** list.
6. Run the **Topological Morph Creation** command from the menu bar.

   Your new morph map is created.
7. Select it in the **Vertex Maps** list.

   The morph is applied to the target but it remains in position.

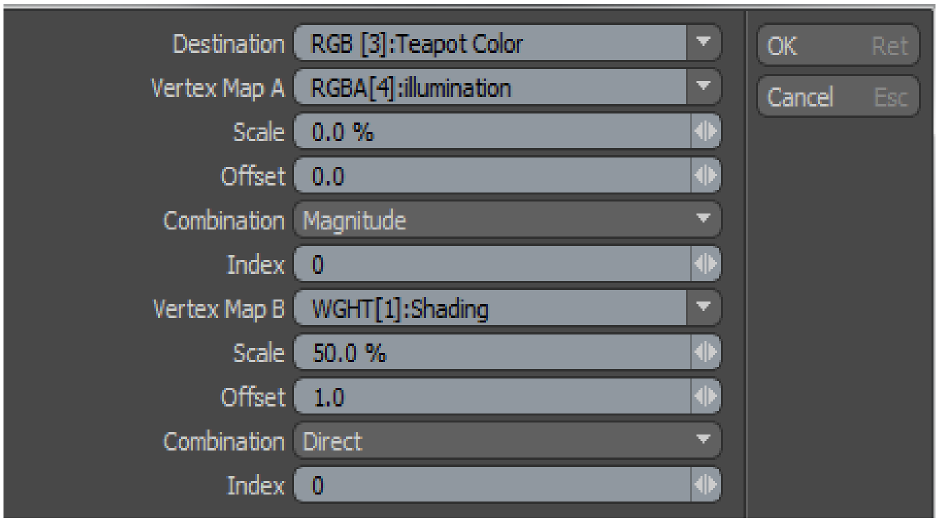


Vertex Map Math

The Vertex Map Math tool is a generalized command that operates on multiple Vertex Maps. With this function, you can scale values, compose maps in different ways, and convert map types. Specific examples include: generating a Weight Map from a U or V coordinate and making a Weight Map from the magnitude of a morph map. You can also create a map with the maximum, minimum, sum, or difference of two maps, or clear one map's entries based on the values in another map.

With the Vertex Map Math tool, you can make a formula for the destination (assuming **Amount** is **100%**) similar to:

`dest = ((sourceA * scaleA) + offsetA) Operation ((sourceB * scaleB) + offsetB)`
with the operation being **+**, **-**, or similar operators.



Destination: RGB [3]:Teapot Color
Vertex Map A: RGBA[4]:illumination
Scale: 0.0 %
Offset: 0.0
Combination: Magnitude
Index: 0
Vertex Map B: WGHT[1]:Shading
Scale: 50.0 %
Offset: 1.0
Combination: Direct
Index: 0

The options for the Vertex Map Math tool are:

Destination: Specifies the name of the destination Vertex Map. Modo creates a map if needed.

Amount: Specifies the amount (as a percentage) of the computed value that goes into the dest, blended like alpha.

Combine: Specifies the math operation used to combine the terms (SourceA, SourceB).
Then for each source (#):
map# - the vmap index
scale# - a multiplier to apply to the offset vmap value: src# = scale# * (val# + offset#)
offset# - an amount to add to the vmap value
mode# - the way to deal with multi-dimensional vmap values. if src# is morf and dest is uv:
"Direct" - copy direct component to component, so morf.x->uv.u, morf.y->uv.v
"Component" - pick a single component (index#) so morf.index#->uv.u, morf.index#->uv.v
"Magnitude" - The 'vector length' of the source vmap value
"Maximum","Minimum" - The max. or min. of the components of the source vector
index# - the component of the source map that is used

You can use this tool to make Weight Maps of morph magnitude, go between weights and UVs, invert vmaps, and so forth. The destination map can also be the source. If you do not specify a, the value is 0, but Modo applies the offset and scale values for a constant value term to be available.

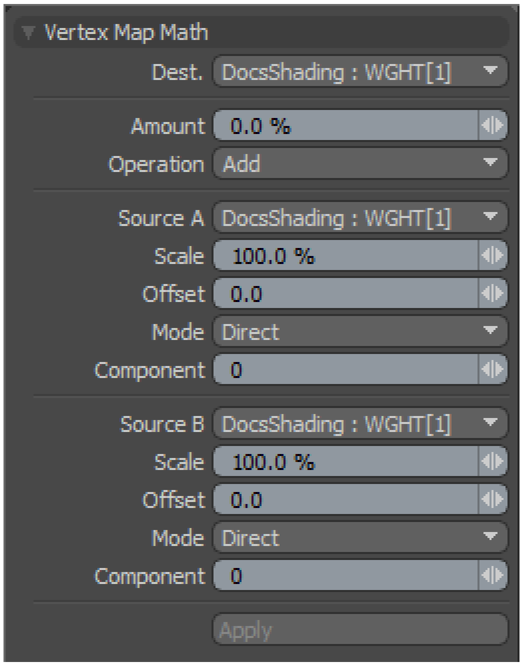
Vertex Map Math Tool

The Vertex Map Math tool is a generalized tool that operates on multiple Vertex Maps. You can use it to scale values, compose maps in different ways, and convert map types. Specific examples include: generating a Weight Map from a U or V coordinate and making a Weight Map from the magnitude of a morph map. You can also create a map with the maximum, minimum, sum, or difference of two maps, or clear one map's entries based on the values in another map.

With the Vertex Map Math tool, you can make a formula for the destination (assuming **Amount** is **100%**) similar to:

```
dest = ((sourceA *scaleA) + offsetA) Operation ((sourceB *scaleB) + offsetB)
```

with the operation being **Add**, **Subtract**, or similar operators.



Dest. DocsShading : WGHT[1]
Amount 0.0 %
Operation Add
Source A DocsShading : WGHT[1]
Scale 100.0 %
Offset 0.0
Mode Direct
Component 0
Source B DocsShading : WGHT[1]
Scale 100.0 %
Offset 0.0
Mode Direct
Component 0

The options for the Vertex Map Math tool are:

Destination: Specifies the name of the destination Vertex Map. Modo creates a map if needed.

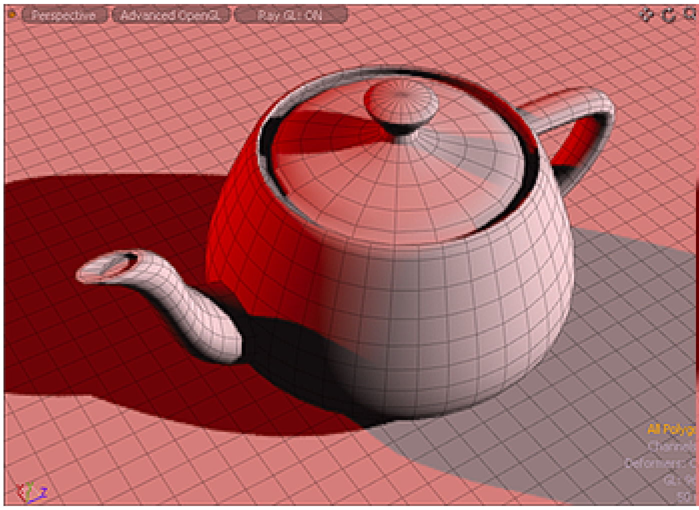
Amount: Specifies the amount (as a percentage) of the computed value that goes into the dest, blended like alpha.

Operation: Specifies the math operation used to combine the terms Source A and Source B.
Then for each source (#):
map# - the vmap index
scale# - a multiplier to apply to the offset vmap value: src# = scale# * (val# + offset#)
offset# - an amount to add to the vmap value
mode# - the way to deal with multi-dimensional vmap values. if src# is morf and dest is uv:
"Direct" - copy direct component to component, so morf.x->uv.u, morf.y->uv.v
"Component" - pick a single component (index#) so morf.index#->uv.u, morf.index#->uv.v
"Magnitude" - The 'vector length' of the source vmap value
"Maximum","Minimum" - The max. or min. of the components of the source vector
index# - the component of the source map that is used

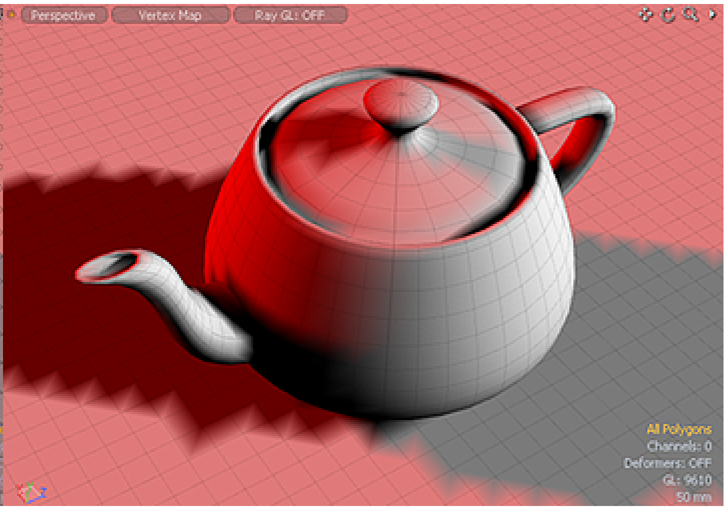
You can use this tool to make Weight Maps of morph magnitude, go between weights and UVs, invert vmaps, and so forth. The destination map can also be the source. If you do not specify a, the value is 0, but Modo applies the offset and scale values for a constant value term to be available.

Vertex Shader Baking

With the **Vertex Shader Baking** function you can bake (or burn in) the illumination colors of the scene directly to the vertices of the geometry as a vertex color map. This type of map is often used in real-time environments, such as games, as a way to limit processing requirements by pre-computing lighting in a scene. Because the color map stores values with the vertices themselves, the resolution of the resulting map is dependent on the resolution of the mesh.


Perspective
Advanced OpenGL
Ray GL: ON

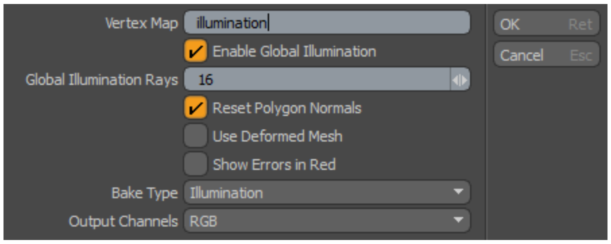
RayGL in-viewport render preview


Perspective
Vertex Map
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 9610
50 mm

The resulting baked Vertex Color map

Usage

Before baking, set up the target surfaces with the desired colorations. The resulting vertex color map takes not only illumination (all lighting including global illumination) into account, but is a combination of the surfacing and the lighting in the scene. Make sure that the target items are selected either in the 3D viewport or in the **Items** list. Then, from the menu bar, click **Vertex Map** > **Vertex Shader Baking** and set the function's options.



Vertex Map: illumination
☑ Enable Global Illumination
Global Illumination Rays: 16
☑ Reset Polygon Normals
☐ Use Deformed Mesh
☐ Show Errors in Red
Bake Type: Illumination
Output Channels: RGB

OK Ret
Cancel Esc

- **Vertex Map** - Provides a way to name the resulting vertex color map. You can accept the default name or type a different name. If the name already exists in the scene, Modo overwrites the values from the existing map with the new values.
- **Enable Global Illumination** - When enabled, Modo takes all global illumination (GI) from the scene into account. When disabled, Modo only uses illumination from direct light sources.
- **Global Illumination Rays** - Determines the number of rays used to calculate the global illumination. Vertices in a mesh are generally at a lower resolution than any bitmap that could be applied to them; therefore, very low values are generally acceptable. However, if there are obvious variances between vertex colors, increase this value from the default.
- **Reset Polygon Normals** - When enabled, Modo automatically fixes any degenerated polygon normals that might otherwise cause errors in the resulting vertex color map.
- **Bake Type** - Provides two options to control what Modo bakes (based on the **Bake Type** set) and where it stores the result of the bake (based on the **Output Channels** set):
  - **Illumination** - Bakes illumination on all surfaces in the scene using the light in your scene.
  - **Selected Render Output** - Bakes the render output based on your selection in the Shader Tree. Modo computes the effect of the selected render output at every vertex. For example, if you select a render output using the Segment ID effect, the vertices are colored by the segment color.
- **Output Channels** - Sets the output channels. The following options are available: **RGB**, **RGBA**, **R**, **G**, **B**, and **A**.

Tip: You can set different effects in each channel. For example, if you set the Output Channel to red, the blue and green channels are not affected. You can bake one effect into one channel and another effect into another channel. This is useful when scripting.

Viewing a Vertex Color Map

To view any vertex color map in the 3D viewport, set the viewport display option to **Vertex Map** and then select the map from the **Lists** viewport under **Other Maps**. Depending on the particular lighting and the desired effect, you may also apply vertex smoothing to the vertex illumination result to provide a softer result. For more information on smoothing, see Smooth Vertex Map.

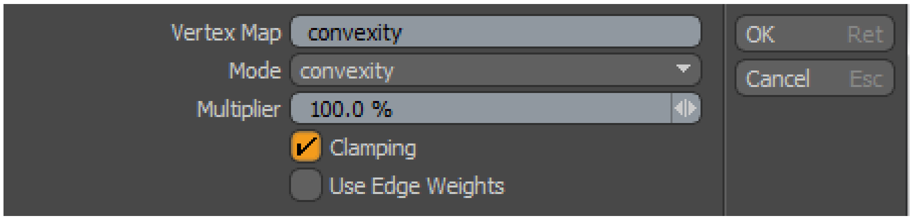
Note: The illumination colors are baked in to the individual vertices themselves, so, for the best results, the geometry needs to be present for the effect to work properly. To visualize the illumination baking, enable the viewport RayGL rendering option to provide an in-viewport reference of the results. When you apply the **Vertex Shader Baking** command, Modo samples the color at each vertex point in 3D space (including cast shadows and the effects of global illumination and surfacing) and applies the color at that particular point to the vertex. With well-modeled geometry, this can produce very realistic results that take little time to render or display. Many real-time gaming engines use these maps to simulate lighting in an environment without the processing overhead of calculating the lighting per frame.

Tip: Game engines usually display vertex color maps differently than Modo does. Therefore, test geometry in your target application to confirm the expected results.

Vertex Curvature Baking

With the Vertex Curvature Baking function, Modo generates vertex Weight Maps based on the concavity or convexity of a surface. Concave surfaces are the interior of a curve; convex surfaces are on the exterior of a curve. You could use the maps produced by this command as mask layers in the Shader Tree when used with the Weight Maps Texture layer. You may also convert the map to another type of map (such as a Vertex Color map) by using the Vertex Map Math Tool.

To view any of the resulting Weight Maps, set the viewport display option to **Vertex Map** and then select the named map from the **Lists** viewport. When generating multiple maps, be sure to enter a unique name for each instance you want to retain. Modo overwrites maps with the same name each time you apply the function.



Vertex Map: convexity
Mode: convexity
Multiplier: 100.0 %
☑ Clamping
☐ Use Edge Weights

Vertex Map: Provides a way to name the resulting Vertex Map.

Mode: Specifies the type of map generated.
Convexity- Generates a map from the outer areas of surfaces, especially curved surfaces.

Concavity- Generates a map from the inner recessed areas of surfaces.
Both- Generates a map for both convex and concave surfaces.

Multiplier: Controls the strength of the final result. If a map seem too weak, modifying this value proportionally adjusts all values in the map.

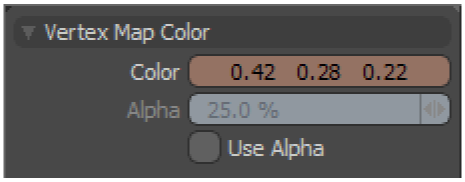
Clamp: When enabled, forces all values in the generated map to be greater than **0** to prevent negative weight values.

Use Edge Weight: When enabled, strengthens edge values.

Tip: Game engines usually display vertex color maps differently than Modo does. Therefore, test geometry in your target application to confirm the expected results.

Vertex Color Tool

By using the Vertex Color tool you can have Modo assign RGB values to a Vertex Map for Vertex Map-related shading effects. Before to editing with this tool, you must have an RGB or RGBA Vertex Map available and selected in the **Vertex Map** list. The application of the tool is done on a selection basis. If you select the **Use Alpha** checkbox, you can adjust the **Alpha** amount in the viewport. You can apply color to a map interactively by using the Airbrush Falloff falloff to directly paint color on the Vertex Map.



Vertex Map Color

Color 0.42 0.28 0.22

Alpha 25.0 %

☐ Use Alpha

Vertex Map Color

Color: Specifies an RGB color value.

Alpha: Determines the amount of transparency if you select the **Use Alpha** checkbox. A setting of **100%** indicates an opaque color; a value of **0%** creates a totally transparent 'color'.

Use Alpha: When enabled, can assign a transparency value in additional to color values.

Force Discontinuous: When enabled, blending between multiple color values no longer occurs.

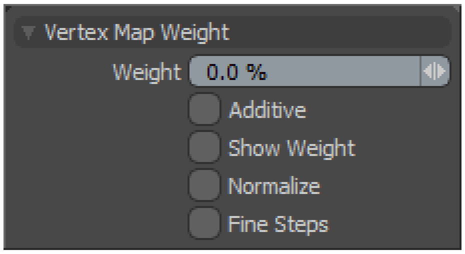
Force discontinuous disabled.

Force discontinuous enabled.

Vertex Map Weight Tool

With the Weight tool, you can adjust the **Weight** percentage for the Vertex Map to control edge weighting values. Weight Maps are discontinuous one-dimensional Vertex Maps that hold weight percentage values. You can use these Vertex Maps to modulate the strength of deformers when you apply the deformers to animated characters, to control the strength of tools when combined with a weight falloff operator, and to control edge creasing for Subdivision Surface geometry. When you set a 3D viewport to a Vertex Map shaded mode, Modo shows the weight shading (hot > medium > cold = red > green > blue) in the viewport.

To use the Weight tool, first select the appropriate map in the Vertex Map list. Modo applies adjustments only to selected vertices.



Vertex Map Weight

Weight: Displays the currently applied percentage. You can adjust this interactively by clicking and dragging in the 3D viewport with the tool active to see the effects of the value as you adjust it. You may also type a **Weight** value directly in the box and click in the viewport to apply the value to the selected vertices.

Additive: When enabled, adds the **Weight** value adjustment to those existing on the map. When disabled, adjusted values replace those previously in the map.

Show Weight: When enabled, displays a numeric readout of Weight values for each vertex.

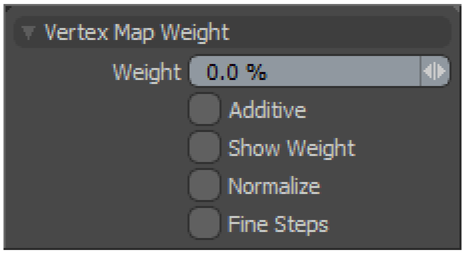
Normalize: When enabled, equalizes weights among the multiple Weight Maps assigned to joints.

Fine Steps: When enabled, controls the strength of the application of the Weight tool to allow for finer and more precise adjustments.

Tip: The Weight tool works well with all the Using Falloffs types. For example, try using the Linear falloff with the Weight tool to blend creases into the rest of the model.

Edge Weight Tool

With the Edge Weight tool, you can add creasing to subdivision surfaces without adding additional geometry. You do this by assigning a weight to various edges. To use the tool, select an edge (or edges). Then click the **Edge Weight Tool** button in the **Vertex Map** menu and drag in the 3D viewport to specify a value. Setting the value to **100%** creates a completely hard edge. With a value of **0%**, Modo uses the default interpolation of the limit surface through the control vertices. It is possible, with Modo, to set edge weights to negative values to push the limit surface away from the control vertices.



This tool is actually a special use of the generic Weight tool. Before activating this tool, first select the Subdivision Vertex Map so that Modo adjusts the appropriate Vertex Map.

Setting Edge Weight Procedurally

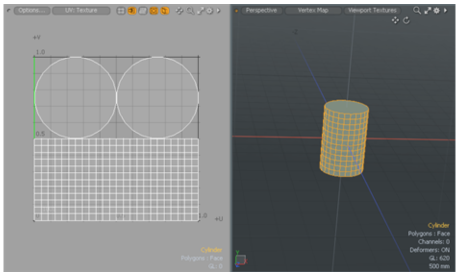
You can create a vertex map procedurally using the Set Edge Weight mesh operation.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

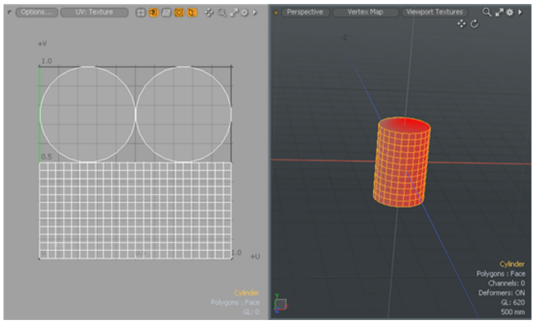
UV to Weight

Use the **Vertex Map** > **UV to Weight** tool to convert a UV map to a weight map.

For more information, see Working with Weight Maps.

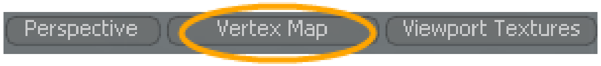


Before



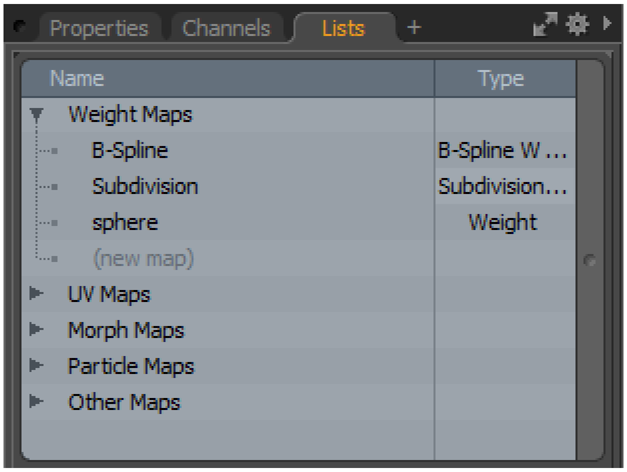
After - displaying the weight map in 3D viewport

1. Download our example scenes and open **sping-uv_weight.lxo**.
1. In the **Setup** layout, open the **Items** list on the right panel, and select the **Mesh** item.
2. In the 3D viewport, set the viewport style to **Vertex Map**.



3. On the right panel, under the **Items** list, open the **Lists** tab, and expand the **Weight Maps** item.

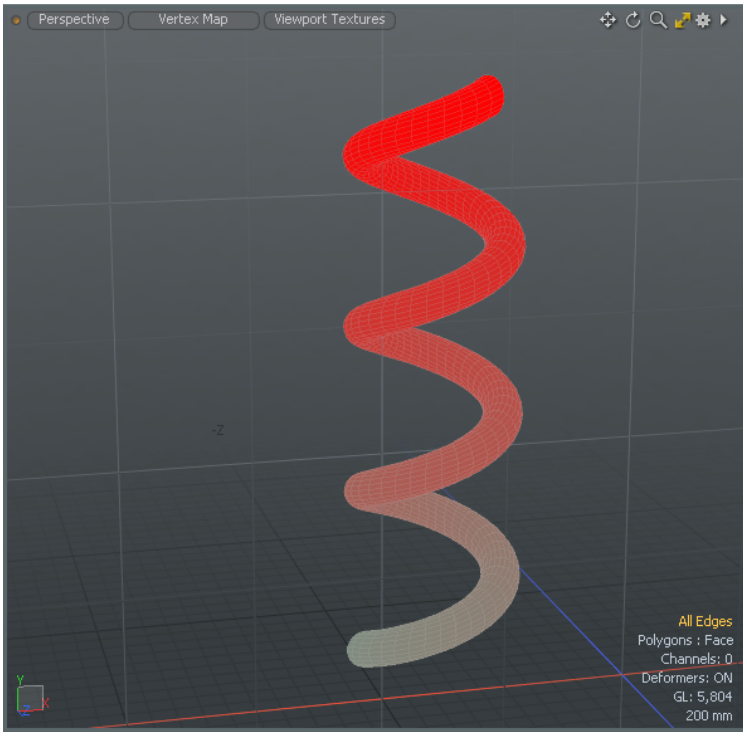
   Only the B-Spline and Subdivision weight maps are displayed.



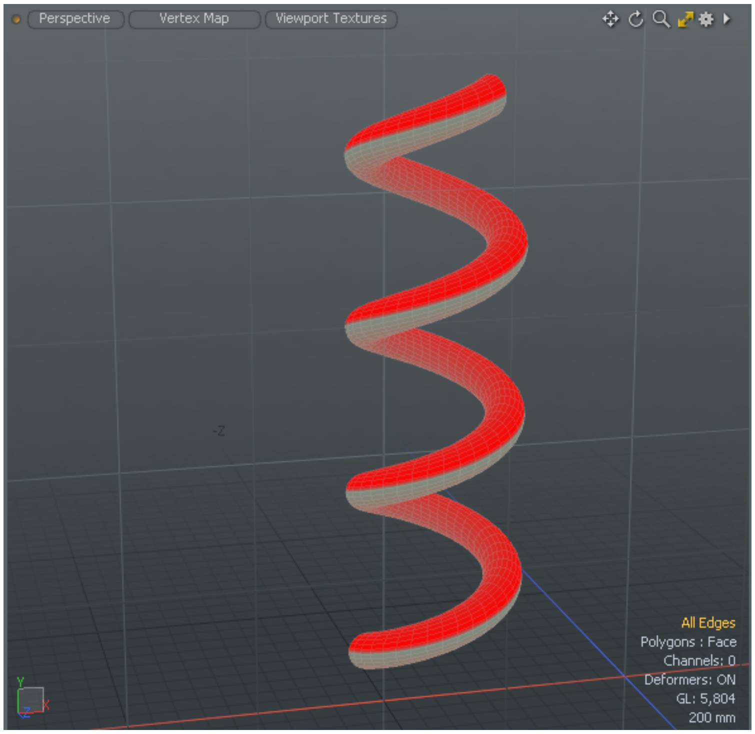
Name	Type
Weight Maps	
B-Spline	B-Spline W ...
Subdivision	Subdivision...
sphere	Weight
(new map)	
UV Maps	
Morph Maps	
Particle Maps	
Other Maps	

4. In the **Lists** tab, expand **UV Maps**, and select **Texture**.
5. On the menu bar, click Vertex Map > **UV to Weight** > **UV U to Weight**.

   The spring model displays red markings for the applied weight map along the U direction. In the **Lists** tab, under **Weight Maps**, a new **Texture2Weight** item is listed.


Perspective
Vertex Map
Viewport Textures
-Z
All Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 5,804
200 mm

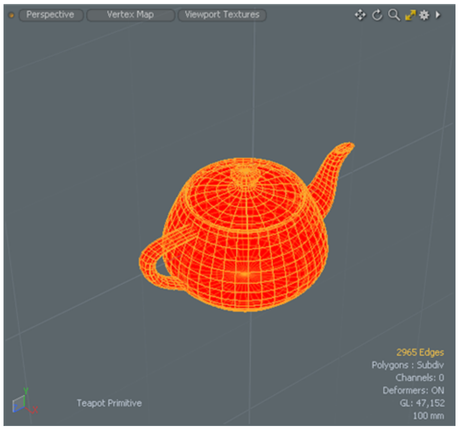
6. On the **Lists** tab, under **Weight Maps**, right-click **Texture2Weight** and select **Delete**.
7. On the menu bar, click **UV to Weight** > **UV V to Weight**.


Perspective
Vertex Map
Viewport Textures
-Z
All Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 5,804
200 mm

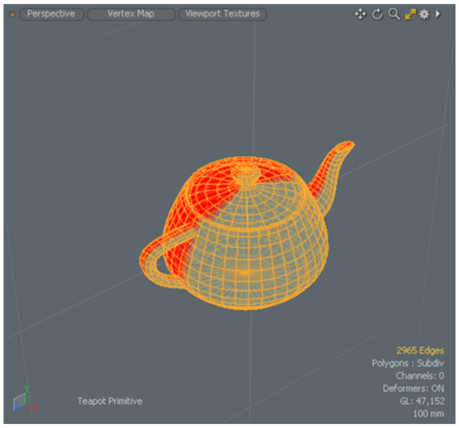
Weight Map Slicer

Use the **Vertex Map > Weight Map Slicer** tool to create new weight maps on the left and right side of the created weight map using symmetry.

For more information, see Vertex Map Weight Tool and Symmetry Tool.

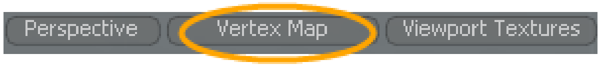

Perspective
Vertex Map
Viewport Textures
Teapot Primitive
2965 Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
100 mm

Before

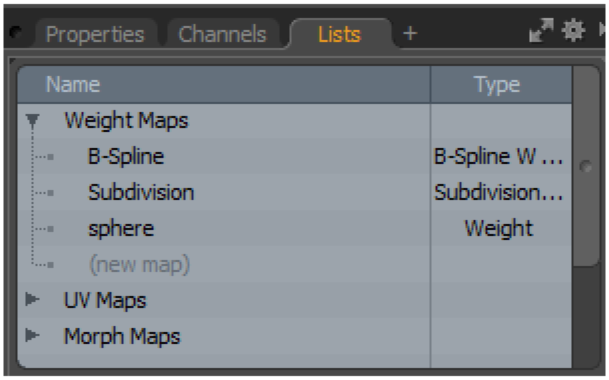

Perspective
Vertex Map
Viewport Textures
Teapot Primitive
2965 Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
100 mm

After displaying the left weight map

1. Download our example scenes and open **weight_map_slicer.lxo**.
1. In the **Model** layout, open the **Items** list on the right panel, and select the **Sphere**.
2. In the 3D viewport, set the viewport style to **Vertex Map**.


Perspective
Vertex Map
Viewport Textures

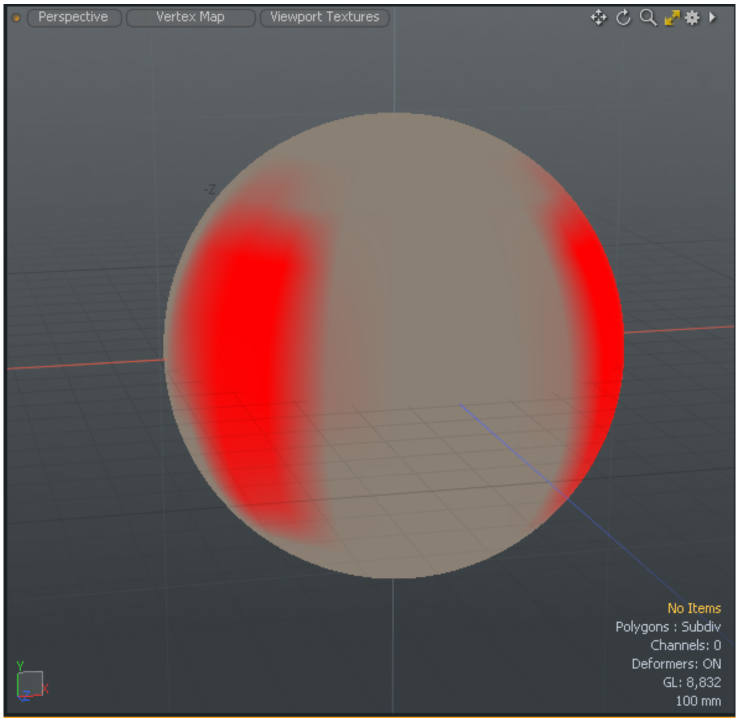
3. On the right panel, open the **Items** list, under the **Lists** tab, and expand the **Weight Maps** item.


Properties
Channels
Lists
+

Name	Type
Weight Maps	
B-Spline	B-Spline W ...
Subdivision	Subdivision...
sphere	Weight
(new map)	
UV Maps	
Morph Maps	

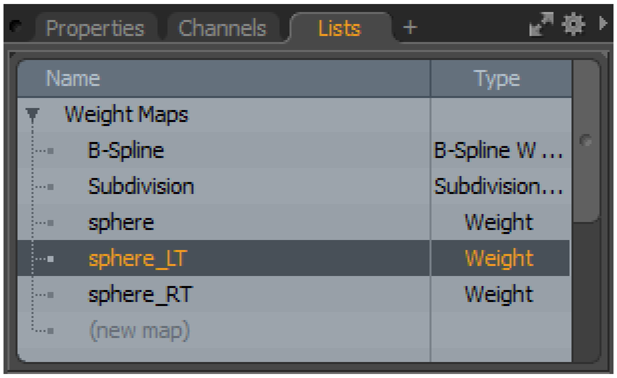
4. Select the Weight Map called **Sphere**.

   The sphere displays red marking for the applied weight map.


Perspective
Vertex Map
Viewport Textures
-Z
No Items
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

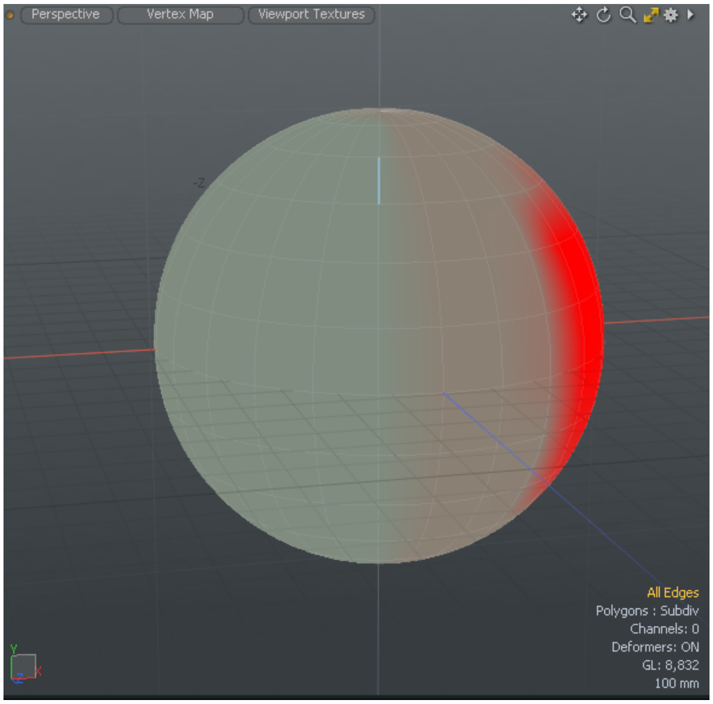
5. On the menu bar, click **Vertex Map** > **Weight Map Slicer**.

   Left and right weight maps, using the original weight map, are created.


Properties
Channels
Lists
+

Name	Type
Weight Maps	
B-Spline	B-Spline W ...
Subdivision	Subdivision...
sphere	Weight
sphere_LT	Weight
sphere_RT	Weight
(new map)	

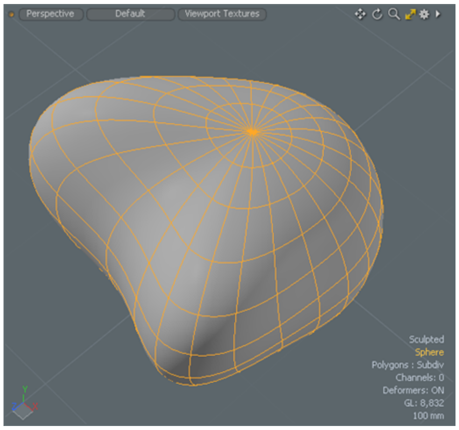
6. In the **Lists** tab, select the **sphere_RT** to display the right weight map in the 3D viewport.


Perspective
Vertex Map
Viewport Textures
-Z
All Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

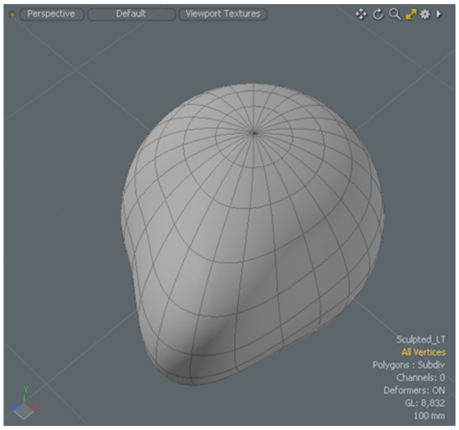
7. In the **Lists** tab, select the **sphere_LT** to display the left weight map in the 3D viewport.

Morph Slicer

Use the **Vertex Map > Morph Slicer** to copy an existing morph using symmetry to create a left and right version of it.


Perspective
Default
Viewport Textures
Sculpted
Sphere
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

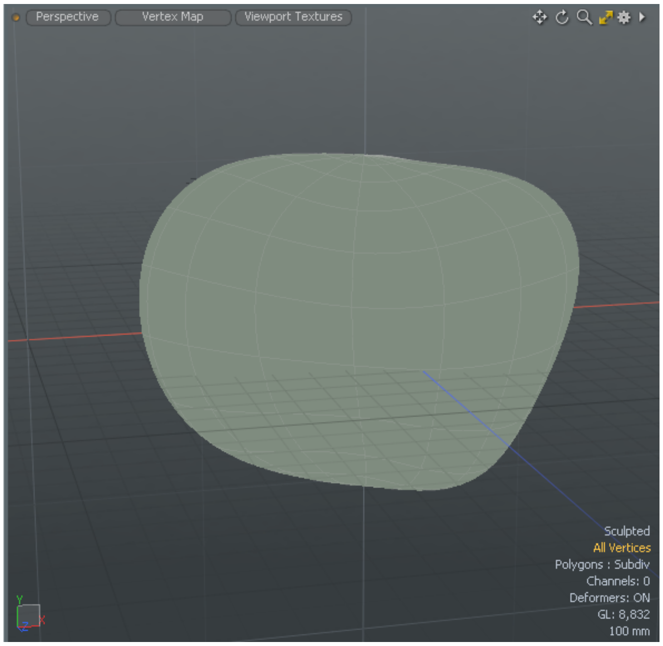
Before with a polygon selected


Perspective
Default
Viewport Textures
Sculpted_LT
All Vertices
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

After displaying the right side of the morph map

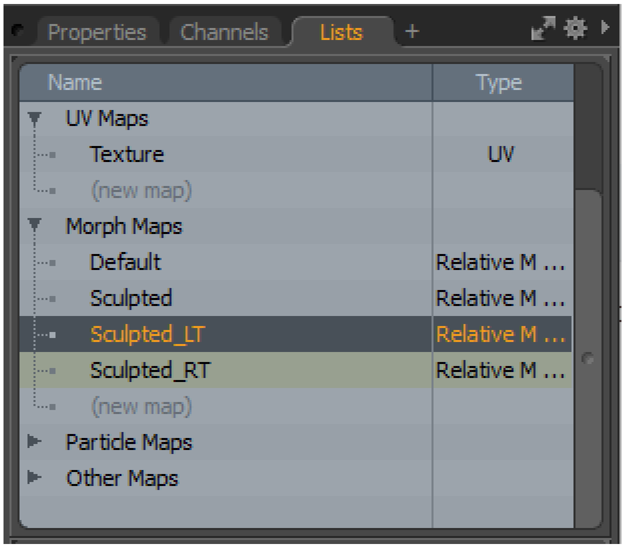
1. Download our example scenes and open **morphslicer.lxo**.
2. In the **Model** layout, open the **Items** list on the right panel, and select the **Sphere**.
3. Open the **Lists** tab, expand the **Morph Maps** item, and select **Sculpted**.

The morph map is applied to the Sphere and displayed in the 3D viewport.


Perspective
Vertex Map
Viewport Textures
Sculpted
All Vertices
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

4. On the menu bar, click **Vertex Map** > **Morph Slicer**.

   New **Sculpted_LT** and **Sculpt_RT** items are listed under Morph Maps.


Properties
Channels
Lists
+

Name	Type
UV Maps	
Texture	UV
(new map)	
Morph Maps	
Default	Relative M ...
Sculpted	Relative M ...
Sculpted_LT	Relative M ...
Sculpted_RT	Relative M ...
(new map)	
Particle Maps	
Other Maps	

5. In the **Items** list, under **Morph Maps**, select **Sculpted_LT**.

   The new morphed map is applied to the sphere and displays only on the left morph influence on the sphere.

Falloff Weight

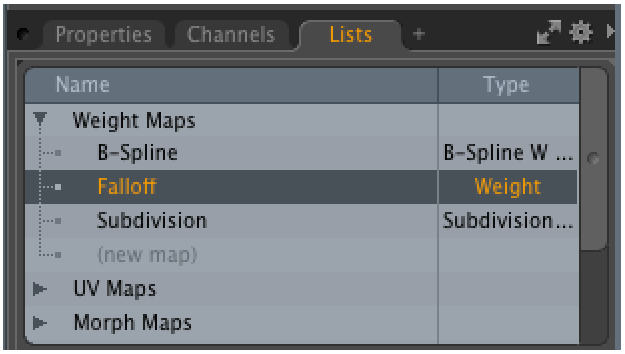
The **Falloff Weight** tool is used to create a weight map and has similar functionality as the Soft Selection Falloff tool. Using the weight map generated by the **Falloff Weight** tool is very useful for modeling, texturing, dynamics, rigging, and more. This tool provides real-time feedback to make it extremely easy to adjust the option values to your liking. Once you are happy with the values, simply drop the tool and start using the Weight Map.

Video: Watch this video for an overview.
- Falloff Weight Tool

To activate the Falloff Weight tool:

1. Open the **Model** layout and select a number of polygons, edges, or vertices on your mesh.
2. From the **Vertex Map** menu, select **Falloff Weight Tool**.

   The **Falloff** item is listed in the **Lists** tab, under **Weight Maps**.



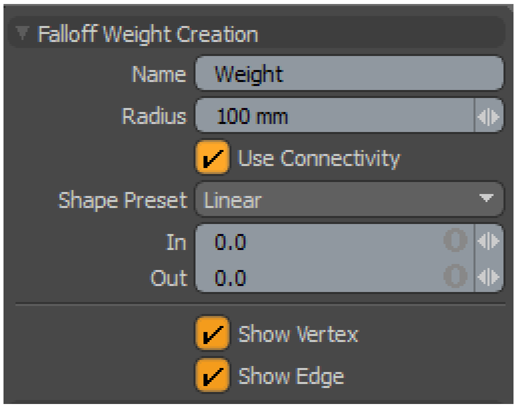
Name	Type
Weight Maps	
B-Spline	B-Spline W ...
Falloff	Weight
Subdivision	Subdivision...
(new map)	
UV Maps	
Morph Maps	

3. Left-click your mouse in the viewport to activate the tool.
4. In the Toolbox properties, in the **Name** field, type a name for your weight map.
5. In the **Lists** tab, on the right panel, expand **Weight Maps**, and select your weight map item.
6. In the viewport, left-click and drag to adjust the radius of the falloff.

   Alternatively, adjust the **Radius** value in the **Falloff Weight Creation** properties, located in the Toolbar on the right panel.
7. Continue to adjust the Falloff Weight Tool Options.

Falloff Weight Tool Options

The following options are available:



Falloff Weight Creation

Name: Weight

Radius: 100 mm

☑ Use Connectivity

Shape Preset: Linear

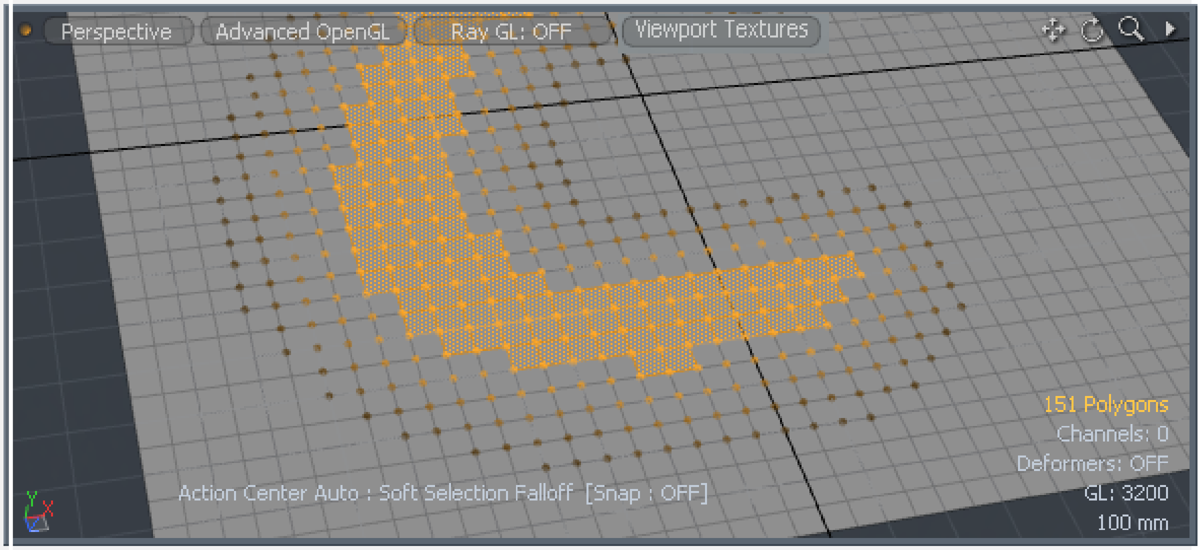
In: 0.0

Out: 0.0

☑ Show Vertex

☑ Show Edge

Option	**Description**
Name	Specifies the name of the weight map to create.
Radius	Determines the range outside of the selected edge to attenuate the falloff.
Use Connectivity	When enabled, only affects single surface/connected elements. Unconnected elements within range are ignored.
Shape Preset	The strength of the falloff's influence can be controlled along the extent using a shape preset. • **Linear** - Attenuation of the falloff occurs evenly across its range. • **Ease-In** - Strength of the falloff is greater toward the **Start** position. • **Ease-Out** - Strength of falloff is greater toward the **End** position. • **Smooth** - Strength of falloff is greater toward the center of the falloff. • **Custom** - You can use the **In**/**Out** options to fine-tune the strength of the falloff.
In/Out	The **In** value determines the strength of the falloff nearer to the **Start** position, and the **Out** value determines its strength nearer the **End** of the falloff.
Show Vertex/Show Edge	You can use these toggles to enable/disable the display of geometry highlights for both vertices and edges, giving a visual reference to the application of the Soft Selection falloff (demonstration of the vertices display mode below).


Perspective
Advanced OpenGL
Ray GL: OFF
Viewport Textures
151 Polygons
Channels: 0
Deformers: OFF
GL: 3200
100 mm
Action Center Auto : Soft Selection Falloff [Snap : OFF]

Convert Falloff Cache

The **Convert Falloff Cache** tool is used to convert internal falloff weights generated in Element Falloff, Airbrush Falloff, Screen Falloff, and Soft Selection Falloff into regular weight map. The internal cache stores falloff weight values for the last operation with these tools.

Video: Watch this video for an overview.
- Convert Falloff Cache

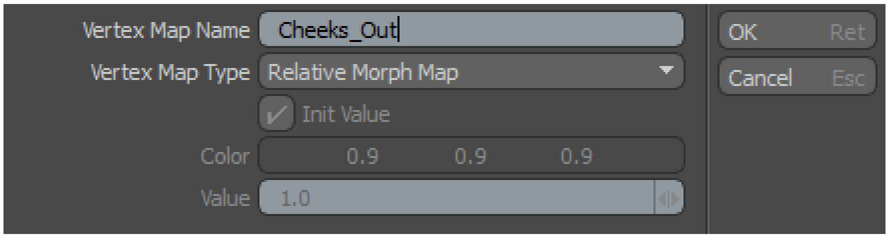
Note: The **Convert Falloff Cache** is enabled only when one of the Using Falloffs tools is used for the active meshes.

The example below demonstrates how to use this tool to create a weight map that you can later use to add color to specific areas of a mesh. We have previously created a Mesh, and a number of Weight Maps. For more information, see Create New Vertex Map.

To activate the Convert Falloff Cache tool:

1. Click **Vertex Map** > **Create**.

   The Create New Vertex Map dialog displays.



Vertex Map Name Cheeks_Out

Vertex Map Type Relative Morph Map

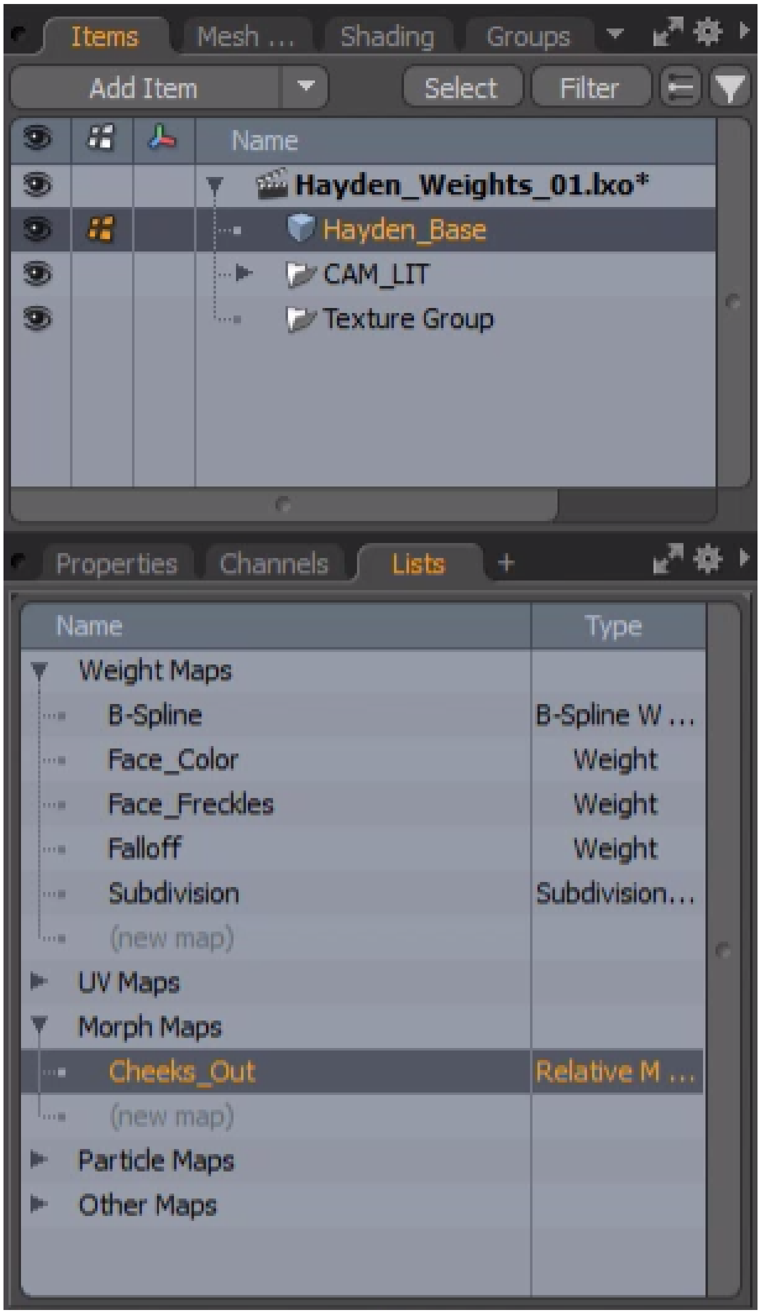
☑ Init Value

Color 0.9 0.9 0.9

Value 1.0

2. In the **Vertex Map Name** field, type **Cheeks_Out**.
3. In the **Vertex Map Type** field, select **Relative Morph Map** and click **OK**.
4. On the Toolbox, use the **Move** tool to adjust the mesh to make the cheeks puff out a little.

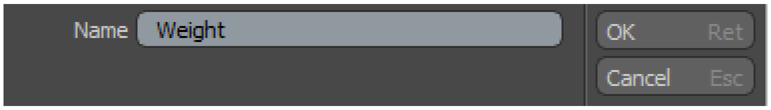
   The element Move tool uses Element Falloff, which stores internal weight data.
5. On the right panel, with the **Items** tab open, click the **Lists** tab, expand **Morph Maps**, and select the Morph Map you created earlier.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
Hayden_Weights_01.lxo*
Hayden_Base
CAM_LIT
Texture Group
Properties
Channels
Lists
+

Name	Type
Weight Maps	
B-Spline	B-Spline W ...
Face_Color	Weight
Face_Freckles	Weight
Falloff	Weight
Subdivision	Subdivision...
(new map)	
UV Maps	
Morph Maps	
Cheeks_Out	Relative M ...
(new map)	
Particle Maps	
Other Maps	

6. Click **Vertex Map** > **Convert Falloff Cache** to transfer the internal data to a new Weight Map.

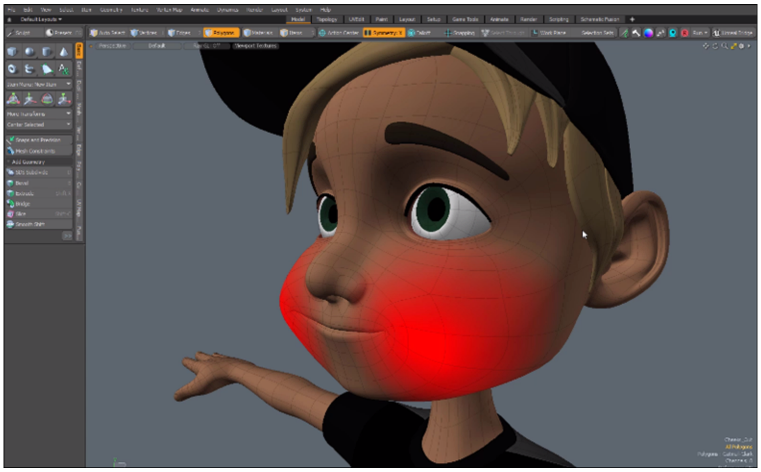
   The **Convert Falloff Cache** dialog displays.


Name: Weight

OK Ret
Cancel Esc

7. Type a name into the **Name** field and click **OK**.

   You now have a Weight Map you can use as a mask to add color to the areas of the mesh affected by the Morph. For example, you can synchronize the colors layers opacity to the morph **Strength** value to have the color rush into the cheeks of the character mesh as he puffs out his cheeks. For more information, see Morph Influence.

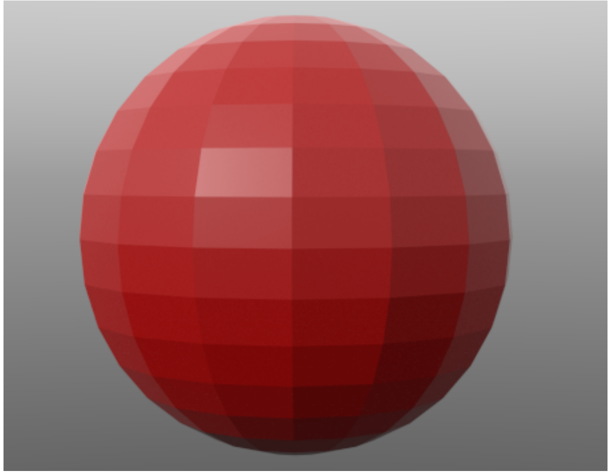


Set Vertex Normals

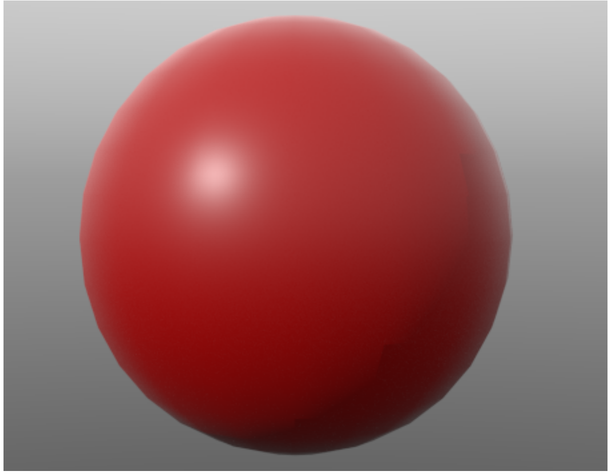
Vertex normals provide an efficient way to smooth shade a polygon mesh, giving it the appearance of a much higher poly mesh. Vertex normal smoothing provides much more control over edges hardness than material smoothing.

Smoothing and Vertex Normals

Smoothing graduates the shading across polygons to produce a surface that appears smooth, without adding any additional geometry.



No Smoothing - the surface is faceted.



Smoothing.

Modo provides a number of ways to smooth an object so that the surface does not appear faceted. For example:

- The mesh can be smoothed in the 3D viewport through Subdivide (press **Tab** or **Shift**+**Tab**).
- The surface's material item has a Smoothing setting.

Once smoothed, you can freeze the smoothing by baking out the current values into a **Vertex Normal Map** (see Vertex Normal Maps). This captures the smoothing that is displayed in the 3D viewport. The map then preserves the shading when used with a lower poly mesh.

Note: Unlike displacement maps, vertex normal maps do not modify the geometry, and the contours of the mesh are unaffected. The map gives the impression that the surface is smooth, but the silhouette is shaped according to the underlying geometry.

Note: A vertex normal map overrides any smoothness settings of the material.

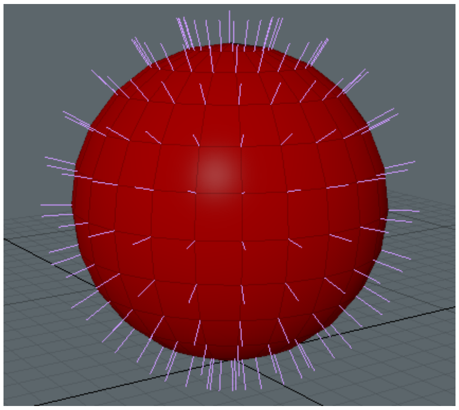
Hard and Soft Edges

A vertex normal is calculated by averaging the surface normals of the faces that surround the vertex. The orientation of the vertex normals provides a means to determine the properties of reflected light (and therefore the shading) at each vertex. By interpolating between the vertex normals, Modo can smooth shade across the surface.

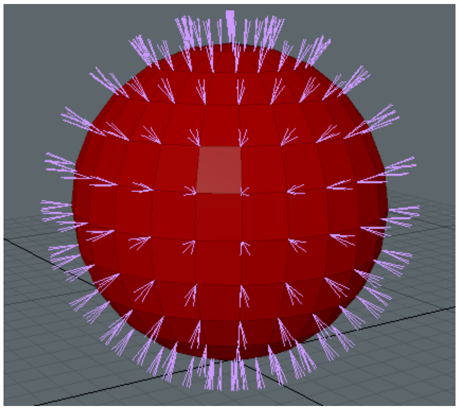
A smooth interpolation across polygon edges creates the appearance of soft edges, with no sharp distinction between adjacent polygons. Alternatively, the vertex normals can be adjusted to create hard edges, with contrasting shading across edge boundaries.

The images below show the same mesh with soft edges and hard edges,

Tip: In Modo, you can display the normals at each vertex by selecting **View** > **Show Vertex Normals**.



A smooth shaded sphere showing vertex normals. A single normal at each vertex indicates smooth shading between the edges.



A sphere with hard edges, resulting in a faceted surface. Multiple vertex normals are drawn at each vertex to indicate the difference in shading angles between adjacent polygons.

Creating a Vertex Normal Map

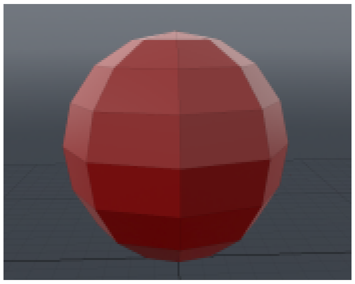
In Modo, there are a few ways to create a **Vertex Normal Map**:

- Create a new, empty Vertex Normal Map using **(new map)** in the **Vertex Maps** viewport. You can add to this map later.
- Bake the current shading to a **Vertex Normal Map** using **Vertex Map** > **Set Vertex Normals**. See Setting Vertex Normals
- Use the Vertex Normal Tools to create or edit a **Vertex Normal Map** with hard or soft edges as required.

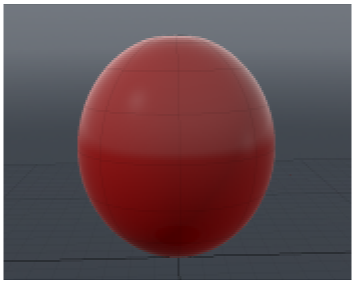
You can edit the values in the fixed **Vertex Normal Map** to vary the shading settings. If you generate a vertex normal map, it saves the map with the model, and you can export it in various formats.

Setting Vertex Normals

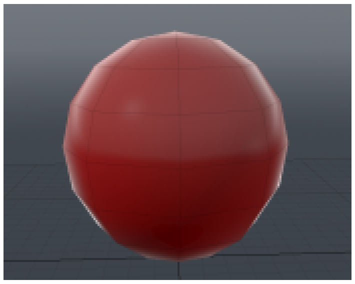
You can generate a **Vertex Normal Map** based on the current selection. The smoothing is then baked into a **Vertex Normal Map** and is automatically applied when you reduce the smoothing. The images below demonstrate this process using subdivision.



A low poly sphere with no smoothing.



The same sphere subdivided by pressing **Tab**. **Set Vertex Normals** is then selected to to bake the shading into a **Vertex Normal Map**.



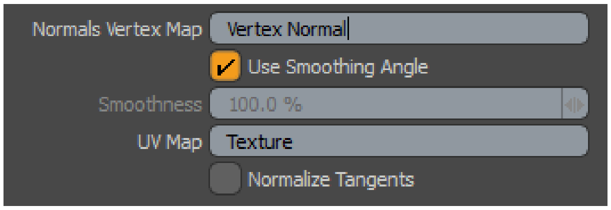
The low poly version, viewed by pressing **Tab**. Notice how the **Vertex Normal Map** is now smooth shading the surface.

Tip: You can apply Vertex Normal Maps to sections of a mesh by selecting the required polygons before generating vertex normals.

To Set Vertex Normals

1. Open your geometry in the **3D viewport**, with the preferred smoothing method applied.
2. Select **Vertex Map** > **Set Vertex Normals**.

   This opens the Normals Vertex Map dialog.



Normals Vertex Map: Vertex Normal

☑ Use Smoothing Angle

Smoothness: 100.0 %

UV Map: Texture

☐ Normalize Tangents

Adjust the parameters:

- **Normals Vertex Map** - Enter a name for the Vertex Map. This is the name used in the **Lists** viewport.
- **Use Smoothing Angle** - Enable this to use the smoothing values from the Material item's **Smoothing** and **Smoothing Angle** settings.

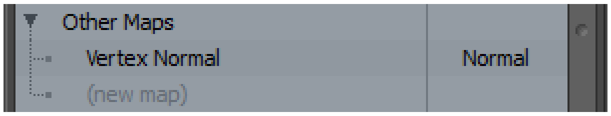
If disabled, you can specify a **Smoothness** value that Modo assigns to all vertices in the target geometry.

- **Smoothness** - Specifies a percentage for smoothing the Vertex Map. This option is available if you clear the **Use Smoothing Angle** checkbox. At **100%**, Modo fully smooths all vertices.

  Once applied, you can then select specific edges and apply the **Split Normals** command to add creases to specific areas of a surface.
- **UV Map** - Modo stores its own tangent basis in a vertex map that it uses behind the scenes. This tangent basis is derived from the normals of the mesh and the UV coordinates. This option lets you choose the **UV Map** that is used to derive the tangent basis.
- **Normalize Tangents** - Check this to normalize the tangent vectors. For normal maps, normalized vectors are usually preferred. For displacement maps, the vectors are not required to be of unit length.

3. Click **OK**.

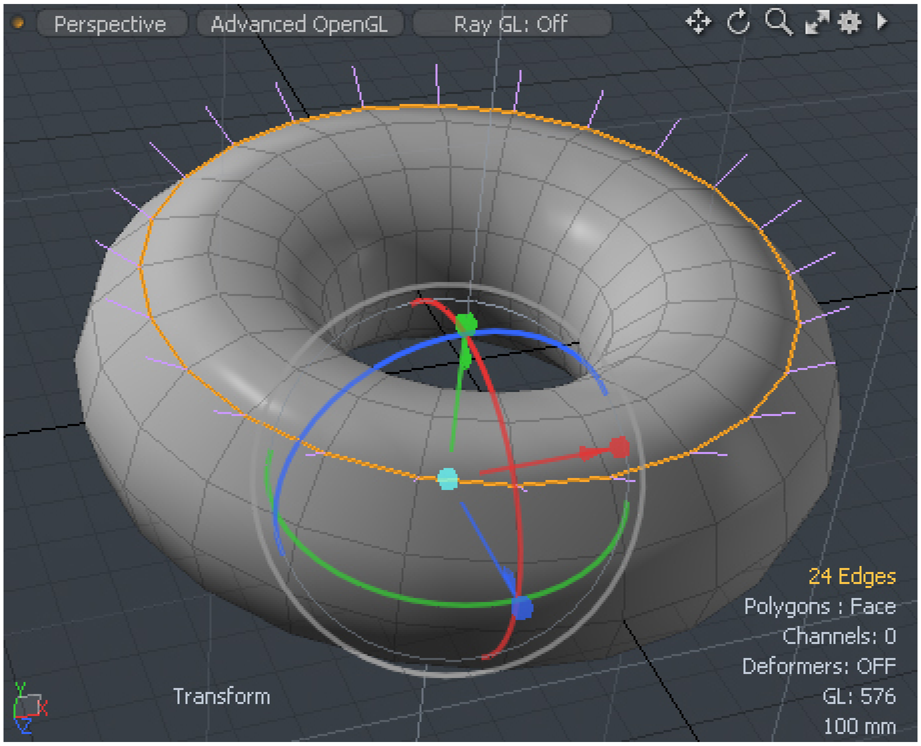
   You can see the new Vertex Normal Map in the **Other Maps** section of the **Lists** viewport.



4. You can now switch to a low poly version or reduce the material smoothing. The Vertex Normal Map is used to shade the selection.

Warning: When Modo associates a vertex normal map with a Mesh Item, modifying the smoothing value for that surface no longer affects the actual smoothing of the surface in the viewports because vertex normal maps supersede any other settings. To modify the values, remove the map and generate a new one with the new values or apply the **Split Vertex Normals** command with an appropriate edge selection.

Transform Vertex Normals

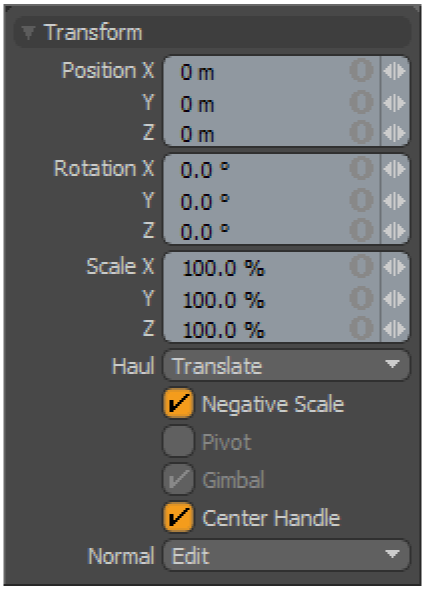

Perspective
Advanced OpenGL
Ray GL: Off
24 Edges
Polygons : Face
Channels: 0
Deformers: OFF
GL: 576
100 mm
Transform

With the Transform Normals tool you manipulate vertex normal vectors within a Vertex Map. This tool is a special preset version of the standard Transform tool that specifically transforms Vertex Map values instead of vertex positions. The **Normal** list provides options specific to working with vertex normal maps. **Edit** transforms only normal vectors in the selected vertex normal map. **Update** works for both vertex positions and normal vectors in all associated vertex normal maps — including unselected maps.

Usage

You can select the target vertex normals by using any of the component selection modes: vertices, edges, or polygons. Modo transforms all the vertex normal map values for the associated vertices. Next, select the target Vertex Map to edit in the **Lists** viewport, and then choose **Vertex Map > Transform Normals**. You can then use the tool's handles in the 3D viewport to adjust the vertex normal values.

Note: You can set vertex normal vectors per vertex corner of a polygon, and Modo stores them in discontinuous Vertex Maps. The Transform Normals tool does not edit discontinuous values on locked vertices or polygons. You can edit a specific value on vertex corner of polygon by locking the neighboring polygons where they are not meant to be updated.



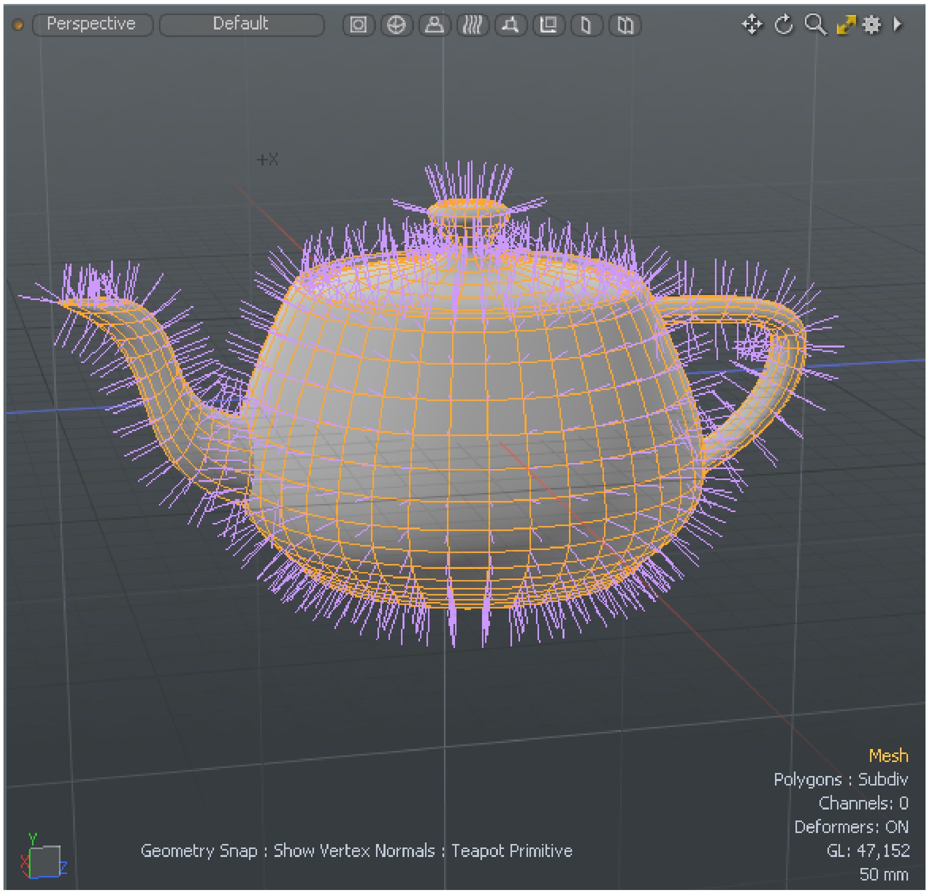
Transform

Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Haul Translate
☑ Negative Scale
☐ Pivot
☑ Gimbal
☑ Center Handle
Normal Edit

Option	Description
Transform	
Position X/Y/Z	Apply specific offset distance values calculated from the position the object was in when the tool was activated. When adjusting an object interactively in the viewport, the **Position** values indicate the current offset distance applied.
Rotation X/Y/Z	Apply specific rotation values. When adjusting an object interactively in the viewport, the **Rotation** values indicate the current offset distance applied.
Scale X/Y/Z	Apply specific scaling values. When adjusting an object interactively in the viewport, the **Scale** values indicate the current scale factor applied.
Haul	Indicates what action Modo does when you drag the pointer when not using the tool handles in the viewport. Selecting **Translate** moves the selection; selecting **Rotate** rotates the selection; and selecting **Scale**

	adjusts the size of the selection.
Negative Scale	When enabled, Modo can apply negative scale values when manipulating the tool handles in the viewport. When disabled, you can input negative values in any of the boxes directly.
Pivot	When enabled, you can modify an item's pivot position without going into Pivots mode. To adjust the **Pivot** checkbox, the **Gimbal** checkbox must be clear.
Gimbal	When enabled, you can assign rotation values numerically, but doing this may create a situation that produces gimbal lock (the loss of one rotation axis when another's value is too close). To avoid this, clear this checkbox, but then you cannot specify input values.
Center Handle	Shows or hides the center handle on the Transform tool.
Normal:	Controls how the map values are affected when working with Vertex Normal maps. **None**- Vertex Normal map values won't be affected at all by the transform operations. **Edit**- Only Vertex Normal map values are modified by the tool and only in the selected Vertex Map. **Update**- All Vertex Normal map values (selected or otherwise) and any selected geometry are modified by the transform operation.

Show Vertex Normals



Mesh
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
50 mm

Geometry Snap : Show Vertex Normals : Teapot Primitive

When you use the **Show Vertex Normals** command, Modo displays small lines that represent the vertex normal facing directions for each vertex in a Vertex Normal Map. Split lines represent creased hard edges, and single lines represent smoothed edges. You must have a Vertex Map present to see anything after using this command.

You can create a map based on the object's current smoothing by using the Set Vertex Normals command. The resulting map appears in the Vertex Map Lists viewport.

Once your map is created, you can display the normals at each vertex by clicking **View** > **Show Vertex Normals**.

Vertex Map Tools

You use the Vertex Map Tools to adjust vertex color maps and vertex Weight Maps by dragging a brush over the surface of the target map.

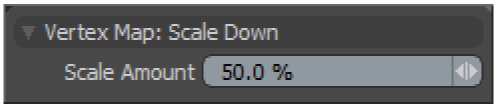
In addition to the Toolbox **Paint** tool, in the **Vertex Map Tools** sub-tab, other vertex map options are available. For more information, see Vertex Normal Tools.

Using the tools

The Vertex Map tools are found in the **Paint** layout. Open the Toolbox on the left panel, expand the **Vertex Map Tools** sub-tab, and click **Paint**. To use a Vertex Map tool, you first need to select a target Mesh Item layer in the Items (Scenes) List. You also need to create and select a vertex color map or a Weight Map in the **Vertex Map** list viewport. See the Working with Vertex Maps topic for information about creating Vertex Maps.

After selecting a tool, its attributes appear in the tool's **Properties** viewport. When you first select a tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to apply the brush's effect to the designated map. Usually, pressing **Ctrl** negates the effects of the brush.

Scale Down



Scale Amount 50.0 %

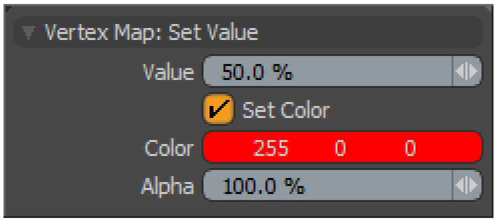
The Scale Down tool adjusts the values in a map down toward **0** (zero) — slowly reducing the values. Pressing **Ctrl** scales up the values (acting as a Scale Up tool). The overall effect is similar to dodging and burning in an image editor.

Scale Amount: Determines the strength of the scaling applied to the Vertex Map values.

Erase

The Erase tool removes any values in the target map that the brush touches. This tool doesn't just lower the values of the map to **0%**, but it completely removes the value for the vertex. There are no user-adjustable options available.

Set Value



Vertex Map: Set Value

Value 50.0 %

☑ Set Color

Color 255 0 0

Alpha 100.0 %

The Set Value tool is similar to a paint tool combined with an erase tool. It affects any vertex it touches **100%** (changes the color or the weight as specified) with no fading or attenuation. Pressing **Ctrl** erases the value.

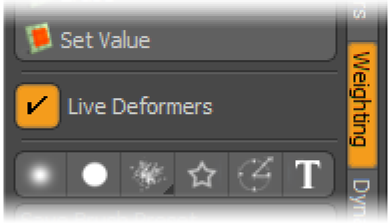
Value: Determines the weight value to apply when editing Weight Map values.

Set Color: Enabled automatically when painting onto vertex color maps for you to specify a color to apply to the map.

Color: Defines an RGB value to apply to the target RGB or RGBA vertex color map.

Alpha: Defines an Alpha (transparency) value to apply to the target RGBA vertex color map.

Live Deformers



☑ Live Deformers

Controls the way Modo updates the geometry in a 3D viewport when weight painting. When enabled, Modo updates deformations on the fly while you paint or adjust the weights. (On-the-fly updates affect performance.) When disabled, Modo updates the deformations after you release the mouse button when weight painting or making other adjustments. Having this disabled increases performance.

Vertex Map Paint

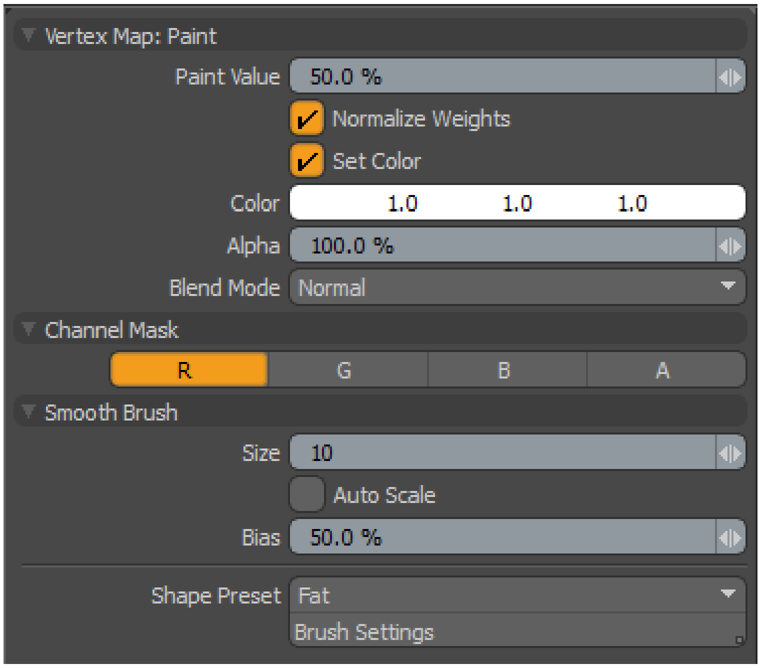
You can use the Vertex Paint tool to define and adjust vertex color maps and vertex Weight Maps interactively. By using the familiar painting workflow already established for creating bitmap-based textures, you can select the Vertex Paint tool and paint directly onto the surface in the 3D viewport with either a tablet-based input or a mouse.

Using the tool

The Vertex Paint tool is in the **Paint** interface tab on the **Vertex Map Tools** sub-tab of the toolbox. To use the Vertex Paint tool, you first need to select a target Mesh Item layer in the Items (Scenes) List. You also need to create and select a vertex color map or a Weight Map in the **Vertex Map** list viewport. See the Working with Vertex Maps topic for information about creating Vertex Maps.

After selecting the tool, its attributes appear in its **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to apply the specified values to the designated map. Pressing **Ctrl** reduces the values applied to Weight Maps (instead of increasing them). Pressing **Shift** applies the Smooth Vertex Map function.

Paintbrush


Vertex Map: Paint
Paint Value 50.0 %
☑ Normalize Weights
☑ Set Color
Color 1.0 1.0 1.0
Alpha 100.0 %
Blend Mode Normal
Channel Mask
R G B A
Smooth Brush
Size 10
☐ Auto Scale
Bias 50.0 %
Shape Preset Fat
Brush Settings

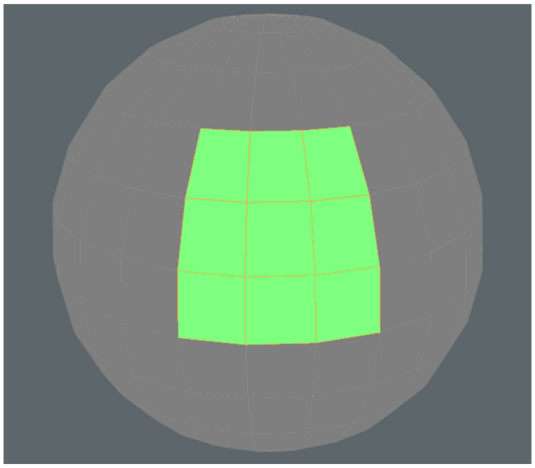
Option	Description
Paint Value	Determines the strength of the painting action. Higher values saturate the canvas more quickly than lower values. Set this to **100%** to quickly overlay an area, or set to **5%** or less to make subtle adjustments.
Normalize Weights	In the case of Weight Maps, this option ensures that the values of all Weight Maps add up to exactly 100% at each vertex. So if you paint a Weight Map at full strength (100%), and paint another one at half-strength (50%), this option reduces the first map to 50%, so the total of the two is 100%. This option is useful for rigged meshes. For example, if you have two bones trying to move a vertex two different ways, normalizing weights makes sure that the total control each bone has isn't more than what it should safely have.
Set Color	Enabled automatically when applying values to a vertex color map. When enabled, you can define a **Color** value and apply it to the target map.
Color	Defines an RGB value to apply to the target vertex color map.
Alpha	Determines the opacity of the color applied to the RGBA vertex color map. **Tip:** Modo stores the Vertex Map values for the vertices themselves of the mesh's surface. Therefore, it is not possible to resolve any finer detail than the resolution of the target geometry (un-subdivided). If you don't want to subdivide the mesh, then using a bitmap-based texture may be a better option to assign colors to the surface.
Blend Mode	This specifies how multiple layers interact with each other. The standard blend modes are available for vertex color painting. For more information on blend modes, see Layer Blend Modes.
Channel Masks	
R/G/B/A	Vertex color painting can now be masked per channel. To mask a channel, click on the appropriate button. The button highlights orange and the corresponding channel doesn't get affected when painting.
Smooth Brush	
Size	This value sets the radius of the brush, in pixels, from screen space. This is

Option	Description
	important to understand, as zooming the view changes the result of the brush. With the brush size being fixed, zooming in on the model decreases the relative scale of the brush, whereas zooming out increases it. You can interactively adjust the tip **Size** value by right-clicking in the viewport and dragging to adjust an on-screen display representing the brush's size.
Auto Scale	When enabled, the brush **Size** scales automatically to remain constant, regardless of the zoom percentage.
Bias	This option is like a curve adjustment to the brush that favors one side of the range over the other, acting much like a gamma curve would affect the shades of an image.
Shape Preset	This dropdown includes several preset brush options for varying effects. This setting drives the profile of the airbrush. You can also adjust the **In**/**Out Weight** and **Slope** settings to produce customized brushes.

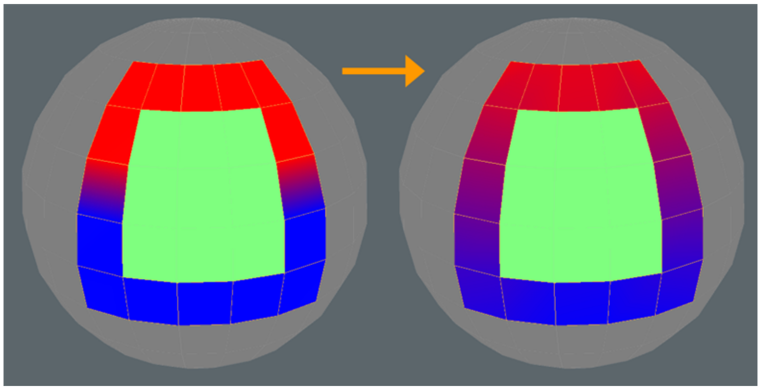
Polygon Masks

Vertex color painting and smoothing is now masked to the selected, unlocked, and visible polygons, so painting discontinuous values is easier.

Vertex color painting masked to polygon selection:



Vertex color blending (smoothing) masked to polygon selection:



Working with UV Maps

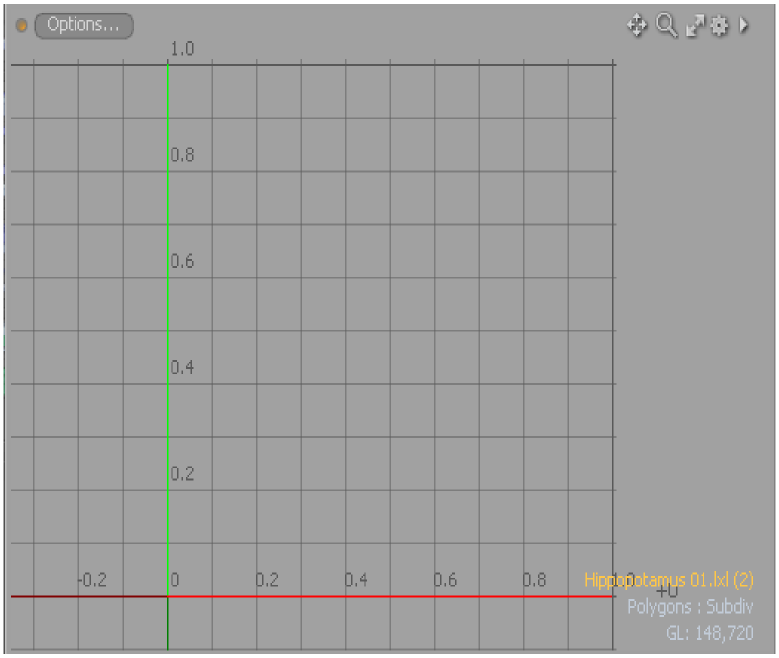
UV mapping is the process of translating a 3D surface with volume and shape onto a flat 2D texture image. A way to visualize how that works is to consider an object that was wrapped as gift. A UV map is similar to the careful unwrapping of the object and pressing the wrapping paper flat (much like a map of the earth that has the same constraints as a UV map).

The actual execution of the UV mapping process requires some science and much art, but, like modeling, texturing, or even rendering, understanding the tools is an important part of understanding the process. Modo provides a range of functions that simplify the UV mapping process to make a possibly tedious task much easier.

Video: Watch this video on UVing.

Understanding UVs

UV map values are position coordinates on a 2D texture that are stored within an associated vertex. In Modo, these coordinates are floating-point values (numbers with decimal points) for the extra precision that floating-point numbers provide. The **UV Map** viewport plots the 2D position values on a grid and includes the edges and polygons so it's easier to see how the 2D geometry map relates to its 3D counterpart. Modo typically confines these numbers to the 0-1 area of the grid as the normal space (or home) position of the texture. Any texture applied using a UV map always fills the entire 0-1 space. Any part outside of this area simply repeats the texture. (You control how the map gets tiled with the Texture Locator.) By confining the map values within this single quadrant, you can control positioning and avoid any unnecessary overlapping because overlapping can cause errors when baking textures. By scaling UV values larger than the 0-1 space, you can precisely control the tiling of the image on the surface. Modo has no specific constraints regarding the position of geometry within UV space.



A possible point of confusion is the shape of the UV editing viewport and how it relates to the image's aspect ratio. The UV window is square, so you may assume that UV-mapped images need to be square as well. This is not the case. Images can be any size, any aspect ratio, and any resolution. Modo still always maps the image to the 0-1 space of the UV grid. For a very wide or very tall image, Modo positions the values proportionately across the texture. This is why you may want to think of the values as percentages. For example, a floating-point value of .5 is effectively a 50% position, and Modo positions that vertex in the center of the image — regardless of the image's size or shape. Doing that makes UV mapping precise and valuable. When displaying non-square images in the UV viewport, Modo distorts the image to fill the 0-1 space. You can also enable the **Use Aspect Ratio** in the viewport's options to distort the UV space view to display the image's aspect ratio properly.

At the bottom right corner of the UV viewport, Modo presents coverage information which represents the amount of uniform UV space of the total region that is covered by geometry. This information is useful when trying to maximize the amount of texture space being used by the target surface. For more information, see Reviewing UV Coverage.

Making UV maps

For many instances, the texture projection modes of the Texture Locator provide the needed control for applying texture layers to a surface. UV maps are only necessary for surfaces that require high-precision texture mapping (such as for modeling organic characters, applying a logo to a bottle in a specific way, or working with a model that has a peculiar shape). UV maps are also required with any real-time game engine and any object-to-object baking. If you don't plan to apply a texture layer to a surface, then a UV map is unnecessary.

You generate and edit maps in the UV editing tab of the main (default) interface. The dedicated UV editing viewport has quick access to frequently used tools and commands. When working in the **UV** viewport, almost all the same tools, modes, and functions of the 3D Viewports apply. You can use transform tools, modeling tools, and even sculpting tools to edit UV maps. Most tools and functions are the same, but some tools behave differently. For example, regardless of the geometric topology, the **Select Connected** function only selects the connected area of a UV island when applied in the **UV** viewport. (A UV island is a connected group of polygons in a UV map.)

To make UV maps, first select or define a Vertex Map to store values into. UV maps are just another type of Vertex Map that you manage within the **Lists** viewport. For information, see the Working with Vertex Maps topic. Once you specify the proper map type in the **Lists** viewport, you can select the polygons to map and invoke any tools or commands. For interactive tools, such as **Unwrap**, you can adjust the values directly in the 3D viewport and see real-time updates of the UV map in the **UV** viewport.

The tool or process you use to produce the map depends on the shape of the surface. With geometric shapes (such as a can or a box) use the Projection tool, which generates UV values with options similar to those found in the Texture Locator (such as **Spherical** and **Planar**). For organic shapes, use the UV Unwrap tool to produce good results immediately. Long, winding objects (such as a belt) can leverage the UV Peeler tool's ability to produce precise maps with uniform spacing. Often, the result produced by a given tool is adequate, and you can move to other tasks. At other times, the result provides a starting point for further refinement. Depending on your model, the default map generated by Modo's primitive geometry may not be appropriate or optimum for your intended use.

You can view a basic example of creating a UV map for a can in the video. As seen, the layout of the map, itself, is largely dependent on how the geometry is used. In general, set things up to make surfacing easy. There are no set rules to follow for layouts. You can position UV islands (grouped areas of geometry in a UV map) anywhere in the UV map. There are some guidelines. These guidelines are not hard and fast rules, but they can be helpful when creating a UV map.

Minimize Seams

Seams are the edges (or breaks) in geometry between UV islands and are inevitable for most models. Sometimes seams can cause problems with painting or image-based sculpting. If at all possible, define seams in inconspicuous places. For example, with human characters, this may be the inseam of the legs and thighs, under the arms, behind the head, and down the back. When viewed from the front, most of these seams would be hidden from view. When your model has seams that can't be hidden from view, use image commands like Expand UV Border to minimize the seam's appearance. Also, having as few islands as necessary reduces the number seams.

Managing Distortions

Two goals when UV mapping are: eliminate (or reduce) seams and minimize distortion from stretching. Unfortunately, these two are not mutually exclusive. Distortion is a fact of life with UV mapping. Completely eliminating distortion would require an increase in UV seams. Fortunately, Modo has several tools and functions to help you manage distortion. First, select the **Show Distortions** checkbox for the **UV** viewport. When enabled, Modo shows a color overlay on the geometry that indicates a polygon's distortion (for example, a scaled or stretched polygon) in relation to neighboring polygons. Polygons that are the closest in relative size to others in the map with the least distortion display a middle gray/green color. Distorted and smaller relative polygons fade toward red; distorted and larger relative polygons display as blue. You can also see the distortion colors in the 3D viewport by selecting the **Vertex Map** viewport style and then selecting the **UV Distortion**-named Vertex Map generated under the **Other Maps** heading of the Vertex Map List viewport. Once you can see where the problem areas are, you can take action to minimize issues. The UV Relax tool is the best tool for reducing distortions. Use it to massage a map by relaxing the data to tug and pull neighboring polygons until you can achieve a mid-range situation. You can limit the application of the relax using Using Falloffs as well.

Tip: Use the **Distortion Compensation** option when you create a primitive that does not have equal radius values. The aspect ratio and related area size of UV polygons conforms to the 3D polygons. By default, the UVs that are generated for the primitive are distorted and displayed in red and blue in the UV viewport of the **UV** layout. To view distortion, in the UV viewport, click **Options...** and then enable **Show Distortion**.

Enabling the **Distortion Compensation** option reducing the distortion automatically and produced cleaner UVs. This feature also works in real-time as you edit the attributes of the primitive before committing to the final values.

This option applies to the following primitives: Cube, Cylinder, Capsule, Sphere, Torus, and Ellipsoid.

Consistent Texture Density

For the most part, models look their best when an image map's pixel density is spread evenly across the surface. For example, with a sphere, an image map would need to be twice as wide as it is high for an even pixel density because the circumference is twice the distance of how far the image wraps from top to bottom. Also, you would want to have the UV islands as close as possible to get the most out of the number of pixels in the image. Large gaps between islands waste texture space. You may want to map a temporary grid or checkerboard pattern onto the object to get direct visual feedback of how consistent the texture's density is. You could use the **UV Check** image in the **Materials > Miscellaneous** folder (from the Modo content download) for this purpose.

Take the Image Map into Account

When creating a UV map, it's important to consider the texture map(s) that you plan to apply to the surface. What is a texture map's size and aspect ratio? For example, if you plan to create a UV map for a rectangular sign that is much wider than it is high, it is best to make the final image the proper size and aspect ratio. This is especially true if you might need to edit it later. In this case, the geometry appears squashed in the square **UV** viewport, but the image renders exactly as expected.

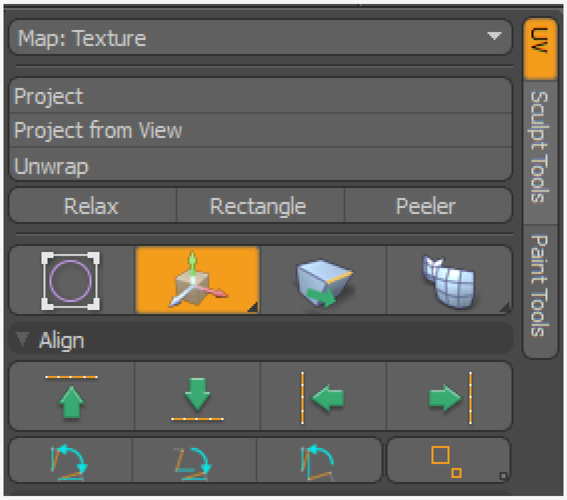
Is Orientation Important?

In most cases, the orientation of individual islands is arbitrary, but in some cases, it can be important. For example, when texturing a face directly in 3D, the paint tools take care of projecting the pixels into the image properly. In this case, the UV's position within the map is inconsequential. But, if you have to make edits in another program (such as Photoshop), it's easier to edit when the face is straight up and down instead of at an angle.

Overlapping UVs

Overlapping UVs aren't always bad. For example, when each overlapping part has a different material tag, overlaps may be okay because this simplifies the model by reducing the total number of Vertex Maps. Barycentric maps (where Modo maps each individual polygon to the entire 0-1 UV space) are useful to control image tiling. While texture memory in games continues to go up, it was a good idea to model and texture half of a character and then mirror it (making UV overlaps) effectively doubling the texture resolution, while halving the image size. When baking textures out, overlapping UVs cause errors. In those cases, avoid overlaps.

Tip: If you want to move polygons, which are overlapping, use the **Transform** tool.



First select a number of polygons you want to move in the **UV** viewport and then click the **Move selected along any axis** button in the Toolbar on the left panel. The **Transform** tool properties are listed in the Toolbar. Once activated, handles display in the **UV** viewport. You can either move the selected polygons or enable the **Tear Off** option to separate and move your selection of polygons to a new position. Holding down the **Ctrl/Cmd** + **Shift** keys, while dragging, temporarily enables the **Tear Off** mode.

Get to know the tools

Modo has an extensive toolbox — especially for working with Vertex Maps. If you aren't familiar with what Modo has to offer, that might limit what you can do. You may want to read through each tool's explanation from the UV Mapping topic.

Applying the Map

You use UV maps mainly when applying textures, so once you have created the map, it shows up in the **Vertex Maps** window under the **UV Maps** section. To actually use the map, you first apply a texture layer, either as an image map or a procedural texture. Once applied, you can access the associated Texture Locator from the **Item List** tab, or the Shader Tree. (You do this by clicking the **+** icon next to the layer's name.) Within the locator's properties, set **UV Map** as the **Projection Type**. Then you may select the specific map. (Modo tries to automatically select the proper map when you apply an image map to any geometry that has a UV, but you may have to specify an alternate map if Modo selects the wrong one.)

UV maps are also useful to apply anisotropic shading (which can be further controlled by an image map). When you apply a UV map, the **U** (horizontal direction) defines the length that the anisotropic shading follows. You can specify the needed map whenever the **Anisotropy** setting of the material item is above **0%**.

Lastly, you can use UV maps to simplify certain modeling tasks. Often it may be easier to apply a UV map to geometry and act as a visual aid. You can then convert the map to a Vertex Map of another type (such as a Weight Map) and then apply it as a falloff. For example, use the UV Peeler tool on a spiral. When you do, Modo creates a linear-type UV map that follows the contours of the form. You can then convert this to a Weight Map with the Vertex Map Math Tool. Finally, apply the resulting weight as a falloff to the Push tool to scale of the spiral along its length.

Tip: Discontinuous vertices in a UV map are those that are single vertices within the model, but that have two locations in a UV map. (They originate from shared edges that are split into separate UV islands.) Modo indicates discontinuous vertices and edges in the **UV** viewport with a light lavender color to make it easier for you to locate the associated vertex or edge.

UV Mapping

UV mapping is an essential part of precision texture mapping. Also, it can be useful when applying procedural texture and gradient layers as well as anisotropic shading. Modo automatically generates UV maps for many functions and tools, but you may want to modify or replace those maps. The UV mapping tools are useful to create initial maps, but UV map operators modify existing maps. For information about using UV maps in Modo see the Working with UV Maps topic. For information about a specific function, click a link below to see the details about that function.

UV Map Tools

Create UV Tool

View Projection

UV Peeler

UV Unwrap

UV Relax

UV Box Transform Tool

Split, Move, and Sew UVs

Texel Density

UV Map Operators

Align UV

Fit UVs

Select Border

Mirror UVs

Rotate UV

Flip UV

Sew UVs

Split, Move, and Sew UVs

Orient Pieces

Pack UVs

Merge UVs

Symmetrize UVs

Rectangle

Texel Density

Cut, Copy, and Paste UVs

Delete UVs

Export UVs to EPS

Export UVs to SVG

Transform UV Map

Select Error UVs

Select Unmapped Polygons

Create UV Tool

The Create UV tool provides several methods and options for generating the UV maps. If you have not selected any polygons, the tool uses all visible foreground polygons to create the UV map. To isolate certain areas of your model, select the polygons before using the tool. The Create UV tool works by projecting the polygons from 3D space into the 2D (UV) space. As such, it is important to specify the most appropriate projection type. Using the Create UV tool to create a UV map is often only part of the process because this tool generally results in UV maps that need some manual editing to complete.

Modo stores UV maps as discontinuous 2D Vertex Maps for 0 -1 UV coordinate data. You use UV maps to explicitly link vertices in the 3D mesh to locations on a 2D image for accurate texture mapping. The UV map can be considered as a method to lay out a 3D mesh so that it lays flat for the 2D image to be applied to the mesh accurately. Modo provides a UV view to help you visualize the UVs over a bitmap image. You can use the **UV** layout to see mesh geometry in the viewport and the UV map.

Tip: Use the **Distortion Compensation** option when you create a primitive that does not have equal radius values. For more information, see Managing Distortions.

The **Create UV** tool is available in a number of places in the interface:

- **UV** layout: In the Toolbox on the left panel, open the **UV** tab and click **Project**. The Create UV properties are listed in the bottom of the Toolbox panel.
- **Model** layout: In the Toolbox on the left panel, open the **UV** tab and click **Project**. The Create UV properties are listed in the bottom of the Toolbox panel.
- **Procedural Mesh Operation**: Open the **Mesh Operations** tab on the right panel, click **Add Operator**, expand the **UV** category, and click **Create UV Map**. The properties are displayed in the right panel.

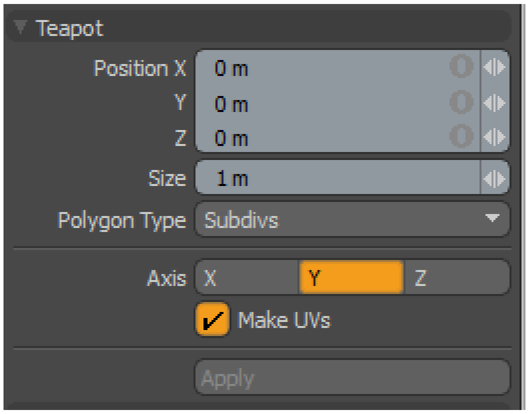
Creating UVs using the Direct Modeling Toolbox tool

The following is an example of creating a UV map using the Teapot primitive.

To generate UVs with the Toolbox Create UV tool:

1. Open the **UV** layout.
2. In the menu bar, click **Geometry** > **Primitives Tools** > **Teapot**.

   This is an example. You can work with other meshes.
3. In the Toolbox on the left pane, set the teapot **Size** to **1m** and then click **Apply**.



Teapot

Position X 0 m

Y 0 m

Z 0 m

Size 1 m

Polygon Type Subdivs

Axis X Y Z

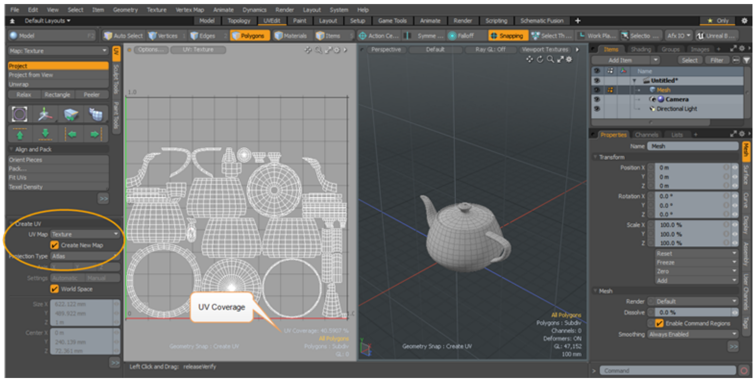
☑ Make UVs

Apply

4. In the Toolbox on the left pane, click **Project**.

   Modo generates a UV map, which appears in the UV Texture view. In the Toolbox in the left pane, the **Create UV** options are displayed.

   Alternatively, you can specify another **UV Map** previously created by selecting it from the **UV Map** dropdown menu. If you want to create a new UV map, click **Create New Map**.



5. Set the **Projection Type** is set to **Atlas**.

Creating UVs in the Schematic Viewport

The following is an example, using the Hippopotamus preset and creating a UV map.

1. Directly below the menu bar, in the switcher bar, click **Layout** > **Model**.
2. Press **F6** on your keyboard to open the Presets, expand **Assets** > **Meshes** > **Animals**, and double-click **Hippopotamus 01**.

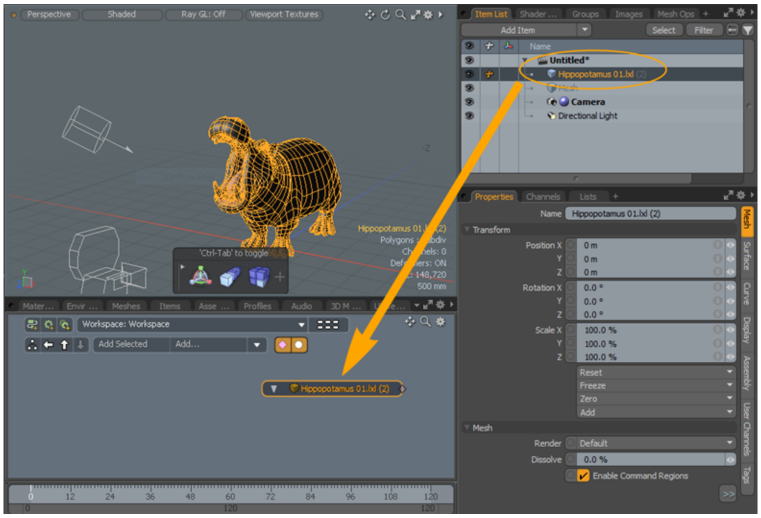
> **Tip:** If working on a macOS, you may need to hold down the **Fn** key when using F keys. For example, to use **F6** while working on a mac, press **Fn** + **F6** on your keyboard.

For more information, see Preset Browser.

3. Close the Preset Browser dialog.
4. In the right pane, open the **Items List** tab and drag the **Hippopotamus 01** item into the Schematic viewport.

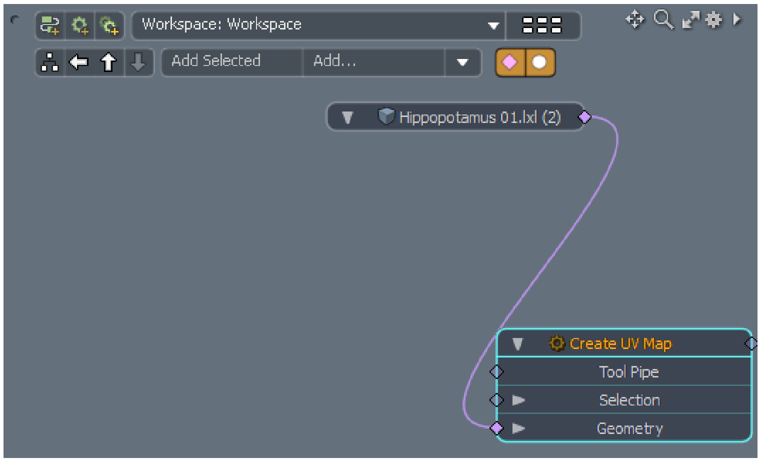
> **Tip:** If the Schematic tab is not displayed in the viewport, click the plus button and select **Utilities** > **Schematic**.

The Hippopotamus displays in the 3D viewport and the Schematic viewport.

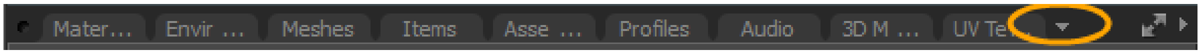

Perspective
Shaded
Ray GL: Off
Viewport Textures
Item List
Shader ...
Groups
Images
Mesh Ops
Add Item
Select
Filter
Name
Untitled*
Hippopotamus 01.lxl (2)
Camera
Directional Light
Properties
Channels
Lists
Name Hippopotamus 01.lxl (2)
Transform
Position X
Rotation X
Scale X
Reset
Freeze
Zero
Add
Mesh
Render Default
Dissolve 0.0 %
Enable Command Regions
Workspace: Workspace
Add Selected
Add...
Hippopotamus 01.lxl (2)

5. In the **Schematic** viewport, click **Add**, expand **Mesh Operations** > **UV**, and double-click **Create UV Map**.

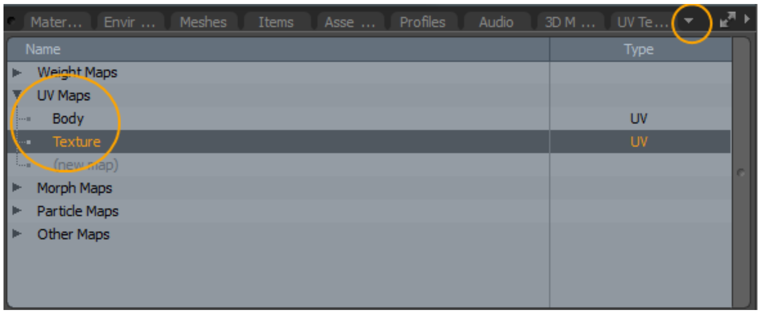
   The Hippopotamus 01 item links to the Create UV Map item.


Workspace: Workspace
Add Selected
Add...
Hippopotamus 01.lxl (2)
Create UV Map
Tool Pipe
Selection
Geometry

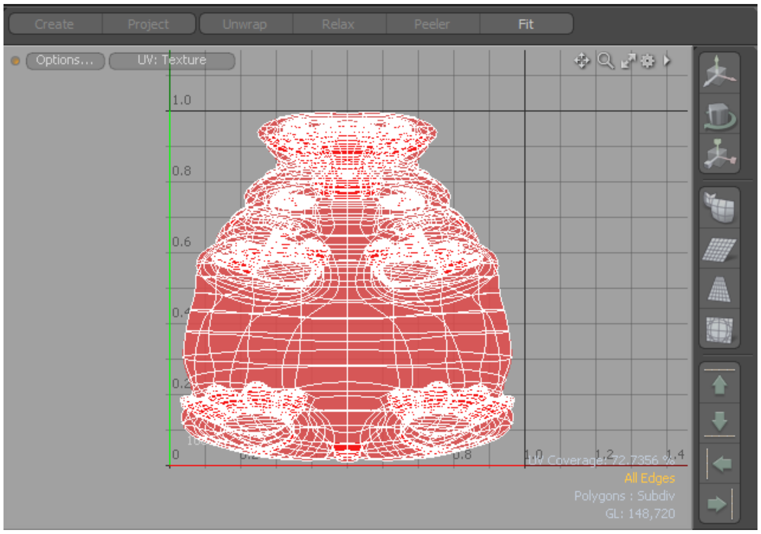
6. In the **Schematic** viewport on the bottom of the view, click the **New Tab** button on the top right corner of the tabs listed.


Mater...
Envir ...
Meshes
Items
Asse ...
Profiles
Audio
3D M ...
UV Te

7. Select **New Tab** > **Data Lists** > **Vertex Map List**.
8. In the Vertex Map List viewport, click **UV Maps** and double-click **Texture**.


Mater…
Envir …
Meshes
Items
Asse …
Profiles
Audio
3D M …
UV Te…
Name
Type
Weight Maps
UV Maps
Body
UV
Texture
UV
(new map)
Morph Maps
Particle Maps
Other Maps

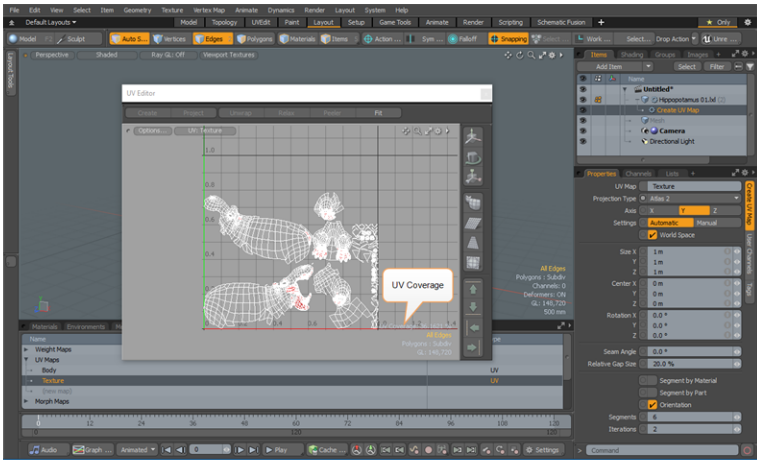
The UV Editor displays the Hippopotamus in the UV space.


Create
Project
Unwrap
Relax
Peeler
Fit
Options…
UV: Texture
1.0
0.8
0.6
0.4
0.2
0
UV Coverage: 72.7356 %
All Edges
Polygons : Subdiv
GL: 148,720

9. In the right pane, open the **Items List** tab, click the plus button beside the **Hippopotamus** item and then click **Create UV Map**.
10. In the **Properties** pane, change the **Projection Type** to **Atlas2**.

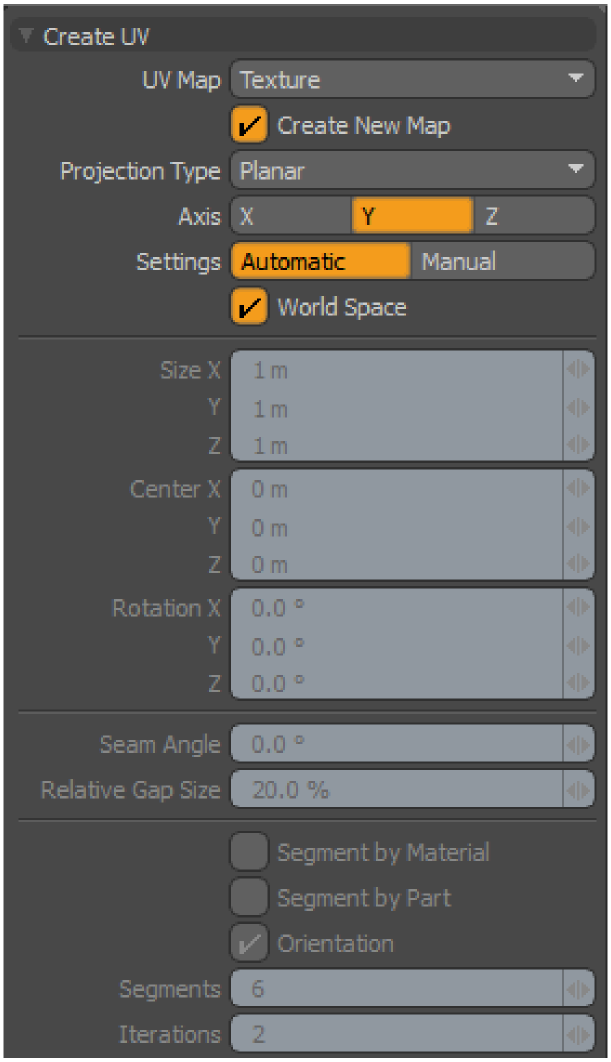
    The Hippopotamus 01 mesh displays and updates in the **UV Editor**.

Tip: The **UV Coverage** value is displayed at the bottom right corner of the **UV Editor**, which shows the amount of uniform UV space of the total region that is covered by geometry. This option is useful to game workflows, when trying to maximize the amount of texture space being used for the target surface. When you use an UV editing tools, the coverage is updated at the end of the editing process.



Create UV Options

The following options are available:



Create UV

UV Map: Texture

☑ Create New Map

Projection Type: Planar

Axis: X | Y | Z (Y selected)

Settings: Automatic | Manual (Automatic selected)

☑ World Space

Size X: 1 m
Y: 1 m
Z: 1 m

Center X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Seam Angle: 0.0 °

Relative Gap Size: 20.0 %

☐ Segment by Material

☐ Segment by Part

☑ Orientation

Segments: 6

Iterations: 2

Option	Description
UV Map	Allows you to select a UV texture map to project UVs. You can explicitly choose the destination texture map from the dropdown menu.
Create New Map	Creates new texture map if the mesh does not have a UV map using the specific name selected in the UV Map option, otherwise it does not create a new map.
Projection Type	Specifies the type of projection used when creating the UV map. **Planar** - Projects the selected polygons from an orthogonal view determined by the **Axis** setting. The resulting UV map actually looks quite similar to viewing the model in a 3D viewport set to **Top**, **Front**, or **Side**. Use this mode for quickly assigning areas of a mesh that are all facing the same direction. For more complicated geometry (such as a head), **Planar** results in a significant amount of UV overlap (such as the projection places polygons on both sides of the head into the same UV coordinates). A **Planar** projection is good for landscapes and some architectural or hard-surface models. **Cylindrical** - Projects the polygons into the UV map based on a virtual cylinder that surrounds the mesh. If you consider the cylinder as a soda can and set the **Axis** to **Y**, the can sits on a table, but if you set the **Axis** to **X** or **Z**, it lies on its side. Use **Cylinder** for mapping specific areas of a model (such as arms) with organic meshes or for tubes and pipe segments with hard-surface meshes. **Spherical** - Projects the UVs from a virtual sphere that surrounds the mesh. Modo places the poles of the sphere based on the **Axis** setting. The **Y** axis setting places the poles up and down; an **X** axis puts them side to side; and a **Z** axis puts them front to back.

	Atlas - Uses a multi-faceted projection method to place all polygons into the UV map and maintains the relative scale based on the 3D volume of the polygons. This is designed to minimize any texture distortion from non-uniform UV data. However, the **Atlas** projection results in a highly discontinuous UV map, which may be useful only for certain algorithmic uses, such as texture baking. **Barycentric** - Forces each individual polygon to fill the 0 to 1 UV space. Quadrangular polygons fit perfectly to the outer bounds of the UV area. Modo places triangles and other polygons within the UV space as best as it can. If you have a texture that you want to tile across every polygon in the mesh, this can be a very useful mode. **Atlas 2**- Attempts to fill the UV space with all selected polygons in the most efficient way without distorting the relative scale of each polygon from 3D to 2D space. This is similar to **Atlas** mode. **Cubic** - Projects the UVs from the polygons depending on their normal direction and segmented by the six planar directions: Top, Bottom, Left, Right, Front, and Back. This mode is similar to a **Planar** projection, however, it avoids the common backward texture errors that can result from **Planar** projections.
Axis	Sets the main axis direction for generating the UV projection.
Settings	Specifies how Modo scales the UV projection. **Automatic** - Sets the scale of the UV projection in a manner to best fill the total available UV space based on the selected mesh's bounding box. **Manual** - Uses the **Size**, **Center**, and **Rotation** settings to define a specific projection that affects the overall size of the final UV map.

Seam Angle	Specifies at what angle a seam appears in the UV map for **Cylindrical** and **Spherical** projections that have a seam. For example, if your model has a face and it happens to be turned in a way that creates a seam across the face, you can adjust the seam angle to place it in the back of the head without changing the geometry.
Relative Gap Size	Produces a greater distance between individual UV islands when using one of the atlas projection types that generate multiple pieces (or islands).
Segment By Material	When enabled, creates additional island segments with atlas projection types so that individual islands never have polygons from more than one material tag.
Segment By Part	When enabled, creates additional island segments with atlas projection types so that individual islands never have polygons from more than one part tag.
Orientation	When enabled, forces UV islands to flip based on their axial direction when used with atlas projection types. For example, if you select this option with a head, Modo maps one side of the head to mirror the other side. This makes it easier to distinguish one side from the other. For **Atlas 2** projections, this checkbox is always selected.
Segments	Determines (roughly) the number of generated UV islands when using the **Atlas 2** projection type.
Iterations	Generates UV islands, with the **Atlas 2** projection type, by grouping together polygons with similar normals to reduce overall distortion. Increasing the **Iterations** value, increases Modo's sensitivity to the normal direction. This generates less distortion with more UV islands.

UV Box Transform Tool

The UV Box Transform tool allows you to quickly move groups of polygons in the UV space. This tool, while active, draws a rectangular bounding box around the component selected in the UV space, to which you can apply transform actions.

Applying UV Box Transform

In this example, add the Teapot model to your scene, and use the UV Box Transform tool to transform a group of polygons in the UV space.

Adding a Primitive to your Scene:

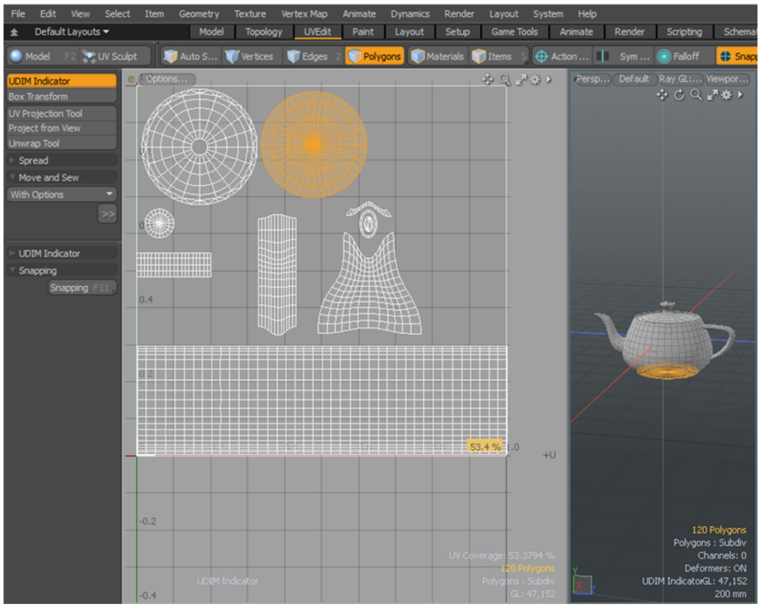
1. Open the **UVEdit** layout.
2. Below the interface layout tabs, click **Polygons** to enable the Polygon Selection Mode.
3. In the menu bar, click **Geometry** > **Primitive Tools** > **Teapot**.
4. In the **Toolbox** on the left pane, under the Teapot properties, click **Apply**.

   The Teapot primitive displays in the viewport.

Selecting a Group of Polygons to Move in the UV Space:

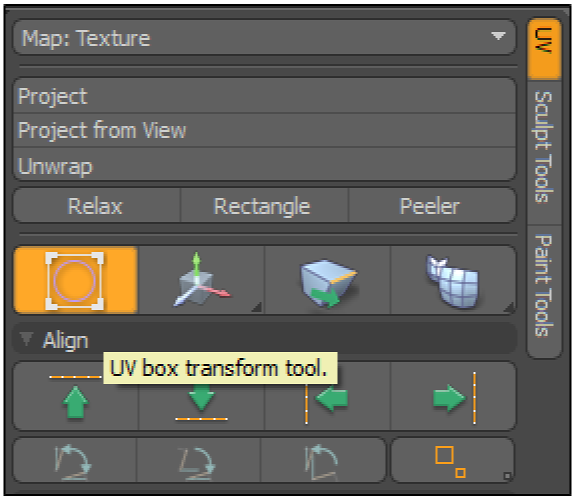
- In the UV window, double-click on a group of polygons.

  The group of polygons is selected and highlighted.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Default Layouts
Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Animate
Render
Scripting
UDIM Indicator
Box Transform
UV Projection Tool
Project from View
Unwrap Tool
Spread
Move and Sew
With Options
Snapping
Polygons
120 Polygons

Activating the UV Box Transform Tool:

- On the left pane, in the Toolbox, click **UV box transform tool**.


Map: Texture
UV
Sculpt Tools
Paint Tools
Project
Project from View
Unwrap
Relax
Rectangle
Peeler
Align
UV box transform tool.

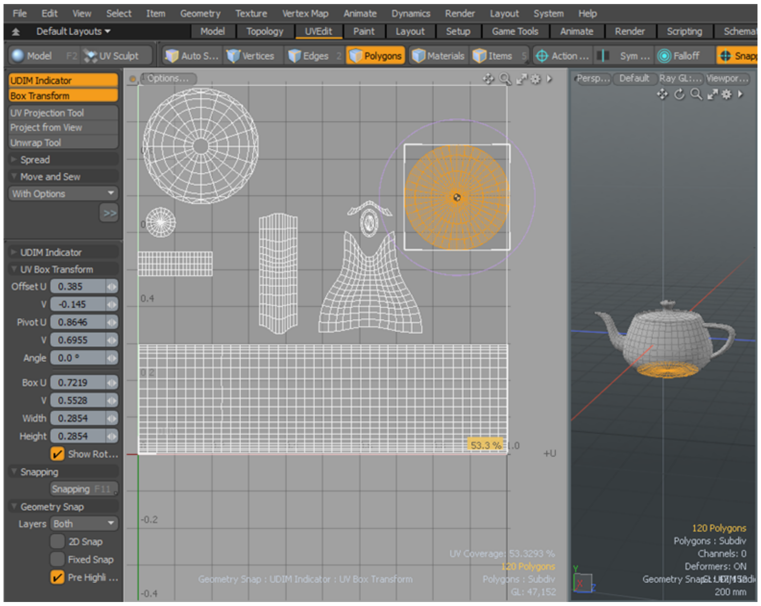
A rectangular bounding box displays around the selected item.

Moving a Group of Polygons:

1. Click inside the UV bounding box of the selection of polygons and drag it to a new position in the UV space.

   A rectangular bounding box displays around the selected item and the pivot does not change position when moving the selection.

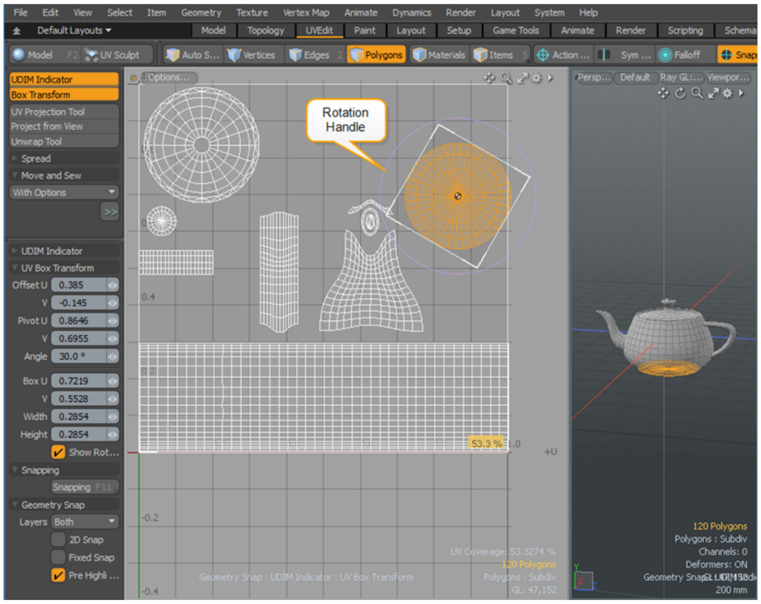
Tip: Press and hold the **Ctrl** key to constrain to the axis (U and V).
Hold the **Shift** key to snap the UV bounding box corners to the major axis lines.



UV Box Transform
Offset U 0.385
V -0.145
Pivot U 0.8646
V 0.6955
Angle 0.0 °
Box U 0.7219
V 0.5528
Width 0.2854
Height 0.2854

Rotating and Repositioning a Group of Polygons:

1. Press the **Ctrl** key and then click and drag the rotation handle to constrain the rotation to fixed 15° intervals.



UV Box Transform
Offset U 0.385
V -0.145
Pivot U 0.8646
V 0.6955
Angle 30.0 °
Box U 0.7219
V 0.5528
Width 0.2854
Height 0.2854

2. Click and drag on the pivot handle to move the pivot position to the side of the UV bounding box.

   The pivot snaps to the side edge of the UV bounding box.

3. Click and drag the polygon selection to reposition the group of polygons.
4. Click and drag the rotation handle to rotate the group of polygons.

   The center of rotation is based on the new position of the pivot. Polygons that overlap on other polygons in the UV space are displayed in red.

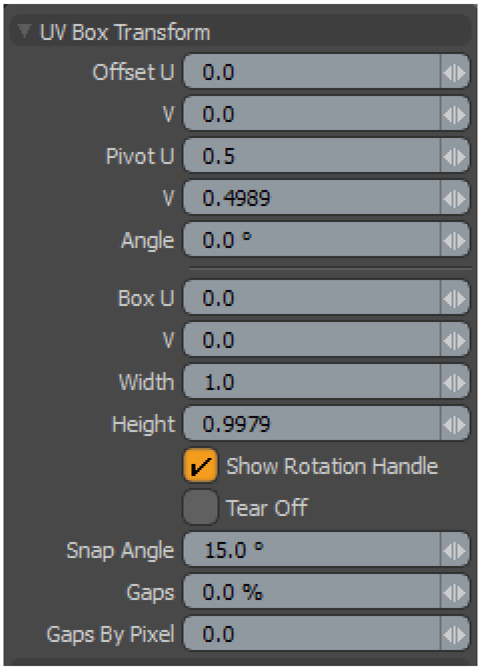
Scaling a Selection of Polygons:

- With a group of polygons selected, right-click and drag a side of the bounding box.

  The group of polygons is scaled from the opposite edge, not from the pivot point. Dragging on a corner scales a group of selected polygons uniformly.

Tip: Dragging on a corner scales the selection uniformly.

UV Box Transform Properties



UV Box Transform

Offset U 0.0

V 0.0

Pivot U 0.5

V 0.4989

Angle 0.0 °

Box U 0.0

V 0.0

Width 1.0

Height 0.9979

☑ Show Rotation Handle

☐ Tear Off

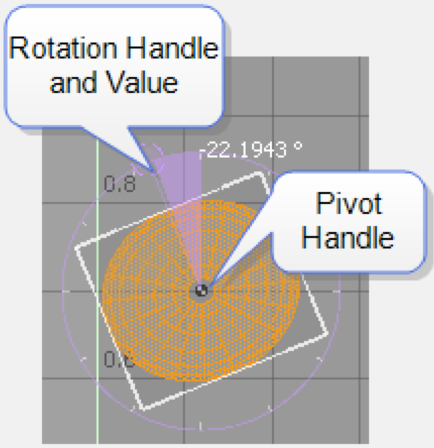
Snap Angle 15.0 °

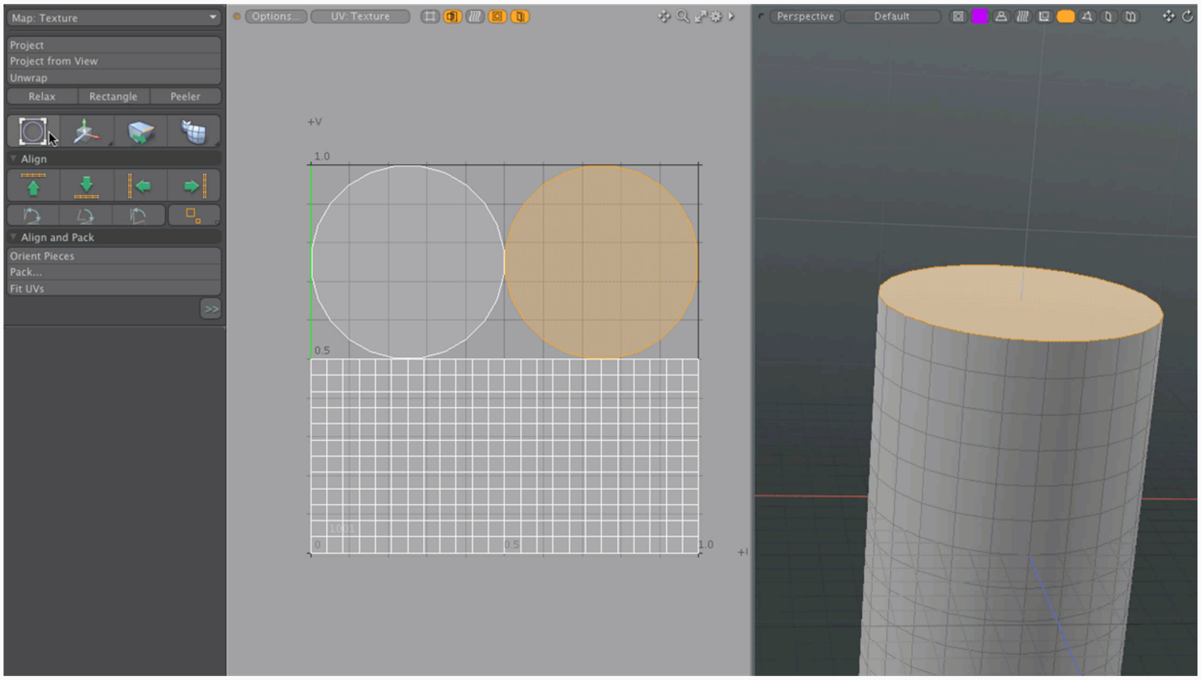
Gaps 0.0 %

Gaps By Pixel 0.0

Offset U and V	Set the offset amount to translate selected UV polygons along the **U** or **V** direction.
Pivot U	Position the pivot for the center of rotation.

and V	When adjusting an object interactively in the 3D viewport, the **Pivot U** and **V** fields display the current offset distance values applied. To view the pivot, enable the **Show Rotation Handle** option. • In the UV viewport, click and drag the pivot handle to move it to a side or to a corner of the UV bounding box. The pivot handle snaps into place.
Angle	Set the angle to rotate the selected UV polygons at the pivot position. The pivot does not move from its position. When adjusting an object interactively in the viewport, and when the **Show Rotation Handle** option is enabled, the angle value is displayed and updates while dragging the rotation handle. **Tip:** Holding the **Shift** key snaps the UV bounding box corners to major axis positions (constraining the rotation to fixed 15° intervals).
Box U and V	Set the UV bounding box position at left-bottom of the box. When adjusting an object interactively in the viewport, the **Box U** and **V** fields display the current position.
Width	Set the width of the UV bounding box. When adjusting an object interactively in the viewport, the **Width** field displays the current UV bounding box size. In the viewport you can do the following: • Adjust the **Width** value only of the UV bounding box by right-clicking and dragging on a side of the UV bounding box. • Adjust both the **Width** and **Height** of the UV bounding box at once by right-clicking and dragging a corner of the UV bounding box. • Scale the selected polygon group within the UV bounding box by right-clicking and dragging a top or bottom of the bounding box. The polygon group is scaled from the opposite edge of the bounding box, not from the pivot point.
Height	Set the height of the bounding box. When adjusting an object interactively in the viewport, the **Height** field displays the current UV bounding box size. In the viewport you can do the following: • Adjust the **Height** value only of the UV bounding box by right-clicking and dragging on a side of the UV bounding box.

	• Adjust both the **Width** and **Height** of the UV bounding box at once by right-clicking and dragging a corner of the UV bounding box. • Scale the selected polygon group within the UV bounding box by right-clicking and dragging a top or bottom of the bounding box. The polygon group is scaled from the opposite edge of the bounding box, not from the pivot point.
Show Rotation Handle	Toggle the visibility of **Rotation** and **Pivot** handles. When adjusting an object interactively in the viewport, the rotation handle displays the angle value and the pivot handle displays and remains in its position.  Rotation Handle and Value -22.1943 ° 0.8 Pivot Handle 0.6
Tear Off	Splits transforming polygons.
Snap Angle	Sets the delta rotation angle for rotation tools.
Gaps	Adds border padding to the UV map. By default the UV map is drawn up to the edge of the UV space, which may cause problems with pixel blending. Entering a percentage gap value corrects this type of problem. The Gaps by Pixel value is automatically calculated.

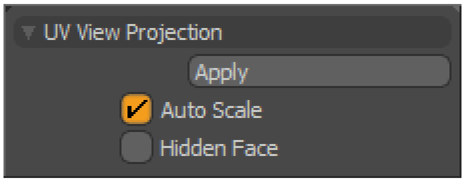
	 Map: Texture Project Project from View Unwrap Relax Rectangle Peeler Align Align and Pack Orient Pieces Pack... Fit UVs Options... UV: Texture Perspective Default +V 1.0 0.5 0 0.5 1.0
Gaps By Pixel	Sets the padding to an image pixel unit. This option automatically updates the **Gaps** value.

View Projection

The View Projection tool (also called the Project from View tool) creates a UV map of the currently selected polygons based on their visibility in the 3D viewport. The final UV map reflects the selected polygons as they appear in the viewport, itself, similar to taking a photograph of them.

Usage

The View Projection tool is in the toolbox of the **UVEdit** layout and is also available as a menu command. Prior to activating the tool, you should select, position, and orient the Target geometry within the viewport because the UV maps should be assigned. Once you assign the UV maps, choose **Texture > UV Tools > View Projection Tool**. After you click in the 3D viewport, Modo generates the map into the currently selected (or the most recently selected) UV map or, if there is no map present, Modo creates a new map named **Texture**. If you are not happy with the resulting map, you can adjust, scale, or rotate the viewport. Clicking again updates the UV map. When the tool is active, options are available in the tool's **Properties** panel.


UV View Projection

Apply

☑ Auto Scale

☐ Hidden Face

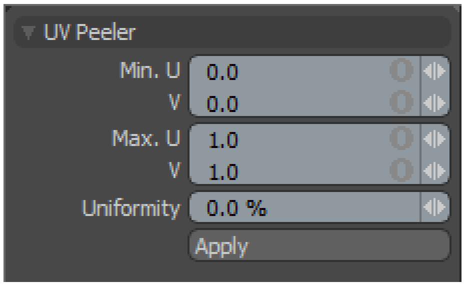
Apply: Applies the current settings. This is the same clicking in the 3D viewport, but you can only click **Apply** when the tool is not in interactive mode.

Auto Scale: When enabled, automatically scales the resulting UV to fit precisely within the full 0-1 UV space.

Hidden Face: When enabled, the resulting UV map ignores any rear-facing geoemtry. (These are polygons that are facing away from the viewport.)

UV Peeler

You use the Peeler tool for manually fixing UV maps. With it, you can cut a seam of edges in a well-behaved quad mesh. The Peeler tool spreads UVs out perpendicular to the seam and tries to loop back to the cut from the other side. You can control the region of UVs produced. You can use this tool for tedious, manual-style Atlas mapping or other cylindrical unwrapping tasks.


UV Peeler

Min. U 0.0

V 0.0

Max. U 1.0

V 1.0

Uniformity 0.0 %

Apply

You can adjust Peeler tool settings:

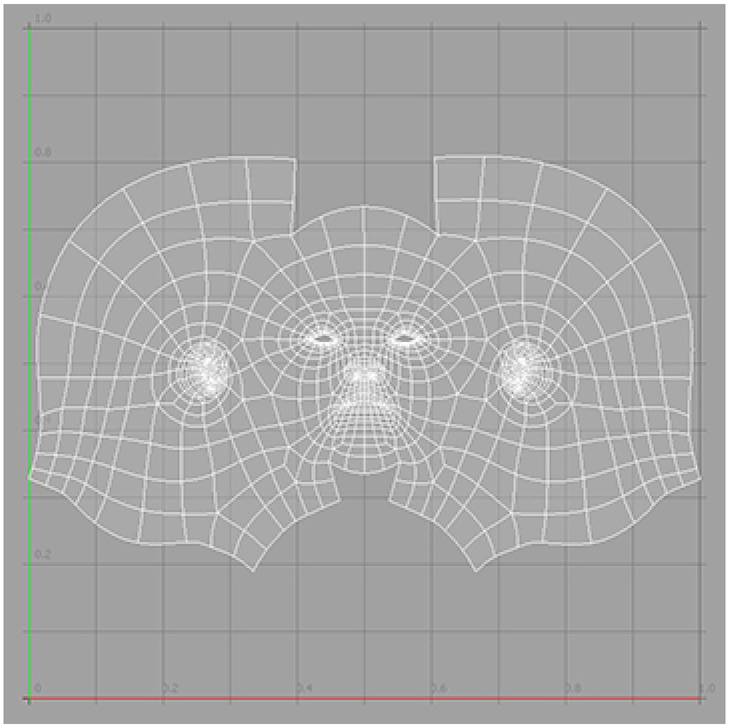
Min U, **Max U**, **Min V**, **Max V**: Define a box in UV space where Modo positions all the grids. Normally, these values are **0**, **0**, **1**, and **1**.

Uniformity: Controls the size of the UV polygons in proportion to the size of the 3D space polygons. At **0%** every polygon's size in UV space is proportional to its size in 3D space. At **100%** the polygons all have the same size to make a regular UV grid.

The Peeler tool works well on meshes that have uniform quadrangular polygons organized with adequate edge loop flow. For meshes that have more complex topology, use the UV Unwrap tool, which uses a more sophisticated algorithm to untangle the UVs.

UV Unwrap

The UV Unwrap tool provides an elegant solution to creating UV maps from almost any kind of mesh, from hard surface to organic. UV unwrapping unfolds the mesh at the seams and lays it out on a flat page.



UV mapping focuses on managing the stretching and the distortion that comes with representing a 3D surface in a 2D plane. Therefore, put some thought into which edges to select for the best results.

For more information on UV maps, see Working with UV Maps.

Tip: Try to hide any seams in the least accessible areas (such as the back of a head under the hair or under the arms). The UV Unwrap tool acts on any visible foreground geometry, so you may want to hide any geometry that should not be affected.

Creating Seams

There are two ways you can create seams for UV Unwrap:

- Select all edges at once and then activate UV Unwrap.
- Use the **Cut Map** tool to gradually add multiple selections as seams. This makes creating complex selections easier.

Using Cut Map

1. Once you have your mesh ready in the viewport, open the **Modo** layout.
2. Make sure you're in **Edges** selection mode and select a series of edges you want to cut the geometry along.

Note: For more information on selection modes, see Selection Mode Types.

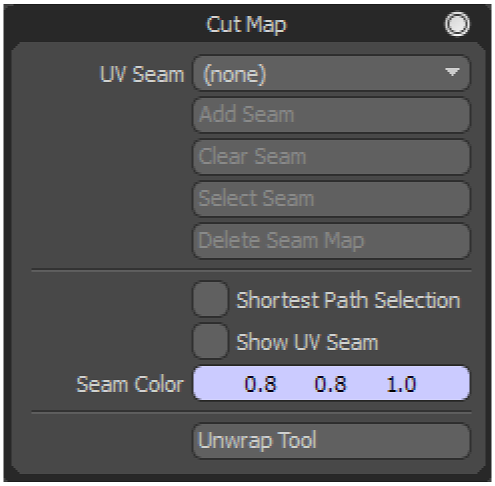
Select a continuous loop of edges over the geometry to define the outer edge for separated UV islands. This makes it easy to unwrap an entire character into multiple islands simultaneously. Hold **Shift** to select multiple edges.

The selected edges become the boundaries of UV islands, which are connected groups of polygons in a UV map.

3. Click the UV tab on the left above the toolbox, and click **Cut Map...**

   The **Cut Map** menu opens.

Tip: You can pin the menu by clicking the circle in the top-right corner.


Cut Map
UV Seam
(none)
Add Seam
Clear Seam
Select Seam
Delete Seam Map
Shortest Path Selection
Show UV Seam
Seam Color
0.8 0.8 1.0
Unwrap Tool

4. Set **UV Seam** to **(new)**.

   The **Create New Vertex Map** dialog opens.

5. Enter a **Vertex Map Name**, then select the **Vertex Map Type**. Additional settings become available depending on which type you select.

6. Click **OK** to close the **Create New Vertex Map** dialog.
7. In the **Cut Map** popover, click **Add Seam**.

   This adds the selection as a new seam to the UV seam map.

Note: If you click **Add Seam** with no edges selected, all edges are added as seams.

8. Continue making selections and clicking **Add Seam** until you have selected all the seams you need.

Tip: **Shortest Path Selection** can help you select tricky areas. It allows you to select elements between the most recently selected and the current element under the cursor position. For more information, see Shortest Path Selection.

Tip: Enable **Show UV Seam** to see which edges you've already marked as seams. The seams are highlighted and you can modify the color in the **Seam Color** field.

To delete a seam, click **Clear Seam**. To delete the entire seam map, click **Delete Seam Map**.

When you have set up all seams, you can run the UV Unwrap tool directly from the menu. Click **Unwrap Tool**.

UV Unwrap Workflow

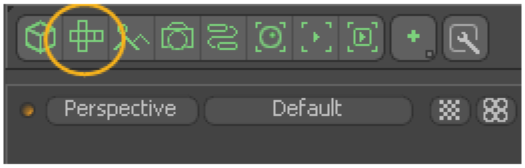
1. Activate UV Unwrap by:
   - In the UV layout, clicking **Unwrap** on the left panel.

Note: You can open the UV layout from the menu bar, by clicking **Layout** > **Layouts** > **UV**.

OR

- In the **Modo** layout, clicking **Unwrap Tool** in the **Cut Map** menu.

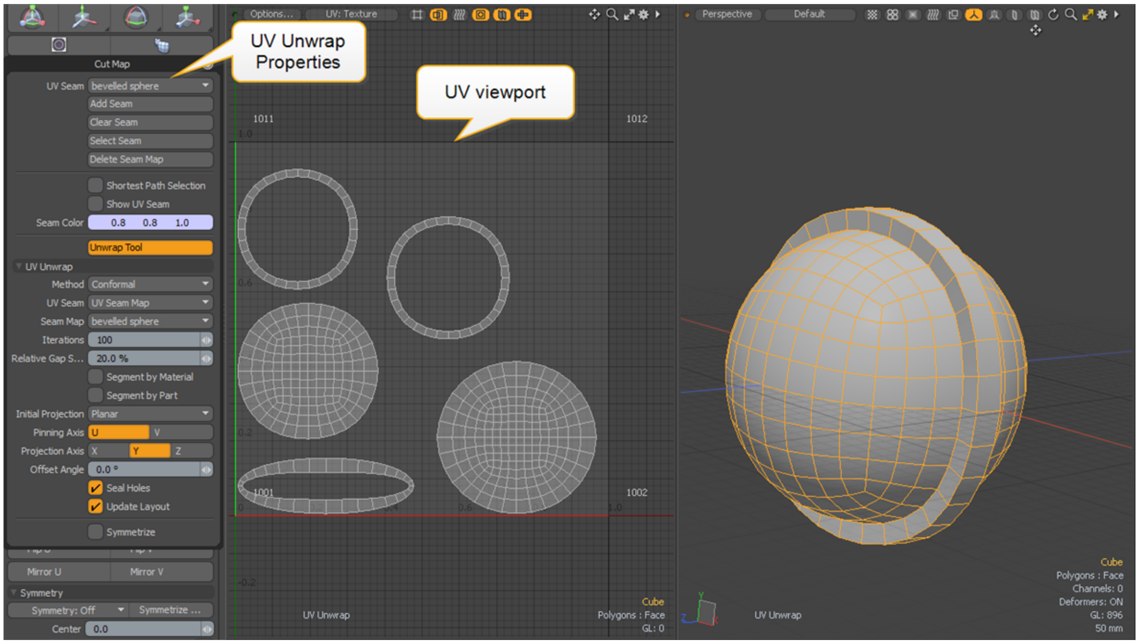
  If you're working in the **Modo** layout, make sure you have the UV viewport open: click the thin line above the 3D viewport to reveal additional controls, and click the UV viewport button.



Note: If you've used **Cut Map** to specify the seams, In the UV Unwrap **Tool Properties**, make sure you have **UV Seam** set to **UV Seam Map**, and the appropriate map selected. This is done automatically when you activate UV Unwrap from the **Cut Map** menu.

2. Click and drag in the UV viewport to increase or decrease the number of **Iterations**.

   The UV map now displays separate UV islands.


UV Unwrap Properties
UV viewport
Cut Map
Add Seam
Clear Seam
Select Seam
Delete Seam Map
Unwrap Tool
UV Unwrap
Mirror U
Mirror V
Symmetry
Symmetrize ...
1011
1012
1001
1002
UV Unwrap
Perspective
Default
Cube
Polygons : Face
Channels: 0
Deformers: ON
GL: 896
50 mm

UV Seam: bevelled sphere
☐ Shortest Path Selection
☐ Show UV Seam
Seam Color: 0.8 0.8 1.0
Method: Conformal
UV Seam: UV Seam Map
Seam Map: bevelled sphere
Iterations: 100
Relative Gap S...: 20.0 %
☐ Segment by Material
☐ Segment by Part
Initial Projection: Planar
Pinning Axis: U
Projection Axis: Y
Offset Angle: 0.0 °
☑ Seal Holes
☑ Update Layout
☐ Symmetrize
Symmetry: Off
Center: 0.0

Tip: Usually, when Modo uses more iterations, the resulting UV map is better, although only up to a point. Red areas in the UV map indicate overlapping UVs. This means that they share the same texture space. For more information on how to deal with overlapping UVs, see Overlapping UVs.

3. Adjust the UV Unwrap Properties to adjust the final unwrap quality.

   The tool works in real-time (depending on the density of the target mesh), and you can try different permutations to achieve the best result.

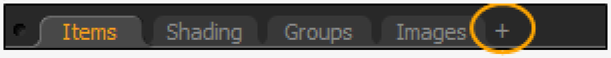
Tip: Once you have defined a map, you can apply additional adjustments with the UV Relax tool to improve the results produced by the Unwrap tool.

Procedural UV Unwrap

Here's how to use the UV Unwrap mesh operation in procedural modeling.

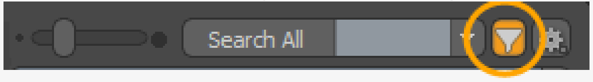
1. Specify your seams as described in Creating Seams.
2. On the right panel, open the **Mesh Ops** tab, and click **Add Operator**.

Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **New Tab** > **Data Lists** > **Mesh Ops**.

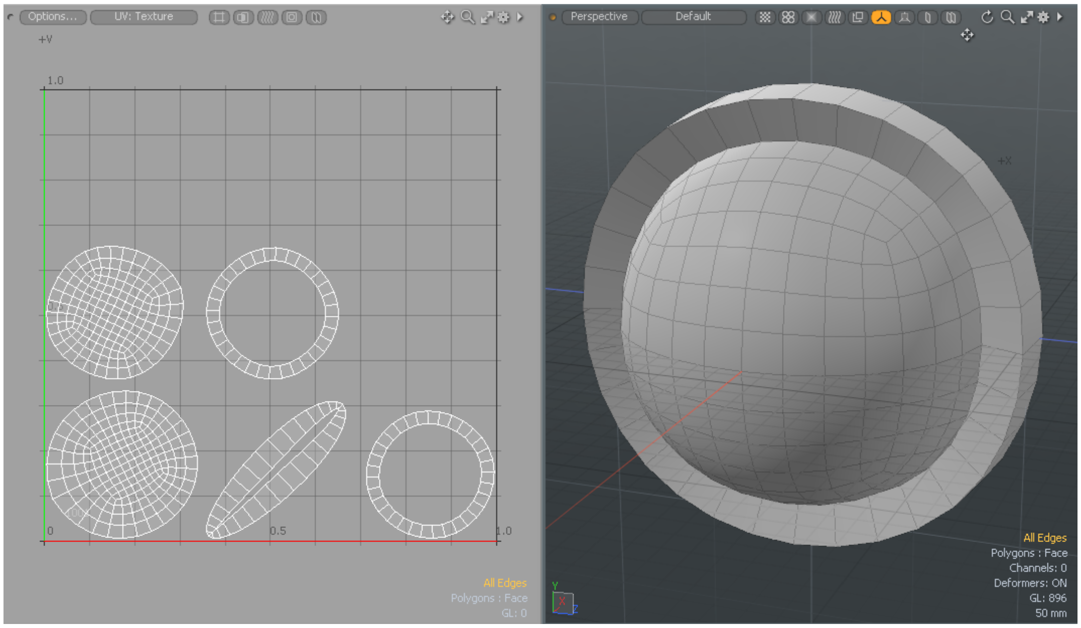

Items
Shading
Groups
Images
+

3. In the **Add Mesh Operations** dialog, click **Mesh Operations** > **UV**, and double-click **UV Unwrap**.

Tip: To quickly locate a mesh operation, click the filter icon and type the first couple of letters of the mesh operation you are looking for in the **Search All** field.

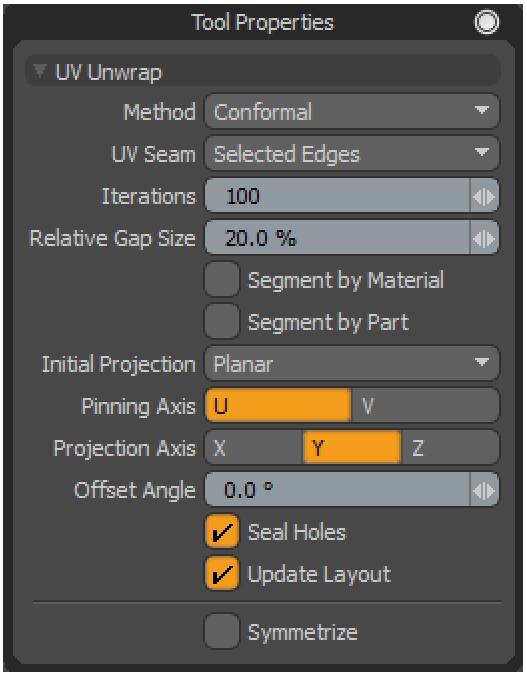


The UV map displays separate UV islands. As a result, the number of seams have reduced in the texture file. This is the goal you want to achieve when creating UV maps.


Options...
UV: Texture
+V
1.0
0
0.5
1.0
All Edges
Polygons : Face
GL: 0
Perspective
Default
All Edges
Polygons : Face
Channels: 0
Deformers: ON
GL: 896
50 mm

UV Unwrap Properties

The direct modeling UV Unwrap properties can be found in the lower left panel when the tool is active. If you're using the UV Unwrap mesh operation, the properties can be found in the lower right panel. Click the **UV Unwrap** sub-tab on the right side of the panel to open them.



Tool Properties

UV Unwrap

Method: Conformal

UV Seam: Selected Edges

Iterations: 100

Relative Gap Size: 20.0 %

☐ Segment by Material

☐ Segment by Part

Initial Projection: Planar

Pinning Axis: U V

Projection Axis: X Y Z

Offset Angle: 0.0 °

☑ Seal Holes

☑ Update Layout

☐ Symmetrize

UV Map	**Procedural modeling only.** Defines the name of the UV texture map used for the mesh operation.
Method	Specifies the UV unwrapping evaluation methods for Modo to use. There are two different UV unwrapping evaluation methods: • **Conformal** - uses the ABF++ algorithm for angle-based flattening, and is good for organic shapes. • **Angle Based** - uses the Least Square Conformal Map algorithm and may be better for hard-surface types of geometry. Try both and see which option produces the most desirable result.
UV Seam	Determines the way UV seam edges are specified. **UV Seam Map** - uses the seam edges from a specific UV seam map. Use this if you set up your edges using Cut Map. For more information, see Creating Seams. If you

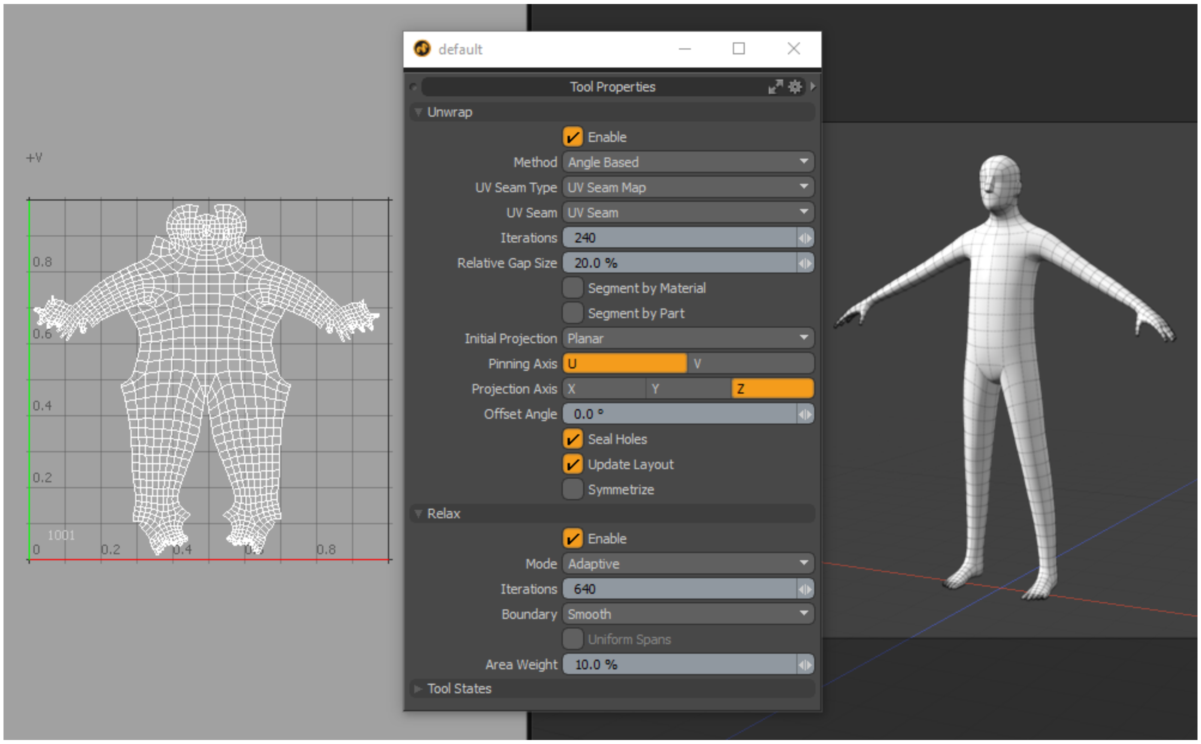
	select this, an additional **Seam Map** field appears, where you can choose the map to be used. **Selection Set** - uses the seam edges from a specific edge selection set map. If you select this, an additional **Selection Set** field appears, where you can choose the map to be used. For more information on selection sets, see Using Selection Sets. **Selected Edges** - uses the currently selected edges as seam edges.
Iterations	Determines the number of iterations for the global solver to run across the UV data. This is the amount of unwrapping. With the value set to **0** (zero), the UV view appears as if you just used a UV projection method based on the **Initial Projection** and the **Projection Axis** settings for the tool. Increasing the number of **Iterations** unwraps the data. The number of iterations necessary is largely dependent on the complexity of the target mesh.
Relative Gap Size	Dictates how much space Modo places between the UV boundaries. If your UV data has multiple islands (contiguous geometry pieces), there needs to be a gap between these UV chunks. Increasing this value pushes the UV data further apart as well as scaling the resulting UV data down to accommodate the needed space.
Segment by Material	When enabled, if you are unwrapping a contiguous mesh that has different materials assigned to various polygon groups, Modo uses these groups of materials to create continuous UV sections while unwrapping. This option provides a quick way to unwrap a single mesh into several UV segments that can either be left in a single UV map or pasted into separate UV maps.
Segment by Part	When enabled, this works like **Segment by Material**, but Modo uses the Part tags from the polygons.
Initial Projection	Determines how Modo creates the initial UV projection: **Planar**, **Cylindrical**, **Spherical**, **Use Current Map**, and **Group Normal**. The first three settings rely on the **Projection Axis** setting. • **Planar** - projects the 3D model into UV space based on a simple plane that is projected across the model in the user-defined axis. • **Cylindrical** - creates a virtual cylinder around the mesh. • **Spherical** - creates a virtual sphere for the projection. For more information about the projection modes, see the Create UV Tool topic. • **Use Current Map** - uses the current state of the UV map as a starting point rather than projecting new UVs. This is useful if you already have created a reasonable

	UV map and want to use the Unwrap tool to further refine the UVs without destroying your existing UV map. • **Group Normal** - uses the average normal of the selected polygons to determine the initial projection plane. This can be useful on geometry that is mostly facing the same direction, but this may not produce the right results if the mesh is complex and has many overlapping areas. To see the axis in 3D that Modo would use for the **Group Normal** projection, move the pointer over the 3D viewport and press **Ctrl/Cmd+A**. Doing this aligns the viewport to the average normal of the selected polygons.
Pinning Axis	Determines the UV axis to favor when shuffling the UV boundaries during the unwrap. **U** is the horizontal axis, and **V** is the vertical axis.
Projection Axis	Defines the axis' orientation for the **Planar**, **Cylindrical**, and **Spherical** projections.
Offset Angle	Adds some rotation to the resulting unwrapped UVs. This can be useful if your original mesh is not aligned to the World Axis. By adjusting this value, you can see the effect on the map in real time.
Seal Holes	When enabled, creates a virtual polygon to close any holes and improve the unwrapping effect. If the target mesh is not water tight (such as a head with open eye sockets), the unwrap algorithm can sometimes have difficulty resolving these open areas.
Update Layout	When enabled, forces the UV layout to remain within the 0 to 1 UV coordinate space.
Symmetrize	When enabled, produces a perfectly symmetrical UV map when Modo unwraps a perfectly symmetrical mesh (such as a character). To use, select the **Symmetrize** checkbox and then specify the **Symmetry Axis** and the **Mirror Axis**. You may need to increase the number of **Iterations** to make the mesh fully symmetrical.
Symmetry Axis	Defines the axis that the originating mesh is symmetrical across.
Mirror Axis	Defines the symmetry axis that the resulting unwrapped UV should mirror across. Generally, this is the same as **Pinning Axis**.

Unwrap & Relax

The Unwrap & Relax tool combines the Unwrap Tool and UV Relax into one tool, allowing you to perform two operations on your UVs in one step to create a UV pelt. The Unwrap & Relax tool can be found inside the

UV menu on the **Modo** and **Model** layouts. The Unwrap & Relax **Tool Properties** share the same attributes as the Unwrap Tool and the UV Relax tool.



Tool Properties

Unwrap

- ☑ Enable
- Method: Angle Based
- UV Seam Type: UV Seam Map
- UV Seam: UV Seam
- Iterations: 240
- Relative Gap Size: 20.0 %
- ☐ Segment by Material
- ☐ Segment by Part
- Initial Projection: Planar
- Pinning Axis: U V (U selected)
- Projection Axis: X Y Z (Z selected)
- Offset Angle: 0.0 °
- ☑ Seal Holes
- ☑ Update Layout
- ☐ Symmetrize

Relax

- ☑ Enable
- Mode: Adaptive
- Iterations: 640
- Boundary: Smooth
- ☐ Uniform Spans
- Area Weight: 10.0 %

Tool States

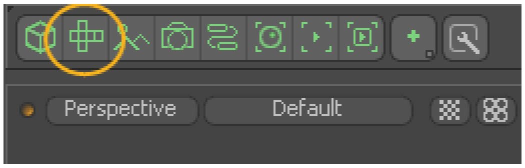
Using the Unwrap & Relax **Tool Properties** to create a single pelt map from this 3D object.

> **Tip:** The Unwrap & Relax tool cannot be used interactively in the viewport for relaxing UVs. Use the UV Relax tool when you need to use **Interactive** mode. See UV Relax.

Unwrap & Relax Workflow

1. Activate Unwrap & Relax by:
   - In the **Modo** layout, clicking **Unwrap & Relax** in the **UV** menu.

     If you're working in the **Modo** layout, make sure you have the UV viewport open: click the thin line above the 3D viewport to reveal additional controls, and click the UV viewport button.

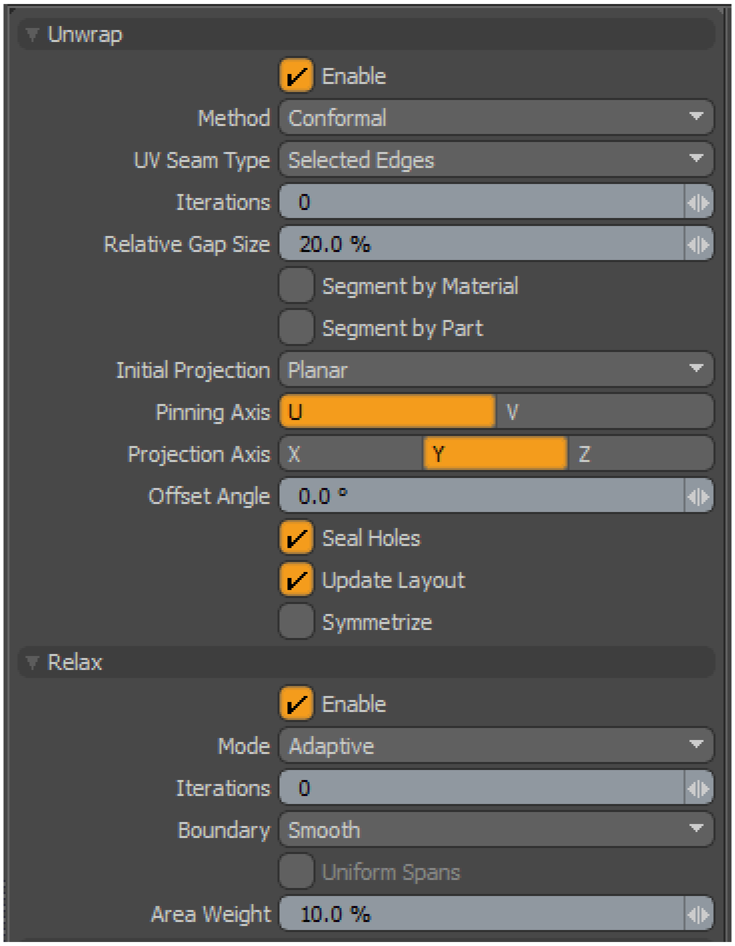


Note: If you've used **Cut Map** to specify the seams, In the **Unwrap Tool Properties**, make sure you have **UV Seam Type** set to **UV Seam Map**, and the appropriate map selected.

2. Click once in the 3D or UV viewport.

   The Unwrap & Relax tool is applied to your object.
3. Adjust the UV Unwrap Properties to adjust the final unwrap quality.
4. Adjust the Relax Properties to adjust the final relax quality.

Unwrap & Relax Tool Properties



Unwrap

☑ Enable

Method: Conformal

UV Seam Type: Selected Edges

Iterations: 0

Relative Gap Size: 20.0 %

☐ Segment by Material

☐ Segment by Part

Initial Projection: Planar

Pinning Axis: U | V

Projection Axis: X | Y | Z

Offset Angle: 0.0 °

☑ Seal Holes

☑ Update Layout

☐ Symmetrize

Relax

☑ Enable

Mode: Adaptive

Iterations: 0

Boundary: Smooth

☐ Uniform Spans

Area Weight: 10.0 %

The **Unwrap & Relax** tool properties panel.

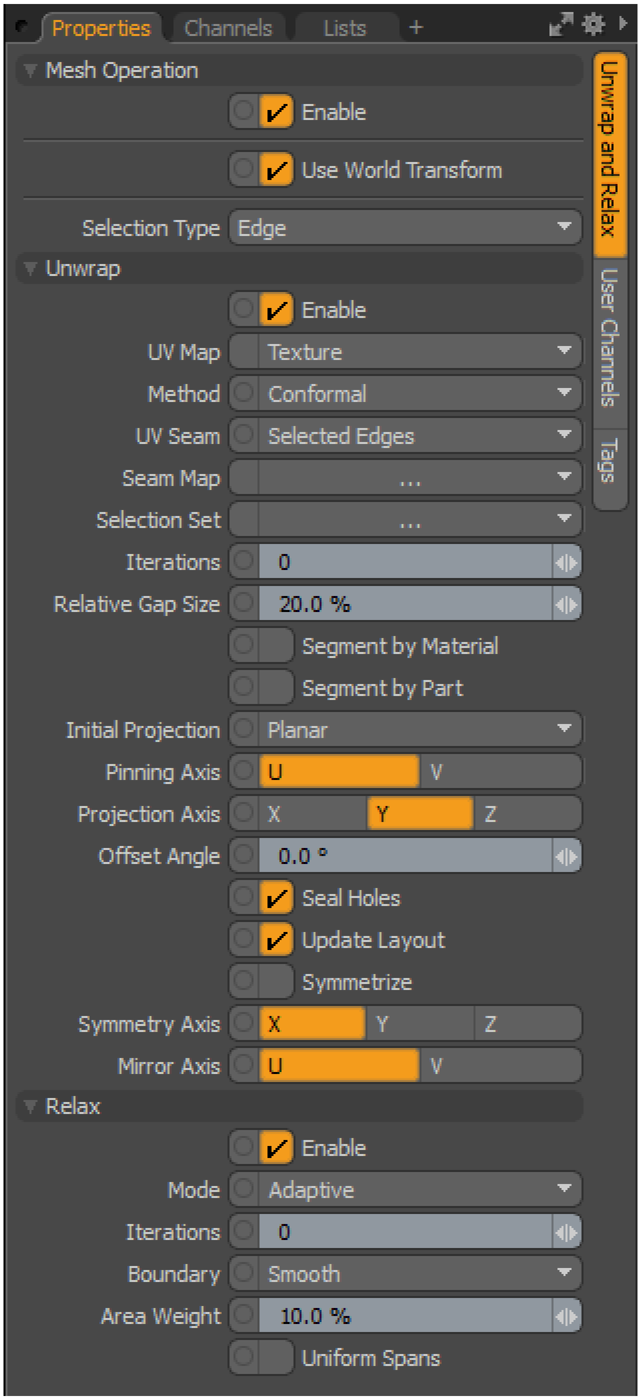
For **Unwrap** properties, see UV Unwrap Properties. For **Relax** properties, see UV Relax.

Note: **Ignore Edge Selection** in the **UV Relax** properties is enabled by default on the **Unwrap & Relax** tool.

Procedural Unwrap & Relax

The **Unwrap & Relax** mesh operation is the procedural version of the **Unwrap & Relax** tool.

The **Unwrap & Relax** mesh operation properties are the same as the **Unwrap & Relax** tool properties, which are a combination of the **Unwrap** tool properties and the **Relax** tool properties.



Mesh Operation

☑ Enable

☑ Use World Transform

Selection Type: Edge

Unwrap

☑ Enable

UV Map: Texture

Method: Conformal

UV Seam: Selected Edges

Seam Map: ...

Selection Set: ...

Iterations: 0

Relative Gap Size: 20.0 %

☐ Segment by Material

☐ Segment by Part

Initial Projection: Planar

Pinning Axis: [✓ U] V

Projection Axis: X [✓ Y] Z

Offset Angle: 0.0 °

☑ Seal Holes

☑ Update Layout

☐ Symmetrize

Symmetry Axis: [✓ X] Y Z

Mirror Axis: [✓ U] V

Relax

☑ Enable

Mode: Adaptive

Iterations: 0

Boundary: Smooth

Area Weight: 10.0 %

☐ Uniform Spans

For **Unwrap** properties, see UV Unwrap Properties. For **Relax** properties, see UV Relax.

Select Error UVs

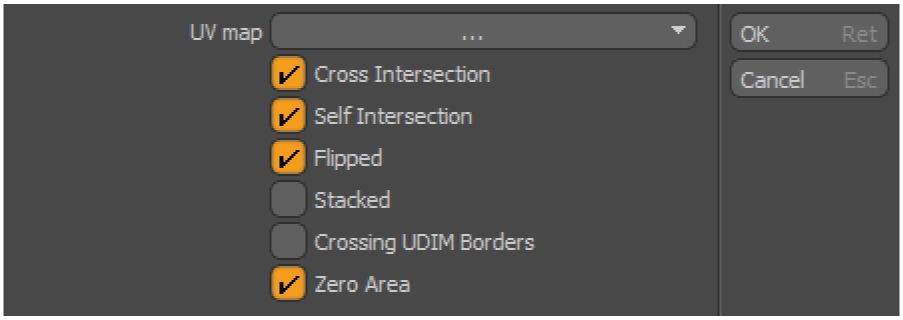
The **Select Error UVs** tool is useful for detecting polygon errors in UV maps . This tool provides a number of options to assist you in locating these errors.

Video: Watch this video for an overview.
- Overlapping Selection

You can access this tool in the **UVEdit** layout from the following locations:

- In the Toolbar, open the **UV** sub-tab, expand **Utilities**, and click **Select Error UVs**.
- In the **Texture** menu, click **UV Operators** > **Select Error UVs**.

Once activated, the **Select Error UVs** dialog displays.

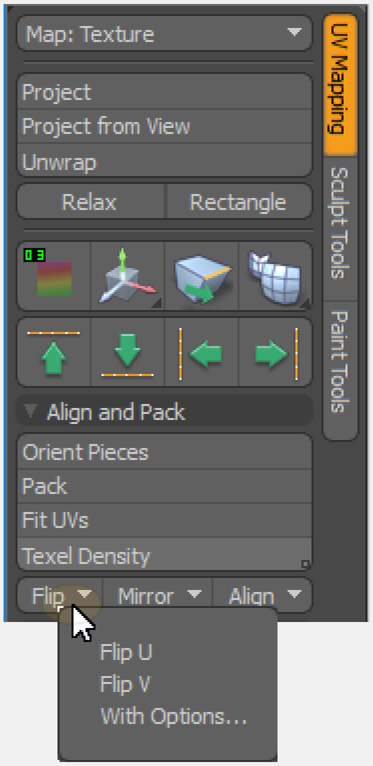


UV map ...
☑ Cross Intersection
☑ Self Intersection
☑ Flipped
☐ Stacked
☐ Crossing UDIM Borders
☑ Zero Area

Select Error UVs Options

The following options are available:

Option	Description
UV Map	Selects the UV map to locate overlapping polygons.
Cross Intersection	Selects the UV polygons overlapping or intersecting with other UVs.
Self Intersection	Detects twisted UV polygons that intersect themselves, which looks like a bow tie shape.
Flipped	The **Flipped** option refers to the direction in which UV texture coordinates are stored on the surface mesh for a particular face. This direction can be either clockwise or counter-clockwise. Selecting this option detects and selects the UV polygons using the clockwise order. Once found, you can repair selected flipped UVs using the Flip UV tools on the toolbar.

	 Map: Texture UV Mapping Sculpt Tools Paint Tools Project Project from View Unwrap Relax Rectangle Align and Pack Orient Pieces Pack Fit UVs Texel Density Flip Mirror Align Flip U Flip V With Options...
Stacked	Selects identical UV polygons which are located at the same location in UV space.
Crossing UDIM Borders	Detects UVs crossing UDIM borders.
Zero Area	Selects shrunken polygons in the UV space. These are shrunken to the line or point with a zero area size.

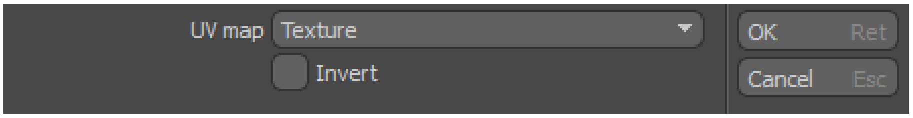
Select Unmapped Polygons

The **Select Unmapped Polygons** tool is useful for locating polygons which do not have UVs in a given UV map.

You can access this tool in the **UV** layout from the following locations:

- In the Toolbar, expand **Utilities**, and click **Select** > **Unmapped**.
- In the **Texture** menu, click **UV Operators** > **Select Unmapped Polygons**.

Once activated, the **Select Unmapped Polygons** dialog displays.


UV map
Texture
Invert
OK Ret
Cancel Esc

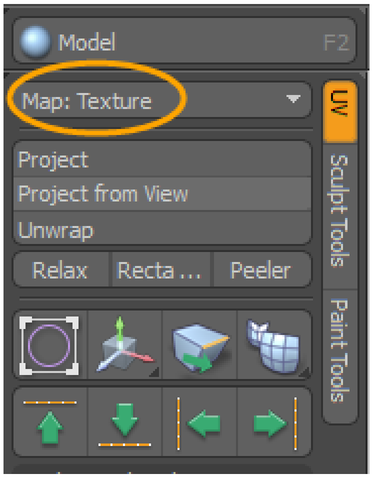
In addition to locating unmapped polygons, you can quickly identify UV distortion by enabling the **3D Viewport Properties** > **Active Meshes** > **Shading** > **Show UV Distortion** option. In addition, you can enable the **UV Viewport Properties**, **Visibility** > **Show Distortion** option to view distortion colors in the UV viewport. Use these options in your workflow to eliminate or reduce seams, and to minimize distortion from stretching. For more information, see UV Viewport and 3D (OpenGL) Viewport. The following image demonstrates a proposed workflow.

Using the Select Unmapped Polygons Tool

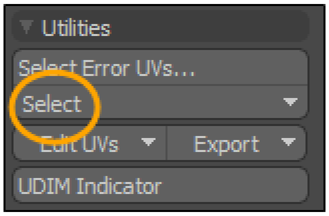
The following example walks you through the process of using the Select Unmapped Polygon tool.

To view unmapped polygons:

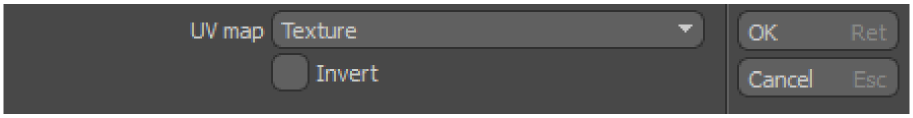
1. Open your scene in the **Model** layout.
2. With the mesh item selected, change to the **UV** layout, and then select the texture map from the **UV** tab in the Toolbox.


Model F2
Map: Texture
UV
Sculpt Tools
Paint Tools
Project
Project from View
Unwrap
Relax
Recta ...
Peeler

3. In the Toolbox, under **Utilities**, click **Select** > **Unmapped**. Alternatively, click **Texture** > **UV Operators** > **Select Unmapped Polygons** from the menu bar.


Utilities
Select Error UVs...
Select
Edit UVs
Export
UDIM Indicator

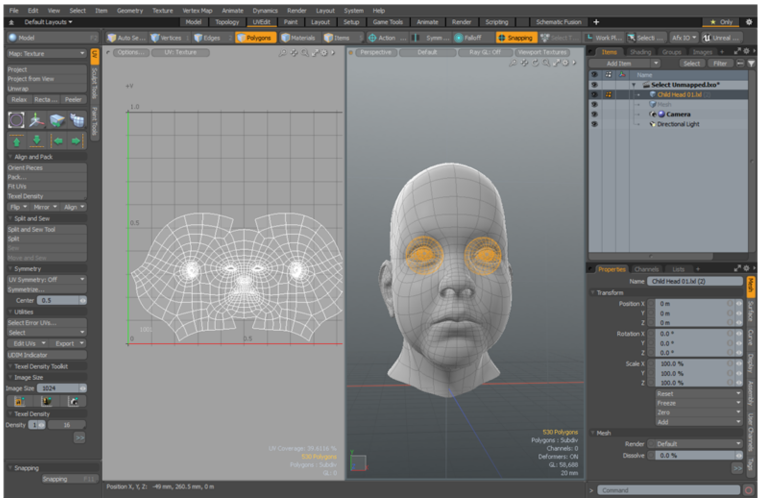
The **Select Unmapped Polygons** dialog displays.


UV map Texture

☐ Invert

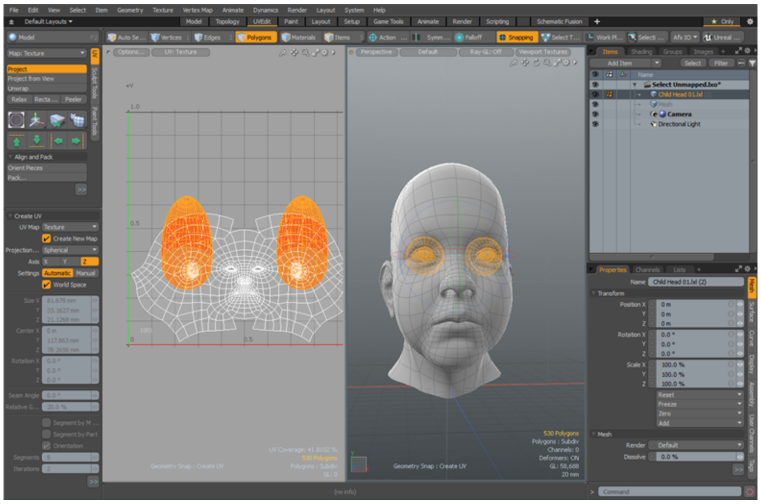
4. From the **UV map** dropdown menu, select your UV map.
5. Ensure **Invert** is disabled and click **OK**.

   The unmapped polygons are highlighted in the 3D Viewport. These polygons haven't been assigned UV coordinates.



6. In the Toolbox, **UV** tab, click **Project**.

   The unmapped polygons are added to the UV map. Apply the appropriate Create UV Tool properties.



Select Unmapped Polygons Options

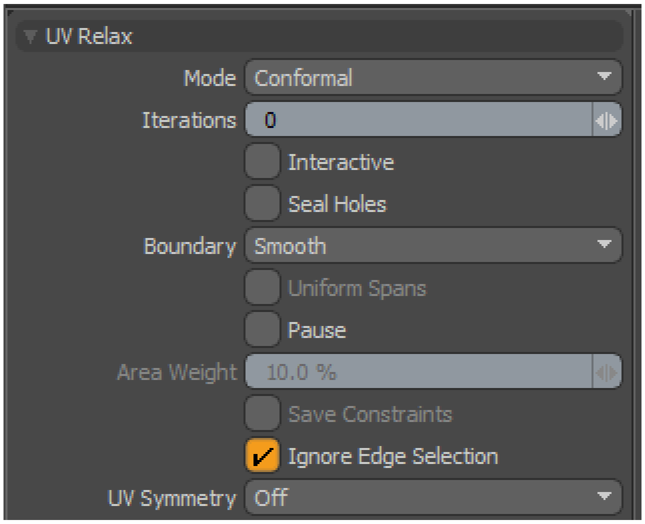
The following options are available:

Option	Description
UV Map	Selects the UV map to locate overlapping polygons.
Invert	When enabled, mapped polygons are selected.

UV Relax

The **UV Relax** tool can be used to refine an existing UV map in much the same way as the smooth tool is used to soften and regularize a model's surface.

Any geometry with a UV map can be relaxed. The only prerequisite is to make sure the target UV map is selected, and that only those polygons you intend to relax are visible. The **UV Relax** tool is found in the **UV** tools menu of the **Modo** and **Model** layouts, and the results of its usage can be viewed in real-time in the UV viewports. To activate, click the **UV Relax** tool button. To relax the UV map, click in the UV viewport and drag the cursor to increase/decrease the **Iterations** value. When the tool is active, additional options for refining the UV map are found in the tool's properties panel.



UV Relax

Mode: Conformal

Iterations: 0

☐ Interactive

☐ Seal Holes

Boundary: Smooth

☐ Uniform Spans

☐ Pause

Area Weight: 10.0 %

☐ Save Constraints

☑ Ignore Edge Selection

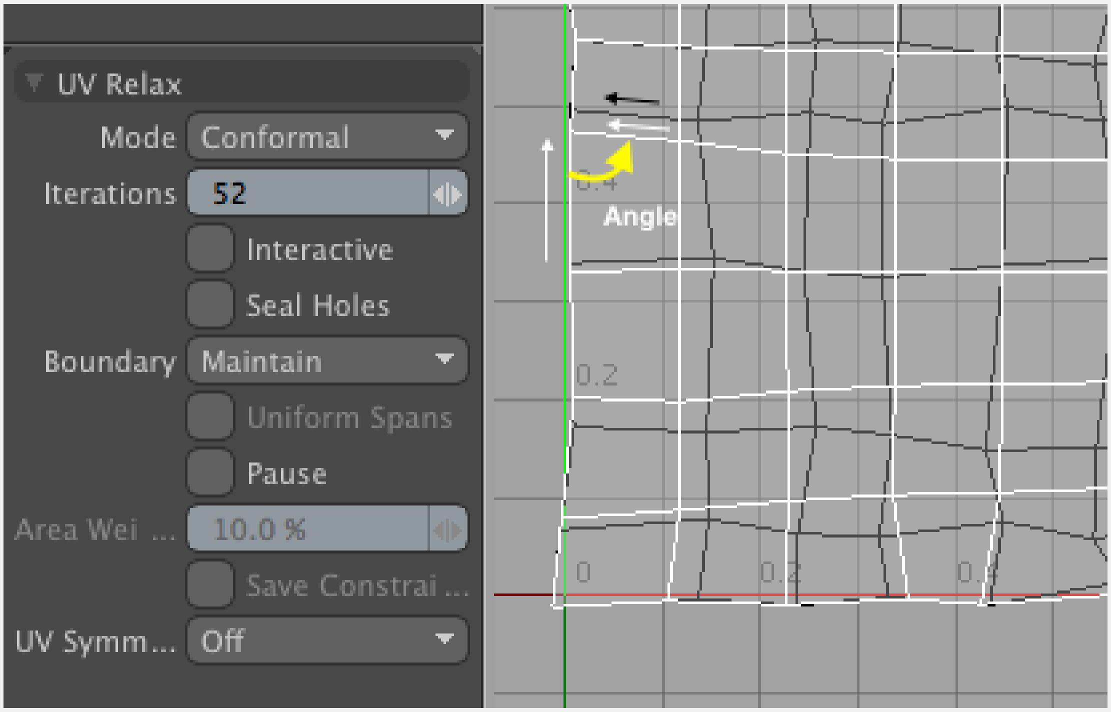
UV Symmetry: Off

The **UV Relax** tool properties panel.

UV Relax Properties

The following **UV Relax** options are available:

Options	Descriptions
UV Map	**Procedural modeling only.** Defines the name of the UV texture map used for the mesh operation.
Mode	Allows you to choose from various modes that control how the **Relax** tool acts upon the target UVs. • **Spring** - Generates a 2D dynamic simulation that makes all the edges in the UV map act like springs, each pulling on the others to find their equilibrium. This method doesn't deal with overlapping UVs very well. • **Conformal** - Relaxes the mesh using the ABF++ (Asynchronous Forward Bounding) algorithm to smooth out the UV map, as with the **Unwrap** tool. • **Adaptive** - An adaptive attempt to retain the original relative polygon size from the 3D mesh surface. • **Angle Based** - Uses the LSCM (Least Square Conformal Maps) algorithm to smooth out the UV map, as with the **Unwrap** tool.
Iterations	Determines the number of iterations for unchanged iterations. Increasing the **Iterations** relax/smooth the geometry; the degree of iteration necessary is largely dependent on the complexity of the target mesh.
Interactive	The interactive mode offers you an amazing way to edit a UV map using **Pins**. When **Interactive** is enabled, you can click over any vertex in the UV map, adding a blue square handle to it. When there is more than one handle in the map (and a sufficient number of **Iterations**) you can interactively drag around the UV map to reshape the layout, as if it were a sheet of rubber that twists and pulls between the your added **Pins**. Removing pins is done by **Ctrl/Cmd**+clicking over an existing pin. You can add as many handles as necessary, but fewer handles usually provide the best results. Use the **Save Constraints** checkbox if you want Modo to remember which vertices were set as pins each time the tool is used.
Seal Holes	If the target mesh isn't "water-tight" (such as a head model with open eye sockets) the **Relax** algorithm can sometimes have difficulty resolving these areas. The **Seal Holes** option creates a virtual polygon to close the holes and improves the unwrapping effect.
Boundary	Determines how to relax boundary UV vertex positions. The following options are available: **Smooth** - Locks boundaries off and relaxes both the boundary edges and the inside UV vertex positions.

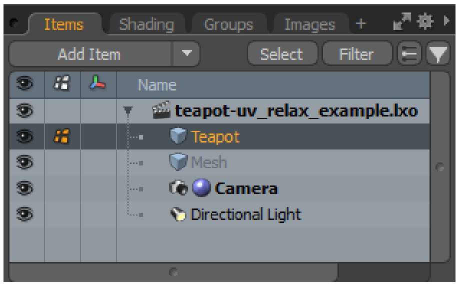
	Locked - Locks the outer boundary edge of all UV islands and only relaxes the inside areas of geometry; the outer edges act like pins constraining the connected edges. (A UV island is a connected group of polygons in a UV map.) **Maintain** - Ensures boundary points only move along the boundary to keep the angle between boundary edge and incoming inner edge. In other words, it ensures that the boundary points only are moved along the boundary, not towards the interior of the surface. For example, when starting with a symmetrical UV map, such as a model created by the **Unwrap Symmetrize** function, enabling the **Maintain** option retains the symmetrical properties of the map while relaxing.  UV Relax Mode: Conformal Iterations: 52 ☐ Interactive ☐ Seal Holes Boundary: Maintain ☐ Uniform Spans ☐ Pause Area Wei ...: 10.0 % ☐ Save Constrai ... UV Symm...: Off **Straighten** - When this option is enabled, boundary edge sections of the UV islands are straightened and the internal points are relaxed inside the new boundary. This is useful when aligning UVs for piped shapes or hairs.
Uniform Spans	Creates uniform spans for UV edges. This option is enabled when the **Boundary** has been set to **Straighten**.
Pause	Temporarily disables the iterative updates to allow pinning to be applied without continual updating of the relaxed results. Disable the **Pause** function to evaluate the relaxation with the updated settings.

Area Weight	Using the **Adaptive** relaxation mode, the **Area Weight** option determines a weighting applied to an area to allow it to resist changes from the relaxing of the UV vertices of the modified UV island.
Save Constraints	Using the **Interactive** option to pin vertices, the **Save Constraints** option saves which vertices have handles and keeps them intact for the next time the tool is applied.
Ignore Edge Selection	When this checkbox is toggled, the UV Relax operation ignores any currently selected edges. This option is enabled by default.
UV Symmetry	Sets the Symmetry state. The following options are available: • **Off** - Sets the Symmetry state to off. • **U** - Sets the Symmetry state to use the U axis. • **V** - Sets the Symmetry state to use the V axis.

Applying Different Boundary Options using the UV Relax Tool

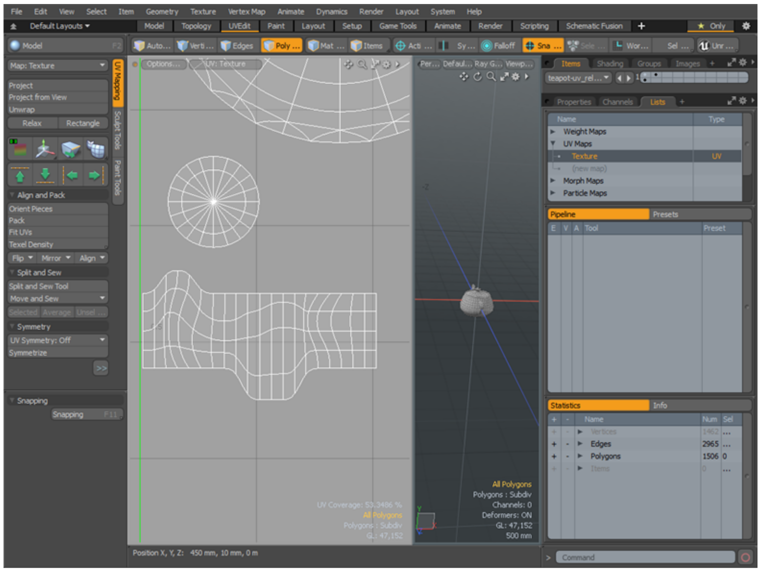
The following example demonstrates how to set different **Boundary** options to relax UVs for a selected UV island.

1. Download our example scenes and unpack the files.
2. Below the menu bar, click the **UV** tab on the switcher bar.
3. Open **teapot-uv_relax-example.lxo**, which has a UV Map to modify.
4. In the **Items Lists** on the right panel, expand the **teapot-uv_relax-example.lxo** item and select **Teapot**.

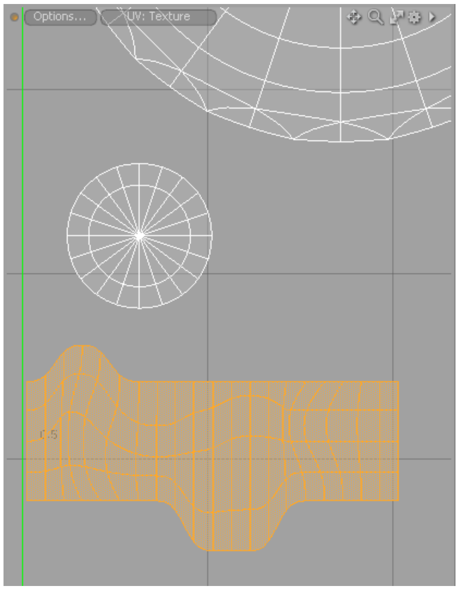


5. In the **Lists** tab, expand **UV Maps**, and select **Texture**.

   The UV map is displayed in the UV Viewport. Alternatively, you can view the UV map in the **Texture** > **Open UV Editor** window.


File Edit View Select Item Geometry Texture Vertex Map Animate Dynamics Render Layout System Help
Default Layouts
Model Topology UVEdit Paint Layout Setup Game Tools Animate Render Scripting Schematic Fusion
Map: Texture
Project
Project from View
Unwrap
Relax
Rectangle
Align and Pack
Orient Pieces
Pack
Fit UVs
Texel Density
Flip
Mirror
Align
Split and Sew
Split and Sew Tool
Move and Sew
Symmetry
UV Symmetry: Off
Symmetrize
Snapping
Options...
UV: Texture
Items
Shading
Groups
Images
Properties
Channels
Lists
Name
Type
Weight Maps
UV Maps
Texture
UV
Morph Maps
Particle Maps
Pipeline
Presets
Tool
Preset
Statistics
Info
Edges
Polygons
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 47,152
500 mm
Position X, Y, Z: 450 mm, 10 mm, 0 m
Command

6. Select the Polygon Selection Mode and draw a lasso around the UV map to select a UV island.

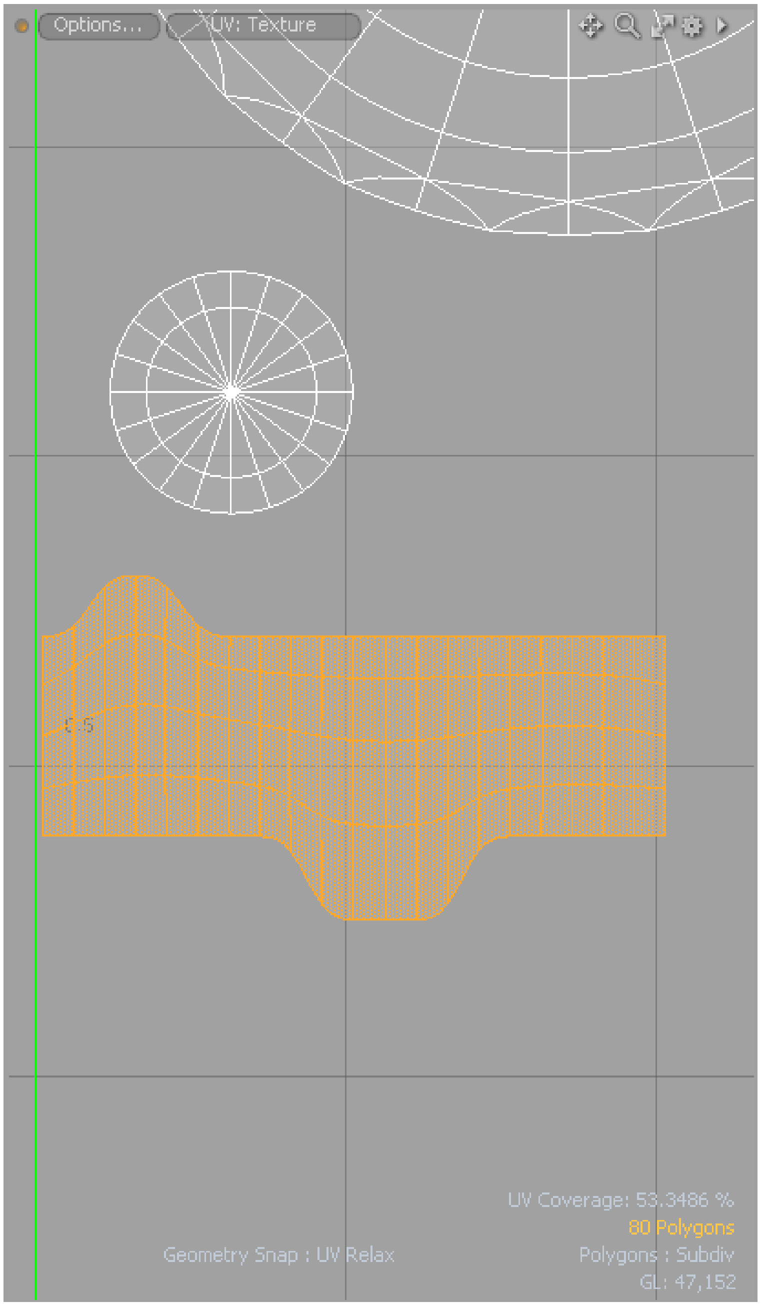

Options...
UV: Texture
0.5

7. On the Toolbox, open the **UV Mapping** tab, and click **Relax**.

   UV Relax properties are displayed at the bottom of the Toolbox.

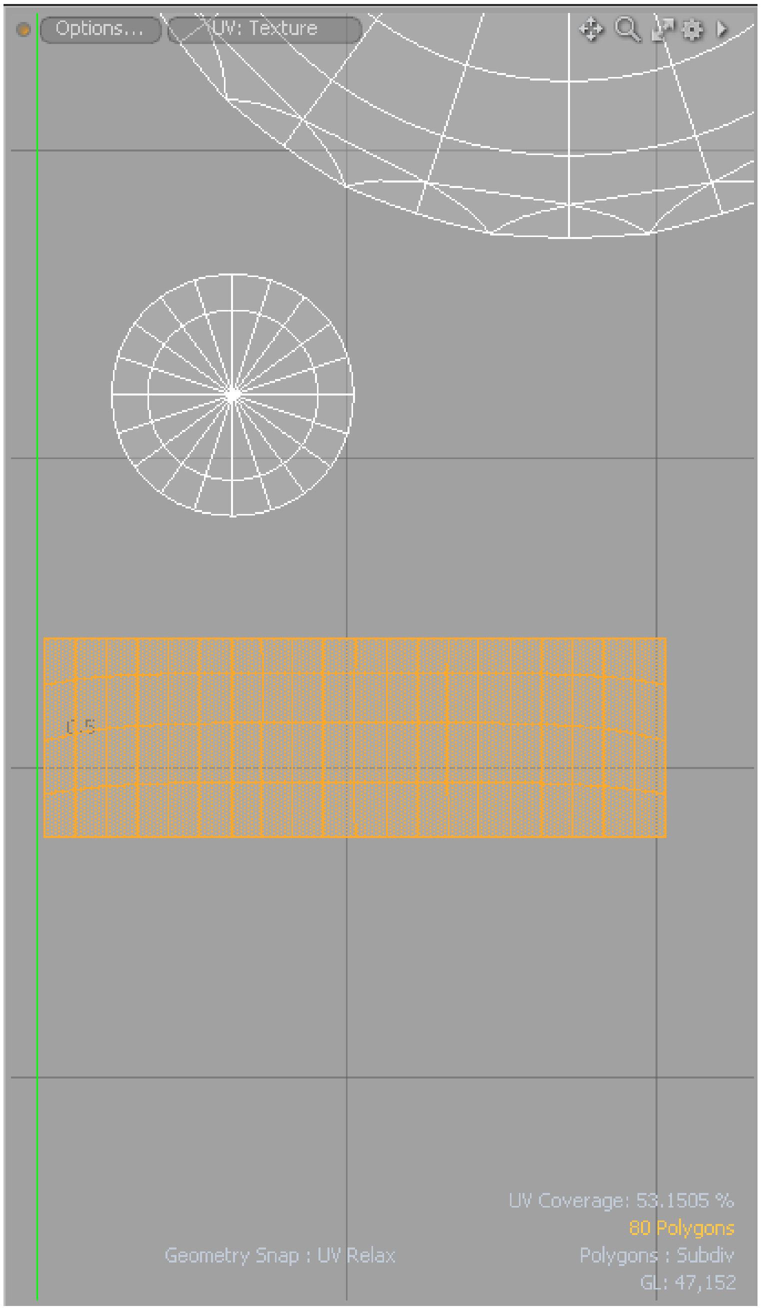
8. Set **Boundary** to **Locked**.

   Only the UVs inside the selected UV island are relaxed within the locked boundary edges.


Options...
UV: Texture
UV Coverage: 53.3486 %
80 Polygons
Geometry Snap : UV Relax
Polygons : Subdiv
GL: 47,152

9. Set **Boundary** to **Straighten** and set the **Iterations** to **9**.

   The boundary edges of the UV island are straightened and the internal points are relaxed inside a new boundary.


Options...
UV: Texture
0.5
UV Coverage: 53.1505 %
80 Polygons
Geometry Snap : UV Relax
Polygons : Subdiv
GL: 47,152

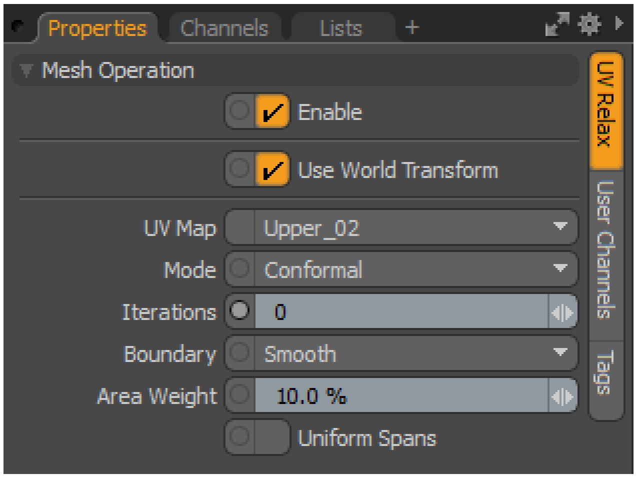
10. Experiment with the other **Boundary** options and **UV Relax** options.

Procedural UV Relax

The **UV Relax** mesh operation is the procedural version of the **UV Relax** tool.

The **UV Relax** mesh operation can be used to refine an existing UV map in a similar way as the **Smooth tool**, which is used to soften and regularize a model's surface.

The **UV Relax** mesh operation properties are the same as the properties of the **UV Relax** tool.


Properties
Channels
Lists
+
UV Relax
User Channels
Tags

Mesh Operation

☑ Enable

☑ Use World Transform

UV Map: Upper_02

Mode: Conformal

Iterations: 0

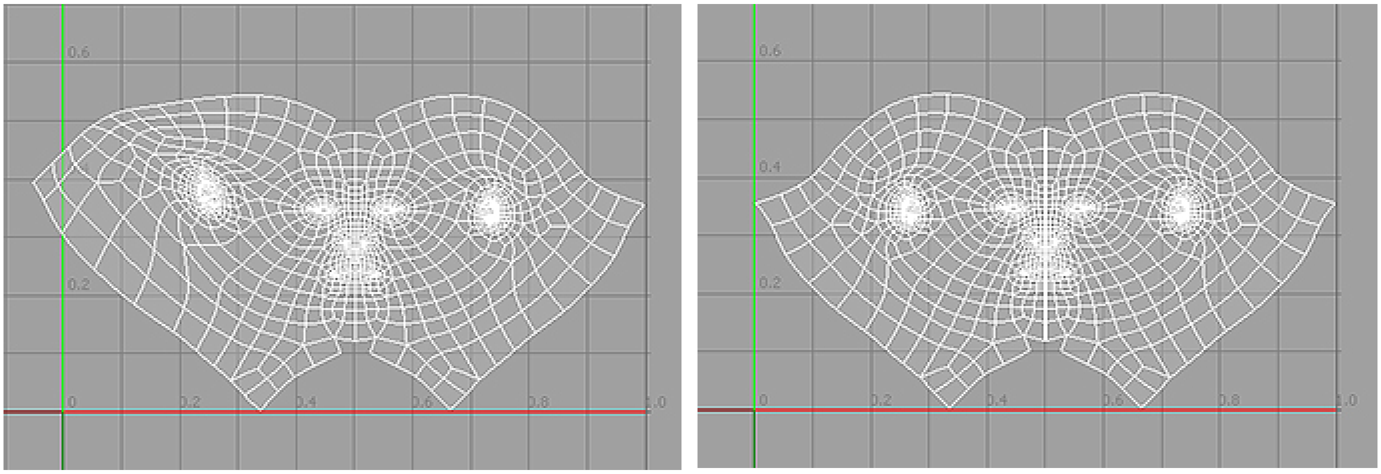
Boundary: Smooth

Area Weight: 10.0 %

☐ Uniform Spans

For **UV Relax** properties, see UV Relax Properties.

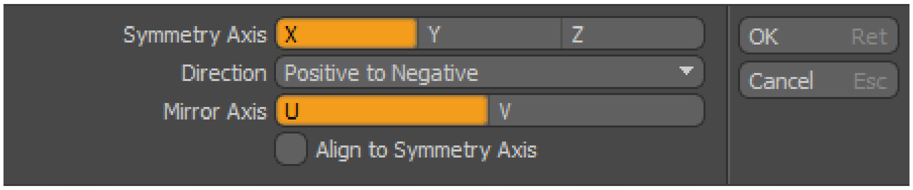
Symmetrize UVs


0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0

Before

After

With the **Symmetrize UVs** function Modo mirrors UVs of polygons from one side to another in UV space. The pairs of polygons to mirror must be symmetrical in 3D space. In the **UVedit** layout of the interface, click the **Symmetrize UVs** button to open the dialog.


Symmetry Axis
X
Y
Z
Direction
Positive to Negative
Mirror Axis
U
V
Align to Symmetry Axis
OK Ret
Cancel Esc

Symmetry Axis: Indicates the axis for symmetry in 3D space. Select either **X**, **Y**, or **Z**. The 3D mesh must be symmetric on this axis.

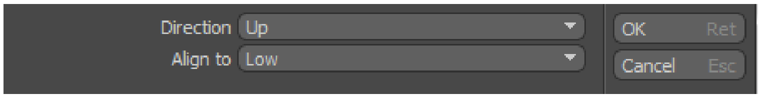
Direction: Indicates the mirroring direction in UV space. Selecting **Positive to Negative** copies UVs from the right side to the left side.

Mirror Axis: Specifies the mirror axis in UV space. Modo mirrors the specified component: **U** or **V**.

Align to Symmetry Axis: When enabled, rotates UV islands to align to the specified **Symmetry Axis**.

Align UV

The **Align** tool on the **Texture** > **UV Operations** menu takes the bounding box of the selected UV vertices and moves all the vertices in the specified **Direction** to the line in the box indicated by the **Align To** option.

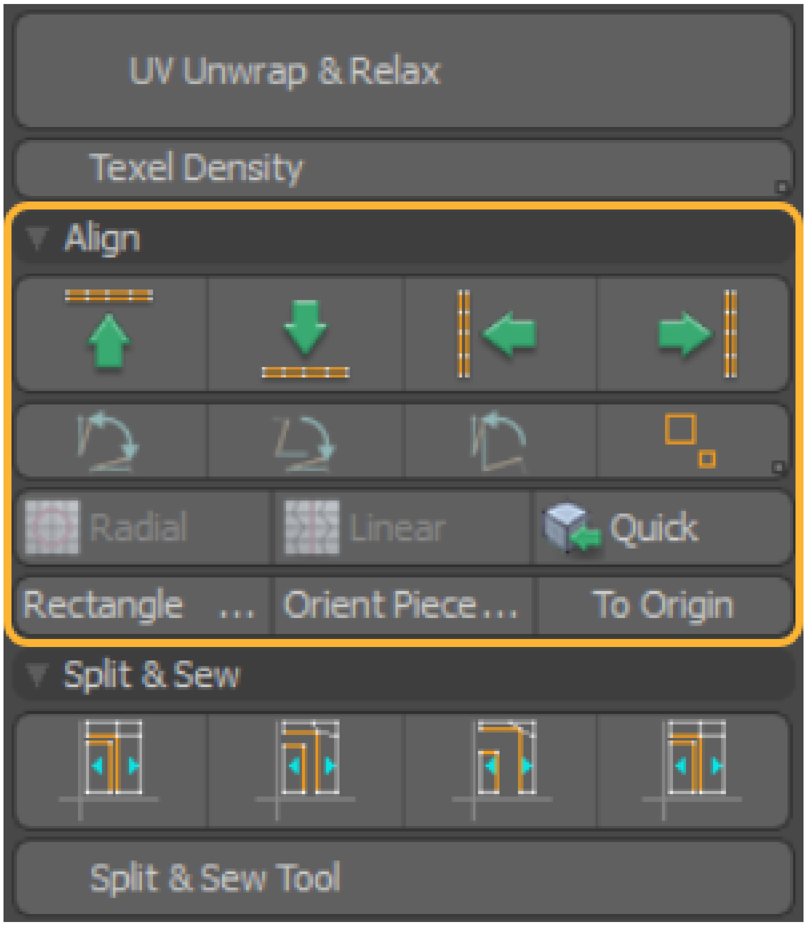

Direction
Up
Align to
Low
OK Ret
Cancel Esc

The options in the **Align UVs** dialog include:

Direction	Moves the vertices in the specified direction. Both **Up** and **Down** move the vertices in the **V** direction. Both **Left** and **Right** move vertices in the **U** direction.
Align to	Determines how to align the vertices in relation to the bounding box. The following options area available: • **Low** is the minimum value of the box. • **Average** is the average of all the values. • **Center** is the center of the bounding box. • **High** is the max value of the box.

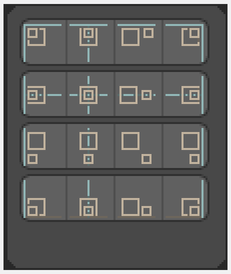
Other Useful Alignment Tools

On the left panel, under the **UV** tab, displays tools to quickly align UVs up, down, left, or right.


UV Unwrap & Relax
Texel Density
Align
Radial
Linear
Quick
Rectangle ...
Orient Piece...
To Origin
Split & Sew
Split & Sew Tool

Option	Icon	Description
Up	⬆	Flattens selection to the farthest up vertex point position. **Tip:** Press **Alt** and click to align multiple edge selections.
Down	⬇	Flattens selection to the farthest down vertex point position.

		Tip: Press **Alt** and click to align multiple edge selections.
Left	←	Flattens selection to the farthest left vertex point position. **Tip:** Press **Alt** and click to align multiple edge selections.
Right	→	Flattens selection to the farthest right vertex point position. **Tip:** Press **Alt** and click to align multiple edge selections.
Quick Align (Auto)		Rotate the UV island so that your first selected edge or the line between your two selected vertices is aligned to the nearest grid axis. (A UV island is a connected group of polygons in a UV map.)
Quick Align (Horizontal)		Rotate the UV island so that your first selected edge or the line between your two selected vertices is aligned to the horizontal axis. (A UV island is a connected group of polygons in a UV map.)
Auto Align (Vertical)		Rotate the UV island so that your first selected edge or the line between your two selected vertices is aligned to the vertical axis. (A UV island is a connected group of polygons in a UV map.)
Island Align	□	Moves selected UVs so that their extremities are aligned vertically (along V axis) and/or horizontally (along U axis). Each axis (U or V) has options for minimum, maximum, and center. The following **Island Align** options are available:

		
Radial Align		**Radial Align** moves selected UV positions along a circle or polygon. When the tool is activated, Modo calculates the center and radius of selected elements based on the location of the first click in the viewport. Before **Radial Align** is activated. After **Radial Align** is activated. **Note:** For more information on the tool properties available within the **Radial Align** tool, see Radial Align.
Linear Align		**Linear Align** aligns selected UV positions along a line. Before **Linear Align** is activated. After **Linear Align** is activated. **Note:** For more information on the tool properties available within the **Linear Align** tool, see Linear Align Properties.
Quick Align		**Quick Align** allows you to manipulate the alignment of UV sections by interacting with them in the 3D viewport. **Note:** For more information on the tool properties available within the **Quick Align** tool, see Quick Align Properties.
Rectangle		**Rectangle** allows you to straighten quadrangle sections of UVs into rectangular sections.

		Before **Rectangle** is activated. After **Rectangle** is activated.
Orient Pieces		**Orient Pieces** automates the process of organizing UV islands. **Orient Pieces** has three modes: • **Auto** - automatically aligns your UV sections. • **Horizontal** - aligns your UV sections horizontally. • **Vertical** - aligns your UV sections vertically. UV sections before aligning Auto alignment Horizontal alignment Vertical alignment
To Origin		Moves UV sections to the origin of a chosen UDIM. The UDIM whose origin the UV section should be aligned to can be set in the **Move UVs to Origin** window that opens when the tool is activated.

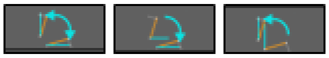
Aligning a UV Island

Modo provides a number of UV alignment tools, which allows you to rotate a UV Island by making a selection of vertices or edges and selecting the appropriate rotate tool. You can align the UV island to the nearest grid axis, horizontal axis, or vertical axis. (A UV island is a connected group of polygons in a UV map.)

To apply an align UV island tool:

1. Open the **UV** layout and select a mesh item from the **Items** list on the right panel.
2. From the Toolbox on the left panel, click **Map** and select **Texture** or any other UV Map you have previously created.

   The UV Island displays in the UV Viewport.
3. Enable either the **Vertices** or the **Edges** selection mode and make a selection.
4. In the Toolbox on the left panel, open the **UV** sub-tab, expand the **Align** tools, and click one of the following icons.

   The UV island is aligned.

Aligning a Selection of UV Islands

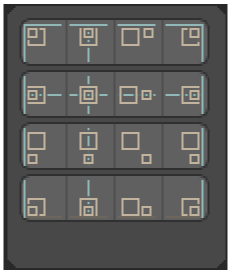
Modo provides a simple workflow to align a selection of UV islands at once.

To apply alignment to a selection of UV Islands:

1. Open the **UV** layout
2. From the **Items** list, on the right panel, select two mesh items.

   Two UV Islands display in the UV Viewport.
3. Enable the **Polygons** selection mode and make a selection.
4. In the Toolbox on the left panel, open the **UV** sub-tab, expand the **Align** tools, and click the **Island Align** icon.

   The dropdown menu displays, listing a number of different alignment icons.

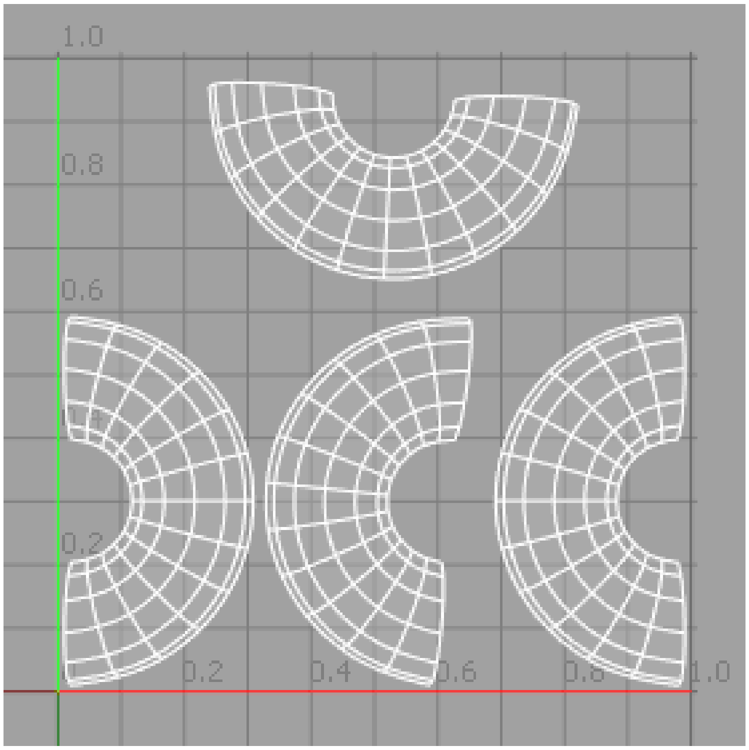


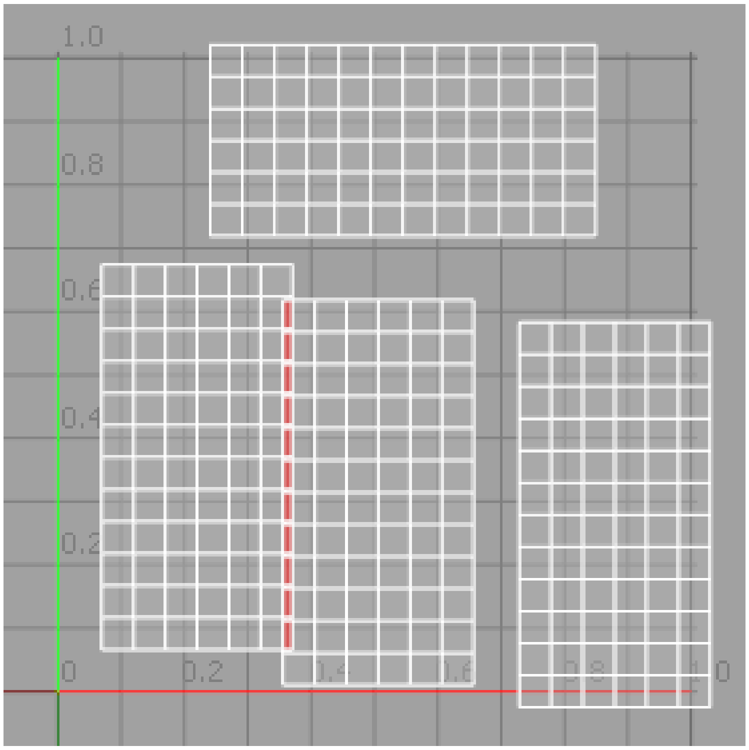
5. Click one of the Island Align tools.

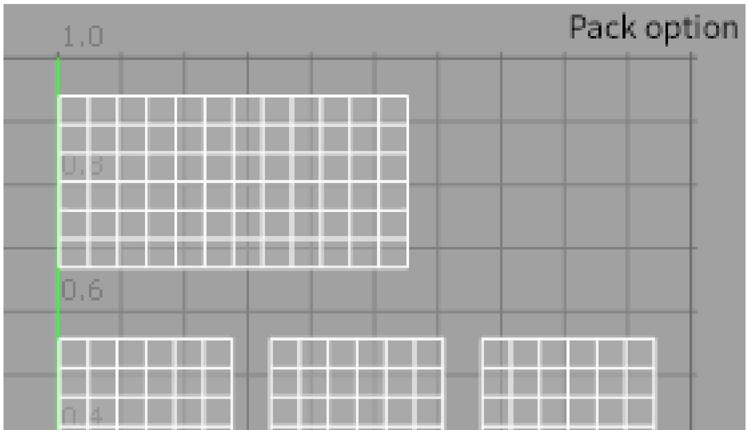
   The selection of UV Islands for your mesh items are repositioned to the specified location.

Rectangle

The **Rectangle** tool straightens quadrangle selections of UVs into rectangular sections. This tool has options to automatically position and regularize the spans into even grids.


1.0
0.8
0.6
0.2
0.2
0.4
0.6
0.8
1.0


1.0
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0

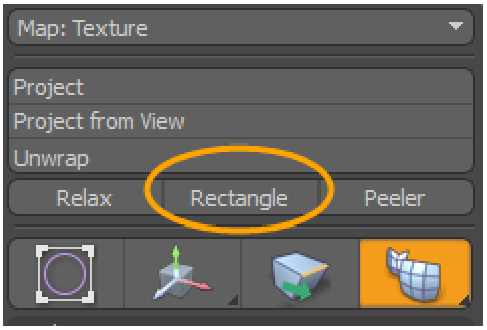

Pack option
1.0
0.8
0.6
0.4

Note: You must apply the **Rectangle** tool to one or more sections of full quadrangles with an equal number of spans along the length in both the vertical and horizontal directions.

Activating the Rectangle Tool

Direct Modeling:

- Using the **UV** layout, on the left panel, click **Rectangle**.



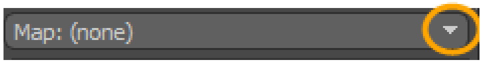
- From the main menu, click **Texture** > **UV Operators** > **Rectangle...**.

Procedural Modeling:

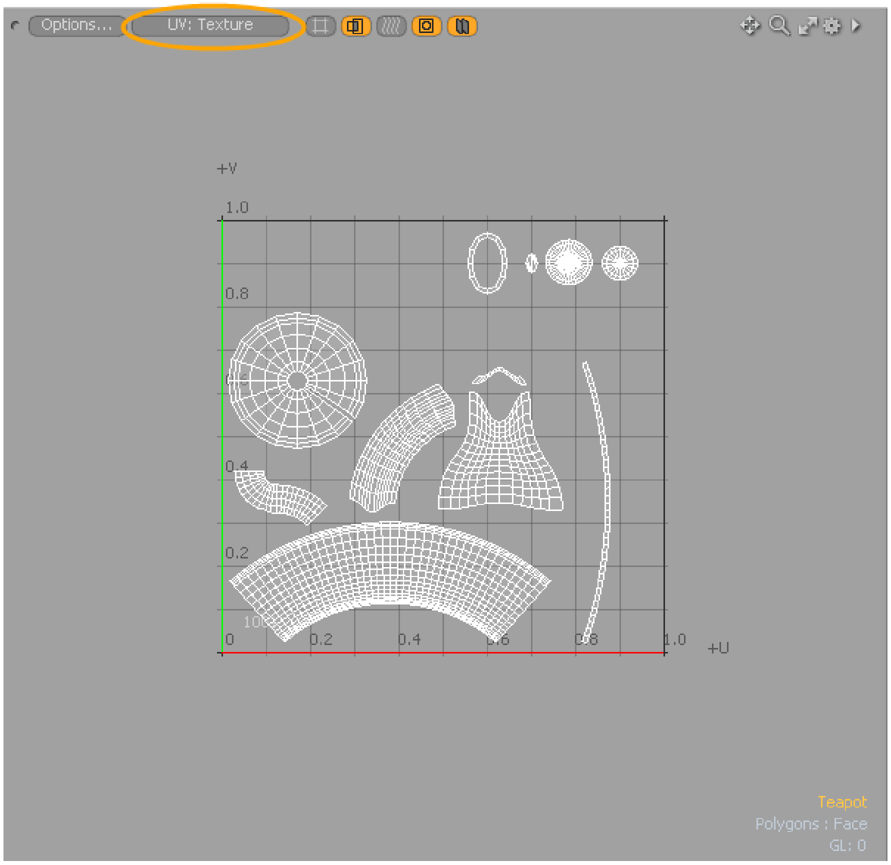
- In the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **UV** > **UV Rectangle**.
- In the **Schematic** view, click **Add...**, and double-click **Mesh Operations** > **UV** > **UV Rectangle**.

Direct Modeling Rectangle Tool Example

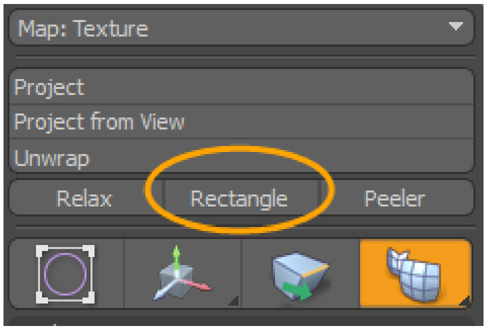
1. Download our example scenes and unpack the files.
2. Open **UV Rectangle mop.lxo** using the **UV** layout.

   The Teapot mesh item is selected in the 3D viewport.
3. On the top left-side of the interface, click the  **Map: (none)** dropdown menu and select **Texture**.

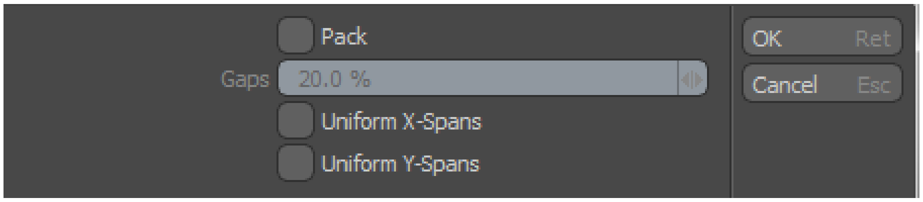
   The UV map is displayed in the UV view.


Options...
UV: Texture
+V
1.0
0.8
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
+U
Teapot
Polygons : Face
GL: 0

4. On the left panel, click **Rectangle**.

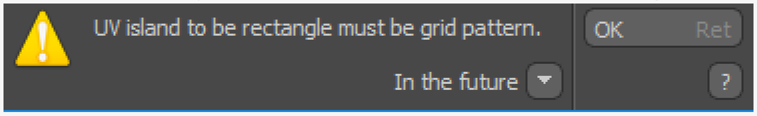

Map: Texture
Project
Project from View
Unwrap
Relax
Rectangle
Peeler

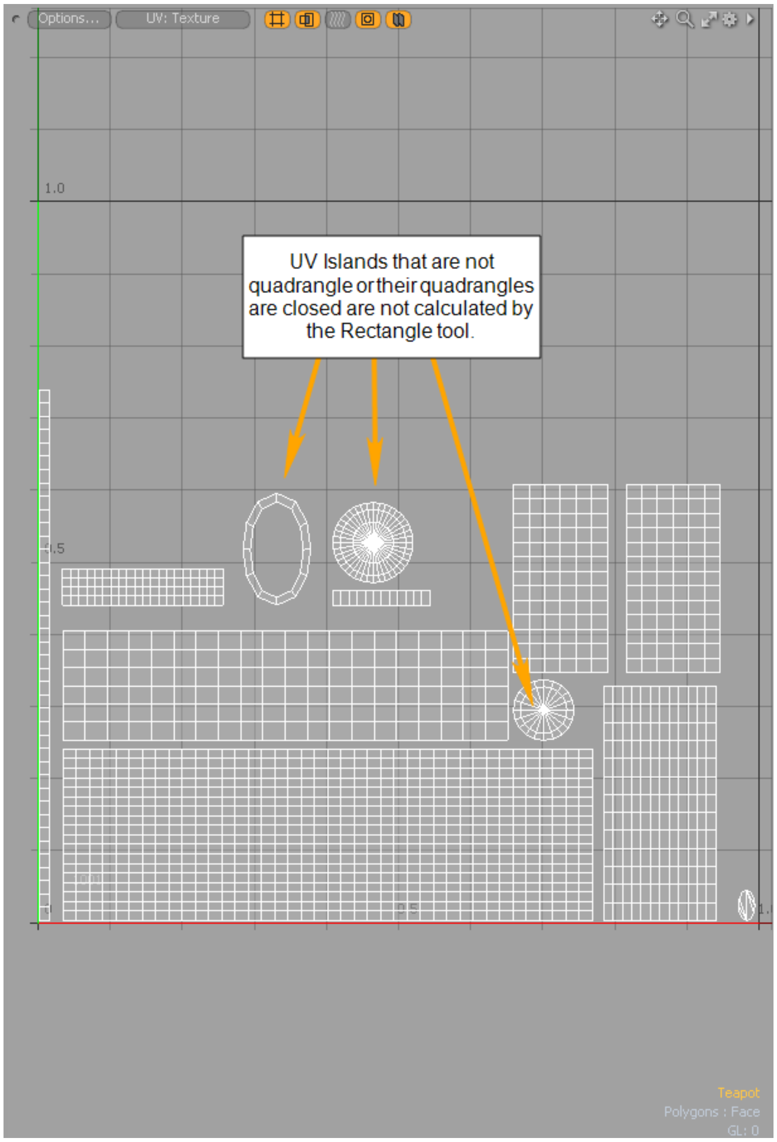
5. In the **UV Rectangle** dialog, enable all of the Rectangle Properties, and click **OK**.


☐ Pack
Gaps 20.0 %
☐ Uniform X-Spans
☐ Uniform Y-Spans
OK Ret
Cancel Esc

The UV map is updated in the UV view displaying the quadrangle selections of UVs straightened into rectangular sections.

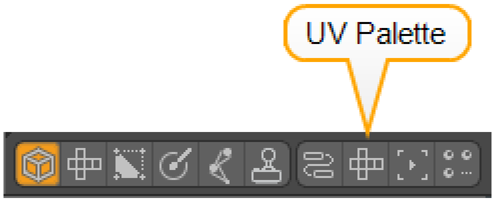
Note: For those UV islands that are not quadrangle or their quadrangles are closed, they are displayed as is in the UV view and you are prompted with the following error message. Click **OK** to dismiss the dialog.


UV island to be rectangle must be grid pattern.
OK
Ret
In the future
?


UV Islands that are not quadrangle or their quadrangles are closed are not calculated by the Rectangle tool.
Options...
UV: Texture
1.0
0.5
0.5
Teapot
Polygons : Face
GL: 0

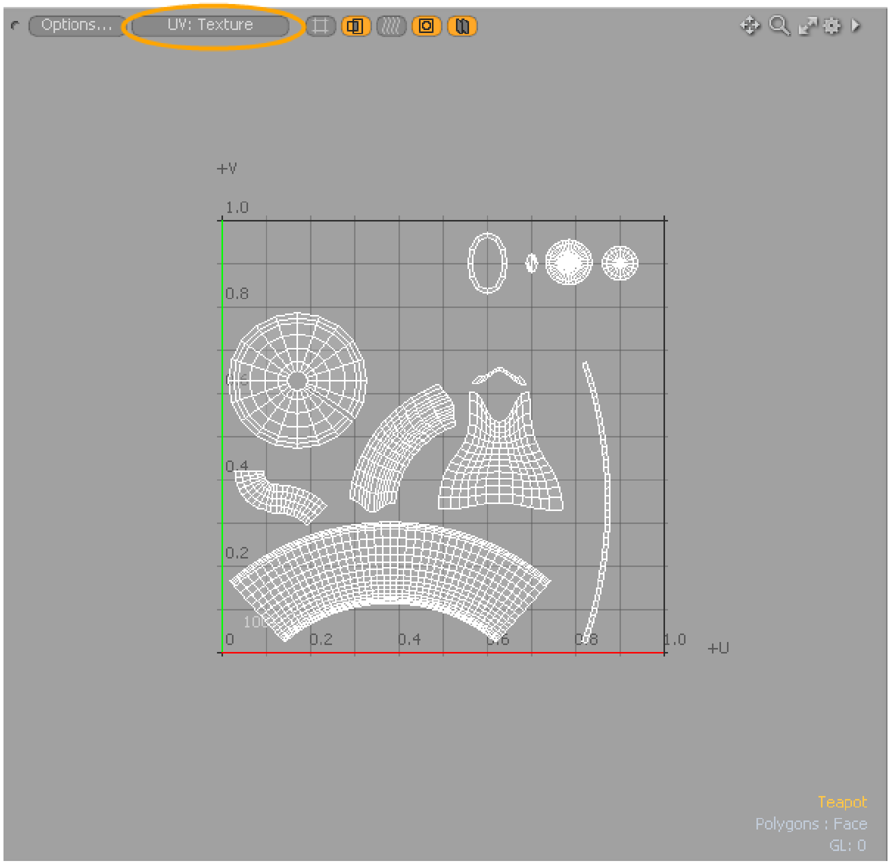
Procedural Modeling Rectangle Tool Example

1. Download our example scenes and unpack the files.
2. Open **UV Rectangle mop.lxo**.
3. On the top left-side of the interface, click the **UV Palette** button.



4. In the UV viewport, click the **UV: (none)** button and select **Texture**.

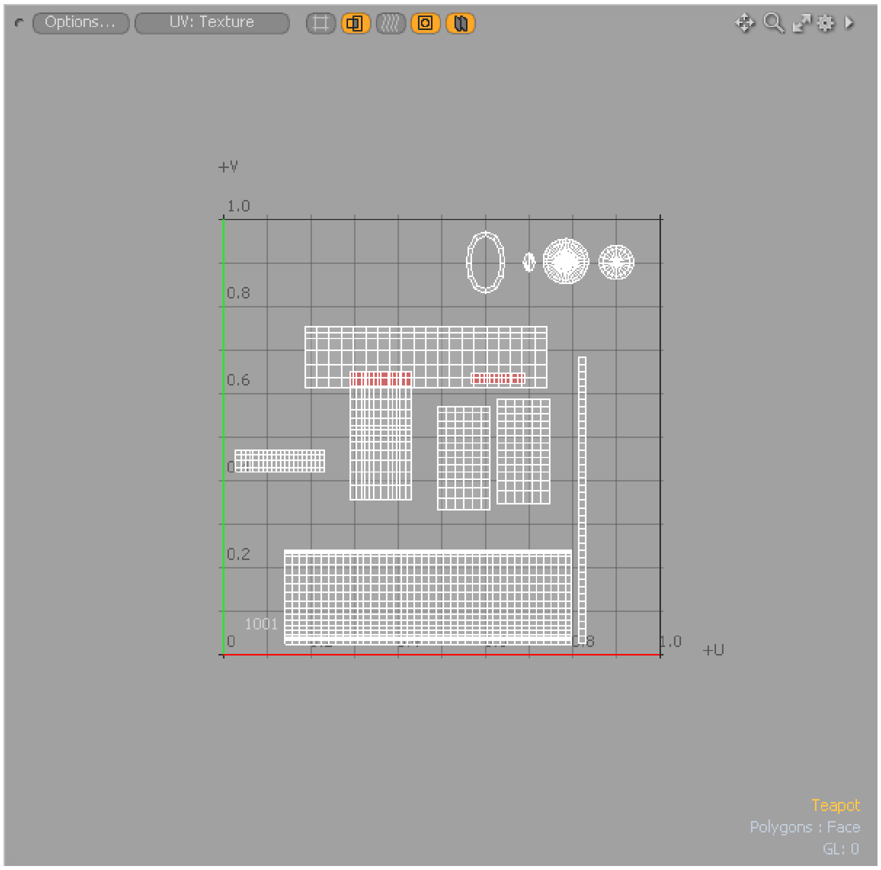
   The UV map is displayed in the UV view.


Options...
UV: Texture
+V
1.0
0.8
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
+U
Teapot
Polygons : Face
GL: 0

5. On the right panel, in the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **UV** > **UV Rectangle**.

   The UV map is updated in the UV view displaying the quadrangle selections of UVs straightened into rectangular sections.

> **Note:** For those UV islands which are not quadrangle or their quadrangles are closed, they are displayed as is in the UV view.



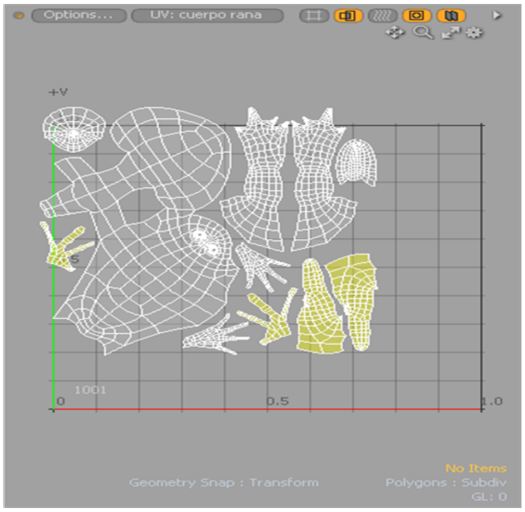
6. Set the appropriate Rectangle Properties to modify the UV map.

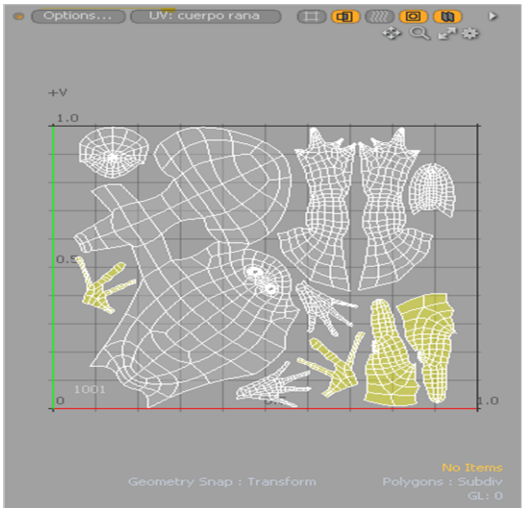
Rectangle Properties

- **Enable** - Toggles the Rectangle tool off and on. This option is only available for a procedural modeling mesh operation.
- **Use World Transform** - When enabled, the mesh operation maintains the world space on the input layer. This option is only available for a procedural modeling mesh operation.
- **Name** - Specifies the output UV texture name. This option is only available for a procedural modeling mesh operation.
- **Pack** - Fits the resulting rectangular UV sections automatically within the target UV space without overlapping. Modo does this by scaling the islands. When enabled, the **Gaps** value defines the amount of spacing between UV islands. The third example, above, enables **Pack** with the default **Gaps** value.
- **Gaps** - Determines the amount of spacing added between UV islands when **Pack** is enabled. The higher the **Gaps** value, the greater the distance between islands.
- **Uniform X-Spans/Uniform Y-Spans** - When enabled, Modo regularizes irregular spacing of sections into even spans. Generally, based on the target surface's topology, when disabled, Modo retains relative span spacing on the resulting section to reduce distortion on resulting textures applied to the surface.

Fit UVs

The **Fit UVs** tool scales the currently-selected UV data to fill the normalized 0 to 1 UV space. This is a very useful tool for optimizing UVs.

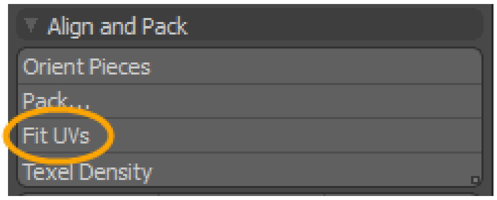

Options...
UV: cuerpo rana
+V
0.5
1001
0
0.5
1.0
No Items
Geometry Snap : Transform
Polygons : Subdiv
GL: 0


Options...
UV: cuerpo rana
+V
1.0
0.5
1001
0
1.0
No Items
Geometry Snap : Transform
Polygons : Subdiv
GL: 0

Activating the Fit UVs Tool

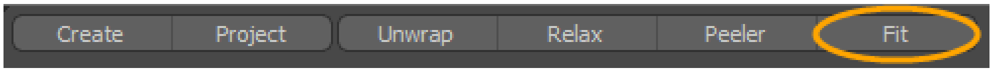
Direct Modeling:

- With the **UV** layout open, on the left panel, expand **Align and Pack**, and do one of the following:
  - click **Fit UVs** to apply the default properties. If these properties are changed, Modo preserves your settings and applies them the next time you use the tool.


Align and Pack
Orient Pieces
Pack...
Fit UVs
Texel Density

  - press **Ctrl**/**Cmd** + click **Fit UVs...** to open the properties dialog box and customize your settings.
  - press **Shift** + click **Fit UVs (Islands)** to fit UV islands into 0-1 space of UDIM 1001 based on the size of the bounding box of each UV island (polygons overlap on top of each other).
  - press **Alt** + click **Fit UVs (Polygons)** to fit UV islands into 0-1 space of UDIM 1001 based on the size of bounding box of every polygon individually (polygons overlap on top of each other).

- Alternatively, from the menu bar, click **Texture** > **Open UV Editor**, and click **Fit** to open the properties.



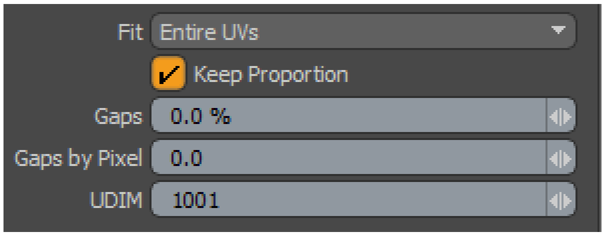
Procedural Modeling:

- In the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **UV** > **UV Fit**.
- In the **Schematic** view, click **Add...**, and double-click **Mesh Operations** > **UV** > **UV Fit**.

Direct Modeling Fit UVs Example

1. With the **UV** layout open, on the right panel, select the mesh item in the **Items** tab you want to work with.
2. On the left panel, expand **Align and Pack**, and press **Ctrl/Cmd** + click **Fit UVs...**

   This opens the **Fit UVs** dialog. For more information, see Fit UVs / UV Fit Properties.



Fit Entire UVs

☑ Keep Proportion

Gaps 0.0 %

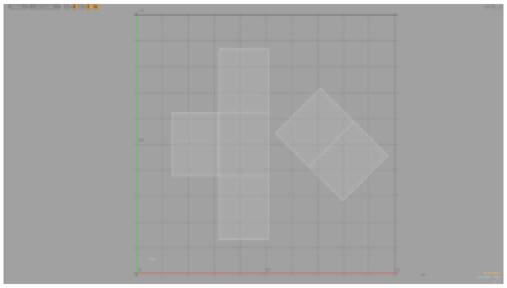
Gaps by Pixel 0.0

UDIM 1001

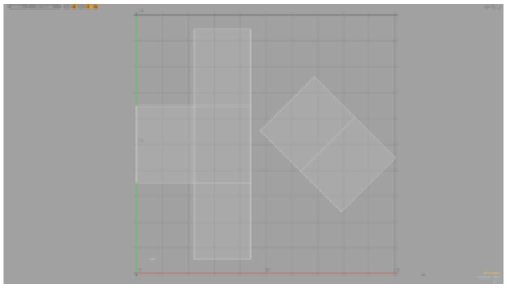
3. Edit the properties and click **OK**.

   The following is an example of a cube using the default Fit UVs properties.

Before:



After:



Procedural UV Fit Example

1. Under the main menu, click **UV** to open the UV layout.
2. On the right panel, open the **Mesh Ops** tab, and select the mesh item you want to work with.
3. On the lower right panel, open the **Lists** tab, expand **UV Maps**, and select a UV map.

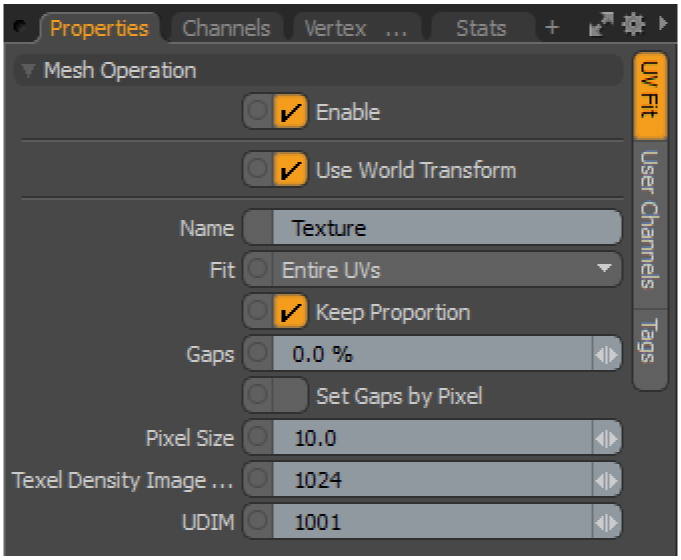
   The UV data is displayed in the UV viewport.

Tip: If you do not have a UV map created for your mesh item, on the left panel, click the dropdown arrow on **Map: (none)**, and select **New**. For more information, see Working with UV Maps.

Map: (none)

4. Click **Add Operator** and double-click **Mesh Operations** > **UV** > **UV Fit**.
5. Open the **Properties** tab on the right panel and edit the properties.

   For more information, see Fit UVs / UV Fit Properties.



Mesh Operation

☑ Enable

☑ Use World Transform

Name: Texture

Fit: Entire UVs

☑ Keep Proportion

Gaps: 0.0 %

☐ Set Gaps by Pixel

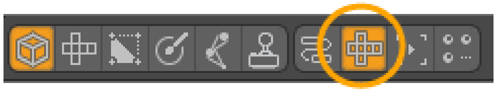
Pixel Size: 10.0

Texel Density Image ...: 1024

UDIM: 1001

Schematic UV Fit Example

1. Under the main menu, click **Setup** to open the Setup layout.
2. On the top left corner of the interface, click UV View icon.

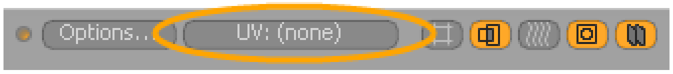


The UV viewport opens as a separate palette.

3. On the right panel, open the **Mesh Ops** tab, and drag-and-drop the mesh item you want to work with onto the Schematic viewport.

   The selected node is added to the Schematic viewport.

4. If the UV map is not displayed in the UV View, click **UV:(none)**, and select the UV map from the drop-down menu.



5. In the Setup Schematic viewport, click **Add...**, and double-click **Mesh Operations** > **UV**> **UV Fit**.

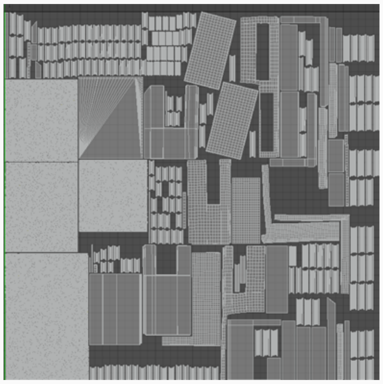
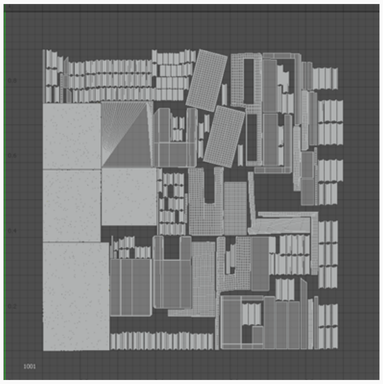
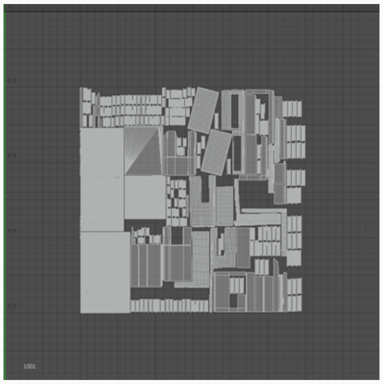
6. Open the **Properties** tab on the right panel, enter the name of your UV map, and edit the properties.

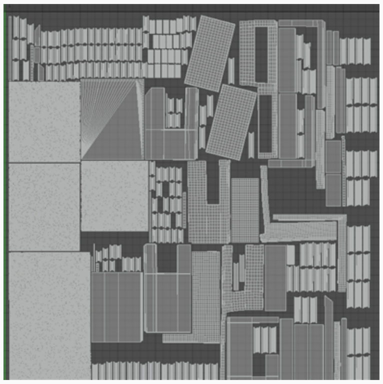
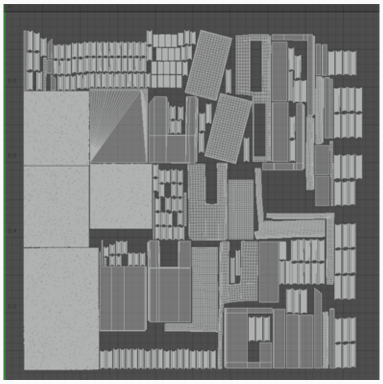
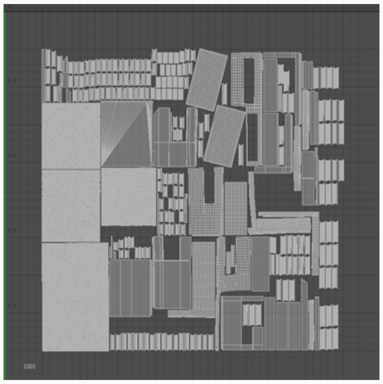
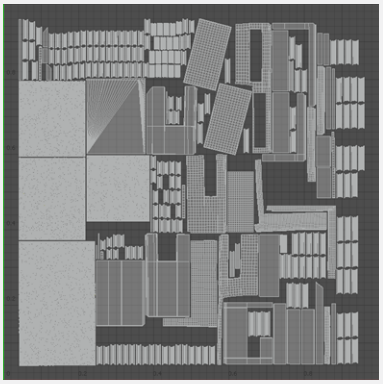
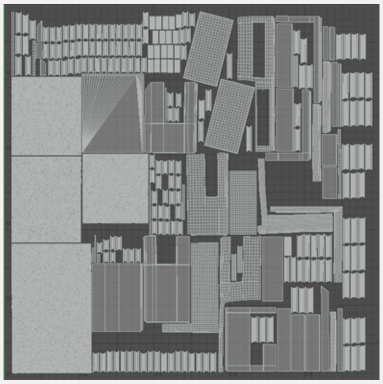
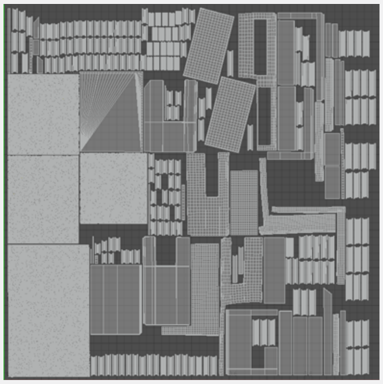
   For more information, see Fit UVs / UV Fit Properties.

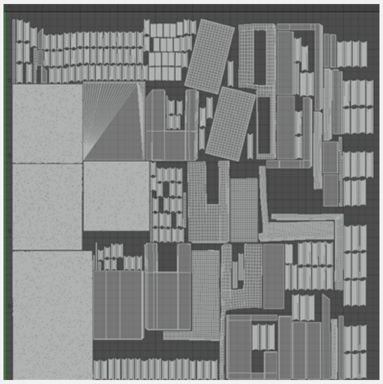
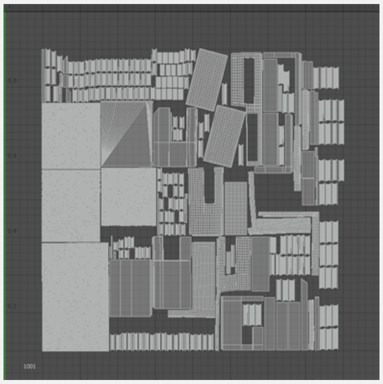
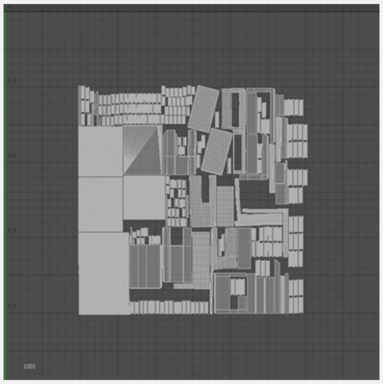
   The UV data in the UV View palette is updated.

Fit UVs / UV Fit Properties

Enable	Toggles the mesh operation off and on. This option is only available for a procedural modeling mesh operation.
Use World Transform	When enabled, the mesh operation maintains the world space on the input layer. This option is only available for a procedural modeling mesh operation.
Name	Specifies the output UV texture name. This option is only available for a procedural modeling mesh operation.
Fit	Controls how the UV data is handled. The following options are available: • **Entire UVs** - treats all the UV data as a single grouped element. • **per UV Polygon** - treats every polygon individually (resulting in a Barycentric UV layout). • **per UV Island** - treats connected islands of polygons as individual groups.
Keep Proportion	When active, the UV data scales proportionally so that the resulting UV data has the same ratio of height to width. The data scales until one of the maximum UV sizes is fit on one or

	both axes, but the data is not deformed to accomplish this.
Gaps	Adds border padding to the UV map. By default the UV map is drawn up to the edge of the UV space, which may cause problems with pixel blending. Entering a percentage gap value corrects this type of problem.  **Gaps**: 0 **Set Gaps by Pixel**: Off  **Gaps**: 10 **Set Gaps by Pixel**: Off  **Gaps**: 20 **Set Gaps by Pixel**: Off
Gaps by Pixel	Sets the padding to an image pixel unit. This option automatically updates the **Gaps** value. The **Gaps by Pixel** value is automatically calculated using the following equation: **Gaps** = **Pixel Size** / **Texel Density Image Size**. **Tip:** The pixel size is extracted from the **Texel Density Image Size** setting. To ensure that you have the correct image size, set the **Texel Density Get Image Size** option to use the background image. At the top of the **UV Editor**, click the **Show Background Image** icon to display the background image in the **UV Editor**. Setting the **Gaps by Pixel** value to the number of pixels you want to add as a gap places a border around the entire UV map. **Gaps by Pixel** is converted to **Gaps** when setting the pixel gaps in the command dialog. The **Gaps by Pixel** property is only available for the direct modeling **Fit UVs** tool.
Set Gaps by Pixel	When enabled, **Pixel Size** is used instead of the **Gaps** attribute. This option is only available for a procedural modeling mesh operation.
Pixel Size	The measurement of a single pixel in a UV unit. This value allows you use images scaled to

	real world size. This option defines the **Gaps by Pixel** based on the **Texel Density Image Size**.  **Pixel Size**: 10 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024  **Pixel Size**: 50 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024  **Pixel Size**: 100 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024 This option is only available for a procedural modeling mesh operation.
Texel Density Image Size	The size of the image that is used for the Texel Density to be mapped to. A **Texel Density Image Size** of 512 represents a Texel Density of 512 pixels per UV unit.  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 2048

	 **Pixel Size**: 10 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512  **Pixel Size**: 50 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512  **Pixel Size**: 100 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512 For more information on Texel Density, refer to the Texel Density documentation. **Note:** When using the direct modeling version of the **Fit UVs** command, the **Texel Density Image Size** can be adjusted from the **Image Size** option in the **Texel Density** panel on the **UV Tab**. This option is only available for a procedural modeling mesh operation.
UDIM	Positions the UDIM indicator frame to the defined UDIM address location in the UV Editor. The UDIM property is only available for the direct modeling Fit UVs tool.

Select Border

The function of the **UV Select Border** tool is similar to that of the Boundary Edges command — only for UV mapping. Modo takes the UVs that appear in the UV Texture window and selects the border around the UV islands. It also selects the border edges on the geometry in the 3D Model view. In this way, you can see how the flattened sections fit together on the mesh geometry.

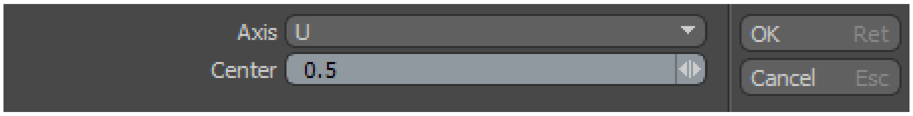
To use the **Select Border** tool:

1. With the Create UVs tool, create UVs from mesh geometry.
2. On the **UV** menu, choose **Select Border**.

Modo indicates the borders surrounding the UVs with an orange color on the mesh geometry and in the UV view.

Mirror UVs

With the **UV Mirror** command you can quickly flip the orientation of the selected UV in the UV Texture view. In the **Mirror UV** dialog, you specify which **Axis** to mirror around (**U** or **V**) and set the position of the **Center** of the mirroring operation.



Axis U

Center 0.5

To mirror a UV:

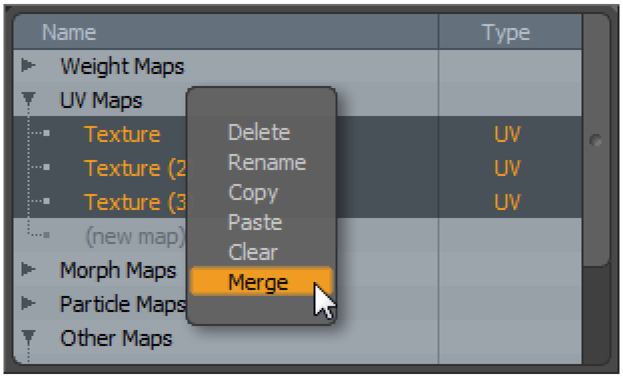
1. In the UV Texture view, select the UV(s) to mirror.
2. On the **UV** menu, choose **Mirror**.
3. On the **Mirror UVs** dialog, specify the **Axis**.
4. Then, specify the **Center** position.
5. Click **OK**.

Modo flips the UVs to their new orientation and position in the UV Texture view.

Rotate UV

If you invoke the Rotate UVs command, Modo turns the UV data around each selected UV polygon to rotate the UV order. This can be useful when working on low-polygon meshes where the orientation of the texture is critical.

Merge UVs


Name
Type
Weight Maps
UV Maps
Texture
Texture (2
Texture (3
(new map)
Morph Maps
Particle Maps
Other Maps
UV
UV
UV
Delete
Rename
Copy
Paste
Clear
Merge

By using the **Merge UVs** command, you can merge multiple UV maps on a single Mesh Item into a single map. The first selected UV map in the **Lists** viewport is the destination map, and Modo delete the other source maps after it merges their values. If Modo determines that a polygon is mapped in both the destination and source maps, it uses the UVs on the destination map and ignores the one on the source map.

Usage

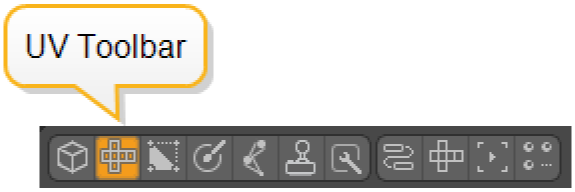
Select the UV maps to merge in the **Lists** viewport. Be sure to select the Target map first as the destination for the others. With the maps selected, either right-click one of the maps and choose **Merge UVs** or click **Merge UVs** in the **UVedit** layouts main toolbox. You can also choose **Texture > UV Operators > Merge**.

Flip UV

The **Flip UV** command allows you to horizontally or vertically flip UV orientations for an entire map or a selected area of it. This is useful for quickly changing the position of the texture applied to the UV.

To access the tool:

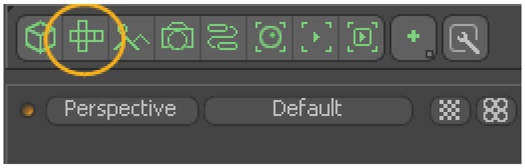
1. Open the UV toolbar by clicking the UV tab at the top of the left panel:



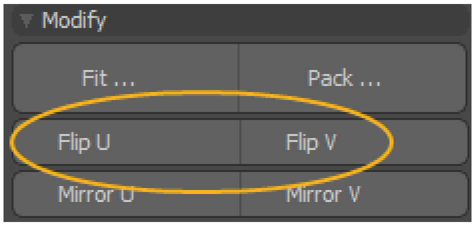
Alternatively, open the **UV** layout from the Switcher bar.

Note: You may need to disable the Starred Only ★ Only button on the right to reveal the **UV** layout button.

2. If working in the **Modo** layout, click the gray line above the 3D viewport to reveal additional controls, and click the UV viewport button.

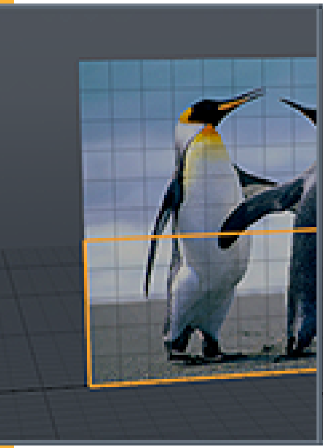


The **Modo** layout has separate buttons for **Flip U** and **Flip V**. To flip a selection per polygon, hold **Alt** when clicking either of these buttons.

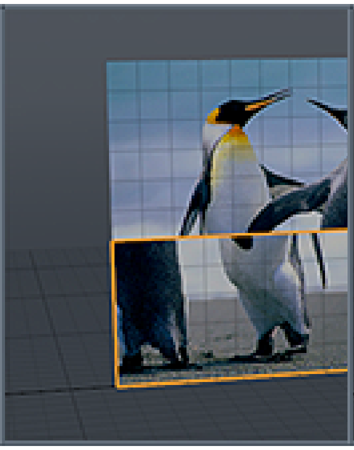


If you're working in the UV layout, click the **Flip** button and select the appropriate option.

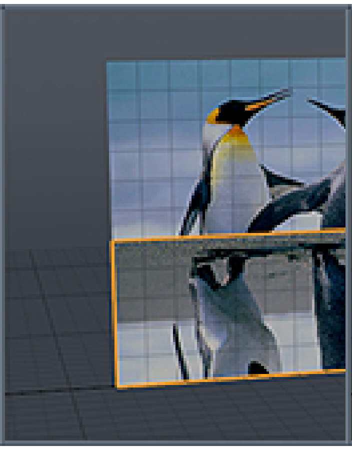
- **Flip U** - flips the entire selected area on the horizontal axis, based on the selection's bounding box center location.
- **Flip V** - flips the entire selected area on the vertical axis, based on the selection's bounding box center location.
- **With Options** - opens a dialog that allows you to select an additional **Per Polygon** option that flips the UV values independently per polygon.



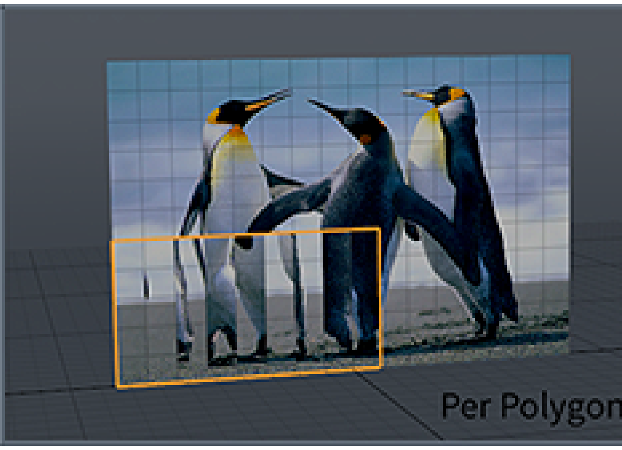
Original



Flip U



Flip V



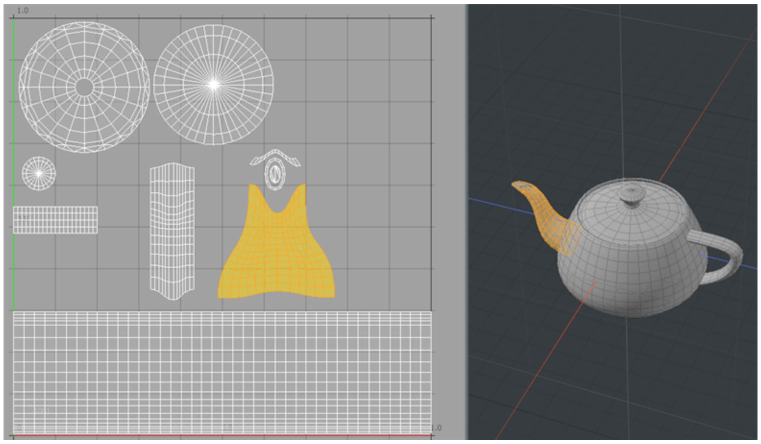
Per Polygon

Procedural UV Flip

The UV Flip mesh operation is the procedural equivalent of the Flip UV tool.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

1. In **Polygon** selection mode, select the required polygons on your UV map in the UV viewport.



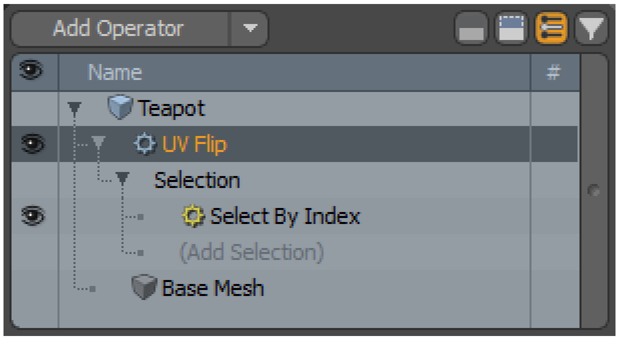
2. On the right panel, click the **Mesh Ops** tab.

Note: If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

3. Click the **Add Operator** button, and under **Mesh Operations** > **UV**, double-click **UV Flip**.

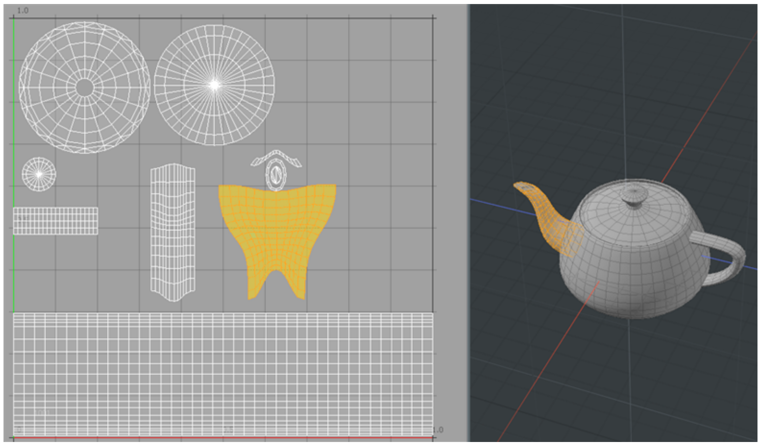
This adds the UV Flip mesh operation to the list and opens its properties in the **Properties** panel.

You can expand the **UV Flip** mesh operation in the list by clicking the arrow in front of it. This reveals the **Selection** input the operation uses. For more information on procedural selection, see Procedural Selection.



4. Set the **Axis** along which to flip the selection, and adjust any additional options as needed. For more information on the available properties, see UV Flip Properties.

   The selected area is flipped along the specified axis. In the image below, **Axis** is set to **V**.



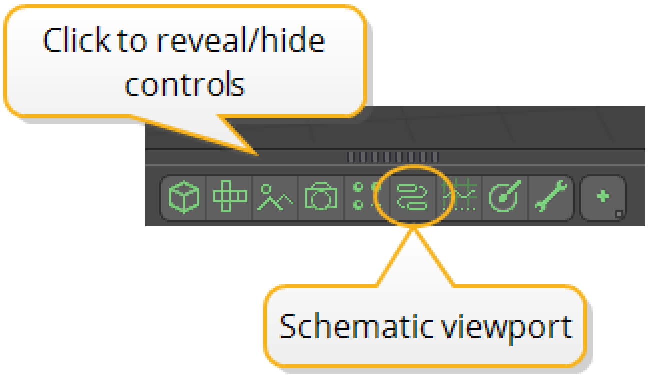
UV Flip in the Schematic

You can also work with the UV Flip mesh operation in the Schematic viewport.

Note: For more information on working with Schematic viewport in general, see Schematic Viewport.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.


Click to reveal/hide controls
Schematic viewport

>

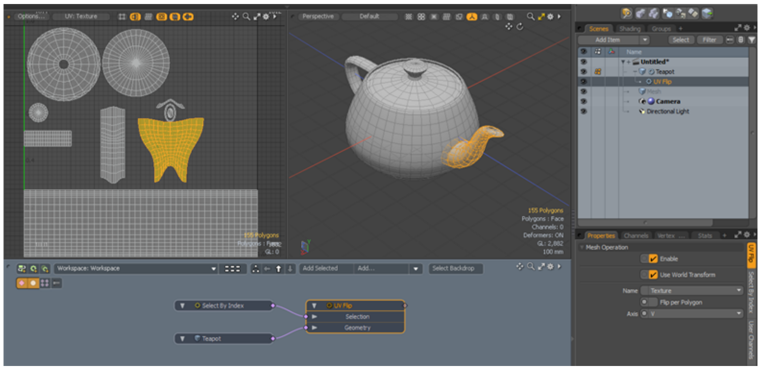
MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

To use UV Flip:

1. Click **Add...**, and under **Mesh Operations** > **UV**, double-click **UV Flip**.

   This adds the UV Flip node to the viewport and opens its properties on the lower right panel.


Select By Index
Teapot
UV Flip
Selection
Geometry

The node has the following inputs:

- **Selection** - Select polygons or modify your existing selection.
- **Geometry** - Any geometry that is affected by the tool.

UV Flip Properties

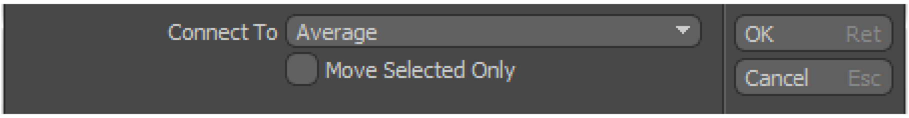
You can adjust the following properties for the UV Flip mesh operation.

Name	The texture map the UV Flip operation uses.
Flip per Polygon	When enabled, the UV orientation is flipped individually for each selected polygon.

Axis	The axis used by the operation: • **U** - Flips the selection on the horizontal axis, based on the selection's bounding box center location. • **V** - Flips the selection on the vertical axis, based on the selection's bounding box center location.

Sew UVs

With the **Sew UV** command you can quickly connect matching pairs of discontinuous UVs. To use the command, select an edge or vertex that is currently discontinuous and click the **Sew** button. Modo moves both matching UVs to a set of averaged UV coordinates and converts the UVs into a single UV. This is useful for mending UV maps that have been broken apart either manually or by using the mapping tools (such as an **Atlas** projection).



Connect To: Average

☐ Move Selected Only

When working with edge selections, select one of the two corresponding UV pairs to use the **Sew UV** command. However, to use the **Sew UV** command on vertex selections, you must have both vertices selected before applying the command.

It is helpful to adjust the UVs prior to using the **Sew UV** command so that the UVs are not too far apart before they are sewn back together.

Split, Move, and Sew UVs

The Split and Sew tools allow you to merge UVs together. The sew operation converts discontinuous UV data into continuous data to stitch UVs together. The move operation translates the entire UV island to facilitate the merger of separate UV shells. (A UV island is a connected group of polygons in a UV map.)

Use the Split and Sew tools to complete a number of operations:

- Separate one shell into multiple shells to correspond to different parts of a texture.
- Correct UV texture coordinates, which may be distorted or are overlapping shells in the UV Editor.
- Join small shells to reduce the number of seams to minimize distortion.

For more background information, see Working with UV Maps.

Video: Watch this video for an overview.

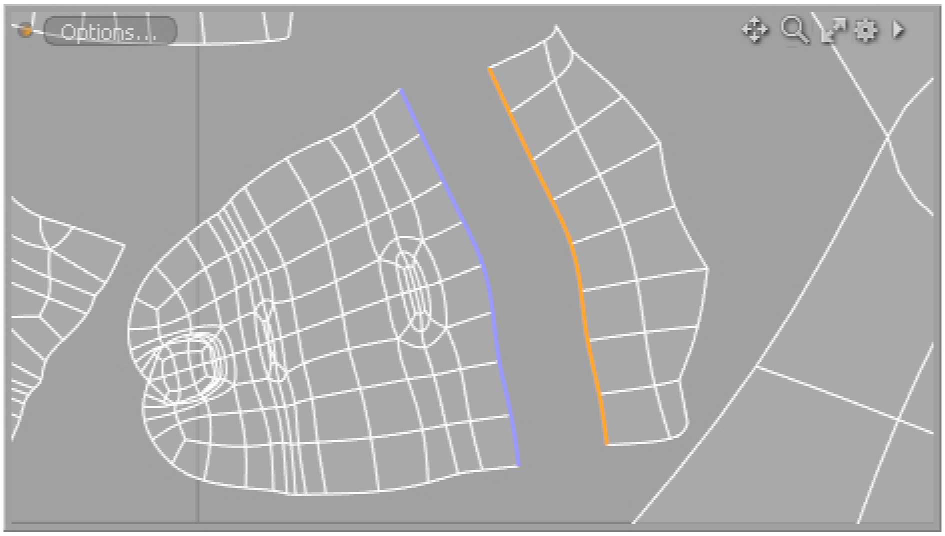
To activate the Move and Sew tools:

1. With your mesh already loaded, open the **UV** layout.
2. Under the layout tabs, click **Edges**.

Tip: Alternatively, you can use the **Vertices** or **Polygons** selection modes.

3. Select a number of edges on one island in the UV Editor.

   The purple edges indicate that they are the geometric equivalent of the selected orange edges.

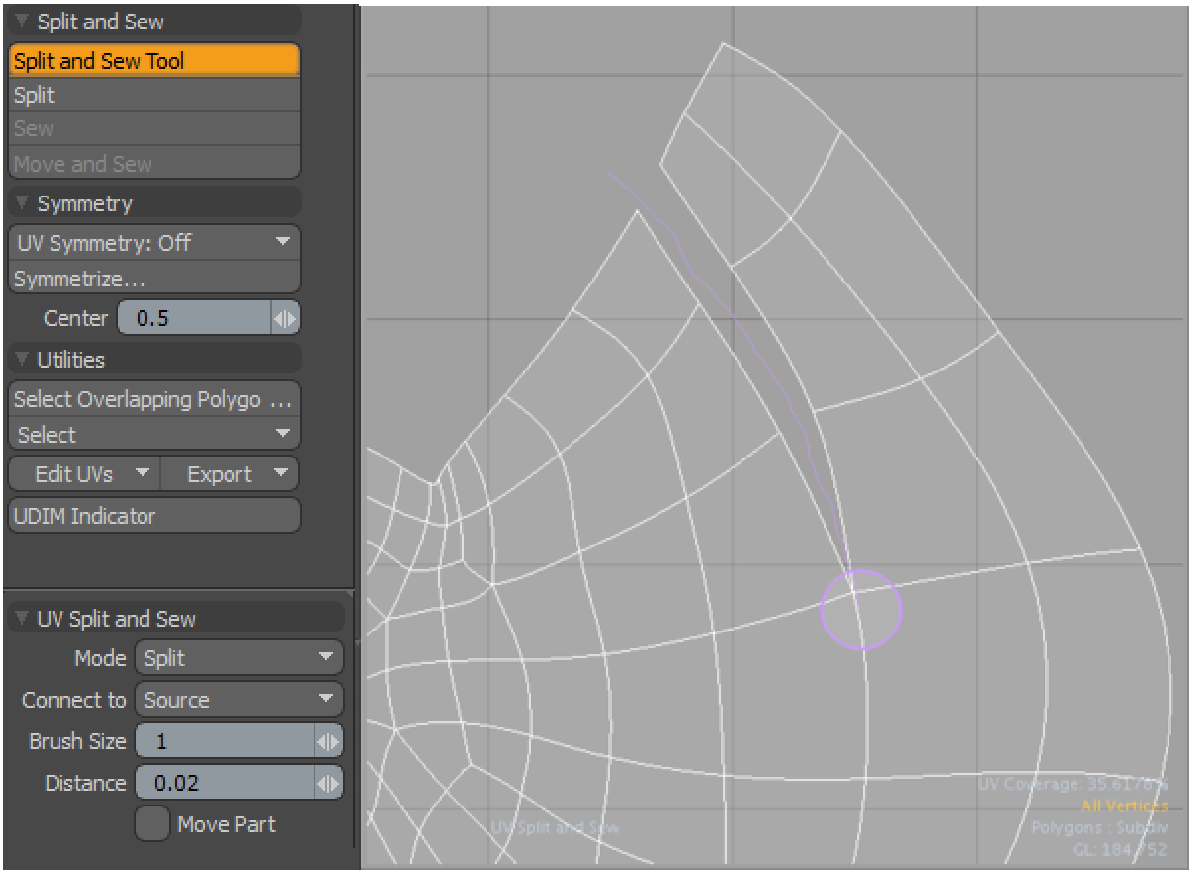


4. In the Toolbox, select **Split and Sew Tool**.
5. Adjust the **Split and Sew** options.

   For more information, see Split and Sew Tool Options.

Using the UV Split and Sew Tool

The **Split and Sew Tool** allows you to change the layout of UVs by dragging your mouse over them. This tool acts like a brush, with the radius controlling the results. Applying the tool over vertices in the UV space, with the Mode set to **Split**, separates them apart, with the brush radius controlling how far they are offset from their starting position. Holding the **Shift** key while right-clicking and dragging your mouse on the vertices, sews the edges together. Alternatively, setting the Mode to **Sew** and clicking on vertices sews the edges together.


Split and Sew
Split and Sew Tool
Split
Sew
Move and Sew
Symmetry
Symmetrize...
Utilities
Select Overlapping Polygo ...
Select
Edit UVs
Export
UDIM Indicator
UV Split and Sew
UV Split and Sew
All Vertices

UV Symmetry: Off

Center 0.5

Mode Split

Connect to Source

Brush Size 1

Distance 0.02

☐ Move Part

UV Coverage: 35.6176%

Polygons : Subdiv

GL: 184,752

To activate the Split and Sew tool:

1. With your mesh already loaded, open the **UV** layout.
2. Under the layout tabs, click **Edges**.

Tip: Alternatively, you can set the selection mode to **Polygons** or **Vertices** to make a selection.

3. Select a number of edges on one island in the UV Editor.

   The purple edges indicate that they are the geometric equivalent of the selected orange edges.

4. In the Toolbox, select **Split & Sew Tool**.

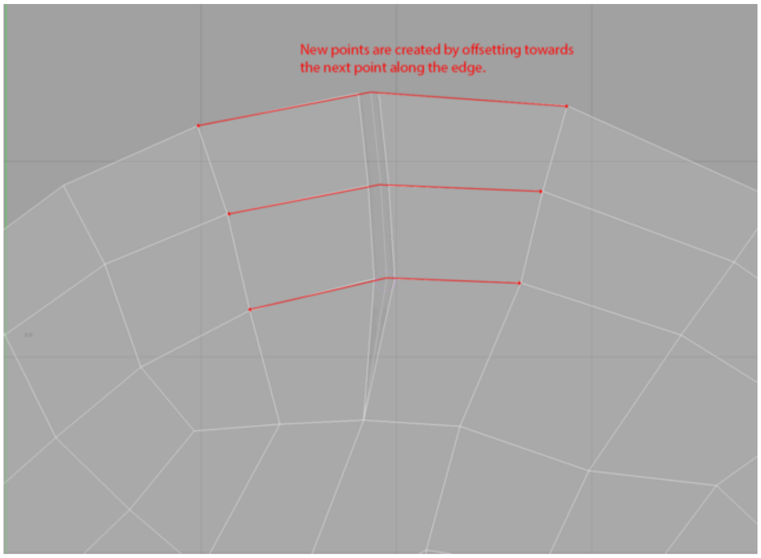
   The **UV Split & Sew** options are displayed in the lower portion of the Toolbox.

To split and sew edges of an UV island:

1. From the **UV Split and Sew** options, set the **Mode** to **Split**.
2. Right-click and drag the mouse to change the radius of the brush.

   For more information, see Customizing Paint Tools.
3. With the **Edges** selection mode enabled, click and drag your mouse over a UV edge.

   The UV island splits apart, with the offset controlled by the brush radius. Newly created UV points are positioned along the connected edge that is perpendicular to the cut.

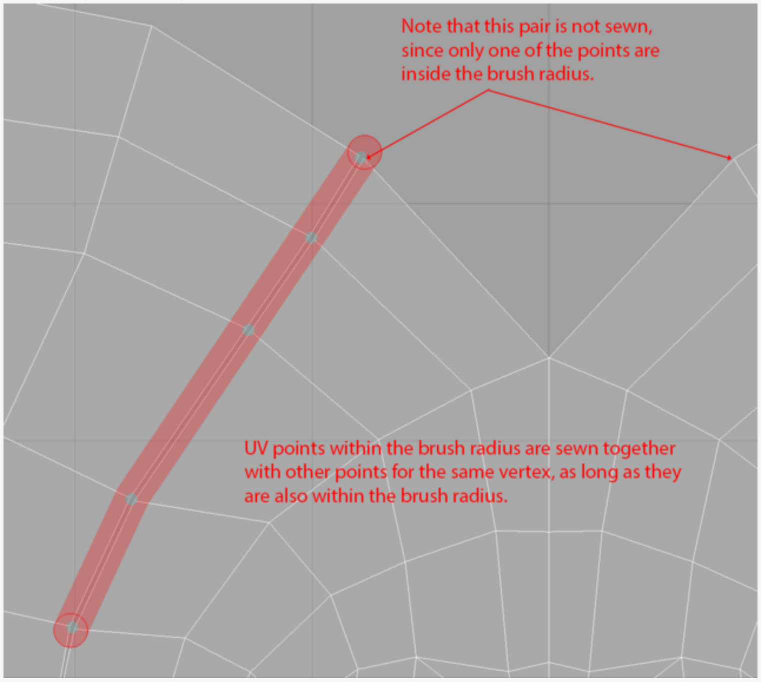

New points are created by offsetting towards the next point along the edge.

4. From the **UV Split & Sew** options, set the **Mode** to **Sew**.
5. Click and hold the mouse on a UV edge and drag it along the split seam.

Tip: Alternatively, to automatically switch from Split to Sew mode, hold the **Shift** key and click and drag on the vertices to sew the edges together.

The UV islands are sewn together.

Note: UV points within the brush radius are sewn together with other points for the same vertex, as long as they are also within the brush radius.


Note that this pair is not sewn, since only one of the points are inside the brush radius.
UV points within the brush radius are sewn together with other points for the same vertex, as long as they are also within the brush radius.

Note: If a brush stroke only covers a single vertex, or a set of unconnected vertices, those are split along all connected directions.

Split and Sew Tool Options

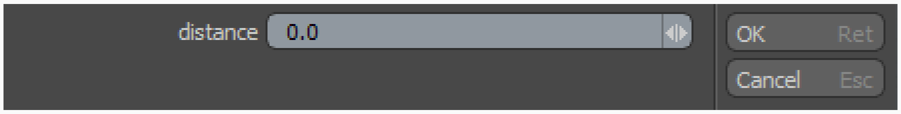
The following Split and Sew options are available:

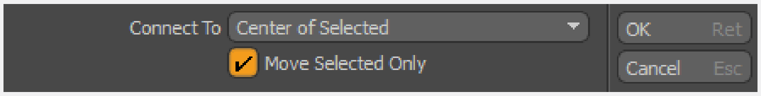
Depending on the command key and option selected, the interface updates, allowing you to quickly adjust your UVs.

Click the following	Results
Shift + Move and Sew	Connect to selected.
Alt + Move and Sew	Connect to unselected.
Ctrl = Move and Sew	Options dialog with scale toggle.
Shift + Sew	Connect to the center of selected.

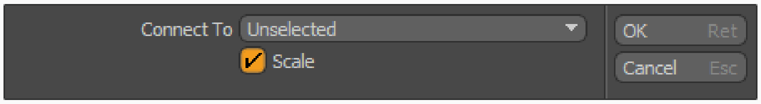
Alt + Sew	Connect to the center of unselected.
Ctrl +Sew	Options dialog with move toggle.

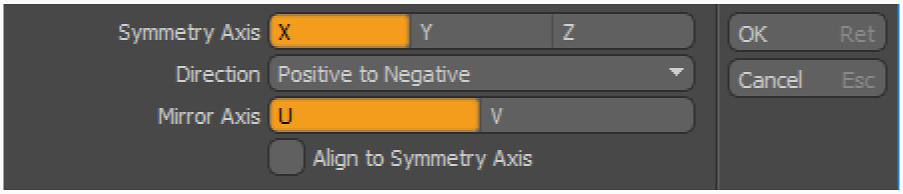
Option	Description
Split & Sew Tool	Splits and sews UVs interactively with a brush by moving one shell towards another in the specified direction. • **Mode** - Sets the type of operation. The following mode options are available: • **Split** - Separates UVs from each other along the edges connected to the selected UV points, creating borders using brush strokes. The radius of the brush controls the offset. Right-clicking and dragging controls the brush radius. • **Sew** - Connects UV vertices along border edges using brush strokes. The radius of the brush controls the offset. Right-clicking and dragging controls the brush radius. • **Connect to** - Specifies how to sew UV vertices when the **Mode** is set to **Sew**. The following options are available: • **Source** - Moves paired edges to the border edges with brush strokes. • **Average** - Moves brush stroked source edges and the paired edges to the averaged position. • **Destination** - Moves border edges on brush strokes to paired edges. • **Split from** - Controls the split direction when **Mode** is set to **Split**. The following options are available: • **Source** - Aligns the split direction to the source side. • **Average** - Averages the split between the source and the destination. • **Destination** - Aligns the direction to the destination side. • **Brush Size** - Sets the size of the brush in pixels. • **Distance** - Sets the distance to split UV parts in UV coordinate space when the **Mode** is set to **Split**. • **Move Part** - Moves the entire UV seam to the border edges. By default, this option is off.
Split	Allows you to splits UVs along the selected geometry (either along selected edges, around polygon selection boundaries, or on vertices). You can specify the distance of the UV split. Click **Shift** and then click **Split** on the Toolbox. The **UV Split** dialog displays. Specify the distance and then draw on the UVs to split. It doesn't offset the UVs but once split you can move the polygons independently.

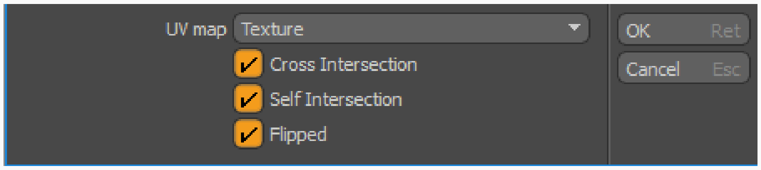
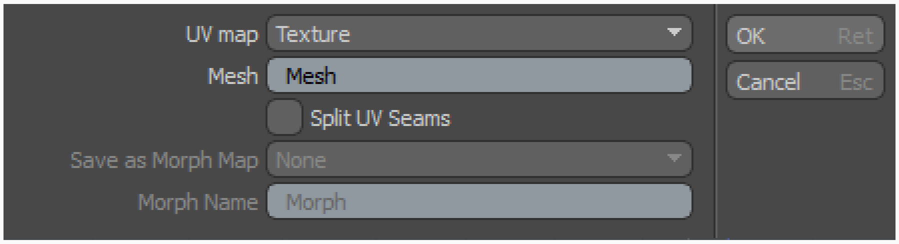
	 distance 0.0 **Tip:** Hold the **Shift**, **Ctrl**, or **Alt** key while you select the **Split** option to view other options. **Shift** + **Split** - Creates a large split. **Ctrl** + **Split** - Creates a medium split. **Alt** + **Split** - Creates a custom split.

Sew	Connects matching pairs of discontinuous UVs.
	To use the tool, select an edge, vertex, or polygon that is currently discontinuous and click the **Sew** button. Modo moves both matching UVs to a set of averaged UV coordinates and converts the UVs into a single UV.
	For more information, see Sew UVs.
	When enabled the **Sew UVs** dialog displays.
	
	Connect To: Center of Selected
	☑ Move Selected Only
	
	• **Connect to:**
	  • **Center of Selected** - Moves the unselected geometry to the center of the position and orientation of the selected geometry in the UV space.
	  • **Average** - Moves and rotates both pieces to an averaged center position of the transform.
	  • **Center of Unselected** - Matches the selected geometry to the center of the position and orientation of the unselected geometry.
	• **Move Selected Only** - Moves only the selected UVs to the position specified by the **Connect to** setting.

	Tip: Hold the **Shift**, **Ctrl**, or **Alt** key while you select the **Sew** option to view other options. **Shift** + **Sew** - Moves the selected side of the seam to the unselected points or edges. **Ctrl** + **Sew** - Sews together UV seams. When enabled the Sew UVs dialog displays. **Alt** + **Sew** - Moves the unselected side of the seam to the selected points or edges.

Move and Sew	Attaches UVs along the selected borders and moves them together in the UV space.
	When enabled the **Move and Sew UVs** dialog displays.
	
	Connect To: Unselected
	☑ Scale
	
	• **Connect to:**
	  • **Selected** - Moves the unselected geometry to match the position and orientation of the selected geometry in the UV space.
	  • **Unselected** - Matches the selected geometry to the position and orientation of the unselected geometry in the UV space.
	  • **Average** - Moves and rotates both pieces to an averaged transform position in the UV space.
	• **Scale** - Modo scales the geometry as needed to create a stitched, continuous UV map. This option is enabled by default.

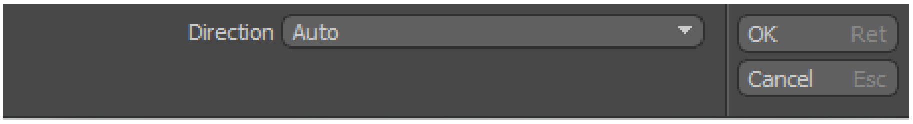
	Tip: Hold the **Shift**, **Ctrl**, or **Alt** key while you select the **Move and Sew** option to view other options. **Shift** + **Move and Sew** - Move and sew selected island towards the unselected one. **Ctrl** + **Move and Sew** - Move and sew UV seams. When enabled the Move and Sew UVs dialog displays. **Alt** + **Move and Sew** - Moves and sew unselected island towards the selected one.
Symmetry	
UV Symmetry	Mirrors corresponding components across the U or V axis. This allows you to quickly match component layouts you've made on one side of the mesh to another and to tidy up UV layouts to make texture mapping easier. • **Off** - Disables UV Symmetry. By default this option is disabled. • **U** - Uses the U position to apply a mirrored action. • **V** - Uses the V position to apply a mirrored action.
Symmetrize...	Allows you to select the how the symmetry is applied for the UV map.  Symmetry Axis: X Direction: Positive to Negative Mirror Axis: U ☐ Align to Symmetry Axis
Center	Joins the discontinuous UVs at the line of UV symmetry.
Utilities	
Select Overlapping Polygons	Allows you to detect polygon errors in UV maps. This tool provides three options to assist you in locating these errors; **Cross Intersection**, **Self Intersection**, and **Flipped**. For more information, see Select Error UVs
Select	Specifies the type of selection mode. The following options are available: • **Select UV Border** - Modo takes the UVs that appear in the UV Editor and selects the border around the UV islands. It also selects the border edges on the geometry in the 3D viewport. In this way, you can see how the flattened sections fit together on the mesh geometry.

	- **Invert Selection** - Selects all of the UVs except your UV selection. - **Convert** - Changes the selection mode to Vertices, Polygons, or Edges. - **More** - Selects additional UVs. - **Overlapping** - Selects overlapping UVs. To use this tool, specify a texture you have added to your scene and specify how the overlapping is executed.  UV map Texture ☑ Cross Intersection ☑ Self Intersection ☑ Flipped
Edit UVs	- **Cut UVs** - Cuts the selected UVs. - **Copy UV positions** - Copies the selected UV positions. - **Paste UV positions** - Pastes the selected UV positions. Select the mode **Connectivity** to respect polygon connectivity or select the mode **Selection** to respect selection order. - **Transfer UV positions** - Allows you to transfer all UVs from selected mesh items to another. This is useful when you want to copy the UVs you created for an original mesh to other duplicated mesh items in your scene. - **Delete UVs** - Deletes the selected UVs. - **Set Value** - Sets the Vertex Map value. Specifies the Vertex Map Type (UV or Position), Component (U or V), and specifies the Value.
Export	Allows you to export your UVs using the following options: - **Export UVs to EPS** - The **Save As** dialog displays. For more information, see **Importing and Exporting File Formats**. - **Export UVs to SVG** - The **Save As** dialog displays. For more information, see **Importing and Exporting File Formats**. - **Convert UVs to Mesh** - The **Convert UVs to Mesh** dialog displays. For more information, see **Convert UVs to Mesh**.  UV map Texture Mesh Mesh ☐ Split UV Seams Save as Morph Map None Morph Name Morph

UDIM Indicator	Allows you to target the current UDIM space to easily move UV elements around. For more information, see UDIM Indicator.

Orient Pieces

This command automates the process of organizing UV islands. UV islands are continuous areas of polygons in UV space that are disconnected from other UV data. This commonly occurs in Atlas-mapped UV sets as well as with UV unwrapped data that was accomplished by isolating areas of the mesh for focused unwraps.


Direction Auto

The **Orient Pieces** dialog has a single list for specifying the **Direction** to orient each island: **Auto**, **Horizontal**, and **Vertical**. Modo specifies **Auto**, by default, to automatically rotate the pieces in either direction according to which direction creates the algorithically-optimal layout. Selecting **Horizontal** forces the islands into a more horizontal orientation based on the lengths of the islands. Selecting **Vertical** forces the islands in an orientation that is perpendicular to the **Horizontal** method.

Texel Density

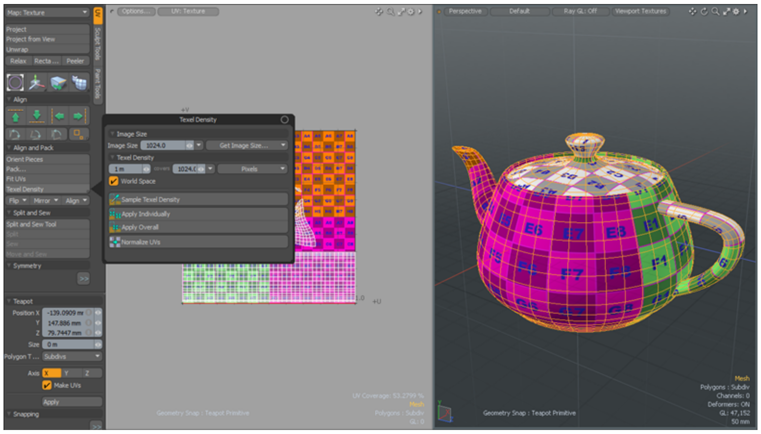
The **Texel Density** tool allows you to quickly scale the size of UV islands or individual polygons such that they span roughly the same number of texture pixels per meter of real size. (A UV island is a connected group of polygons in a UV map.) This is useful when you are dealing with texture density mismatches; in some cases you may either have differently sized UVs or textures, resulting in visibly over-detailed or under-detailed areas relative to the rest of the asset or scene. Making the texel density more uniform can help maintain a standardized level of texture detail across an asset or environment.

The **Texel Density** tool allows you to specify the resolution of the texture used, allowing the texel density to be calculated appropriately for different texture sizes but still remain visually consistent. For example, a texel density of 512 pixels per meter would need a 1 meter real size to span 2 UV units for a 256 x 256 texture, but only 1/2 UV units for a 1024 x 1024 texture.

This tool is useful in games development and environment texturing workflows.

Note: The **Texel Density Toolkit** is available from **UV** layout in the Toolbox **UV** sub-tab. This toolkit is useful for game art. It includes scripts to pick/apply texel density to meshes and UI functions to normalize texel density for UVs. For details on the options available, see Texel Density.

The **Texel Density** tool is found in the **UV** layout, on the Toolbox, under **Align and Pack**.


Texel Density
Image Size
Get Image Size...
Texel Density
Pixels
World Space
Sample Texel Density
Apply Individually
Apply Overall
Normalize UVs
Align and Pack
Texel Density
Perspective
Default
Viewport Textures

The Benefit of Using this Tool

• To make all UVs applied to a mesh use approximately the same texel density for the texture size of their applied material.

• To apply a uniform level of texture detail across many assets which might share the same scene, allowing them to better visually fit together.

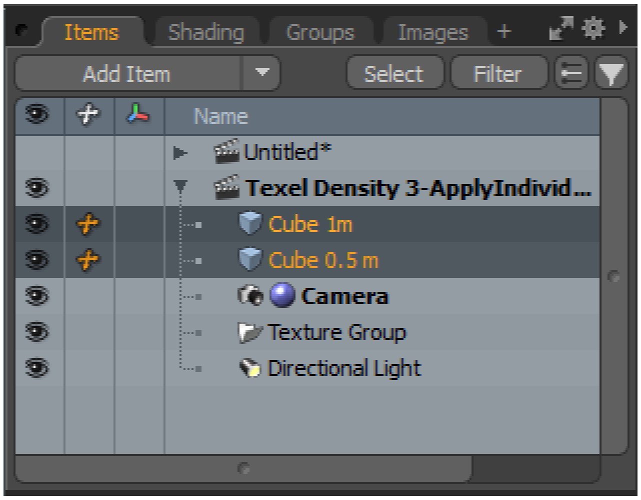
Tip: Textures are represented by texels in the same way that pictures are represented by pixels. They are the smallest "puzzle piece" of a texture. They are not the same thing as pixels because a texel is a container for the pixels. Think of them as a texture element, which acts as a single point of color on the surface of your mesh.

Applying Texel Density Tools

The Texel Density, **Apply Individually**, tool allows you to quickly scale each selected UV Island to use the average texel density value that matches the target. The following example that demonstrates the workflow of applying this tool plus the results of applying the **Apply Overall** and **Normalize UVs** tools.

To apply individual texel density values:

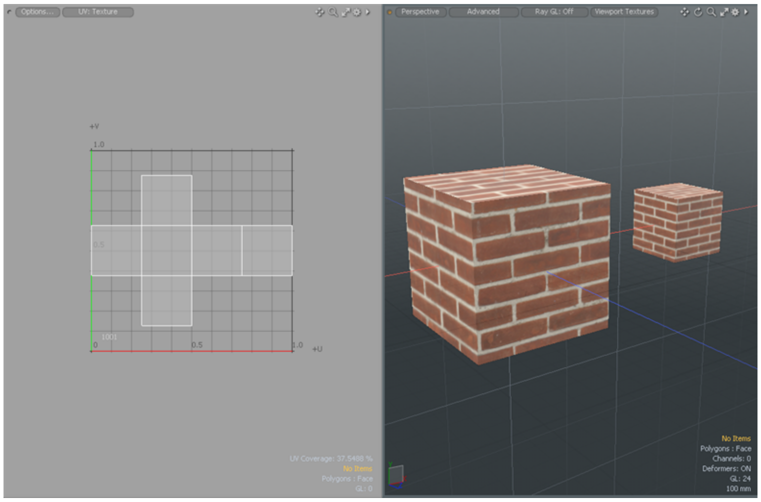
1. Download our example scenes and unpack the files.
2. Extract **example_texel_density.zip** and open the **.lxo** file in the **UV** layout.
3. In the **Items** list, on the right panel, select both mesh items; **Cube 1m** and **Cube 0.5m**.


Items
Shading
Groups
Images
+
Add Item
Select
Filter
Name

Untitled*
Texel Density 3-ApplyIndivid...
Cube 1m
Cube 0.5 m
Camera
Texture Group
Directional Light

4. In the Toolbox on the left panel, open the **UV** sub-tab, and set **Map** to **Texture**.

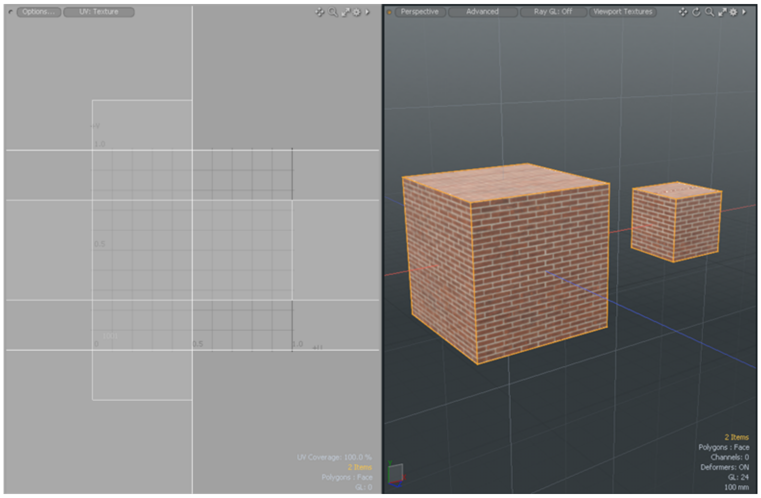
   The UV islands display in the UV Viewport.


Options...
UV: Texture
Perspective
Advanced
Ray GL: Off
Viewport Textures
1.0
0.5
+U
+V
No Items
Polygons : Face

5. In the Toolbox, under **Align and Pack**, click **Texel Density**.

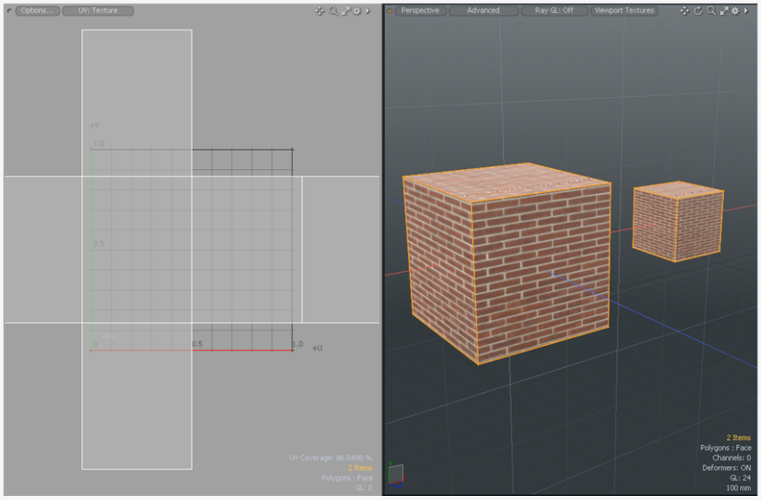
   The **Texel Density** options display.
6. Ensure **World Space** is enabled and click **Apply Individually**.

   The UVs for both meshes updates and display different texture sizes. The smaller cube, which is scaled on all axes by 50%, matches the texel density value to the larger cube which is scaled differently. World Space for both meshes is taken into account. The size of the brick images used are the same for both meshes.

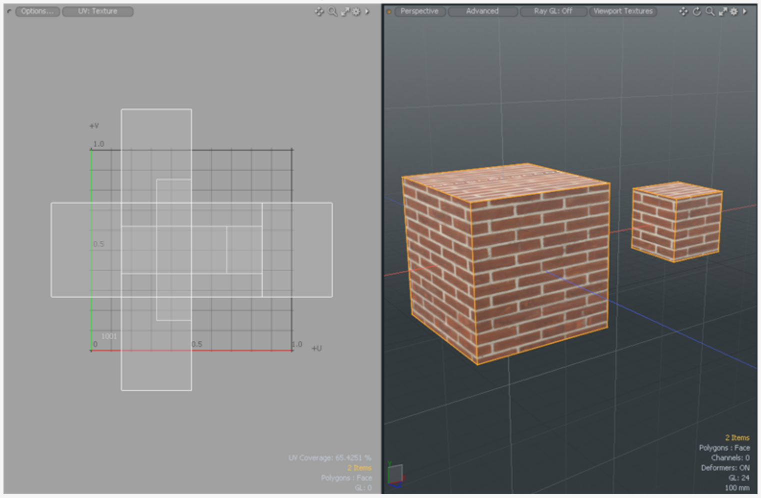

Options...
UV: Texture
+V
1.0
0.5
1001
0
0.5
1.0
+U
UV Coverage: 100.0 %
2 Items
Polygons : Face
GL: 0
Perspective
Advanced
Ray GL: Off
Viewport Textures
2 Items
Polygons : Face
Channels: 0
Deformers: ON
GL: 24
100 mm

Tip: When using the **texel-density-apply-overall_bundled.lxo** example scene, and following the above procedure to apply **Apply Overall**, the UVs for both meshes update and display the same texture size.

For both cubes, the texel density is averaged based on World Space. For example, the size of both mesh items are taken into account and a partial brick image is applied over both meshes. Each have five bricks in the top row but the size of the bricks are different for both meshes.


Options...
UV: Texture
Perspective
Advanced
Ray GL: Off
Viewport Textures
2 Items
Polygons : Face
Channels: 0
Deformers: ON
GL: 24
100 mm

When using the **texel-density-normalize-uvs_bundled.lxo** example scene and following the above procedure to apply **Normalize UVs**, all UVs are scaled such that their texel density is the same as their total average.


Options...
UV: Texture
Perspective
Advanced
Ray GL: Off
Viewport Textures
2 Items
Polygons : Face
Channels: 0
Deformers: ON
GL: 24
100 mm

Texel Density Options

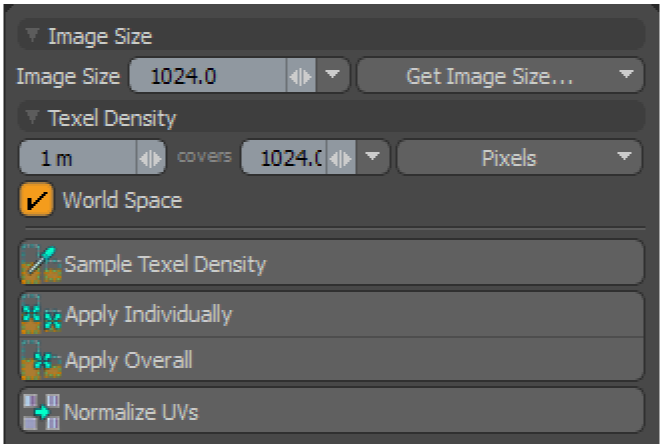
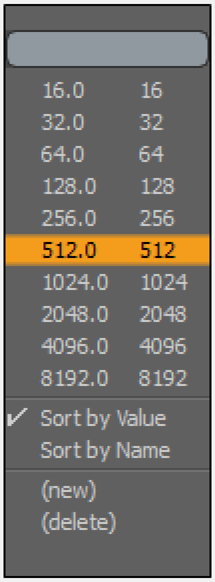


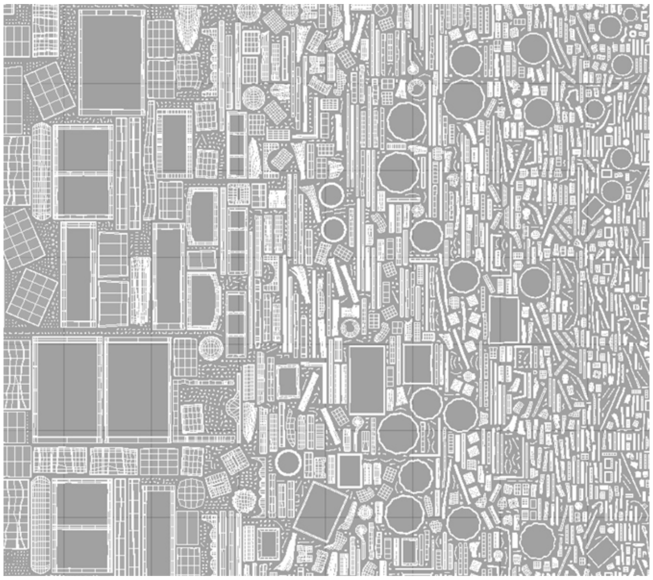
Image Size
Image Size
1024.0
Get Image Size...
Texel Density
1 m
covers
1024.0
Pixels
World Space
Sample Texel Density
Apply Individually
Apply Overall
Normalize UVs

Image Size	Sets the resolution of the texture to calculate for Texel Density. The following predefined options are available:  16.0 16 32.0 32 64.0 64 128.0 128 256.0 256 512.0 512 1024.0 1024 2048.0 2048 4096.0 4096 8192.0 8192 Sort by Value Sort by Name (new) (delete) **Tip:** You can create your own texel density values and sort them by name or value.
Get Image Size	Allows you to select one of the following options from the dropdown menu: • **Polygon Texture -** Sets the resolution to the diffused color texture applied to the first

	selected polygon. • **Selected Texture -** Sets the resolution from the selected texture. • **Render Resolution -** Sets the resolution from the Render output specifications.
Texel Density	Allows you to specify the texel density value, the coverage amount, and the units used.
World Space	Accounts for the item transforms when calculating the 3D sizes. The scaled items can have their texel densities match to unscaled or differently scaled items. **Note:** Texture squares are the same size in the **3D Viewport** as they are in the **UV Viewport**.
Sample Texel Density	Creates a target density sample from the selected UV Islands.
Apply Individually	Scales each selected UV Island so that the average texel density matches the target. For more information, see Applying Texel Density Tools. **Tip:** Press **Alt** to orientate the selected UVs in a vertical position.
Apply Overall	Scales all selected UV Islands so that the overall average texel density matches the target. **Tip:** Press **Alt** to orientate all UVs in a vertical position.
Normalize UVs	Scales all the selected UV islands, such that their texel density is the same as their total average.

Pack UVs

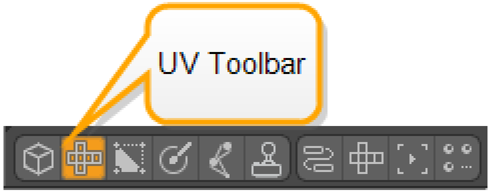
The **Pack UVs** tool automates the process of organizing UV island data to fit into the desired UV space. (A UV island is a connected group of polygons in a UV map.) It could be thought of as an automated puzzle solver. With the default settings, the tool moves, scales, and rotates each individual UV island until they all fit neatly within the designated grid space, as if solving a complex jigsaw puzzle.



Activating the Pack UVs Tool

Direct Modeling Tool:

- From the menu bar, click **Texture** > **UV Operators** > **Pack...**.
- Using the **UV** layout, on the left panel, under **Align and Pack**, click **Pack...**
- With the **UV** palette open, under **Align and Pack**, click **Pack...**



Procedural Modeling Tool:

- In the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **UV** > **Pack UVs**.
- In the **Schematic** view, click **Add...**, and double-click **Mesh Operations** > **UV** > **Pack UVs**.

Applying the Direct Modeling Pack UVs Tool

The Pack UVs tool modifies the currently-selected UV map.

Note: If you have selected multiple UV maps, the first one selected is used to apply the Pack UV tool. Other selected UV maps are not affected.

For more information about how to create UV maps, see Create UV Tool.

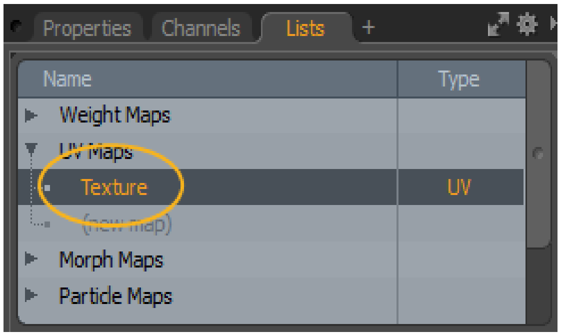
Once your mesh is prepared, follow these steps:

Tip: You can also use the meshes available in Modo's Preset Browser. For more information, see Preset Browser.

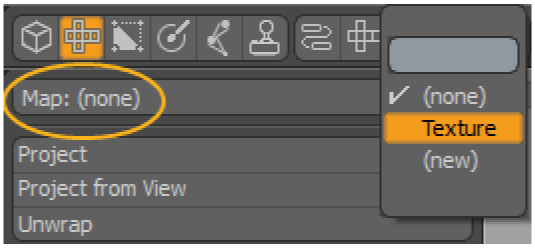
1. In the menu bar, click **Layout** > **Layouts** > **UV** to open the UV layout.

Tip: You can save this layout to be easily accessible from the switcher bar by clicking the **+** button in the switcher bar. In the **New Layout** dialog, name your layout and check **As Copy of Current Layout**.

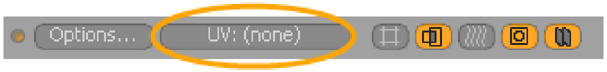
2. Select a texture map. There are a number of ways to do this:
   - On the bottom of the right panel, open the **Lists** tab, expand **UV Maps**, and select it from the list. It's called **Texture** by default.


Properties
Channels
Lists
Name
Type
Weight Maps
UV Maps
Texture
UV
(new map)
Morph Maps
Particle Maps

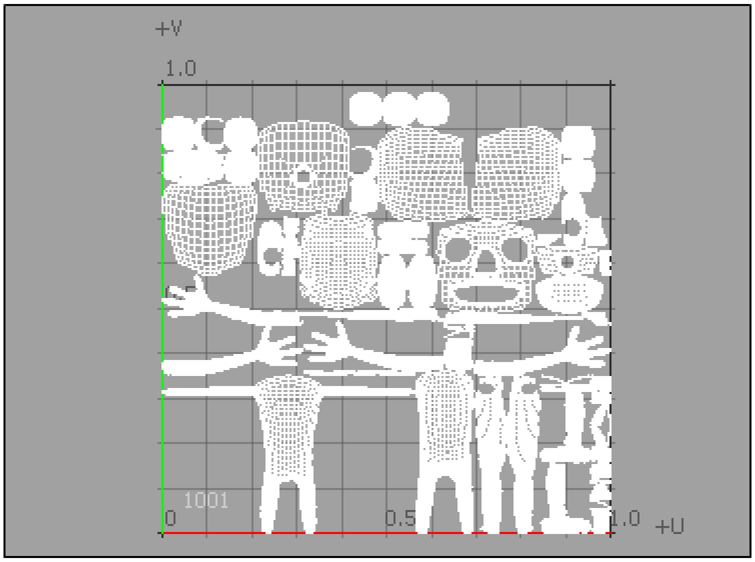
- On the left panel, set **Map: (none)** to the texture map**.**


Map: (none)
Project
Project from View
Unwrap
(none)
Texture
(new)

- On the UV viewport, click on the UV button, and select the texture map from the dropdown menu.

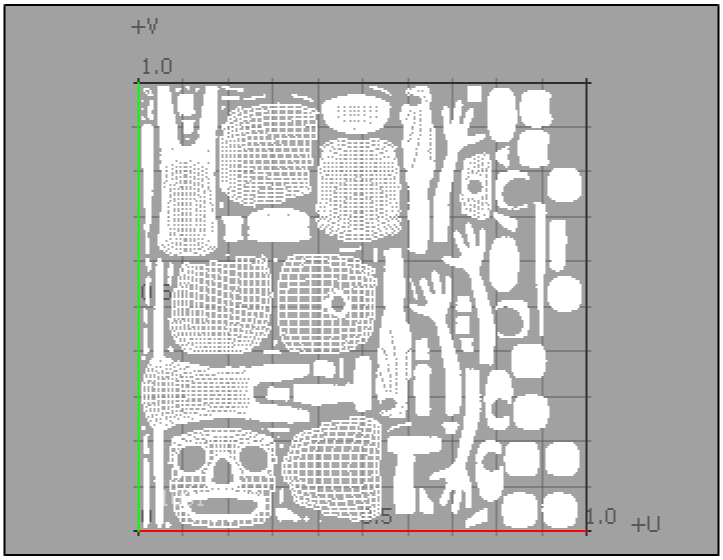

Options...
UV: (none)

The UV viewport displays the UV map.


+V
1.0
1001
0
0.5
1.0
+U

3. On the left panel, under **Align and Pack**, click **Pack...**.
4. On the **Pack UVs** dialog, click **OK**.

   Each individual UV island fits neatly within the designated grid space



Creating UV Island Groups Based on Polygon Tags

You can create individual UV islands and texture maps for each applied polygon tag on a mesh. The following example uses the teapot primitive which has multiple materials (polygon tags) applied to it, and includes a UV map.

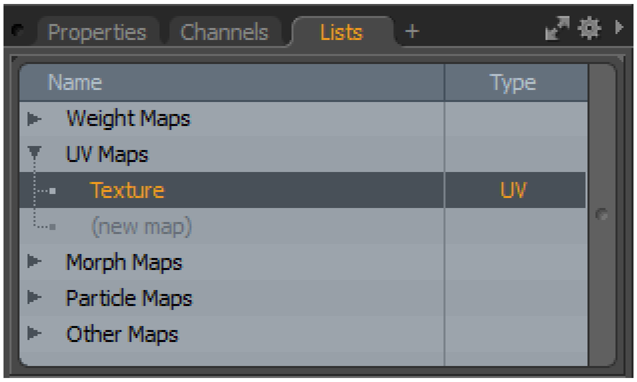
Tip: For more information about how to create your own UV maps, see Create UV Tool.

Tip: For information on using polygon tags, see Polygon Set Material/Part/Smoothing.

1. With your model ready, in the menu bar, click **Layout** > **Layouts** > **UV** to open the UV layout.

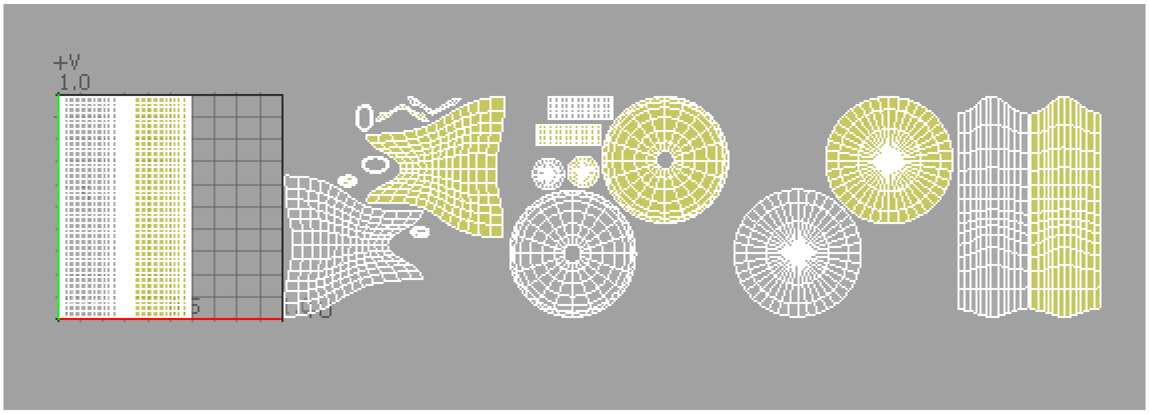
   If you don't have a mesh, you can download our example scenes and use **Teapot_example-polygons_tags.lxo**.

2. On the left panel, set **Map: (none)** to **Texture**.

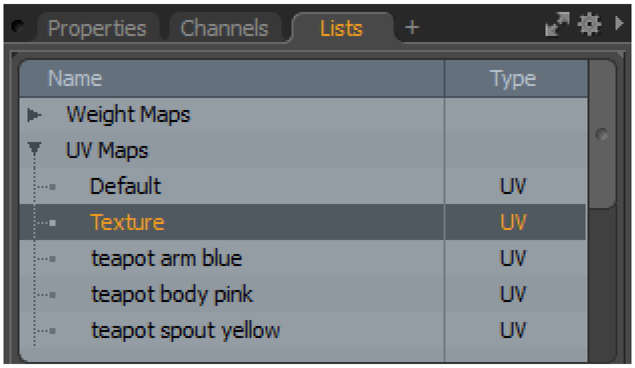

Properties
Channels
Lists
+
Name
Type
Weight Maps
UV Maps
Texture
UV
(new map)
Morph Maps
Particle Maps
Other Maps

3. On the left panel, under **Align and Pack**, click **Pack...**
4. In the **Pack UVs** dialog, set **Pack Region** to **Polygon Tag**, and click **OK**.

   Each polygon tag is packed into separate UV islands and grouped by same polygon tag.



You can pack each polygon tag into separate UV maps by enabling the **Pack UVs to New Map** option. It retains the original UV map and names the separate UV maps based on the material name. These UV maps are listed on the right panel in the **Lists** tab under **UV Maps**. Clicking on each individual UV map displays it in the UV viewport.



Name	Type
▶ Weight Maps	
▼ UV Maps	
Default	UV
Texture	UV
teapot arm blue	UV
teapot body pink	UV
teapot spout yellow	UV

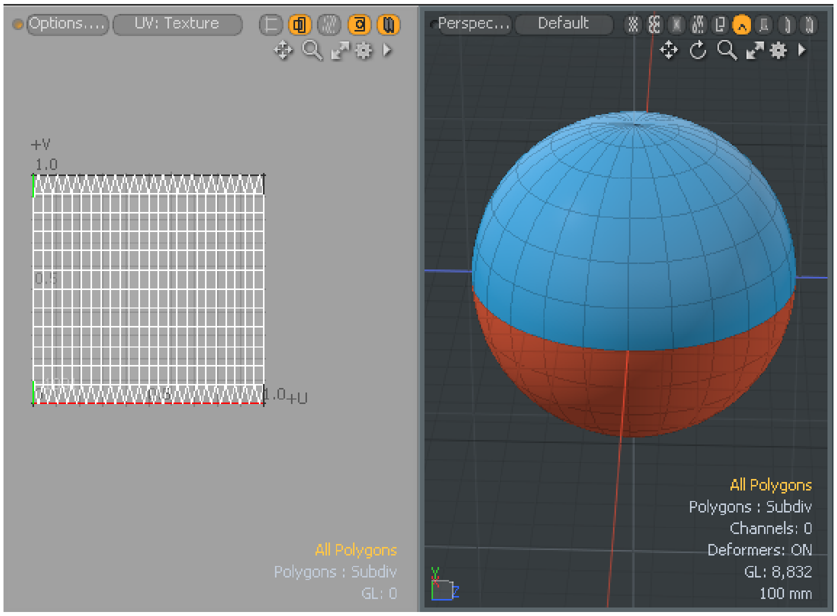
Splitting One UV Island with Multiple Polygon Tags

You can split a UV map which has only one UV island into individual UV maps based on applied polygon tags.

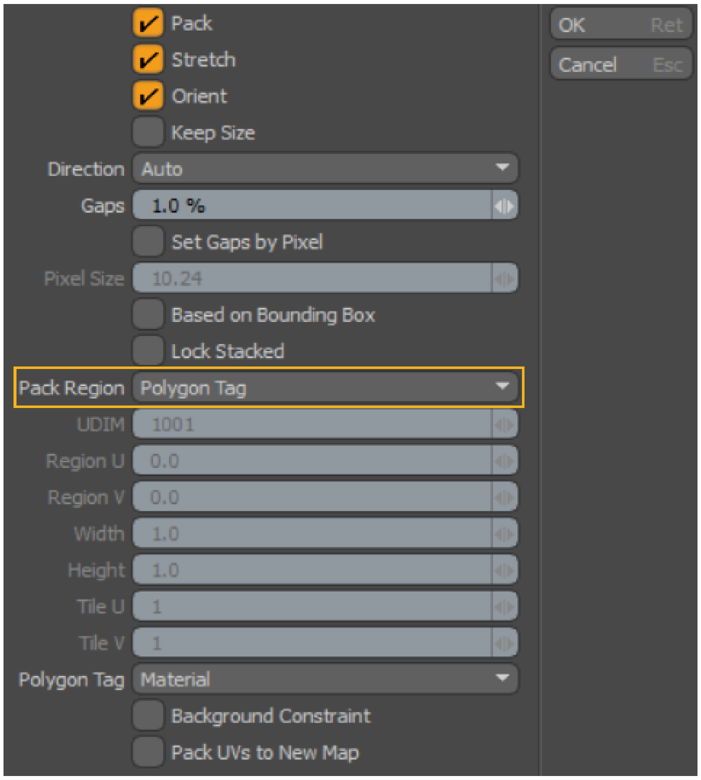
If you don't have a mesh, you can download our example scenes and use **sphere-two_materials-one_uv_island.lxo**.

1. In the **UV** layout, click **Map (none)** and select **Texture**.

   One UV island displays in the UV viewport.


Options...
UV: Texture
Perspec...
Default
+V
1.0
0.5
1.0 +U
All Polygons
Polygons : Subdiv
GL: 0
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

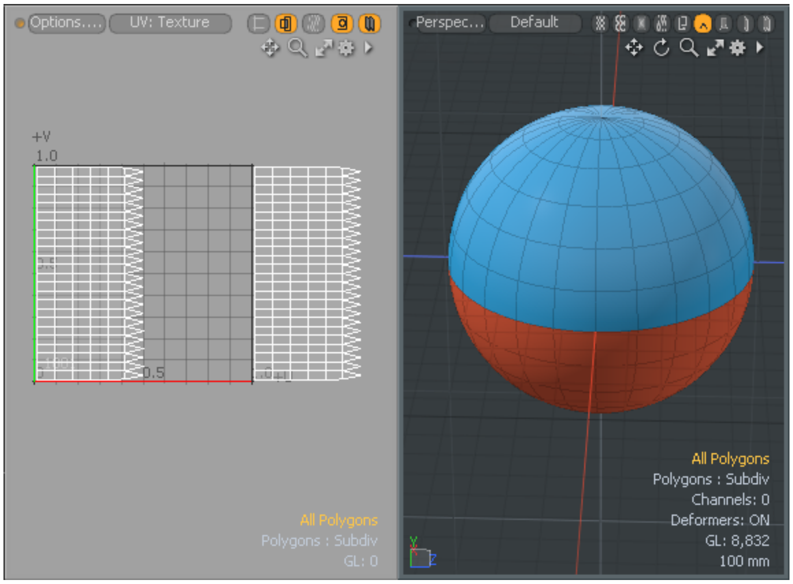
2. On the left panel, under **Align and Pack**, click **Pack...**
3. In the **Pack UVs** dialog, set **Pack Region** to **Polygon Tag** and click **OK**.


OK Ret
Cancel Esc

☑ Pack
☑ Stretch
☑ Orient
☐ Keep Size
Direction Auto
Gaps 1.0 %
☐ Set Gaps by Pixel
Pixel Size 10.24
☐ Based on Bounding Box
☐ Lock Stacked
Pack Region Polygon Tag
UDIM 1001
Region U 0.0
Region V 0.0
Width 1.0
Height 1.0
Tile U 1
Tile V 1
Polygon Tag Material
☐ Background Constraint
☐ Pack UVs to New Map

Two individual UV maps are created.

Tip: If you want the UV islands to retain their original size, enable **Keep Size** in the **Properties** tab. Enabling **Keep Size** disables the effect of **Stretch**.


Options....
UV: Texture
+V
1.0
0.5
All Polygons
Polygons : Subdiv
GL: 0
Perspec...
Default
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 8,832
100 mm

Procedural Modeling Pack UVs Example

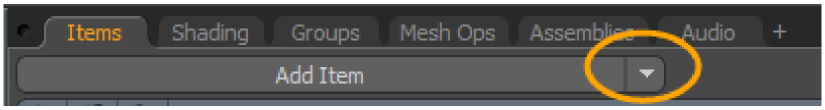
The Pack UVs mesh operation generates procedural UVs and organizes UV island data to fit into the desired UV space. This is particularly useful when you want to reorganize overlapping UV island data for a number of mesh layers, all containing their own UV maps, with a stack of mesh operations applied to your scene. Applying the Pack UVs operation easily reorganizes the data for you.

In the following example we set up simple procedural primitives in a scene, each having their own procedural UV maps. Once done, we'll apply a **Merge Meshes** operation to create one mesh item using all the primitive UV maps. As a result, you will notice how these UV maps overlap and the data is displayed in red in the UV viewport. To resolve the overlapping of UVs, we will apply our procedural Pack UVs tool to unpack the UV maps.

For more information, see Working with UV Maps.

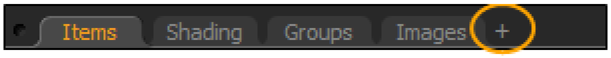
To add procedural primitives:

1. Open the **UV** layout.
2. On the right panel, click the down arrow beside **Add Item**, and click **Mesh**.

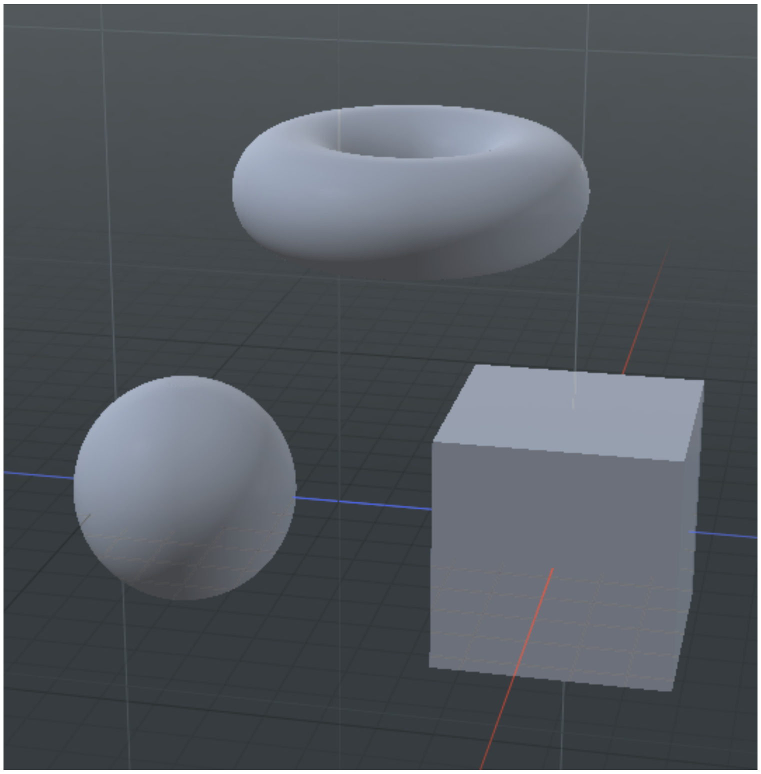

Items
Shading
Groups
Mesh Ops
Assemblies
Audio
+
Add Item

3. On the right panel, open the **Mesh Ops** tab, click **Add Operator**, and double-click **Mesh Operations** > **Create** > **Primitives** > **Cube**.

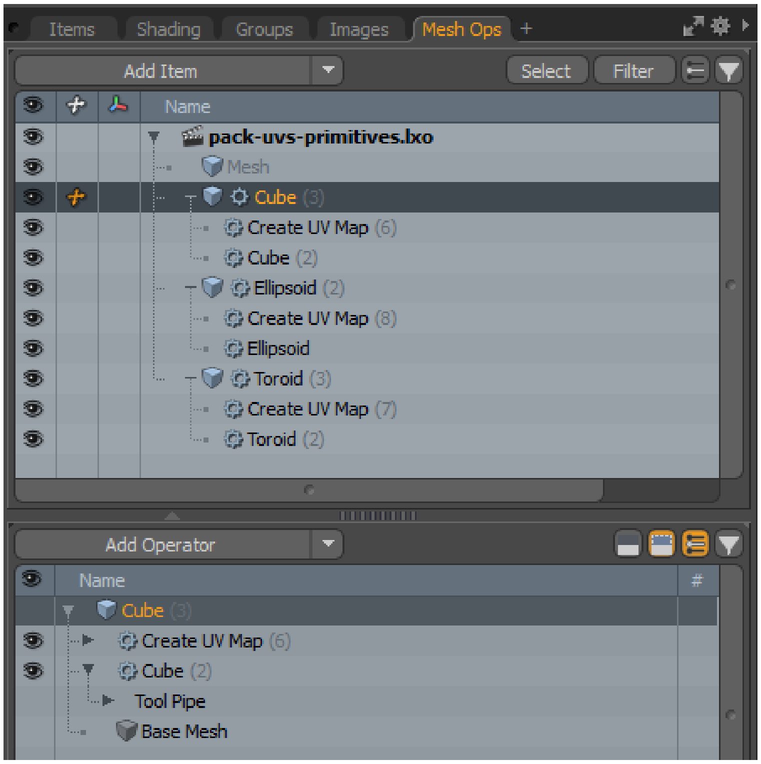
> **Tip:** If the **Mesh Ops** tab is not visible on the right panel, click the **More Tabs** or **New Tab** icon and select **Data Lists** > **Mesh Ops**.



4. In the 3D viewport, click-and-drag the move handles to position the primitive.
5. In the **Items** list, double-click on the mesh item you just created and type **Cube**.
6. With the cube mesh item selected, click **Add Operator**, and double-click **Mesh Operations** > **UV** > **Create UV Map**.
7. In the properties, set **Projection Type** to **Atlas**.
8. Repeat these steps to add an **Ellipsoid** and a **Toroid** procedural mesh item with UV maps.

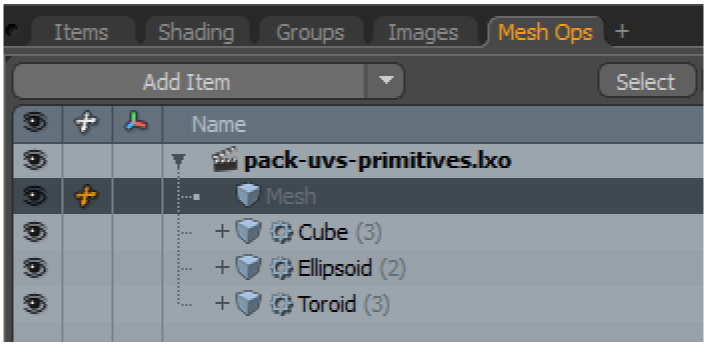


The **Mesh Ops** list displays the three primitives, each with their own UV map.

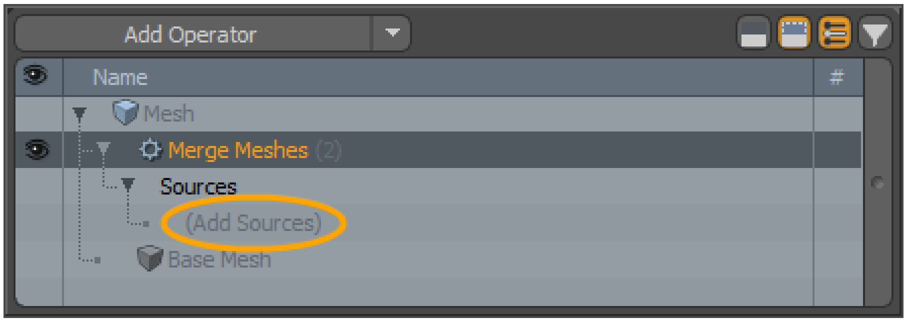

Items
Shading
Groups
Images
Mesh Ops
Add Item
Select
Filter
Name
pack-uvs-primitives.lxo
Mesh
Cube (3)
Create UV Map (6)
Cube (2)
Ellipsoid (2)
Create UV Map (8)
Ellipsoid
Toroid (3)
Create UV Map (7)
Toroid (2)
Add Operator
Name
#
Cube (3)
Create UV Map (6)
Cube (2)
Tool Pipe
Base Mesh

To apply a Merge Meshes operation:

1. Select the first mesh item in the **Items** list.


Items
Shading
Groups
Images
Mesh Ops
Add Item
Select
Name
pack-uvs-primitives.lxo
Mesh
Cube (3)
Ellipsoid (2)
Toroid (3)

2. Click **Add Operator** and double-click **Mesh Operations** > **Edit** > **Merge Meshes**.
3. Expand the **Merge Meshes** operator, and under **Sources** click **(Add Sources)**.

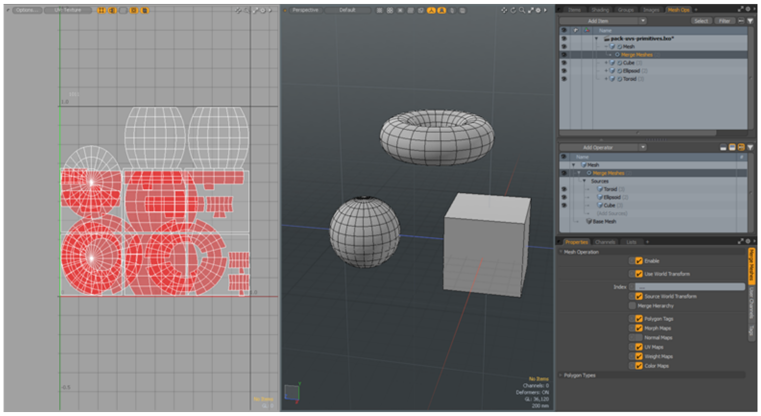

Add Operator
Name
#
Mesh
Merge Meshes (2)
Sources
(Add Sources)
Base Mesh

4. Navigate to **Modifier Mesh Operations** > **Existing** and select the parent source mesh items. In this example, double-click **Cube**, and repeat for **Ellipsoid**, and **Toroid**.

   The three procedural primitives are listed under the **Merge Meshes** > **Sources** item.

5. On the bottom of the right panel, open the **Lists** tab, expand **UV Maps**, and select **Texture**.

   Notice how the UV data from the three UV maps is overlapping and displayed in red in the UV viewport.



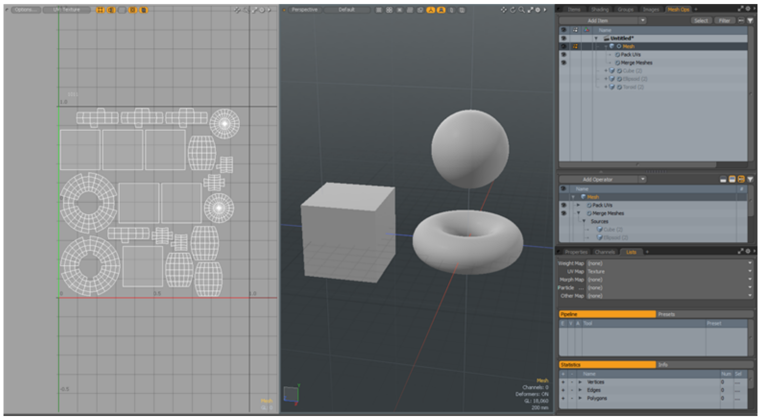
To add a Pack UV:

1. With the **Merge Meshes** mesh operation selected in the right panel, click **Add Operator** and double-click **Mesh Operations** > **UV** > **Pack UVs**.

2. In the **Mesh Ops** list, click the 👁 visibility icon to hide the **Cube**, **Ellipsoid**, and **Toroid** and then select the top mesh layer.

   The UV data is reorganized in the UV viewport.

> **Tip:** If you want the UV islands to retain their original size, enable **Keep Size** in the **Properties** tab. Enabling **Keep Size** disables the effect of **Stretch**.



3. With **Pack UVs** operation selected and the Properties tab open, set the **Gaps** value to **0.3%**.

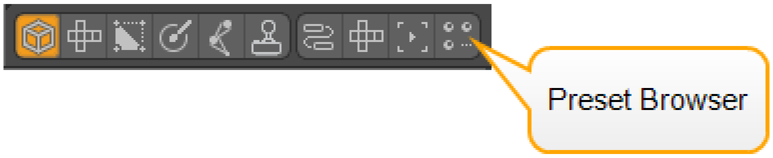
   The UV Island data is updated in the UV viewport.

Schematic Pack UVs Example

1. Open the Setup layout from the menu bar, under **Layout** > **Layouts** > **Setup**.

   At the bottom of the **Setup** layout, the Schematic viewport is displayed.

2. On the top left corner of the interface, click the Preset Browser icon.

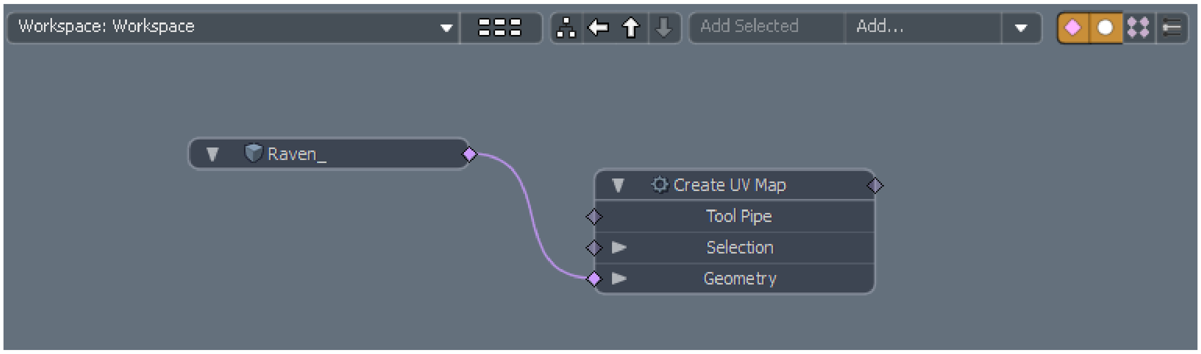

Preset Browser

3. Select an asset to use. This example uses **Cloud Assets** > **Meshes** > **Character** > **mhy3d bird**.

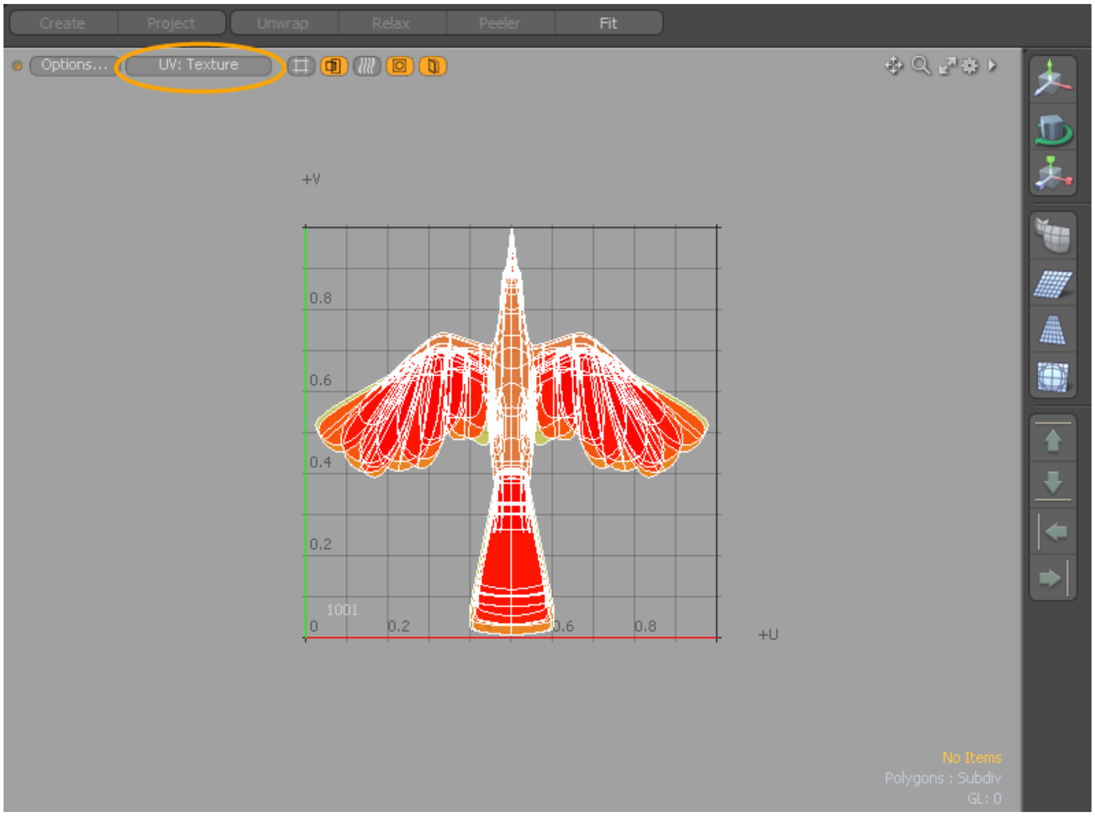
   A **Raven_** item is displayed in the **Items** list on the right panel.

4. On the right panel, from the **Items** list, drag-and-drop this item into the Schematic viewport on the bottom of the **Setup** layout.

5. In the Schematic viewport, click **Add...**, and click **Mesh Operations** > **UV** > **Create UV Map.**

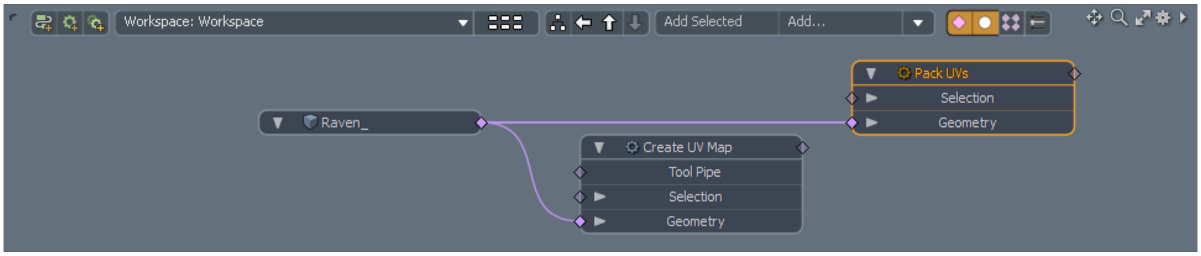

Workspace: Workspace
Add Selected
Add...
Raven_
Create UV Map
Tool Pipe
Selection
Geometry

6. From the main menu, click **Texture** > **Open UV Editor** and set the **UV** button to **Texture**.

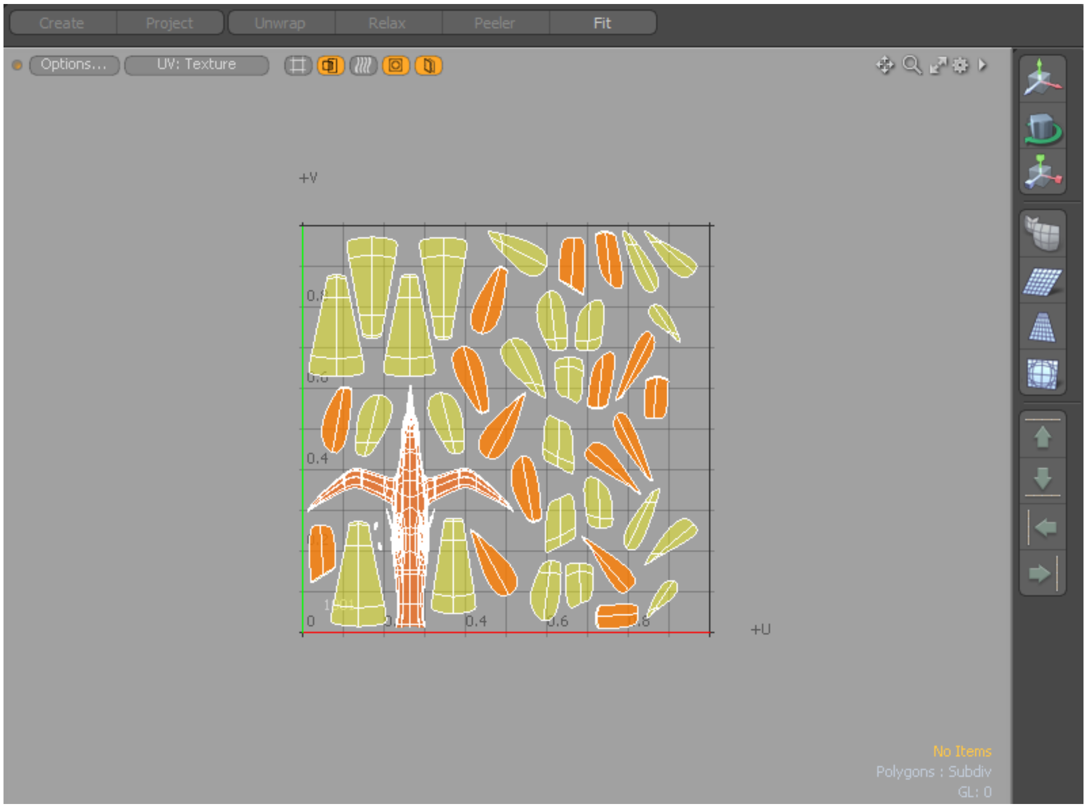

Create
Project
Unwrap
Relax
Peeler
Fit
Options...
UV: Texture
+V
0.8
0.6
0.4
0.2
1001
0
0.2
0.6
0.8
+U
No Items
Polygons : Subdiv
GL: 0

7. On the bottom panel, in the Schematic viewport, click **Add...**, and double-click **Mesh Operations** > **UV** > **Pack UVs**.

   The Schematic viewport now displays the Pack UVs node.


Workspace: Workspace
Add Selected
Add...
Pack UVs
Selection
Geometry
Raven_
Create UV Map
Tool Pipe
Selection
Geometry

The UV Editor updates the UV island data. The Pack UVs operation reorganized the UV island data.

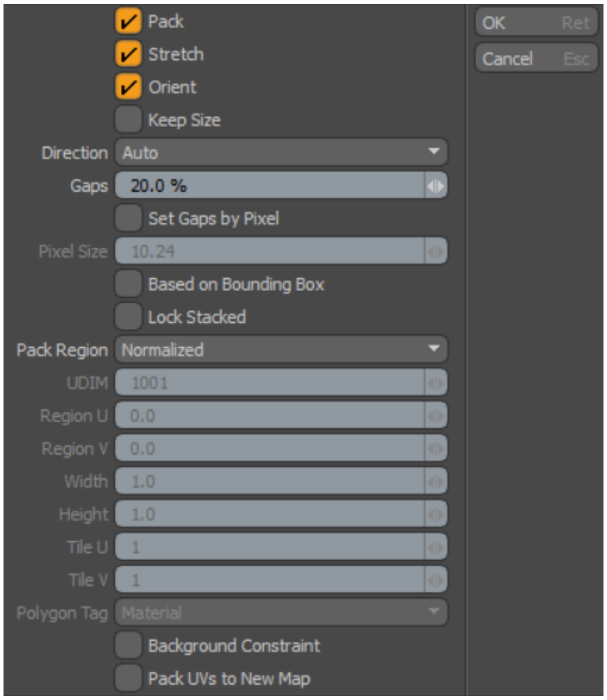

Create
Project
Unwrap
Relax
Peeler
Fit
Options...
UV: Texture
+V
0.4
0
0.4
0.6
+U
No Items
Polygons : Subdiv
GL: 0

8. In the Schematic viewport, select the **Pack UV** node, and update the properties in the right panel.

Tip: If you want the UV islands to retain their original size, enable **Keep Size** in the **Properties** tab. Enabling **Keep Size** disables the effect of **Stretch**.

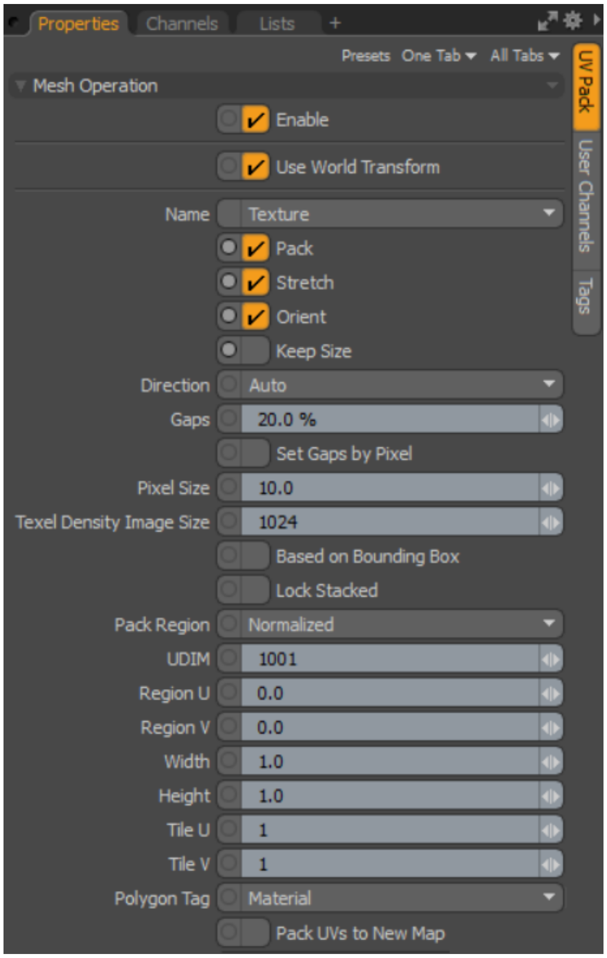
For more information, see Pack UVs Properties.

Pack UVs Properties



☑ Pack
☑ Stretch
☑ Orient
☐ Keep Size
Direction: Auto
Gaps: 20.0 %
☐ Set Gaps by Pixel
Pixel Size: 10.24
☐ Based on Bounding Box
☐ Lock Stacked
Pack Region: Normalized
UDIM: 1001
Region U: 0.0
Region V: 0.0
Width: 1.0
Height: 1.0
Tile U: 1
Tile V: 1
Polygon Tag: Material
☐ Background Constraint
☐ Pack UVs to New Map

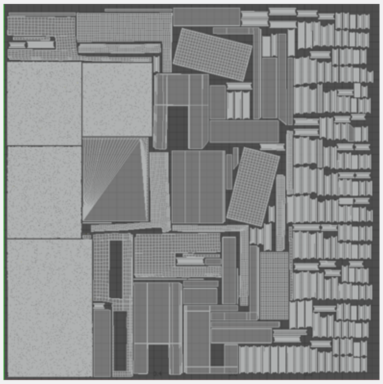
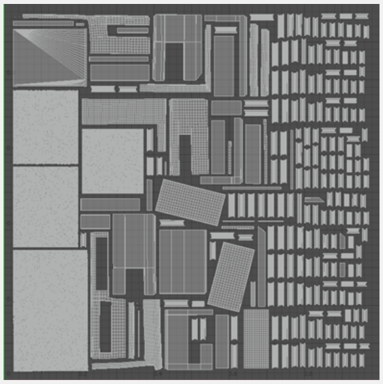
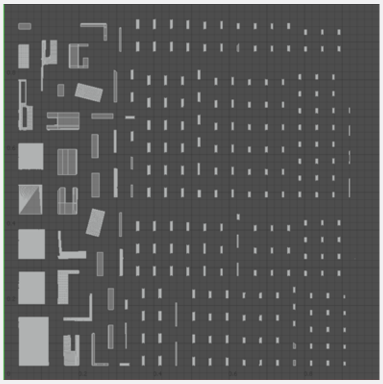
UV Pack tool properties

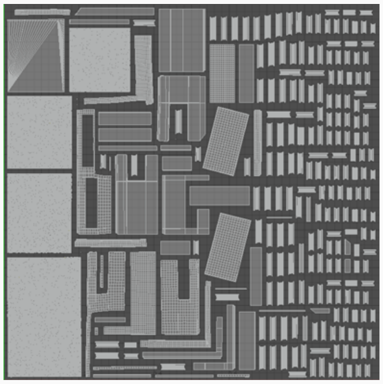
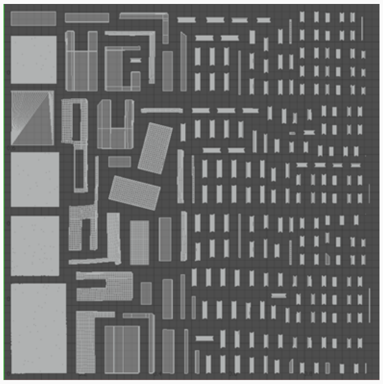
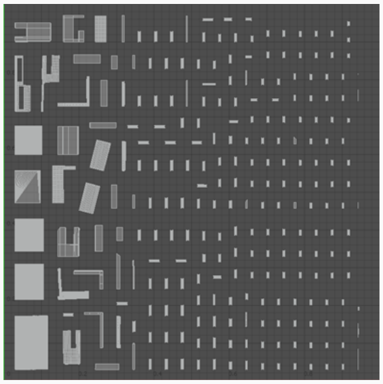
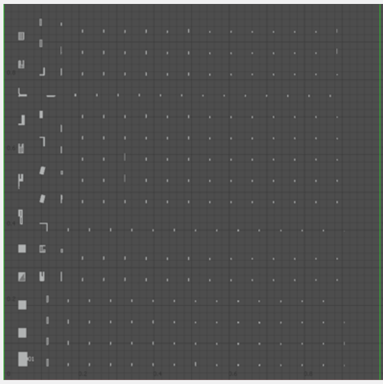
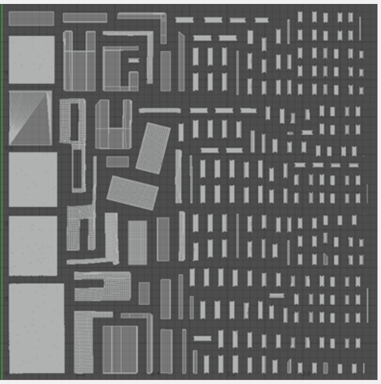
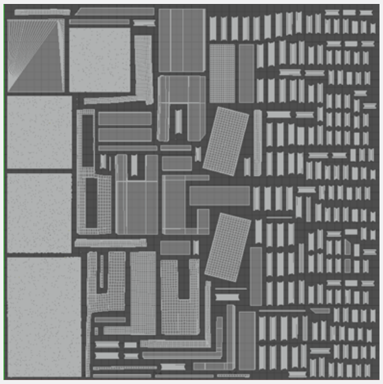


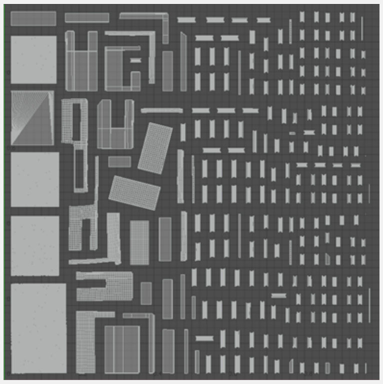
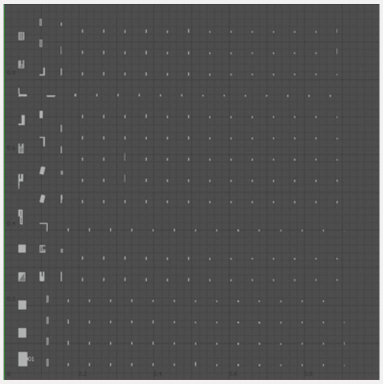
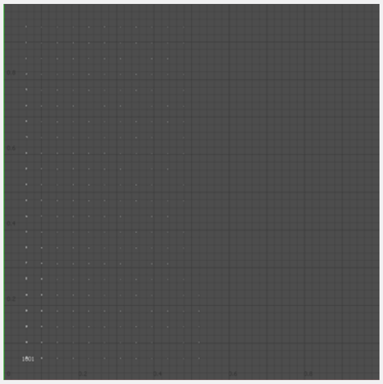
Mesh Operation
☑ Enable
☑ Use World Transform
Name: Texture
☑ Pack
☑ Stretch
☑ Orient
☐ Keep Size
Direction: Auto
Gaps: 20.0 %
☐ Set Gaps by Pixel
Pixel Size: 10.0
Texel Density Image Size: 1024
☐ Based on Bounding Box
☐ Lock Stacked
Pack Region: Normalized
UDIM: 1001
Region U: 0.0
Region V: 0.0
Width: 1.0
Height: 1.0
Tile U: 1
Tile V: 1
Polygon Tag: Material
☐ Pack UVs to New Map

UV Pack procedural properties

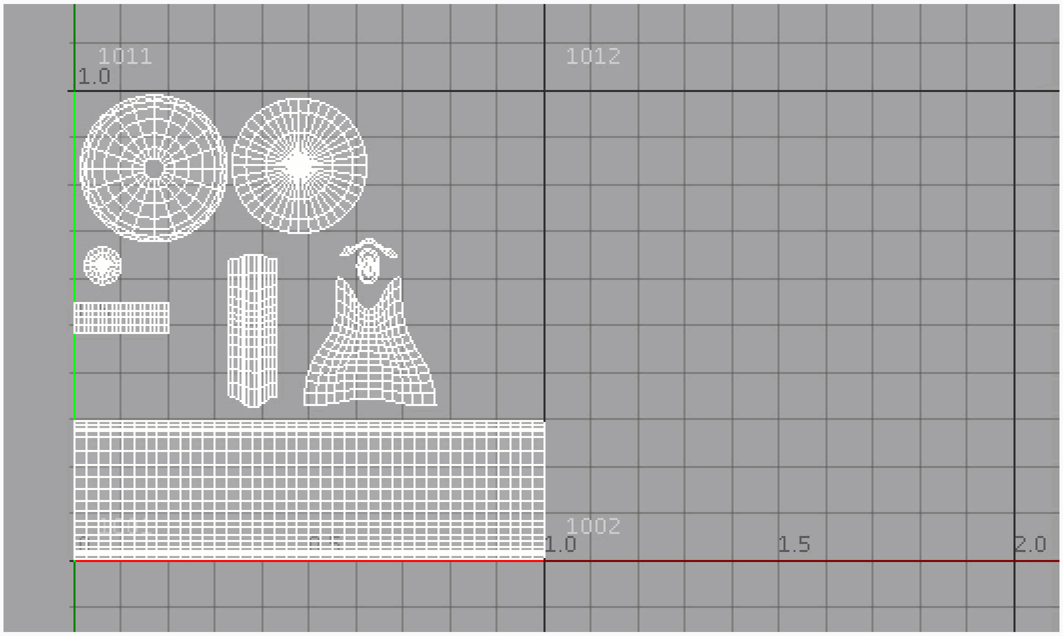
Enable	Toggles the tool off and on. This option is only available for the procedural **Pack UVs** mesh operation.
Use World Transform	When enabled, the mesh operation maintains the world space on the input layer. This option is only available for the procedural **Pack UVs** mesh operation.
Name	Specifies the UV map name. You can easily change it by clicking within the field and typing the new name. This option is only available for the procedural **Pack UVs** mesh operation.

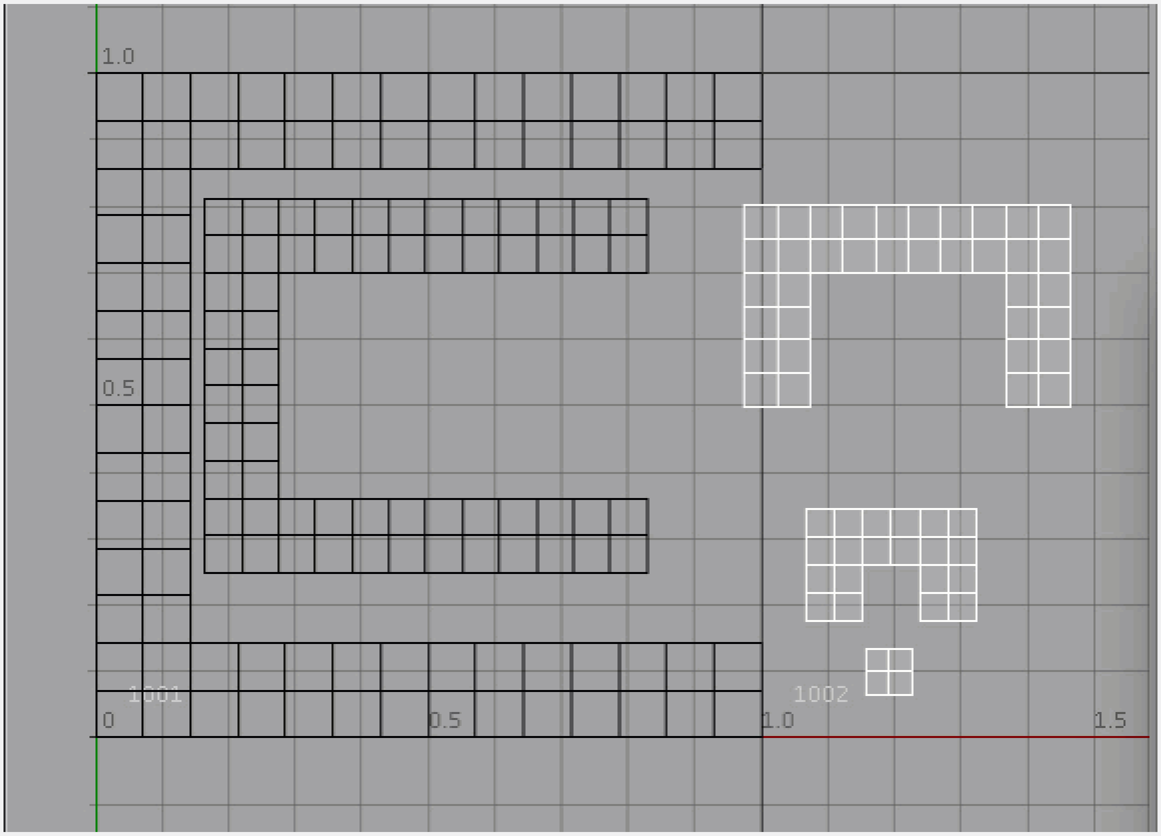
Pack	Controls whether the UV islands can be moved from their current location.
Stretch	Scales and reshapes each UV island to decrease distortions within the UV polygon area/3D polygon area before packing.
Orient	Controls whether the UV islands can be rotated as they are packed.
Keep Size	When enabled, the original size of the UV islands is retained. **Note:** Enabling **Keep Size** disables the effect of the **Stretch** option.
Direction	Provides options for controlling the main orientation of the UV islands. • **Auto** orients them in the direction that best fits. • **Horizontal** orients UV islands in a mostly wide proportion. • **Vertical** orients them in a mostly tall proportion.
Gaps	Controls the final spacing between the islands. Greater values increases the spacing between neighboring UV islands.  **Gaps**: 20 **Set Gaps by Pixel**: Off  **Gaps**: 50 **Set Gaps by Pixel**: Off  **Gaps**: 80 **Set Gaps by Pixel**: Off
Set Gaps by Pixel	When enabled, **Pixel Size** is used instead of the **Gaps** attribute. The **Gaps by Pixel** value is automatically calculated using the following equation: **Gaps** = **Pixel Size** / **Texel Density Image Size**.
Pixel Size	This option defines the gap size in pixel size based on the **Texel Density Image Size**.

	 **Pixel Size**: 10 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024  **Pixel Size**: 30 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024
Texel Density Image Size	The size of the image that is used for the Texel Density to be mapped to. A **Texel Density Image Size** of 512 represents a Texel Density of 512 pixels per UV unit.  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 1024  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 2048

	 **Pixel Size**: 10 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512  **Pixel Size**: 20 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512  **Pixel Size**: 30 **Set Gaps by Pixel**: On **Texel Density Image Size**: 512 For more information on Texel Density, refer to the Texel Density documentation. **Note:** When using the direct modeling version of the **Pack UVs** command, the **Texel Density Image Size** can be adjusted from the **Image Size** option in the **Texel Density** panel on the **UV Tab**. This option is only available for a procedural modeling mesh operation.
Based on Bounding Box	When enabled, treats each island as a rectangular shape, based on the overall width and height.
Lock Stacked	When enabled, keeps stacked islands intact when packing. Overlapping islands must share vertex positions, such as when UV mapped geometry is copied, pasted, and moved in the 3D viewport, or half of a model is mirrored for symmetry.
Pack Region	• Defines the location within the UV space to position the packed UVs. • **Normalized** - Packs UVs into the base 0 to 1 region. • **Nearest** - Packs UVs into the closest UV region based on its initial location. • **UDIM** - Allows you to define a specific UDIM region. When selected, the specific UDIM is specified in the input field below.

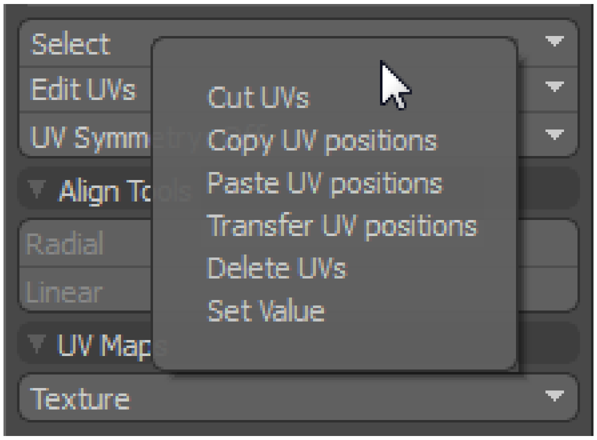
	Tip: UDIM is an automatic UV offset system that assigns an image onto a specific UV tile, which allows you to use multiple lower resolution texture maps for neighboring surfaces, producing a higher resolution result without having to resort to using a single, ultra high-resolution image. For more information, see UDIM Workflow. • **Manual** - Allows you to manually designate a UV space region as a packing area. When selected, the region is specified using the **Region** and **Width/Height** input fields below. • **Tiles** - Defines a maximum rectangular area by number of full regions wide and high, defined with the **Tile U** and **Tile V** values. This targets the packing to fill all the tiled sections within the defined region (without crossing the tile boundaries). • **Polygon Tag** - Groups UV islands by a polygon tag type to separate UDIM spaces. When enabled, you can specify the **Material**, **Part**, **Smoothing Group**, or **Selection Set** option to apply it to. • **By Item** - Packs UVs by mesh item when multiple mesh items are selected, and writes the result into separated UDIM spaces. When **Pack UVs to New Map** is enabled, the new map is named based on the name of the mesh item.
UDIM	When the **Pack Region** option is set to **UDIM**, this option allows you to designate the specific UDIM region as the target for packing the UVs into.
Region U/V and **Width /Height**	When the **Pack Region** option is set as **Manual**, the **Region** and **Width/Height** values designate the target area for packing the UVs into. The **Region U/V** values define the lower-left corner of the packing region and the **Width** and **Height** values define the portion of that area that is used for packing, originating from the defined region position. For example, to fill all of the standard 0-1 UV space, the region would be set to 0,0 and the **Width/Height** would be set to 1.0, 1.0. If the region is set to -1,-1 and the **Width/Height** were set as 2.0, 2.0 then the UVs would fill the entire -1 to 1 on both axes, treating it as one giant UV area to pack into.
Tile U/Tile V	Defines the rectangular tile region by **U** (width) and **V** (height) when using the **Pack Region**'s **Tiles** option.
Polygon Tag	• Groups UV islands by a polygon tag type to separate UDIM spaces. The following options are available: • **Material** - Uses the Polygon Set Material tags applied to the mesh item. It packs

	each polygon tag into a separate UDIM and updates the original texture UV map. • **Part** - Uses the Polygon Set Part tags applied to the mesh item. • **Smoothing Group** - Uses the Polygon Set Smoothing Group tags applied to the mesh item. • **Selection Set** - Uses the selection set specified for the mesh item. For more information, see Using Selection Sets. 
Background Constraint	• When enabled, packs UV islands into the empty spaces available.

	
Pack UVs to New Map	When enabled, the original UV map is retained and individual UV maps for each applied polygon tag are created. If **Pack Region** is set to **Polygon Tag** or **By Item**, UV groups by polygon tag/mesh item are exported to separate new maps per group/item and the new map is named according to the polygon tag or mesh item.
New Texture	Specifies the new texture map name. Use this when you have enabled the **Pack UVs to New Map** option.

Cut, Copy, and Paste UVs

The **Cut**, **Copy**, and **Paste** commands work together to provide you with the means to easily move UV values around between different surfaces. UV editing uses specific **Cut**, **Copy**, **Paste**, and **Delete** commands found in a menu in the default **UV layout** toolbox, under the **Edit UVs** button. Click and hold to open the pop-up with all the command options or right-click to open immediately. All commands work on the currently-selected UV map. Maps can be selected either on the Mesh Item's **Properties** panel, or under the **UVs** section of the **Lists** viewport.



The **Cut UVs** command transfers the UV values for the selected geometry to the internal buffer and then removes the values from the selected geometry. Once values have been stored, they can be applied to a future selection, using the **Paste UV positions** command.

The **Copy UVs** command clones the UV values for the selected geometry into the internal buffer. Once values have been stored, they can be applied to a future selection using the **Paste UV positions** command.

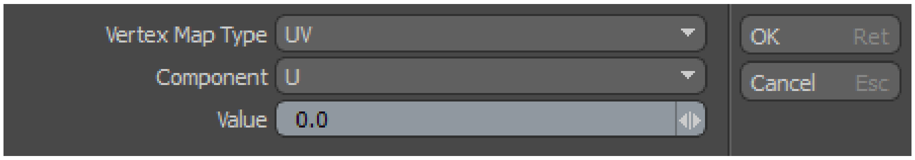
The **Paste UV positions** command applies the UV values stored in the internal buffer to a target selection of geometry. When invoking the **Paste UV positions** command, a dialog box opens, offering you the ability to choose from two different **Modes**: **Connectivity**, which pastes the UV values, based on topological connectivity, and **Selection**, which pastes UV positions.

The **Connectivity** mode allows you to copy UVs from one surface to another with same topology, without worrying about selection order. The second option, **Selection**, pastes UV positions in the buffer to the target polygons, based on their order of selection.

The **Transfer UV positions** command is used to easily transfer UV mapping values between identical topology, eliminating the two step copy and paste process. The transfer process can be applied to multiple surfaces at the same time and is done by way of selecting polygons in a specific order. This is done by selecting two adjoining polygons on the UV source surface, then selecting the exact same two polygons on the target surface in the same order. For multiple targets, continue making the same selections on each subsequent surface. Then, by invoking the **Transfer UV positions** command, the UVs transfer from the source UVs to all the targets at once.

Note: Targets need to have the same topology, meaning the geometry and point count must be identical, but transform, scaling, and deformations can be applied to any of the targets without affecting the transfer results.

The **Delete UVs** command removes the UV values from the selected geometry for the selected UV map.


Vertex Map Type UV

Component U

Value 0.0

The **Set Value** dialog allows you to define a specific UV value for a selection of vertices. This is useful, for example, to force a selection of vertices against a specific UV section edge for instance. To use, select the target vertices and then invoke the command **Set Value** from the menu. In the dialog, the **Vertex Map Type** defaults to the **UV** option. You can then define the component the value is attributed to - the **U** (horizontal direction) or **V** (vertical direction). Then the **Value** input files define the specific UV position along the grid. Press **OK** to assign the value to the target, moving them to the defined location.

Delete UVs

With the UV Delete tool, you can delete UV selections from your UV map. This doesn't affect the mesh selection in the viewport. It only deletes the UV information from the UV map presently displayed in the UV Texture view.

To delete a UV selection:

1. In the 3D Model View, select the geometry.
2. Change the layout to **UV_3D_Split**.
3. In the UV Texture window, select the UV map components to delete.
4. On the **Texture** menu in UV Operators, choose **Delete**.

Export UVs to EPS

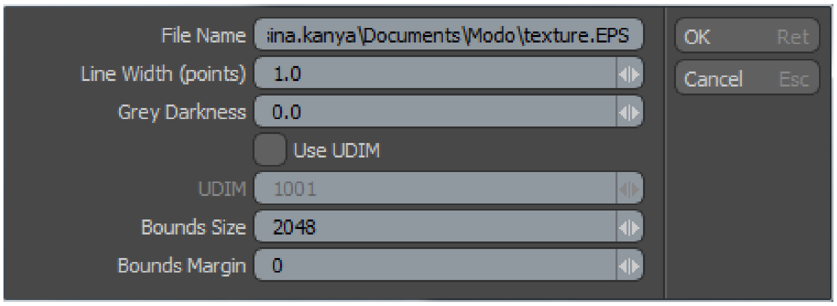
With the **Export UVs to EPS** command, you can export a UV map to an external application for additional processing in the resolution-independent **.eps** vector format. The **.eps** file is useful as a reference point when you open it in an image-editing application (such as Photoshop) to create bitmap textures.

To use this command, follow these steps:

1. Select the target UV map in the **Lists > UV Maps** tab.
2. In the menu bar, click **Texture > Export UVs to EPS**.

   An OS-specific file browser opens.
3. Specify a file name and the location for saving the file.

The **Export UVs to EPS** dialog opens.



File Name :ina.kanya\Documents\Modo\texture.EPS
Line Width (points) 1.0
Grey Darkness 0.0
☐ Use UDIM
UDIM 1001
Bounds Size 2048
Bounds Margin 0

Export UVs to EPS	
File Name	The path and name of the EPS file.
Line Width (points)	The Postscript line width for drawing polygon edges. Measured in points.
Grey Darkness	The Postscript gray level for drawing polygon edges. **0** is white, **1** is black.
Use UDIM	Check this to only export UVs in a specific UV tile.
UDIM	If you have checked **Use UDIM**, choose a UDIM location for the UV source.
Bounds Size	The size of the square EPS bounding box. Measured in points.
Bounds Margin	The size of the horizontal and vertical margins for the square EPS bounding box. Measured in points.

4. Edit the settings you need, then click **OK**.

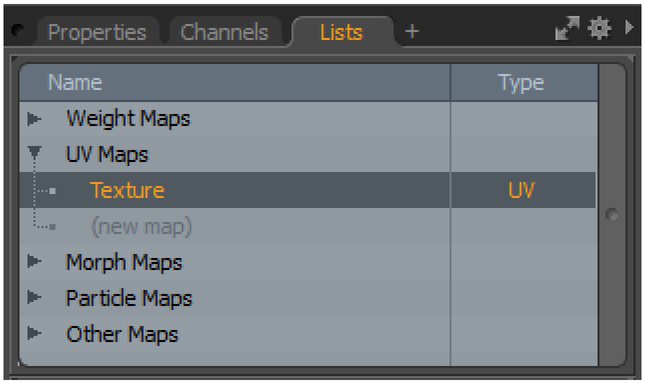
   Modo exports your UV map to **.eps** format.

Export UVs to SVG

With the **Export UVs to SVG** command, you can export a UV Map to an external application for additional processing in the resolution-independent **.svg** vector format. The **.svg** file is useful as a reference point when you open it in an image-editing application (such as Adobe Photoshop) to create bitmap textures.

To use this command, follow these steps:

1. Below the menu bar, click the **UV** layout on the switcher bar.
2. On the right panel, in the **Lists > UV Maps** tab, select the target UV Map.

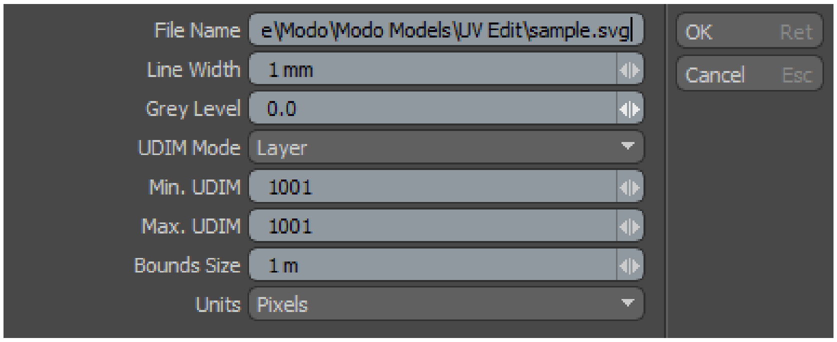

Properties
Channels
Lists
+
Name
Type
Weight Maps
UV Maps
Texture
UV
(new map)
Morph Maps
Particle Maps
Other Maps

3. In the menu bar, click **Texture** > **Export UVs to SVG**.

   An OS-specific file browser opens.

4. Specify a file name, the location to save the file, and click **Save**.

   The **Export UVs to SVG** dialog opens.


OK Ret
Cancel Esc

File Name e\Modo\Modo Models\UV Edit\sample.svg
Line Width 1 mm
Grey Level 0.0
UDIM Mode Layer
Min. UDIM 1001
Max. UDIM 1001
Bounds Size 1 m
Units Pixels

Export UVs to SVG	
File Name	Specifies the path and name of the SVG file.
Line Width	Sets the line width for drawing polygon edges.
Grey Level	Sets the gray level for drawing polygon edges. **0** is white, **1** is black.
UDIM Mode	Allows you to choose how to export UDIM tiles: • **Ignore** - Ignores UDIMs. This exports all UVs, but the SVG view box is set to show only UVs in the **0**-**1** range. • **Layer** - Stacks the UDIM tiles (as specified by **Min. UDIM** and **Max. UDIM**) in separate SVG layers, each within the **0**-**1** range. The SVG view box shows only the **0**-**1** range. • **Tile** - Tiles the UDIMs (as specified by **Min. UDIM** and **Max. UDIM**) and sets the SVG view box to show the entire range.

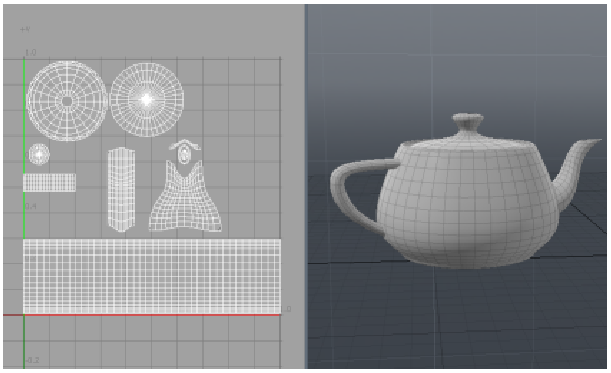
Min. UDIM	Allows you to specify the minimum UDIM value. The first UDIM tile is included in the file export. This represents lower-left corner of the export region. This option is ignored if **UDIM Mode** is set to **Ignore**.
Max. UDIM	Allows you to specify the maximum UDIM value. The last UDIM tile is included in the file export. This represents the upper-right corner of the export region. This option is ignored if **UDIM Mode** is set to **Ignore**.
Bounds Size	Allows you to set a unit value for the output to determine how big the 0 - 1 UV space is in the SVG output. Specifying this option provides better control over the real-world size of the SVG output. This can be helpful for creating templates to use as UV-based textures in external applications and for applications such as packaging or shoe creation, where a pattern with accurate real-world dimensions must be created from a mesh UV Map.
Units	Allows you to specify the units used for the SVG output. The following options are available: • **None** - No units are specified. SVG treats this the same as Pixels. • **Pixels** - Pixels are determined by the importing app. For example, Adobe Illustrator makes this equivalent to Points. • **Points** - Standard typographic points, 1/72 of an inch. • **Millimeters** - Unit of length in the metric system (mm). • **Centimeters** - Unit if length in the metric system (cm). • **Inches** - Unit of length in the (British) imperial system (in). • **Picas** - Obscure typographic standard, 1/6 of an inch or 12 Points.

5. Edit the settings you need and click **OK**.

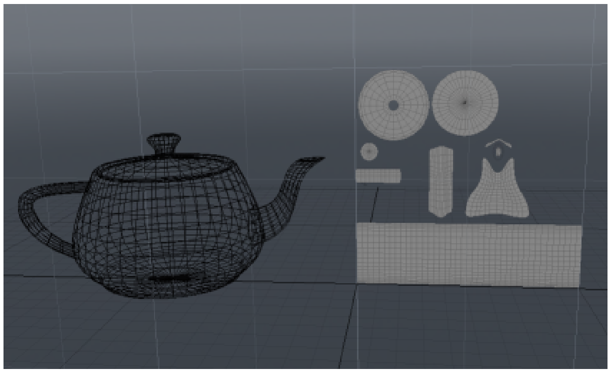
   Modo exports your UV Map to **.svg** format.

Convert UVs to Mesh

The **Convert UVs to Mesh** command converts the currently selected UV map into a new Mesh Item.



A mesh and its UV map.



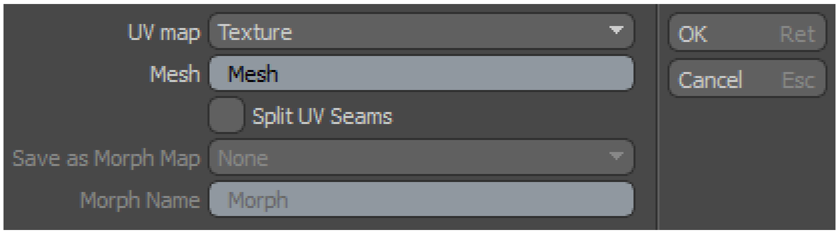
The UV map as a mesh, created by the **Convert UVs to Mesh** command.

For more information on creating UV maps, see Working with UV Maps.

To convert the mesh:

1. Select **Texture** > **Convert UVs to Mesh**.

   This opens the **Convert UVs to Mesh** dialog.



UV map: Texture

Mesh: Mesh

☐ Split UV Seams

Save as Morph Map: None

Morph Name: Morph

2. Select the source map from the **UV map** dropdown.
3. Edit the name of the mesh if needed. Click **OK**.

   The new mesh is added to the **Item List**.

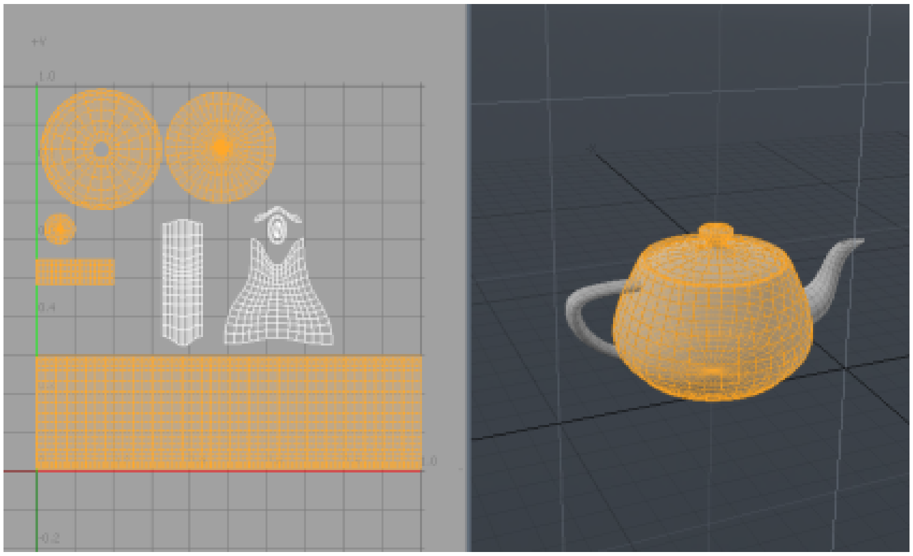
Note: As UV maps have no inherent size or scale, Modo converts the UV 0-1 grid to meters. So when converting from a normalized UV, the resulting Mesh Item fits within the 0m to 1m area of the XY plane.

Preserving Vertex Order

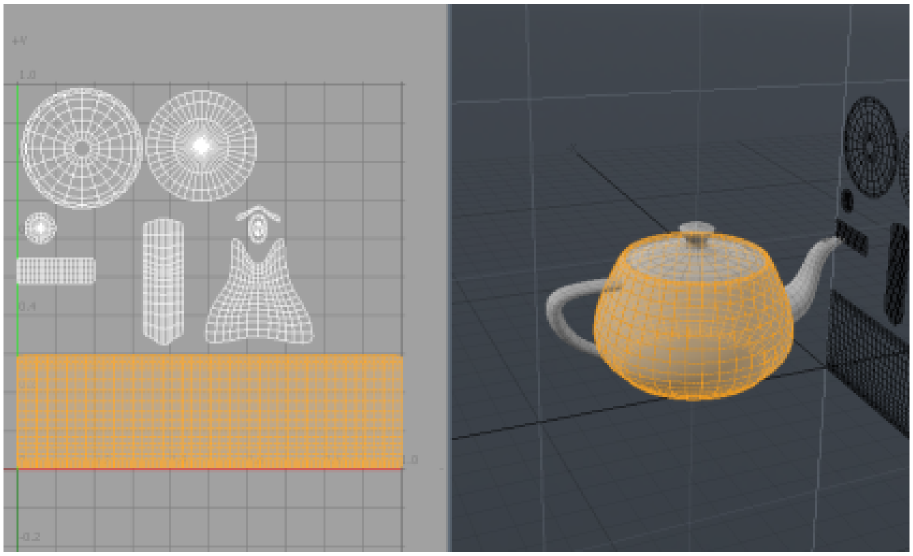
For some operations it is useful to match the order of the vertices in the source mesh with the new (UV) mesh. To preserve the vertex order, the source mesh topology must be the same as the new mesh topology. However, there is usually a mismatch as the UV map consists of a series of islands with different seams to the source mesh. To remedy this, you can split the source mesh to form seams that match the UV mesh, and therefore create a matching topology.

To split the source mesh to match the UV mesh:

- Check the **Split UV Seams** option in the **Convert UVs to Mesh** dialog when converting a mesh.



A UV map of the default teapot primitive. The continuous polygon body is selected and the matching UV islands are shown. You can see that their topology differs.



A new mesh is generated from the UV map using **Split UV Seams**. The source mesh is now split to match the topology of the UV mesh. This is demonstrated by reselecting the body - notice how topology of the body now matches the UV island (the lid and base of the teapot are now split from the body).

Using Morph Maps

Instead of generating the UV map as a new mesh, it can be used to create a morph map for the current mesh. This is made possible by preserving the vertex order when choosing **Split UV Seams**. With the same vertex order, the edges in the source mesh match the edges of the new mesh and the morph is seamless.

Note: The Morph Map can either be **Relative** or **Absolute**. For more information, see Morph Maps

To generate a Morph Map:

1. When converting a UV to a mesh, check **Split UV Seams** on the **Convert UVs to Mesh** dialog.
2. Select the morph map type from the **Save as Morph Map** dropdown: **Relative** offsets a particular vertex relative to its current position. **Absolute** positions a vertex to a specific absolute position
3. If preferred, edit the **Morph Name** to assign a new name to the Morph Map. Click **OK**.

   The new Morph Map is displayed in the **Lists** viewport under **Morph Maps**.

Procedural Convert UV to Mesh

The Convert UVs to Mesh command is available as a mesh operation.

Tip: For more information on procedural modeling and mesh operations, see Procedural Modeling with MeshOps.

To use the Convert UV to Mesh operation:

1. Add a new mesh layer where you want to place the new geometry: open the **Mesh Ops** tab on the right panel and click **Add Item**, then under **Items**, double-click **Mesh**.

   Tip: If the **Mesh Ops** tab is not visible on the right panel, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.

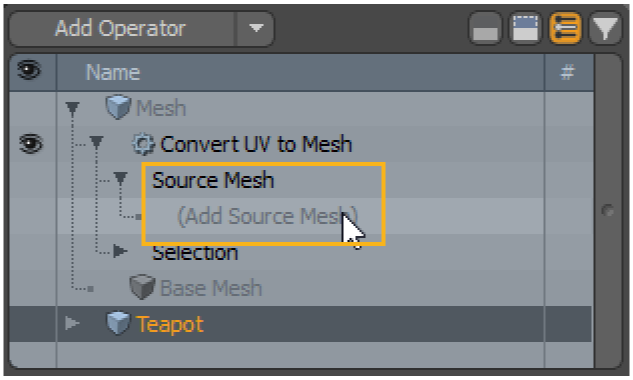
   This adds a new mesh layer to your scene.
2. Select the new mesh layer in the Item List at the top of the **Mesh Ops** tab and click **Add Operator**. Under **Mesh Operations** > **UV**, double-click **Convert UV to Mesh**.

   The **Convert UV to Mesh** operation is added to the Mesh Operations list.

3. To specify what geometry it should use, expand it in the list by clicking the small arrow in front of its name.

   This reveals the mesh operation's inputs: **Source Mesh** and **Selection**.

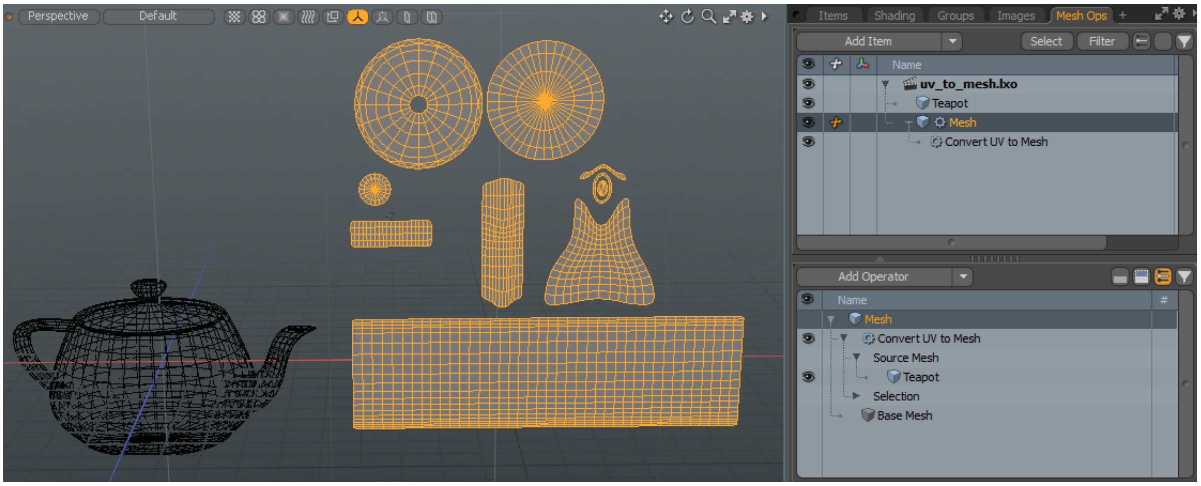
4. Under **Source Mesh**, click **(Add Source Mesh)**.


Add Operator
Name
#
Mesh
Convert UV to Mesh
Source Mesh
(Add Source Mesh)
Selection
Base Mesh
Teapot

   This opens the **Add Operation** dialog.

5. Double-click the mesh you want to use.

   The converted UV map appears in the 3D viewport.


Perspective
Default
Items
Shading
Groups
Images
Mesh Ops
Add Item
Select
Filter
Name
uv_to_mesh.lxo
Teapot
Mesh
Convert UV to Mesh
Add Operator
Name
#
Mesh
Convert UV to Mesh
Source Mesh
Teapot
Selection
Base Mesh

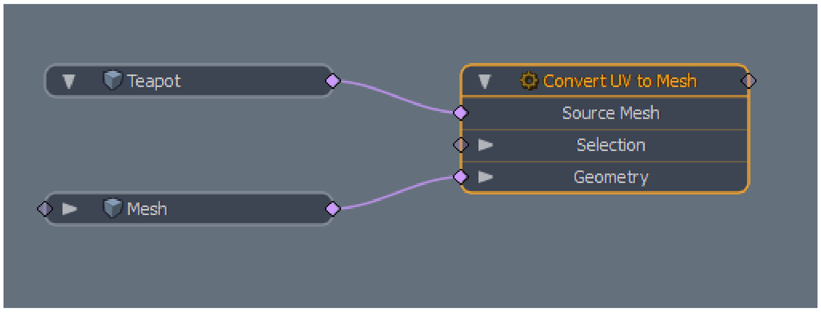
Convert UV to Mesh in the Schematic

In the Schematic viewport, the Convert UV to Mesh node has the same inputs as the mesh operation.

To add the Convert UV to Mesh node, click Add... at the top of the viewport, then click **Mesh Operations** > **UV**, and double-click **Convert UV to Mesh**.

Connect the mesh you want to convert to the **Source Mesh** input of the **Convert UV to Mesh** operator. This places the converted mesh to the same mesh layer as the original mesh. To place it in a new mesh layer, as in the example above, add a new **Mesh** node and connect it to the **Geometry** input of the **Convert UV to Mesh** node.

Note: If nothing is connected to the **Source Mesh** input, Modo reads the UV from the **Geometry** input.



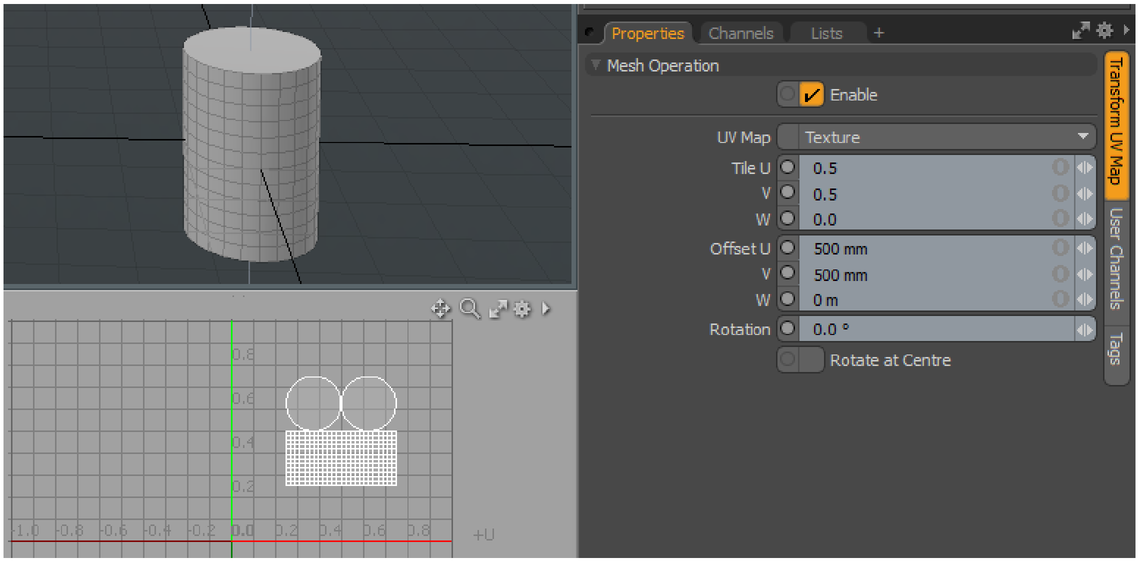
Tip: For more information on working in the Schematic viewport, see Schematic Viewport.

Transform UV Map

The procedural Transform UV Map tool lets you move, scale, and rotate a UV map from within the Mesh Operations List. The tool can be added to the stack from the Procedural Preset Browser and has a **UV Map** property to select the affected UV map.

The tool properties have **Tile** settings to scale the UV Map, **Offset** settings to position the UV map, and a **Rotation** setting.

The image below shows the tool operating on the UV map of a cylinder. The size has been halved and the map has been offset from the origin by 500 mm in U and V.



Mesh Operation

☑ Enable

UV Map: Texture
Tile U: 0.5
V: 0.5
W: 0.0
Offset U: 500 mm
V: 500 mm
W: 0 m
Rotation: 0.0 °
☐ Rotate at Centre

Transform UV Map Properties

Transform UV Map	
UV Map	Selects the UV map to transform. If this is set to **(none)**, the tool has no effect.
Tile U, V, W	The size of the UV Map relative to the original size in 1 m x 1 m x 1 m UVW space.
Offset U, V, W	The distance from the original position in U, V, and W space.
Rotation	The angle of rotation for the UV map.
Rotate at Centre	Check this to rotate the UV map around the centre of the map. If this is unchecked, the UV map rotates around U = 0, V = 0.

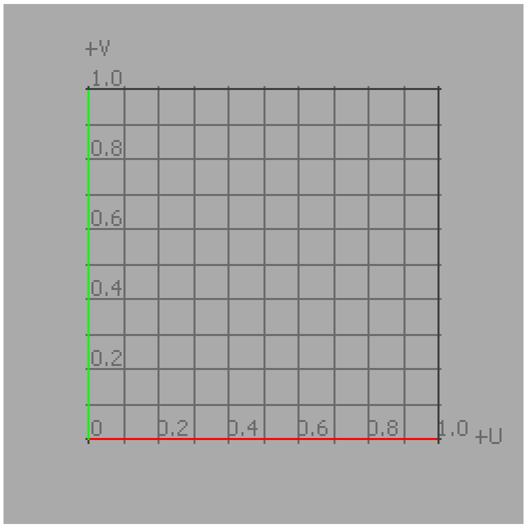
UDIM Workflow

UDIM is an enhancement to the UV mapping and texturing workflow that makes UV map generation easier and assigning textures simpler.

At its core, UDIM is simply an automatic UV offset system that assigns an image onto a specific UV tile, which allows you to use multiple lower resolution texture maps for neighboring surfaces, producing a higher resolution result without having to resort to using a single, ultra high-resolution image. In order to understand the benefits of the UDIM workflow, it's best to have a basic understanding of how UV mapping works.

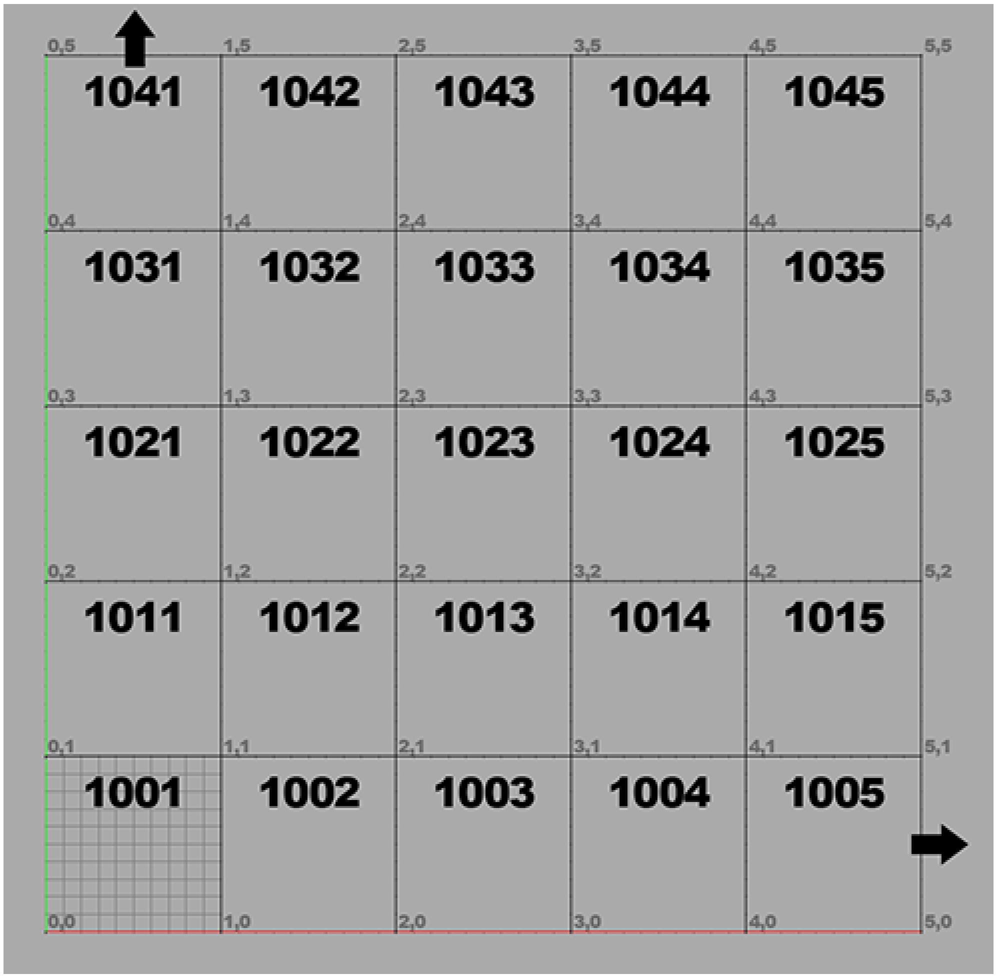
UV Mapping Basics

UV mapping is the translation of a three-dimensional surface into the two-dimensional coordinate system of flat bitmap images (or vice-versa depending on how you look at it). Each vertex of the originating surface is given a coordinate on the UV map, and the pixel values in-between are interpolated across the flat polygon surface for highly accurate texturing control (with the U and V axes names chosen specifically as to not cause confusion with the X, Y, and Z axes for the actual model). The UV map coordinate values are organized into a grid system, where the image itself appears in the 0-1 area of the grid and the area outside of this tile simply repeats the image outward. Because of this fact, traditionally only the UV 0-1 space was ever really used (well, outside of controlling how a surface tiled an image).



UDIM Basics

With a UDIM workflow, each whole tile section, 0-1, 1-2, 2-3, and so on, can have its own independent image map assigned to it, all within the same surface. Each image can have its own resolution as well. The UDIM workflow is simply a notation format that names images accordingly, so they are automatically assigned to a specific UV tile. The numbering starts at the UV origin and moves in the positive direction outward ten tiles (to 1010), it then repeats the rows stacked on top of each other, moving upward 9999 rows. This illustration demonstrates a small section of the overall UDIM tile area, as it relates to the UV space (the upper image corresponding to the lower-left corner of this one).

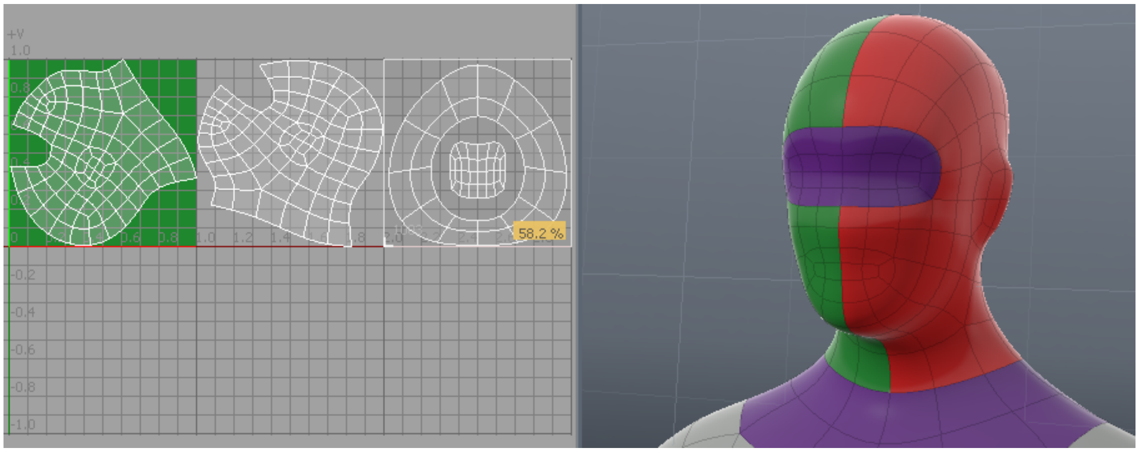


1041	1042	1043	1044	1045
1031	1032	1033	1034	1035
1021	1022	1023	1024	1025
1011	1012	1013	1014	1015
1001	1002	1003	1004	1005

0,5
1,5
2,5
3,5
4,5
5,5
0,4
1,4
2,4
3,4
4,4
5,4
0,3
1,3
2,3
3,3
4,3
5,3
0,2
1,2
2,2
3,2
4,2
5,2
0,1
1,1
2,1
3,1
4,1
5,1
0,0
1,0
2,0
3,0
4,0
5,0

Mapping Your Model

Using a UDIM workflow begins with the UV mapping of the actual geometry. Decisions are made at this stage as to how the model is divided, typically based on what degree of resolution is required for the project. For an average character project, the tiles could be broken up into the head in one tile, the front of the body on another, and the back on another still. Higher resolution requirements might necessitate having hands and feet on separate tiles as well. For information on Modo's UV mapping tools, see the Working with UV Maps or the UV Mapping topic.



For individual parts, it's easiest to create and adjust the initial UV map for a given section in the UV 0-1 tile as many tools default to this locale. Once the map is laid out as desired, it can be moved with the **Transform** tool (**W** keyboard shortcut) to the desired UDIM tile. When working in the **UV** interface layout, it's easiest to simply enter whole number UV values to offset to a specific location, for example U 4.0 and V 3.0 for UDIM 1035, though it's most economical to keep the tiles consecutively placed, but never spanning more than ten tiles in any horizontal row.

The UDIM Indicator tool helps you move geometry to a UDIM tile. You can use the arrows on the indicator to shift the position of the selected geometry. The tool provides a range of useful features to help when UV mapping with UDIMs.

Creating UDIMs Using the UDIM Wizard

UDIM texturing requires a set of images to use as tiles across the UV map. You may already have these images, or you may need to build them from scratch. For the latter, Modo provides a UDIM creation wizard to automatically generate a set of blank, colored tiles that may be customized later using Modo's tools or an external editor. The wizard let's you define key characteristics such as the UDIM image range, image resolution, image format, and color. The resulting tiles are grouped into a clip group in the Images/Clips List.

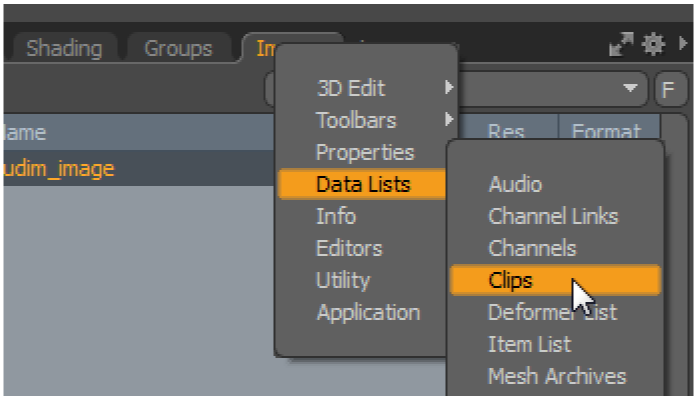
To Use the UDIM Wizard

To create a UDIM group using the wizard, use the **(new udims)** option from the menu under the **Add Clip** button. You can do this by:

- adding it as a layer in the Shader Tree
- Shader Tree

The steps below show you how to generate the UDIMs in the **Clips List**.

1. Open the Clip Browser. By default, the **Clip Browser** is found under the **Images** tab. Alternatively, click **+** next to the viewport tabs and select **Data Lists** > **Clips**.

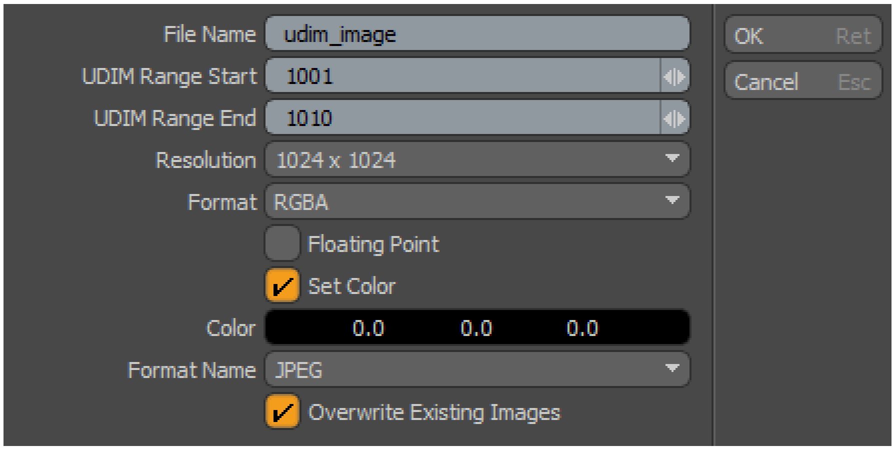

Shading
Groups
Im
3D Edit
Toolbars
Properties
Data Lists
Info
Editors
Utility
Application
Audio
Channel Links
Channels
Clips
Deformer List
Item List
Mesh Archives
ame
udim_image
Res
Format
F

2. Click **Add Clip** and select **(new udims)**.

   This opens a directory browser.

3. Browse to the directory in which to create the UDIMs, and click **Select Folder**.

   The file browser closes, and the **UDIM Wizard** opens.



File Name udim_image
UDIM Range Start 1001
UDIM Range End 1010
Resolution 1024 x 1024
Format RGBA
☐ Floating Point
☑ Set Color
Color 0.0 0.0 0.0
Format Name JPEG
☑ Overwrite Existing Images

OK Ret
Cancel Esc

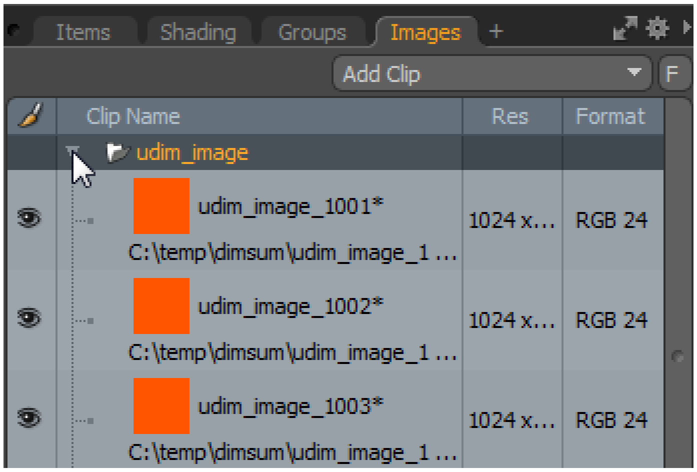
4. Edit the fields in the **UDIM Wizard** as required:

UDIM Wizard	
File Name	The base file name for the created UDIM images (tiles). The UDIM images are named in the format: **[filename]_[UDIM number].[format extension]**
UDIM Range Start	The first UDIM tile number in the range.

UDIM Range End	The last UDIM tile number in the range.
Resolution	The resolution of each tile.
Format	Defines the color format of the tiles. Choose from: • **Gray** - Grayscale. • **RGB** - Opaque RGB color images. • **RGBA** - RGB color images with an alpha channel to support transparency.
Floating Point	Choose whether to use floating-point values or not for the image. Floating point images (also known as HDRI, High Dynamic Range Images) give a broader range of potential color values. By default, this checkbox is off, and Modo creates an integer image that defines colors in 8 bits per channel (0 to 255). A floating point image can use many more values, and are useful for light sources or for detailed displacement maps.
Set Color	Choose whether or not the image has a base color, or base transparency, to start. If **Set Color** is off, new images are completely blank (with no color or alpha pixels). If **Set Color** is on, each RGBA image has an opaque alpha (**100%** so all pixels are visible) and a base color as specified in the **Color** option.
Color	The base color of each tile.
Format Name	The image format used for the tiles.
Overwrite Existing Images	Check this to overwrite any images of the same name in the target folder.

5. Click **OK** to generate the images.

   The clip group is added to the **Clip List**.
6. Open the clip group to inspect the tiles.



Clip Name	Res	Format
udim_image		
udim_image_1001* C:\temp\dimsum\udim_image_1 ...	1024 x...	RGB 24
udim_image_1002* C:\temp\dimsum\udim_image_1 ...	1024 x...	RGB 24
udim_image_1003* C:\temp\dimsum\udim_image_1 ...	1024 x...	RGB 24

Note: You can also generate a clip group in the **Shader Tree** using the **(new udims)** option.

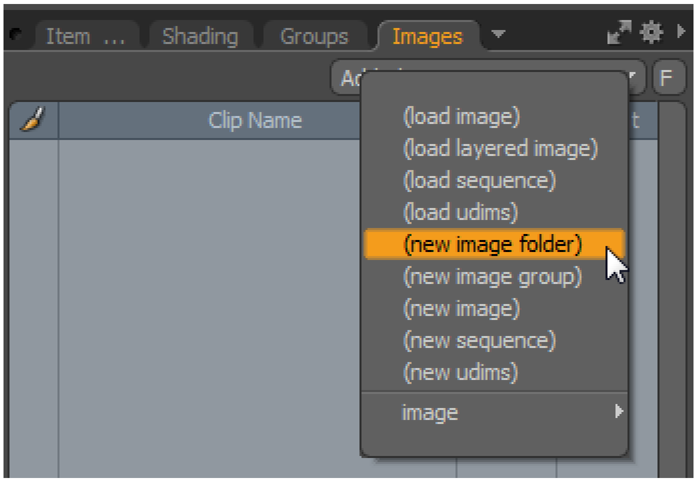
Creating UDIMs Manually

As an alternative to using the UDIM wizard, multiple UDIMs can be imported from an external painting app (such as Mari) or created one-by-one in Modo. This is done as follows:

1. Make an image folder.
2. Import or create the images.
3. Set the UDIM properties.

To Make an Image Folder

You need to make a new folder to hold a set of UDIM images. Click on the **Images** (or **Clips**) tab. This tab is typically found in the same viewport group as the **Shader Tree** viewport. Click the **Add Clip** button and choose the **(new image folder)** option.


Item ...
Shading
Groups
Images
Clip Name
(load image)
(load layered image)
(load sequence)
(load udims)
(new image folder)
(new image group)
(new image)
(new sequence)
(new udims)
image

This action creates a folder in the **Images** viewport.

Note: The **Images** tab and the **Clips** tab show the same viewport. Depending on your set-up, you may only see the **Clips** tab.

To Import UDIM Images

Select the new folder in the **Clips** tab, and use the **(load Image)** option to import multiple images from a single location, or singly import images from multiple locations. You can also drag-and-drop images into the image group folder.

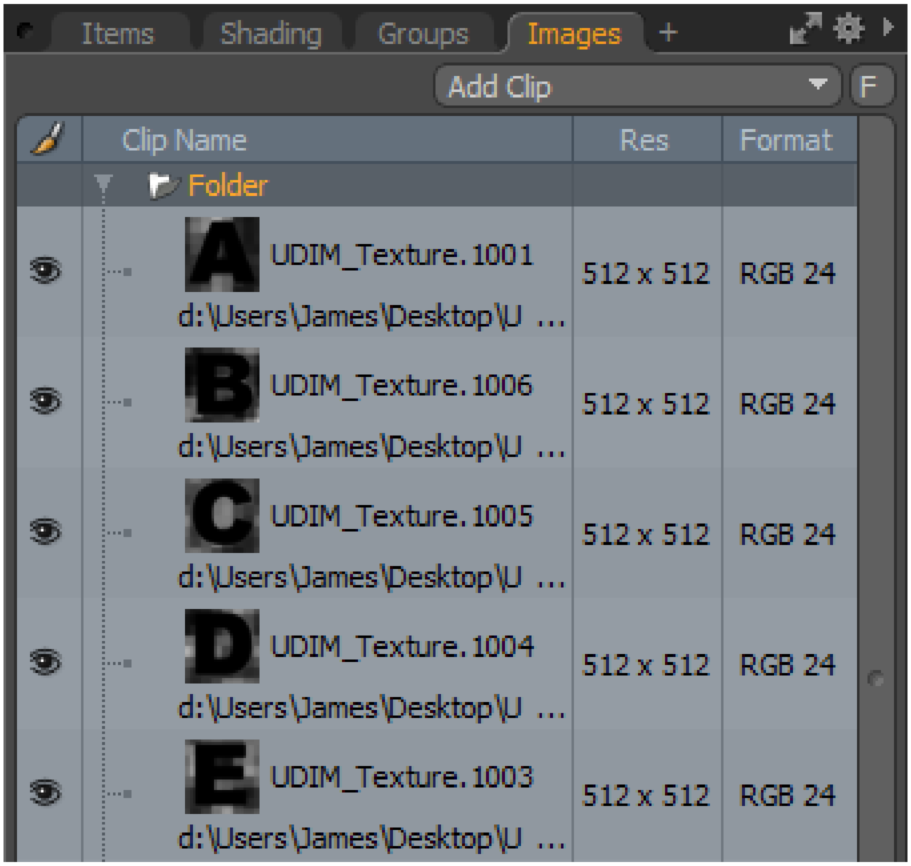
To Create a UDIM Image Manually

Instead of importing images, you can use the **(new image)** option to create blank images. Each image needs to be manually named according to its UDIM tile location. Make sure to adhere to a proper naming convention:

<filename>.<UDIM value>.<format extension>

For example, **hero_wings_diffuse.1012.EXR**

Tip: You can rename the folder if required.



Items
Shading
Groups
Images
Add Clip
F

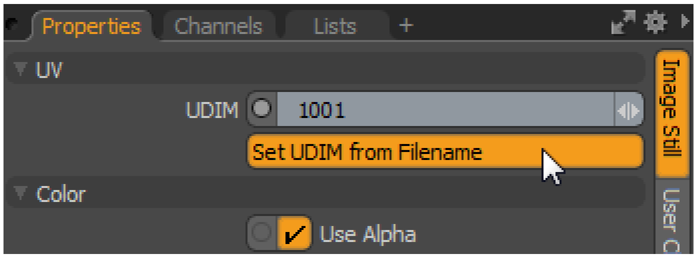
Clip Name	Res	Format
Folder		
UDIM_Texture.1001 d:\Users\James\Desktop\U ...	512 x 512	RGB 24
UDIM_Texture.1006 d:\Users\James\Desktop\U ...	512 x 512	RGB 24
UDIM_Texture.1005 d:\Users\James\Desktop\U ...	512 x 512	RGB 24
UDIM_Texture.1004 d:\Users\James\Desktop\U ...	512 x 512	RGB 24
UDIM_Texture.1003 d:\Users\James\Desktop\U ...	512 x 512	RGB 24

The next step is to assign the proper UDIM values to each bitmap image.

To Set UDIM Properties

In order to use the images as UDIMs, each file must have its **UDIM** property set. This holds the position of the UDIM tile using the UDIM numbering scheme. You can set this property on all images at once using the number in their file name.

1. Select all the images within the folder: click the top-most image to select it, then **Shift**+**click** the image in the bottom layer.
2. In the **Properties** viewport, under the **Image Still** sub-tab, click the **Set UDIM from Filename** button to assign the values automatically.



Properties
Channels
Lists
UV
Set UDIM from Filename
Color
Image Still
User C

UDIM: 1001

☑ Use Alpha

Now the clip folder is ready to be applied to a surface.

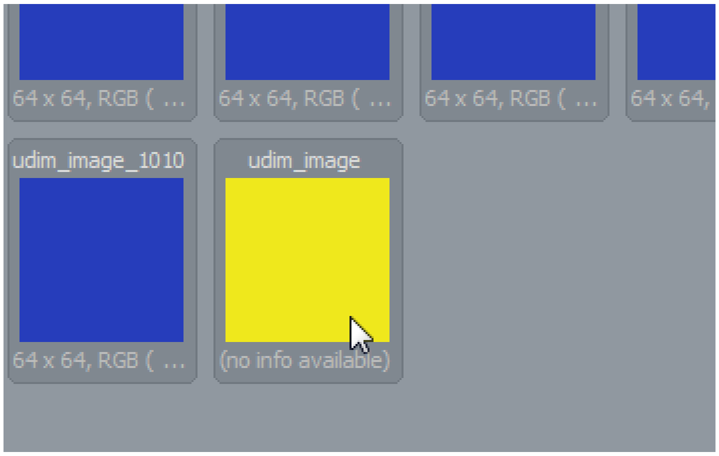
Texture Mapping

All the images in a UDIM folder can be assigned to a surface in the **Shader Tree** as a single layer.

1. In the **Shader Tree**, click the **Add layer** button and select **Image Map** > **(use clip browser)**.

   The **Clip Browser** opens.

2. Select the UDIM folder. This appears as a single icon, labeled with the images' prefix.

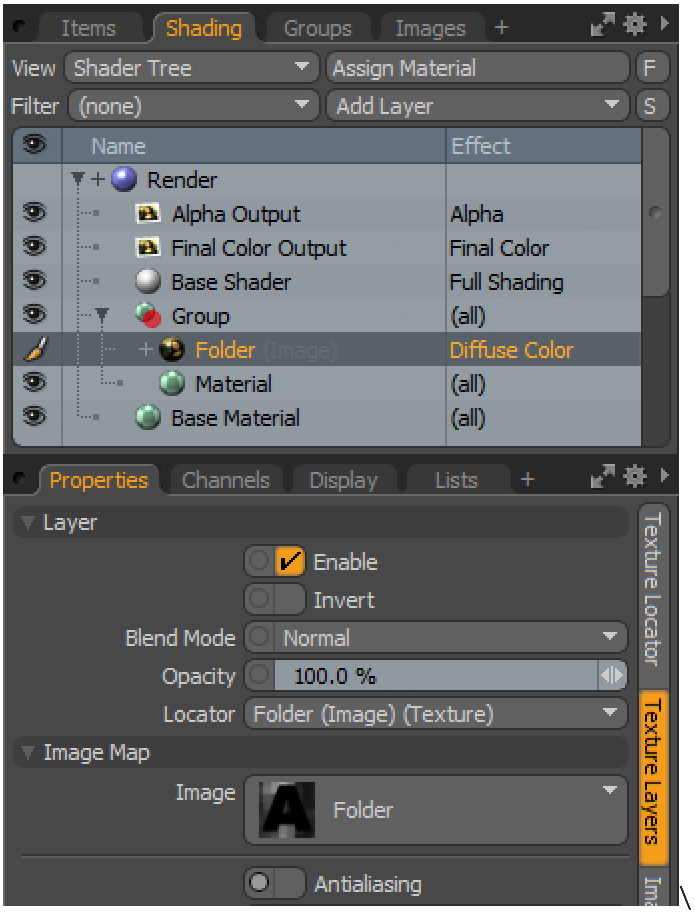


Note: Please refer to the Adding Images topic for information about adding and using texture maps, and the Shader Tree topic for information on working with **Shader Tree** layers.

If your textures aren't visible, it may be necessary to check that the correct UV map is specified in the associated **Texture Locator**.

Note: For older models it may be necessary to make sure the **Use Clip UDIM** option is enabled.

Once assigned with the proper **Effect** type, it is ready for rendering. Or, if this is the beginning of the process, then it is ready for texture painting or baking.


Items
Shading
Groups
Images
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Properties
Channels
Display
Lists
Texture Locator
Texture Layers
Ima

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Group	(all)
Folder (Image)	Diffuse Color
Material	(all)
Base Material	(all)

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Locator: Folder (Image) (Texture)

Image Map

Image: Folder

○ Antialiasing

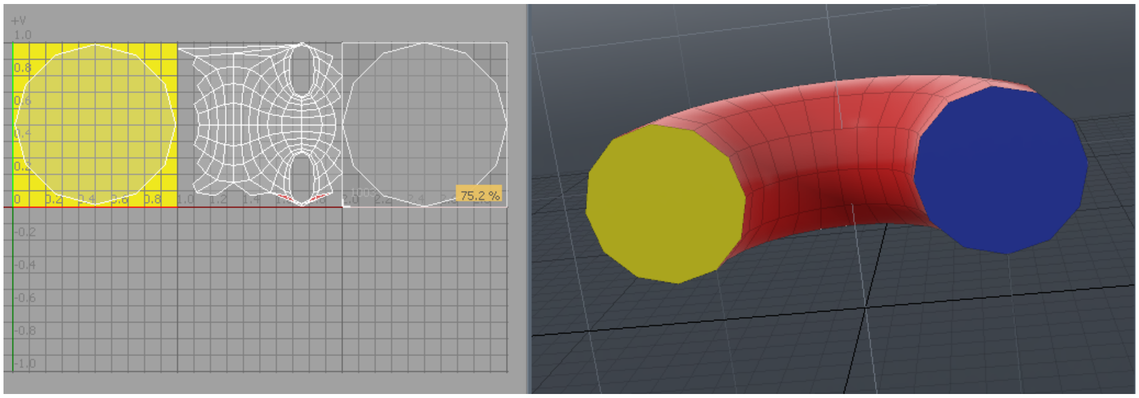
Tip: Each folder that is assigned to the **Shader Tree** using the steps above, can only have a single layer **Effect** (such as **Diffuse Color** or **Bump**). Individual images within a folder can't have their own unique **Effect** setting. And so it's important to make sure that the images are arranged into their own folders, with a folder for each **Effect** type.

Viewing the UDIM Tiles on the UV Map

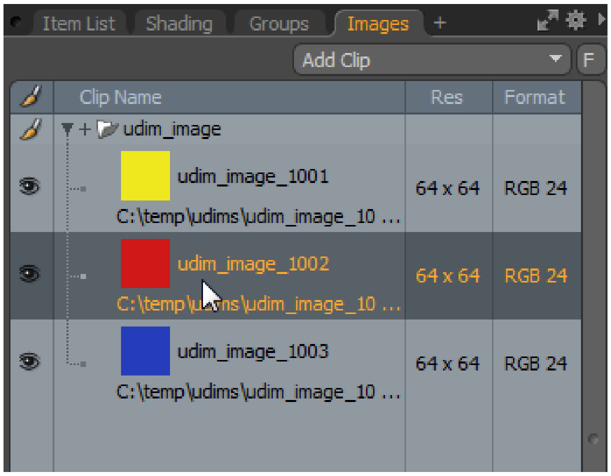
When UDIMs have been added to a material on the **Shader Tree**, you can see the position of the UDIM tiles in the **UV map** viewport. The image below shows a simple mesh that has been UV mapped using three UDIM tiles, each of a different color.

Tip: The UV viewports shows the 0 to 1 space by default. The UDIM tile number display is enabled by default but it is only shown if more than the 0 to 1 space is shown.

Note: The UV viewport in the image below shows a UV area with **U High Span** of **3**, to cover the three UDIM tiles. This can be adjusted in the **UV Viewport Properties** window, which is opened by pressing **O** when in the **UV Viewport**.

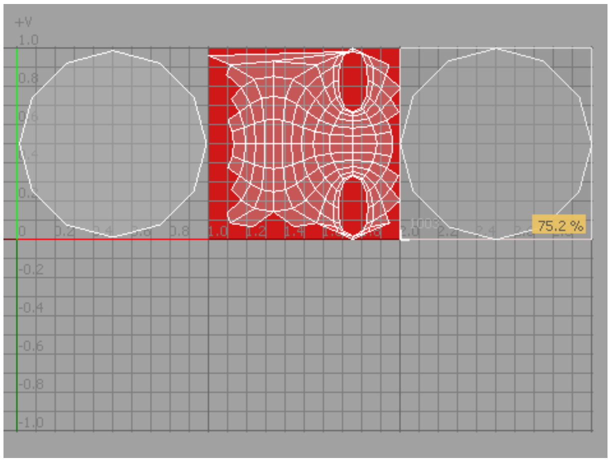


1. Click on the **UVEdit** tab to open the **UV Map** viewport.
2. Open the **Clips List** and click on a UDIM image.



Clip Name	Res	Format
udim_image		
🟨 udim_image_1001 C:\temp\udims\udim_image_10 ...	64 x 64	RGB 24
🟥 udim_image_1002 C:\temp\udims\udim_image_10 ...	64 x 64	RGB 24
🟦 udim_image_1003 C:\temp\udims\udim_image_10 ...	64 x 64	RGB 24

The image displays on the UV map.


+V
1.0
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
2.0
2.2
2.4
75.2 %

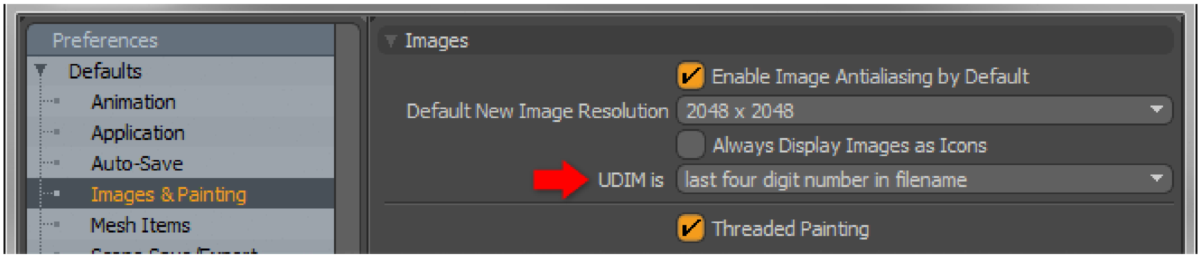
UDIM Baking

If you're using UDIM placement in your UV map, you can bake each tile to a separate UDIM image. For more details, see Bake to UDIM Tiles.

Mari Workflow

Modo's UDIM automation options really shine when working with an external texturing application, such as Mari, where image map UDIM naming is automatic. When working on complex models, where there are possibly hundreds of images associated with it, this is definitely the way to go as Mari saves the images directly, then Modo imports entire folders of images and assigns the UDIM offset values automatically, based on the file names from Mari.

When working with Mari specifically, make sure the **Preferences** option for UDIMs under the **Defaults** > **Images and Painting** menu is set as **last four digit number in filename**. This ensures that the filename UDIMs are read properly when imported.


Preferences
Defaults
Animation
Application
Auto-Save
Images & Painting
Mesh Items
Images

☑ Enable Image Antialiasing by Default

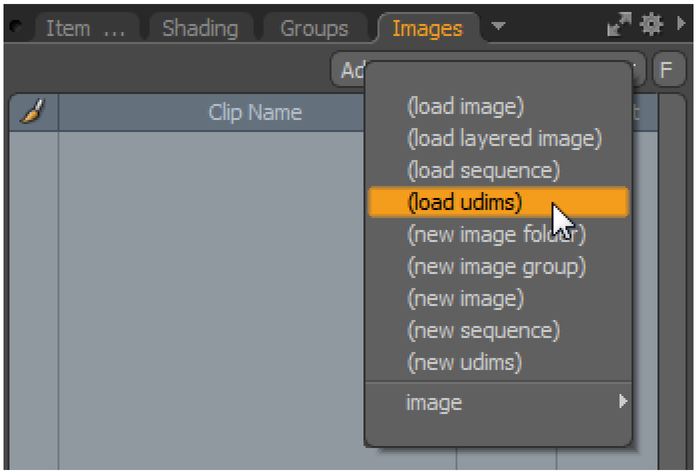
Default New Image Resolution: 2048 x 2048

☐ Always Display Images as Icons

UDIM is: last four digit number in filename

☑ Threaded Painting

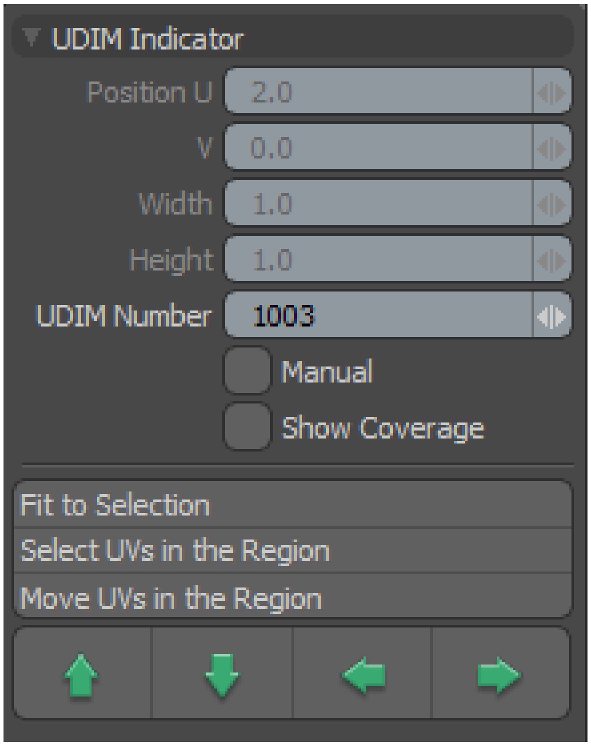
Loading an entire folder of images saved in Mari into an **Image Folder** in Modo and setting the UDIM values for each image is done in one operation. In the **Images** viewport panel, select the **Load UDIMs** option.


Item ...
Shading
Groups
Images
Ad
F
Clip Name
(load image)
(load layered image)
(load sequence)
(load udims)
(new image folder)
(new image group)
(new image)
(new sequence)
(new udims)
image

This opens an OS-specific dialog where you can navigate to any folder of UDIM images and multi-select all the associated images and press the **Open** button. This loads the images into an image group "Folder" and specifies the UDIM value for each automatically. The image can now be assigned in the **Shader Tree** like any other texture layer.

UDIM Indicator

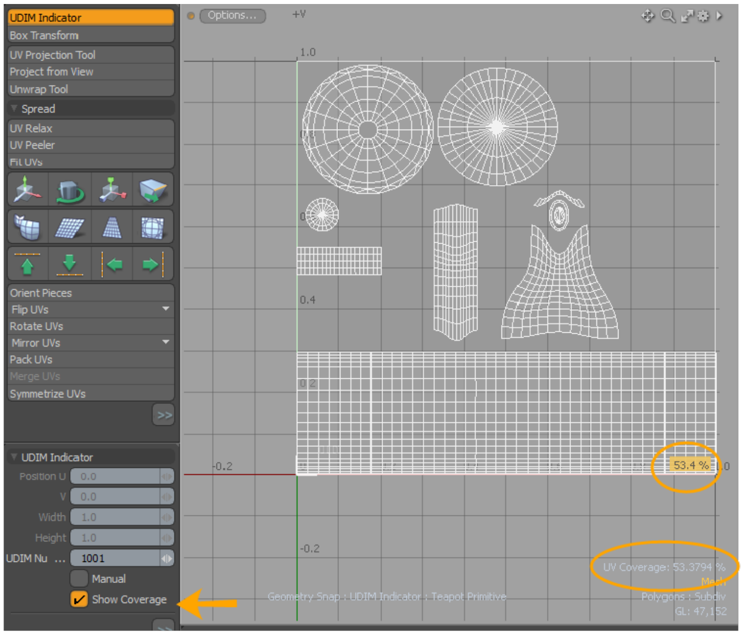
The **UDIM Indicator** is an option that allows for a more fluid UDIM workflow. Its main function provides an outline to help target the current UDIM space, as well as some optional options to easily move UV elements around. The **UDIM Indicator** option is enabled in the **UVEdit** layout, found at the top of the default toolbox. When enabled, a translucent rectangle appears around the active UDIM address location, and the additional options display in the tool **Properties** panel.



UDIM Indicator
Position U
2.0
V
0.0
Width
1.0
Height
1.0
UDIM Number
1003
Manual
Show Coverage
Fit to Selection
Select UVs in the Region
Move UVs in the Region

The following options are available for the UDIM Indicator:

- **Position U/V/Width/Height** - generally, for the most fluid UDIM workflow, you have the **Manual** option disabled and you can enter a **UDIM Number** to position the indicator frame. For specific tasks, you may want to manually position the frame. To do this, enter the **Manual** options and adjust the **Position U** and **Position V** values. These values move the indicator frame originating from its lower-left corner. The **Width** and **Height** values determine the size of the frame itself.
- **UDIM Number** - positions the UDIM indicator frame to the defined UDIM address location in the UV viewport.
- **Manual** - enables you to position the UDIM indicator frame to a specific location, within the UV viewport, which doesn't match up to the UDIM address location standard. Also works in conjunction with the **Fit to Selection** option to allow simplified translations of selected UV elements.
- **Show Coverage** - displays a percentage value in the lower-right corner of the UDIM indicator frame, representing the amount of area of the total region that is covered by the geometry. This option is useful for game workflows, when trying to maximize the amount of texture space being used for the target surface.


UDIM Indicator
Box Transform
UV Projection Tool
Project from View
Unwrap Tool
Spread
UV Relax
UV Peeler
Fit UVs
Orient Pieces
Flip UVs
Rotate UVs
Mirror UVs
Pack UVs
Merge UVs
Symmetrize UVs
Options...
+V
1.0
0.4
0.2
-0.2
-0.2
1.0
53.4 %
UV Coverage: 53.3794 %
Geometry Snap : UDIM Indicator : Teapot Primitive
Polygons : Subdiv
GL: 47,152

UDIM Indicator

Position U 0.0

V 0.0

Width 1.0

Height 1.0

UDIM Nu ... 1001

☐ Manual

☑ Show Coverage

- **Fit to Selection** - fits the UDIM Indicator frame exactly around the outside to the currently selected geometry. The **Manual** option must be enabled first for this option to work. When used as such, the **Fit to Selection** option allows you to use the **Nudge** and **Move** options to easily translate elements around in UV space.
- **Select UVs in the Region** - selects, in the 3D viewport, all the geometry that is fully contained within the UDIM indicator frame.
- **Move UVs in the Region** - opens a dialog box where you can enter a specific UDIM address location and all selected or fully contained geometry within the UDIM indicator frame is moved to the new location, and the **UDIM Number** value is updated to reflect the new UDIM indicator frame location.
- **Arrow Buttons** - works similar to the **Move UVs in the Region** option. The arrow buttons allow you to move the UDIM indicator frame contents over to the adjoining UDIM position, based on the arrow button direction that is pressed.

Painting

Painting in Modo is designed to be simple to use, while providing powerful and intuitive toolsets that empower you to realize your vision with the least amount of strain. Painting Overview provides an overview of how Modo's brushing engine works, while the subsequent sections detail the individual brushes and related tools for texturing and detailing your creation.

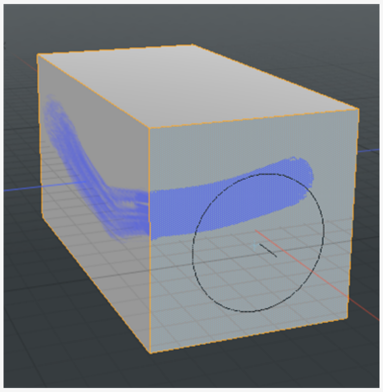
Painting Overview

Modo provides you with the ability to sculpt your models in a variety of ways and then texture them easily in one integrated environment and work non-linearly. This flexibility means you can model, sculpt, apply a basic texture, and then go back and model some more. As with all the tools in Modo, the functions are deeply integrated and take advantage of the core functionality to provide an intuitive and powerful set of options.

Combining a natural blend of 2D- and 3D-painting techniques, you can select a brush and begin painting in any viewport — including UV views! Modo tries to determine the best options automatically when painting or sculpting, but you need to know about a couple requirements necessary to get the best results. Modo's default interface provides a **Paint** layout tab, which is dedicated to painting, sculpting, and Vertex Map-editing tools, and a toolbox with the related tools.

You may benefit greatly from using a pressure-sensitive tablet when painting or sculpting in Modo, but this is not a strict requirement. (Most users find tablets to be a more natural way to work with the pen-based input instead of using a mouse.) The additional control from the pressure-sensitive stylus can provide a greater degree of precision and subtlety when modeling and texturing creations.

Tip: When a brush based tool is activated, a circle is drawn under the mouse when you hover over a surface. This makes it easier to see where the sculpt or paint is applied when the mouse is right-clicked and dragging across the surface of your mesh. To change the size of the brush tip, left-click and drag your mouse.



Painting

To paint in Modo, you need a 3D model and a bitmap texture. You apply the texture map to some (or all) of the model's polygons with the Shader Tree. Modo has several tools for you to quickly do these tasks — including extensive UV mapping tools and functions to add textures to your model.

Modo has two different painting technologies: a standard, 3D-painting method (which paints directly onto the 3D model) and a projection-painting method. Standard painting keeps the brush stroke size consistent with the canvas, but projection painting keeps it consistent with the viewport. Standard 3D painting often results in oddly-shaped brush strokes when painting on complex objects (such as a character). With projection painting you paint in the viewport (similar to painting onto a plane of glass) and Modo projects each stroke onto the 3D model (similar to a movie theater screen). This eliminates the distortions inherent in standard 3D painting.

Warning: You can paint complex objects, but painting on complex objects with any of the various non-UV texture projections (such as **Planar**, **Cubic**, or **Spherical**) usually yields unsatisfactory results.

Particle Paint Tools

Particles have many uses in Modo, but most uses are as point-sources for Replicators and Texture Replicators as well as for blobs, sprites, and volumes. You could create these particles procedurally with Surface Generator or indirectly by generating geometry and then removing the polygon edges between the vertices. The Particle Paint tools provide a way to create and edit particle-point sources directly using the familiar brush-based workflow.

Creating particles is a simple procedure. You only need to prepare by having a geometry surface to spray the particles onto. You use the Particle Paint Tool as the main tool to create the initial particles. Then you use the editing tools to mold and shape the particles for the intended use. When using any constrain actions, the particles always conform to the target surface and Modo knows what is a particle and what is a regular polygon. Therefore, you can't accidentally edit the base geometry to which the particles are being applied.

Customizing Brushes

Like the rest of the Modo toolset, the paint and sculpting tools are modular and composed through the Tool Pipe. If you are not familiar with the Tool Pipe, you don't have to use it. You can do 3D painting and sculpting with the provided tools. Using the standard tools fulfills most (if not all) of your modeling and texturing requirements.

For those who understand the power of the Tool Pipe, you can understand the flexibility and power this brings to the painting system. The Tool Pipe has components that you can add together to create custom tools. These components include the paint tool, a brush tip, custom inks, nozzles, and even falloffs. For basic painting and sculpting, you only need a tool and a brush tip. You can add (or remove) the inks, nozzles, and falloffs depending on the intended effect. With the Tool Pipe you can combine a variety of components to create customized tools specific to your creative needs.

Using Painting Tools

Painting in Modo is an intuitive blend of 2D and 3D painting techniques, where you can paint onto any bitmap image map within any of the GL viewports.
The brushes for painting can be found in the **Paint** tab of the interface, where a focused collection of tools and useful commands are located dedicated to the painting and sculpting functionality. The tools may also be called from the menu bar, under the **Texture** > **Paint Tools** menu. Invoking any of the tools displays the associated properties in the appropriate viewport panel.

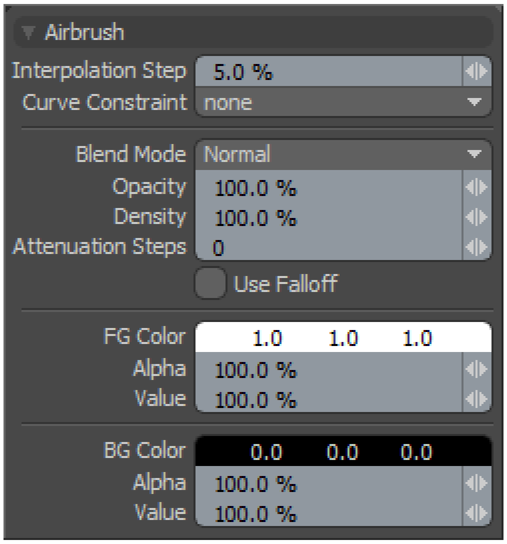
To paint, you need to select both a Tool (brush) and a Tip, which designates how the brush itself applies the color. When working with a mouse, you can click and drag over the surface to apply the stroke. The click is applied automatically when using a pressure sensitive stylus and tablet. You may also right-click and drag in the viewport to interactively scale the size of the brush tip.

Painting		
	Airbrush	The **Airbrush** tool, with the proper **Tip** can be used to mimic a real airbrush which excels at creating soft, even strokes.
	Clone	The **Clone** tool provides a function to duplicate an area of a texture to another area.
	Radial Gradient	The **Radial Gradient** tool applies a flood fill of color that ramps from the FG color to the BG color in a circular fashion.
	Blur	The **Blur** tool averages pixels from under the brush to create the effect of softening any apparent detail.
	Paintbrush	The **Paintbrush** tool, combined with a **Tip** applies solid even strokes to the canvas.
	Line	The **Line** paint tool allows you to quickly create straight lines by clicking a start point, dragging to the end point position, and releasing the mouse button to set.
	Linear Gradient	The **Linear Gradient** paint tool applies a flood fill of ink that ramps from the FG color to the BG color.
	Sharpen	The **Sharpen** tool increases the edge contrast between colors producing an effect that looks like focusing, making images appear sharper.
	Eraser	The **Eraser** tool works by modifying the alpha channel of the currently selected image texture to create the effect of erasing the paint.
	Fill	The **Fill** tool colors all the pixels in the image that fall within the **Fill Tolerance** according to the color of the pixel that was clicked on when initiating the fill.
	Smudge	The **Smudge** tool shift pixels from under the brush in the direction of the mouse's or stylus' movement to create the visual effect of smearing the paint together.

Painting		
	Lasso	The **Lasso** falloff option provides you a means to draw selection areas directly on a viewport that then limit the application of paint strokes to the surface.

Airbrush

The Airbrush tool, with the proper **Tip** can be used to mimic a real airbrush which excels at creating soft, even strokes. You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Airbrush** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin painting directly into the active image map, defined by the selected image in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. You may also select different Customizing Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the airbrush to further customize the way the brush applies to the surface. When selected, the associated attributes display in the associated Tool Properties panel.



Airbrush

Interpolation Step: 5.0 %

Curve Constraint: none

Blend Mode: Normal

Opacity: 100.0 %

Density: 100.0 %

Attenuation Steps: 0

☐ Use Falloff

FG Color: 1.0 1.0 1.0

Alpha: 100.0 %

Value: 100.0 %

BG Color: 0.0 0.0 0.0

Alpha: 100.0 %

Value: 100.0 %

Airbrush	
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the image canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to **5%**. So for an 80 pixel

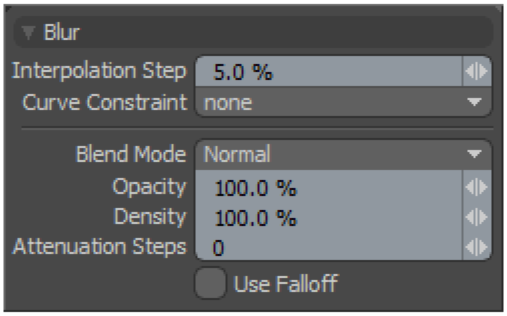
Airbrush	
	sized brush, dabs would be placed every 4 pixels (5% of 80 = 4). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.
Curve Constraint	The **Curve Constraint** option allows you to use pre-drawn curves, either Spline (see Adding Curves) or Bezier, and then constrain the brush strokes to the curve when painting, producing a very smooth, controlled, and repeatable result. It works by having the curve element in the background layer (visible, but not selected). Then the **Curve Constraint** option is set to either **Background 2D** or **Background 3D** mode. Once set, any paint strokes into the canvas near the curve constrain to the nearest curve position. The **None** option disables the constraining, the **Background 2D** projects the curve from any orientation onto the surface through the viewport and the **Background 3D** option constrains the stroke to the curves position in 3D space, only affecting surfaces that are within close proximity (defined by the brushes size).
Blend Mode	The **Blend Mode** defines how the strokes blend into the existing image canvas. The default **Normal** setting applies opaque strokes (as defined by the **Opacity** option) over the top of the existing image canvas. The alternate options works in the same respect as the layer blending options documented with examples on the Layer Blend Modes page of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted is completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value causes the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above 0 causes the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the

Airbrush	
	tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.
FG Color	Sets the foreground color for various paint tools. This can be set through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha transparency value to be associated with the FG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).
BG Color	Sets the background color for paint tools that use it. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha transparency value to be associated with the BG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Blur

The **Blur** tool averages pixels from under the brush to create the effect of softening any apparent detail. You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox.
The **Blur** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin painting directly into the active image map, defined by the selected image in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. You may also select different Customizing

Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.


Blur
Interpolation Step 5.0 %
Curve Constraint none
Blend Mode Normal
Opacity 100.0 %
Density 100.0 %
Attenuation Steps 0
☐ Use Falloff

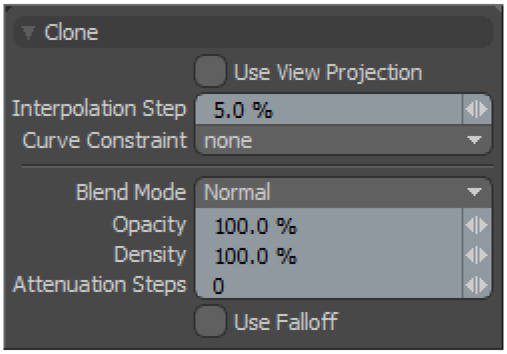
Option	Description
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the image canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to 5%. So for an 80 pixel sized brush, dabs would be placed every 4 pixels (5% of 80 = 4). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.
Curve Constraint	The **Curve Constraint** option allows you to use pre-drawn curves, either Spline (see Adding Curves) orBezier, and then constrain the brush strokes to the curve when painting, producing a very smooth, controlled and repeatable result. It works by having the curve element in the background layer (visible, but not selected). Then the **Curve Constraint** option is set to either **Background 2D** or **Background 3D** mode. Once set, any paint strokes into the canvas near the curve constrain to the nearest curve position. The **None** option disables the constraining, the **Background 2D** projects the curve from any orientation onto the surface through the viewport and the **Background 3D** option constrains the stroke to the curves position in 3D space, only affecting surfaces that are within close proximity (defined by the brushes size).
Blend Mode	The **Blend Mode** defines how the strokes blend into the existing canvas. The default **Normal** setting applies opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options work in the same respect as the layer blending options documented with examples on the Layer Blend Modes page of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque

Option	Description
	the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted are completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value causes the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above **0** cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.

Clone

The Clone tool provides a function to duplicate an area of a texture to another area. Useful for quickly covering imperfection is a photographic image by sampling from another like area of the image or for simply enlarging areas of useful texture. Before applying a stroke, you must define a clone source area. This can be done by holding the **Ctrl** key and clicking the area of the image where the center of the cloning originates from. Each stroke now clones from that center source until a new source is defined.

You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Clone** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin painting directly into the active image map, defined by the selected image in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. You may also select different Customizing Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.



☐ Use View Projection
Interpolation Step: 5.0 %
Curve Constraint: none
Blend Mode: Normal
Opacity: 100.0 %
Density: 100.0 %
Attenuation Steps: 0
☐ Use Falloff

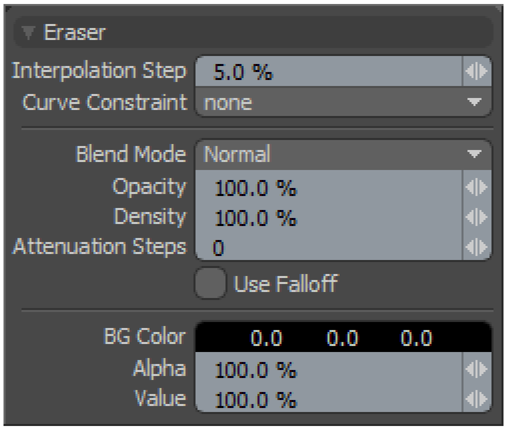
Clone	
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to **10%**. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.
Curve Constraint	The **Curve Constraint** option allows you to use pre-drawn curves, either Spline (see Adding Curves) or Bezier, and then constrain the brush strokes to the curve when painting, producing a very smooth, controlled and repeatable result. It works by having the curve element in the background layer (visible, but not selected). Then the **Curve Constraint** option is set to either **Background 2D** or **Background 3D** mode. Once set any paint strokes into the canvas near the curve constrain to the nearest curve position. The **None** option disables the constraining, the **Background 2D** projects the curve from any orientation onto the surface through the viewport and the **Background 3D** option constrains the stroke to the curves position in 3D space, only affecting surfaces that are within close proximity (defined by the brushes size).
Blend Mode	The **Blend Mode** defines how the strokes blend into the

Clone	
	existing canvas. The default **Normal** setting applies opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options work in the same respect as the layer blending options documented with examples on the Layer Blend Modespage of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted would be completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value causes the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above **0** causes the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.

Eraser

The **Eraser** tool works by modifying the alpha channel of the currently selected image texture to create the effect of erasing the paint. When utilizing an image with no **Alpha** transparency channel present, the Eraser tool then simply acts like a paintbrush tool, painting to the defined background color, instead of the active foreground color.

You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Eraser** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin painting directly into the active image map, defined by the selected image in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to edit a different layer. You may also select different Customizing Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.



Eraser

Interpolation Step 5.0 %

Curve Constraint none

Blend Mode Normal

Opacity 100.0 %

Density 100.0 %

Attenuation Steps 0

☐ Use Falloff

BG Color 0.0 0.0 0.0

Alpha 100.0 %

Value 100.0 %

Eraser	
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the image canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to **5%**. So for an 80 pixel sized brush, dabs would be placed every 4 pixels (5% of 80 = 4). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.
Curve Constraint	The **Curve Constraint** option allows you to use pre-drawn curves, either Spline (see Adding Curves) orBezier, and then constrain the brush strokes to the curve when

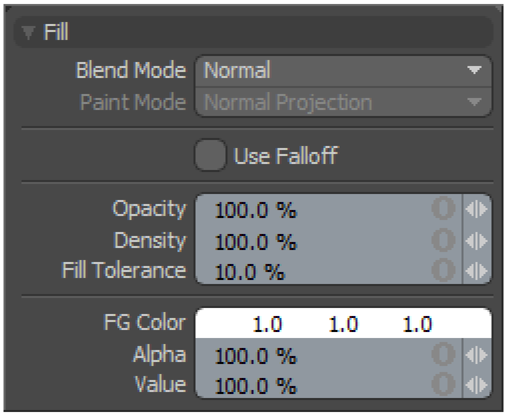
Eraser	
	painting, producing a very smooth, controlled and repeatable result. It works by having the curve element in the background layer (visible, but not selected). Then the **Curve Constraint** option is set to either **Background 2D** or **Background 3D** mode. Once set, any paint strokes into the canvas near the curve constrain to the nearest curve position. The **None** option disables the constraining, the **Background 2D** projects the curve from any orientation onto the surface through the viewport and the **Background 3D** option constrains the stroke to the curves position in 3D space, only affecting surfaces that are within close proximity (defined by the brushes size).
Blend Mode	The **Blend Mode** defines how the strokes blend into the existing canvas. The default **Normal** setting applies opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options work in the same respect as the layer blending options documented with examples on the Layer Blend Modespage of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted are completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value causes the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above **0** cause the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allow any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.

Eraser	
BG Color	Sets the background color for paint tools that use it. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the BG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Fill

The **Fill** tool colors all the pixels in the image that fall within the **Fill Tolerance** according to the color of the pixel that was clicked on when initiating the fill.

You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Fill** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. Once selected, you can begin painting directly into the active image map, defined by the image in the **Shader Tree** with the small paintbrush icon next to it (). Simply select a different target layer if you wish to paint onto a different layer. When selected, the associated attributes display in the Properties panel.



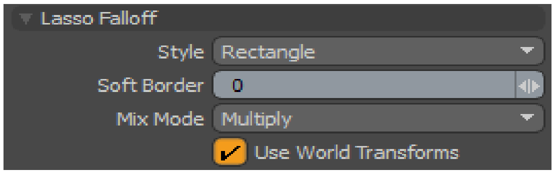
Fill
Blend Mode: Normal
Paint Mode: Normal Projection
☐ Use Falloff
Opacity: 100.0 %
Density: 100.0 %
Fill Tolerance: 10.0 %
FG Color: 1.0 1.0 1.0
Alpha: 100.0 %
Value: 100.0 %

Fill	
Blend Mode	The **Blend Mode** defines how the filled area blends into the existing canvas. The default **Normal** setting applies opaquely (as defined by the Opacity option) over the

Fill	
	top of the existing canvas. The alternate options work in the same respect as the layer blending options documented with examples on the Layer Blend Modespage of the documentation.
Paint Mode	Has no effect over the **Fill** tool (as it always works in image space) and is therefore disabled.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of fill effect.
Opacity	This value determines how the total opacity of the ink distributed by the brush. If the value is less than 100% the ink maxes out at that percentage and never becomes fully opaque no matter how many times you paint over the stroke.
Density	This value determines the amount of ink that comes out during a paint stroke. Unlike **Opacity**, when Density is below 100%, painting over the existing stroke repeatedly continues to increase the amount of paint on the map.
Fill Tolerance	This value defines the threshold area that is flooded with color by the fill tool.
FG Color	Sets the foreground color for various paint tools. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the FG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Lasso

The **Lasso** falloff option is used for drawing selection areas directly on a viewport that then limit the application of paint strokes to the surface. The **Lasso** option is simply a standard falloff type. For more information, see Using Falloffs.



Style Rectangle
Soft Border 0
Mix Mode Multiply
☑ Use World Transforms

Several styles of Lasso selections, drawn in screen space projecting outward toward infinity, producing falloffs that are very specific. Settings allow you to define areas of influence around the lasso shape with the **Soft Border** option. All areas within the Lasso receive 100% of the tools effect and the attenuation of the falloff occurs across the **Soft Border** distance only outside the Lasso area.

Option	Description
Style	Selects from the various **Lasso** styles: • **Lasso** - Draws freeform shapes onto the screen defining the falloff. • **Rectangle** - Drags out rectangular shapes to define the falloff area. • **Circle** - Drags out perfectly circular shapes to define the falloff area. Shape originates from center of the circle. • **Ellipse** - Drags out elliptical shapes to define the falloff area. Shape originates from the upper or lower corner of the ellipse.
Soft Border	The **Soft Border** option defines the actual falloff radius around the Lasso area that attenuates across the distance defined. Calculated in screen-space pixels from the lasso border.
Mix Mode	In instances where there are multiple falloffs applied to a transform, using the **Add** option of the falloff menu, the mix mode defines how each falloff interacts with the other.

Tool Falloffs

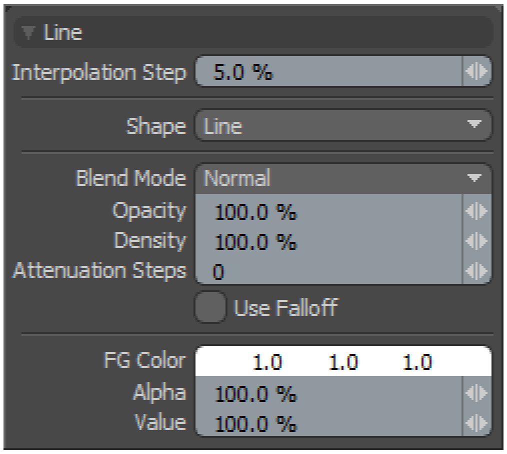
Because of Modo's generalized nature, the same falloff options that are used for geometric edits can be applied to most of the paint tools. In order to utilize falloffs, each paint tool has a **Use Falloffs** option that must be enabled. When using this option, it is easy to confine paint strokes to specific areas of the model (or viewport). Of course, there are some falloffs that are more useful than others in this regard. For instance, the **Screen** and **Airbrush** falloff operators would not be very useful during painting since they have similar attributes to the paintbrushes themselves. Falloffs, such as **Linear**, **Cylinder**, **Radial**, and of course **Lasso** are quite useful when working with the Paint tools.

Typically falloff operators use the right-click for setup. However, since the paint tools use right-click to set the brush size you must select the falloff in the Tool Pipe and then click to define the falloff area before painting. Make sure to select the paint tool prior to painting. Once the falloff has an initial location you can also easily use the falloff handles to further edit their positions without changing tool selections through the tool pipe.

Line

The **Line** paint tool allows you to quickly create straight lines by clicking a start point, dragging to the end point position, and releasing the mouse button to set. The look of the line is defined by the **Tip** shape and size.

You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Line** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin creating lines directly into the active image map, defined by the image in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. You may also select different Customizing Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the paintbrush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.



Line

Interpolation Step 5.0 %

Shape Line

Blend Mode Normal

Opacity 100.0 %

Density 100.0 %

Attenuation Steps 0

☐ Use Falloff

FG Color 1.0 1.0 1.0

Alpha 100.0 %

Value 100.0 %

Line	
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to **10%**. So for an 80 pixel sized brush, dabs would be placed every 8 pixels (10% of 80 = 8). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.

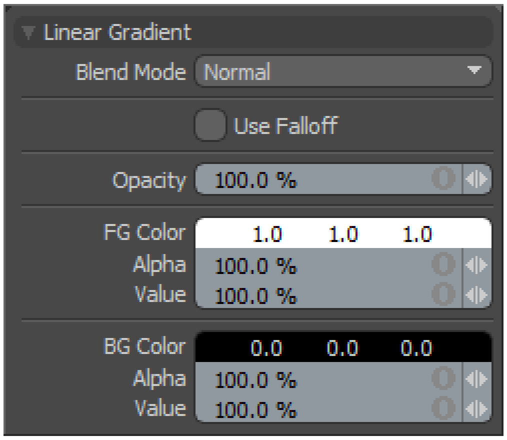
Line	
Shape	The **Shape** option allows you to define some extra geometric shapes for drawing, there is the default **Line** option that generates a line from a defined start and end position (click and drag to define, release to drop the line as pixels). There is also a **Circle** and a **Rectangle** shapes that can be defined in a similar way.
Blend Mode	The **Blend Mode** defines how the strokes blend into the existing canvas. The default **Normal** setting applies opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options work in the same respect as the layer blending options documented with examples on the Layer Blend Modes page of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted are completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value cause the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above **0** causes the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allow any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.
FG Color	Sets the foreground color for various paint tools. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the FG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being

Line	
	applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Linear Gradient

The **Linear Gradient** paint tool applies a flood fill of ink that ramps from the FG color to the BG color. Clicking and dragging in the 3D view creates a line from the cursor. This line is used to define the ramp distance from the FG to BG color. The color ramp is calculated in a linearly according to the line you defined. The ramp can be created in 3D views or directly in the UV view. You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox.

The **Linear Gradient** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. Once selected, new gradients are created in the active image map, defined by the layer in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. When selected, the associated attributes display in the Properties panel.



Linear Gradient
Blend Mode: Normal
☐ Use Falloff
Opacity: 100.0 %
FG Color: 1.0 1.0 1.0
Alpha: 100.0 %
Value: 100.0 %
BG Color: 0.0 0.0 0.0
Alpha: 100.0 %
Value: 100.0 %

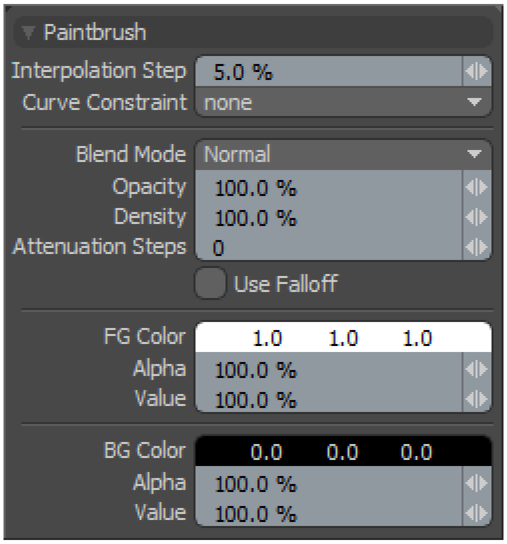
Linear Gradient	
Blend Mode	The **Blend Mode** defines how the filled area blends into the existing canvas. The default **Normal** setting applies opaquely (as defined by the Opacity option) over the top of the existing canvas. The alternate options works in the same respect as the layer blending options documented with examples on the Layer Blend Modespage of the documentation.

Linear Gradient	
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of fill effect.
Opacity	This value determines how the total opacity of the ink distributed by the gradient. If the value is less than 100% the ink maxes out at that percentage and never becomes fully opaque no matter how many times you paint over the stroke.
FG Color	Sets the foreground color for the gradient. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the FG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).
BG Color	Sets the background color for the gradient. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the BG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point (FP) mode active or loaded as a file that supports HDRI).

Paintbrush

The **Paintbrush** tool, combined with a **Tip** applies solid even strokes to the canvas. The main difference between the **Paintbrush** and the **Airbrush** tools is that the **Paintbrush** applies the paint with no internal blending, which is to say the paint comes out at full strength whereas the **Airbrush** layers on the paint daubs as you repeat strokes.

You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Paintbrush** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin painting directly into the active image map, defined by the selected image in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. You may also select different Customizing Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the paintbrush to further customize the way the brush applies to the surface. When selected, the associated attributes display in the Properties panel.



Interpolation Step: 5.0 %
Curve Constraint: none
Blend Mode: Normal
Opacity: 100.0 %
Density: 100.0 %
Attenuation Steps: 0
Use Falloff
FG Color: 1.0 1.0 1.0
Alpha: 100.0 %
Value: 100.0 %
BG Color: 0.0 0.0 0.0
Alpha: 100.0 %
Value: 100.0 %

Paintbrush	
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the image canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to **5%**. So for an 80 pixel sized brush, dabs are placed every 4 pixels (5% of 80 = 4). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.
Curve Constraint	The **Curve Constraint** option allows you to use pre-drawn curves, either Spline (see Adding Curves) or Bezier, and then constrain the brush strokes to the curve when painting, producing a very smooth, controlled and repeatable result. It works by having the curve element in the background layer (visible, but not selected). Then the **Curve Constraint** option is set to either **Background 2D** or **Background 3D** mode. Once set, any paint strokes into the canvas near the curve constrain to the nearest curve position. The **None** option disables the constraining, the **Background 2D** projects the curve from any orientation onto the surface through the viewport and the **Background 3D** option constrains the stroke to the curves position in 3D space, only affecting surfaces that are within close proximity (defined by the brushes size).

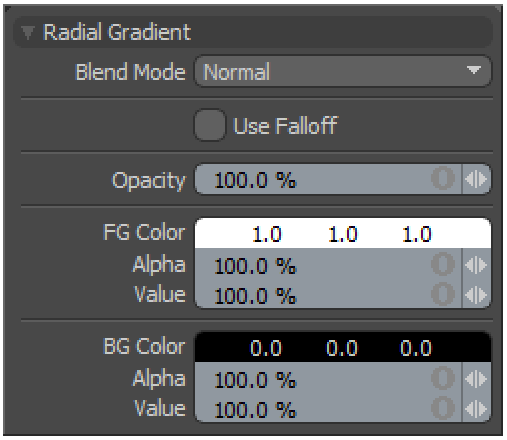
Paintbrush	
Blend Mode	The Blend Mode defines how the strokes blend into the existing canvas. The default **Normal** setting applies opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options works in the same respect as the layer blending options documented with examples on the Layer Blend Modespage of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted are completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value causes the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above **0** causes the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.
FG Color	Sets the foreground color for various paint tools. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the FG paint color.
Value	You ca use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point, FP, mode active, or loaded as a file that supports HDRI).

Paintbrush	
BG Color	Sets the background color for paint tools that use it. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the BG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point, FP, mode active or loaded as a file that supports HDRI).

Radial Gradient

The **Radial Gradient** tool applies a flood fill of color that ramps from the FG color to the BG color in a circular fashion. Clicking and dragging in the 3D view creates a line from the cursor. This line is used to define the ramp distance from the FG to BG color. The color ramp is calculated in a radial direction with the FG color at the center according to the user defined line.

The ramp can be created in 3D views or directly in the UV view. You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Radial Gradient** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. Once selected, new gradients are created in the active image map, defined by the layer in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. When selected, the associated attributes display in the Properties panel.



Radial Gradient

Blend Mode: Normal

☐ Use Falloff

Opacity: 100.0 %

FG Color: 1.0 1.0 1.0

Alpha: 100.0 %

Value: 100.0 %

BG Color: 0.0 0.0 0.0

Alpha: 100.0 %

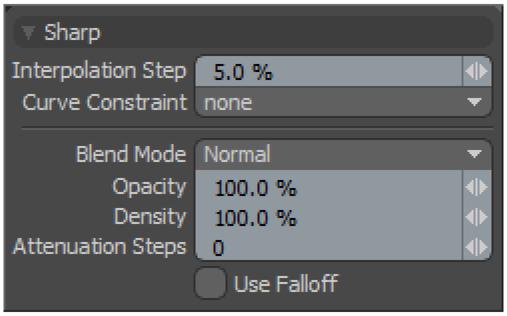
Value: 100.0 %

Radial Gradient	
Blend Mode	The **Blend Mode** defines how the filled area blends into the existing canvas. The default **Normal** setting applies opaquely (as defined by the Opacity option) over the top of the existing canvas. The alternate options works in the same respect as the layer blending options documented with examples on the Layer Blend Modespage of the documentation.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of fill effect.
Opacity	This value determines how the total opacity of the ink distributed by the gradient. If the value is less than 100% the ink maxes out at that percentage and never becomes fully opaque no matter how many times you paint over the stroke.
FG Color	Sets the foreground color for the gradient. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the FG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point, FP, mode active, or loaded as a file that supports HDRI).
BG Color	Sets the background color for the gradient. You can set it through the color component on the tool properties or by clicking on the Color Picker while the tool is active.
Alpha	This option allows an alpha value to be associated with the BG paint color.
Value	You can use this percentage to increase or decrease the overall value of the ink being applied. This value can be set over 100% enabling high dynamic range paint strokes (if your image was created with the floating point, FP, mode active or loaded as a file that supports HDRI).

Sharpen

The **Sharpen** tool increases the edge contrast between colors producing an effect that looks like focusing, making images appear sharper.

You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Sharpen** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin painting directly into the active image map, defined by the selected image in the **Shader Tree** with the small paintbrush icon next to it (🖌). Simply select a different target layer if you wish to paint onto a different layer. You may also select different Customizing Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the brush to further customize the way the tool applies to the surface. When selected, the associated attributes display in the Properties panel.



Interpolation Step: 5.0 %
Curve Constraint: none
Blend Mode: Normal
Opacity: 100.0 %
Density: 100.0 %
Attenuation Steps: 0
☐ Use Falloff

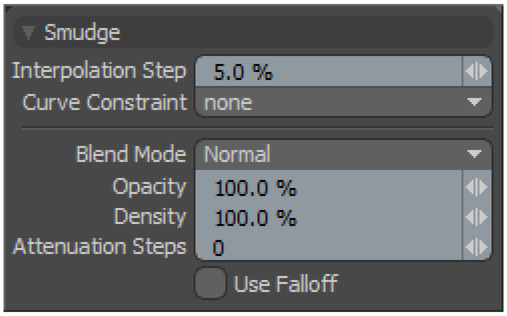
Option	Description
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the image canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to **5%**. So for an 80 pixel sized brush, dabs are placed every 4 pixels (5% of 80 = 4). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.
Curve Constraint	The **Curve Constraint** option allows you to use pre-drawn curves, either Spline (see Adding Curves) or Bezier, and then constrain the brush strokes to the curve when painting, producing a very smooth, controlled, and repeatable result. It works by having the curve element in the background layer (visible, but not selected). Then the **Curve Constraint** option is set to either **Background 2D** or **Background 3D** mode. Once set, any paint strokes into the canvas near the curve constrain to the nearest curve position.

Option	Description
	The **None** option disables the constraining, the **Background 2D** projects the curve from any orientation onto the surface through the viewport and the **Background 3D** option constrains the stroke to the curves position in 3D space, only affecting surfaces that are within close proximity (defined by the brushes size).
Blend Mode	The **Blend Mode** defines how the strokes blends into the existing canvas. The default **Normal** setting applies opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options work in the same respect as the layer blending options documented with examples on the Layer Blend Modespage of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted are completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value causes the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above **0** causes the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.

Smudge

The **Smudge** tool shift pixels from under the brush in the direction of the mouse's or stylus' movement to create the visual effect of smearing the paint together.

You can select the tool from the **Paint** interface tab under the **Paint Tools** sub-tab of the toolbox. The **Smudge** tool can also be activated from the menu bar option under the **Texture** > **Paint Tools** menu. When selected, the tool also auto-selects a **Tip** allowing you to begin painting directly into the active image map, defined by the image in the **Shader Tree** with the small paintbrush icon next to it (). Simply select a different target layer if you wish to paint onto a different layer. You can also select different Customizing Paint Tools and additionally add Ink and Nozzle effects (see Customizing Inks and Tablet Nozzles) to the paintbrush to further customize the way the brush applies to the surface. When selected, the associated attributes display in the Properties panel.



Smudge

Interpolation Step: 5.0 %

Curve Constraint: none

Blend Mode: Normal

Opacity: 100.0 %

Density: 100.0 %

Attenuation Steps: 0

☐ Use Falloff

Option	Description
Interpolation Step	Generally, paint strokes are not a continuous lines, but a series of dots or dabs placed on the image canvas. This option defines the distance between each dab as a percentage of the brush size. By default, this value is set to **5%**. So for an 80 pixel sized brush, dabs are placed every 4 pixels (5% of 80 = 4). You can increase or reduce this value for a variety of effects. Increasing it toward 100% produces individual dots, rather than what looks like a stroked line.
Curve Constraint	The **Curve Constraint** option allows you to use pre-drawn curves, either Spline (see Adding Curves) or Bezier, and then constrain the brush strokes to the curve when painting, producing a very smooth, controlled, and repeatable result. It works by having the curve element in the background layer (visible, but not selected). Then the **Curve Constraint** option is set to either **Background 2D** or **Background 3D** mode. Once set, any paint strokes into the canvas near the curve constrain to the nearest curve position.

Option	Description
	The **None** option disables the constraining, the **Background 2D** projects the curve from any orientation onto the surface through the viewport and the **Background 3D** option constrains the stroke to the curves position in 3D space, only affecting surfaces that are within close proximity (defined by the brushes size).
Blend Mode	The **Blend Mode** defines how the strokes blend into the existing canvas. The default **Normal** setting applies opaque strokes (as defined by the Opacity option) over the top of the existing canvas. The alternate options work in the same respect as the layer blending options documented with examples on the Layer Blend Modes page of the documentation.
Opacity	This value determines the transparency of the brush tip, in turn affecting how opaque the paint generated by the brush is. Values under 100% reduce the transparency, attenuating down toward 0% where the pixels painted are completely invisible.
Density	This value determines the amount of paint that comes out during a paint stroke. A value of 100% applies even pressure across the brush's tip, values under 100% attenuating down toward 0% gradually concentrate the paint toward the center of the brushes tip, producing a results that is similar, but different, than changing the opacity.
Attenuation Steps	This value causes the paint to fade out after the set number of steps is reached in the stroke. This setting is somewhat related to the speed with which you create the stroke as painting quickly spreads out the steps used to paint the stroke. Moving the brush slowly with this value set above **0** causes the paint to run out in a shorter stroke. This is a very useful setting to use if you do not have a tablet to paint with. Tablet users would be better suited to modulate the paint opacity based on the tablet pressure so as to have greater control.
Use Falloff	The **Use Falloff** option allows any of the variousUsing Falloffs to mask the application of brush strokes. Since the right-click brush sizing function supersedes the right-click function to define the falloff, ensure the falloff is defined prior to activating the tool.

Add Complex Details to Your Model with Decal Workflows

Decals are a quick and easy way to apply textures such as graffiti, logos and displacement onto your mesh without impacting your texture maps, or needing to adjust the geometry the decal is being placed on. This allows you to create highly detailed models without needing to increase your poly count, as well as create fully procedural images that can be applied, swapped out and removed from your asset whenever.

RADIOACTIVE III
Contents
Activity
3.2
TRANSPORT INDEX
7

Planar decals allow you to project a flat, square mesh plane containing your decal textures onto an asset so that the images are layered on top of your object instead of stored in your shader. This allows you to move, scale, and rotate your decals as if they were a 3D object whenever you need to, without having to worry about how the decal is impacting your model or any additional materials.

How to Create Decals

Decals are created using flat, square planes and then applying a material containing your decal's textures to that plane.

Projecting Planar Decals on to a Mesh

Project a flat, square mesh plane containing your decal textures onto an asset.

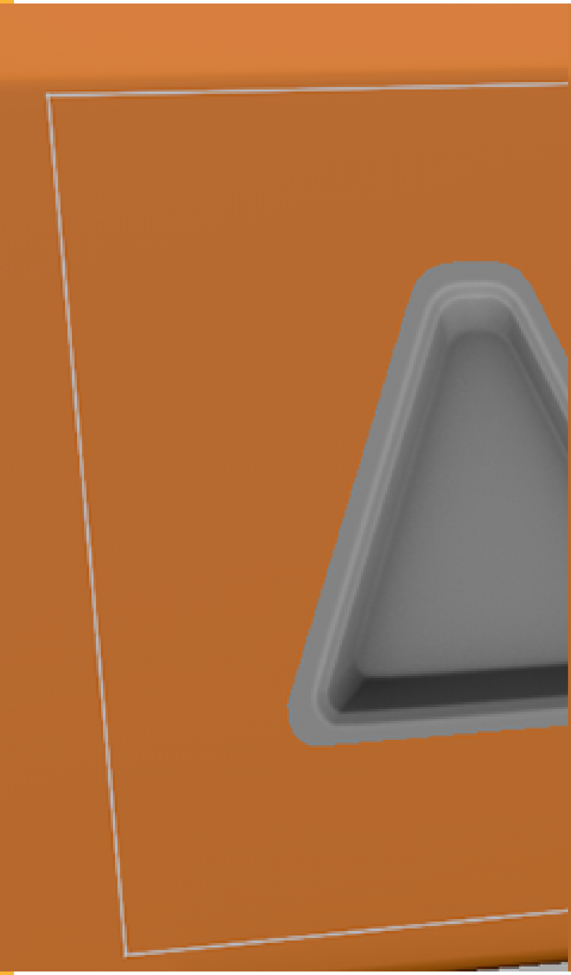
Add Seams to Models With Seam Decals

Seam Decals allow you to use two pieces of intersecting geometry or a curve to create a seam along a mesh.

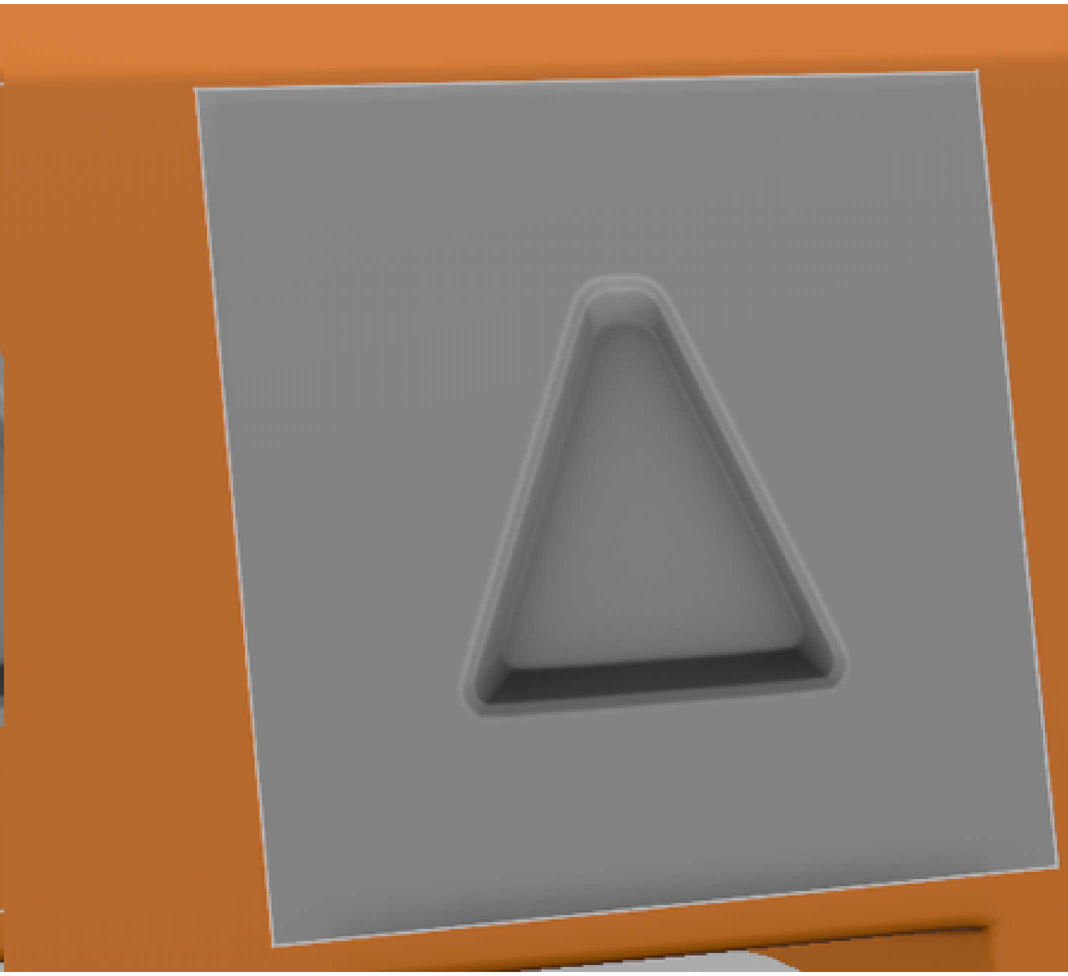
How to Create a Decal That Can be Projected

Decals are easily created by creating a flat, square plane and then applying a material containing your decal's textures to that plane. The plane carrying the decal can then be projected manually via the **Decal Planar** MeshOp, or created automatically by pressing the **Decals...** button found in the create tab.

When creating a decal, there's no set rules on what texture maps should or shouldn't be used, but it is recommended to use a mask as a stencil layer to ensure that the decal appears on the asset correctly.



Decal using a mask. Materials applied to the asset the decal is being projected on are visible in areas that have been masked out.



Decal not using a mask. The edges of the plane carrying the decal are clearly visible.

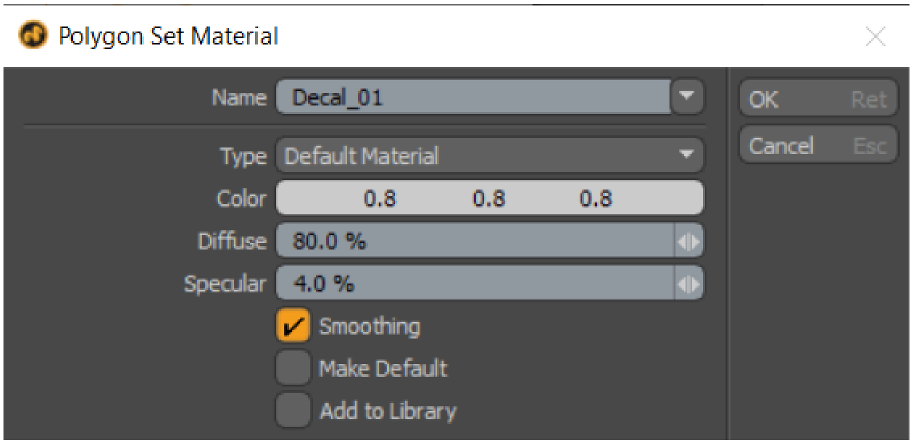
Tip: If using texture maps that follow PBR naming conventions, Modo's PBR Loader can be enabled to speed up the creation of materials. For more information on the PBR Loader, see PBR Loader

To create Decal that can be projected:

1. Hold Shift on your keyboard, and click the **Plane** tool in the **Create** tab.

   This creates a flat 1x1 meter plane at the work plane origin.
2. Rename the plane to Decal_01 or something similar so that it is easily findable in the Items List.
3. With the plane selected, press M on your keyboard to assign a material.

   The **Polygon Set Material** window opens.
4. In the **Polygon Set Material** window, name the material Decal_01 or similar.


Polygon Set Material

Name: Decal_01

Type: Default Material

Color: 0.8 0.8 0.8

Diffuse: 80.0 %

Specular: 4.0 %

☑ Smoothing

☐ Make Default

☐ Add to Library

   Parameters such as **Type**, **Color**, **Diffuse** and **Specular** can be left as default.
5. Once the material name has been set, click **OK**.

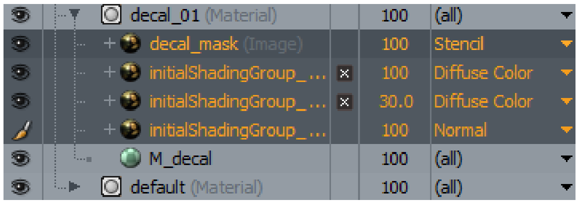
   The material is assigned to the plane and is now available in the Shader Tree.
6. Navigate to the Shader Tree and expand the decal material.
7. Once expanded, select your textures in your system's explorer window and then drag and drop them into the decal material.

   If you are using textures that follow PBR naming conventions, enable the PBR loader before adding textures to the decal material so that layer effects are assigned automatically.
8. Once the textures are present in the decal material, assign their layer effects as needed.

   For example, if using a Normal map, set the texture's layer effect to **Normal**.

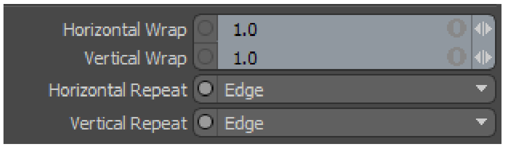
   Mask textures should have their layer effect set to **Stencil** and may need to be inverted.

   Tip: For more information on layer effects in the shader tree, see Layer Effects.
9. Once all textures have had their layer effects set, select all of them by holding shift and then clicking the first and last textures in the list.



decal_01 (Material)	100	(all)
decal_mask (Image)	100	Stencil
initialShadingGroup_…	100	Diffuse Color
initialShadingGroup_…	30.0	Diffuse Color
initialShadingGroup_…	100	Normal
M_decal	100	(all)
default (Material)	100	(all)

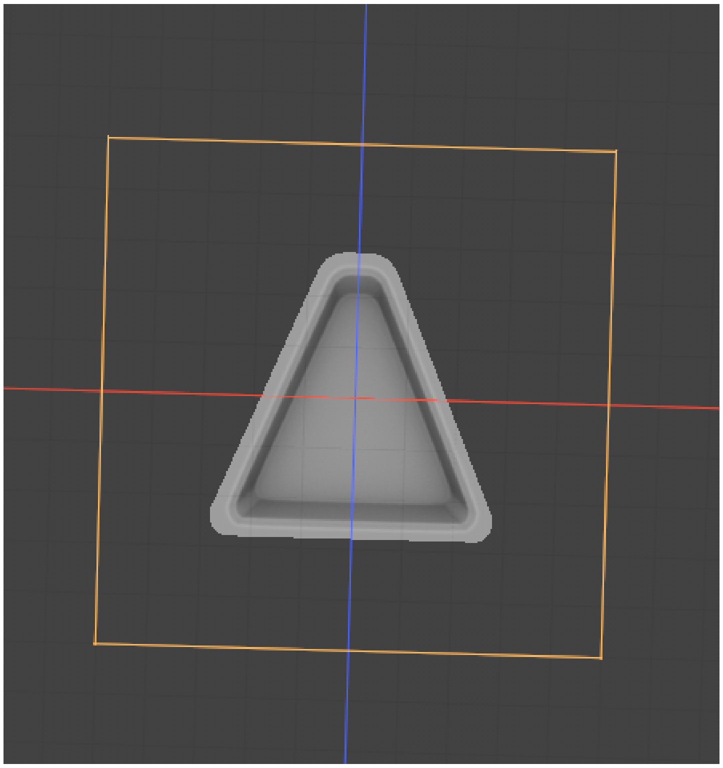
10. Once selected, navigate to **Texture Locator** tab in the **Properties** window. In the **Texture Locator** tab, change the **Horizontal** and **Vertical Repeat** parameters to **Edge**.



Horizontal Wrap 1.0
Vertical Wrap 1.0
Horizontal Repeat Edge
Vertical Repeat Edge

Changing the **Horizontal** and **Vertical Repeat** parameters to **Edge** ensures that your decal does not tile when projected.

Your decal is now created and ready to be projected onto a mesh.



Material containing a decal assigned to a flat plane.
Transparency is caused by the mask texture set to **Stencil**.

It's important to note that if you wish to move your decal from the origin of the work plane, it must be moved while in Item Mode and not while in a Component Mode. For decals to be projected as expected, transformations on both the decal and the asset it is being projected on must be frozen at zero. For more information on freezing transformations and different selection modes in Modo, see Freezing Geometry and Selection Modes.

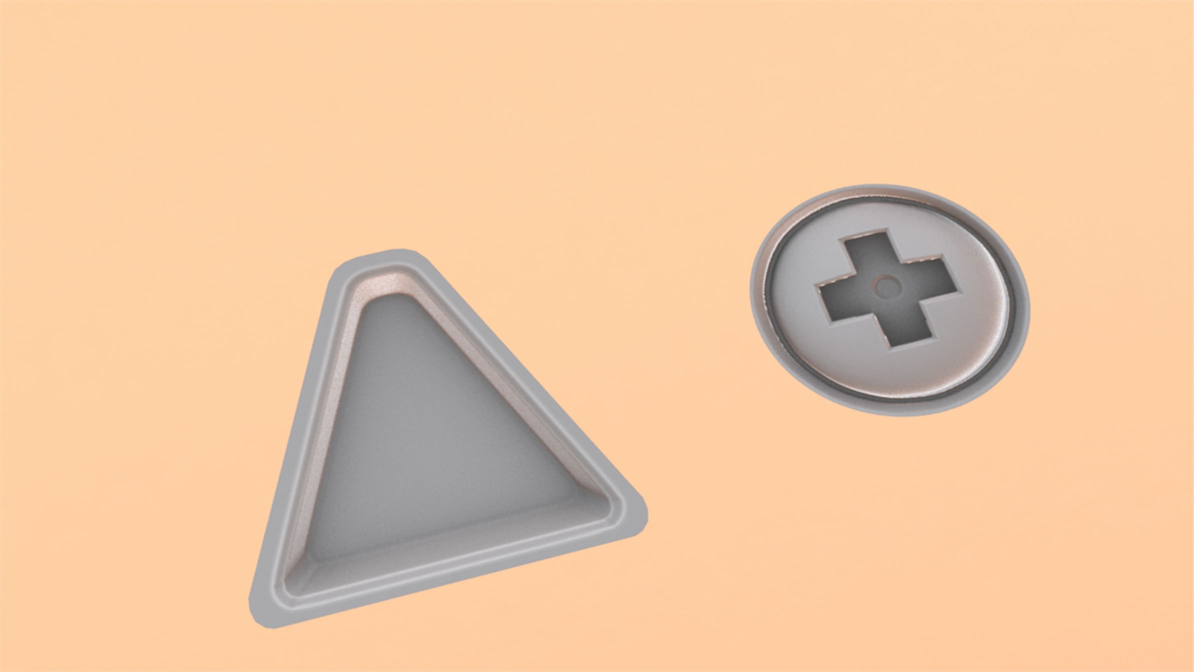
Projecting Decals with Planar Decals

Planar decals allow you to project a flat, square mesh plane containing your decal textures onto an asset so that the images are layered on top of your object instead of stored in your shader. This allows you to move, scale, and rotate your decals as if they were a 3D object whenever you need to, without having to worry about how the decal is impacting your model or any additional materials.

By using decals that are powered by normal and displacement maps, decals are also a brilliant way to add complexity to a model without increasing your polycount, saving vital processing power, something essential for game environments.

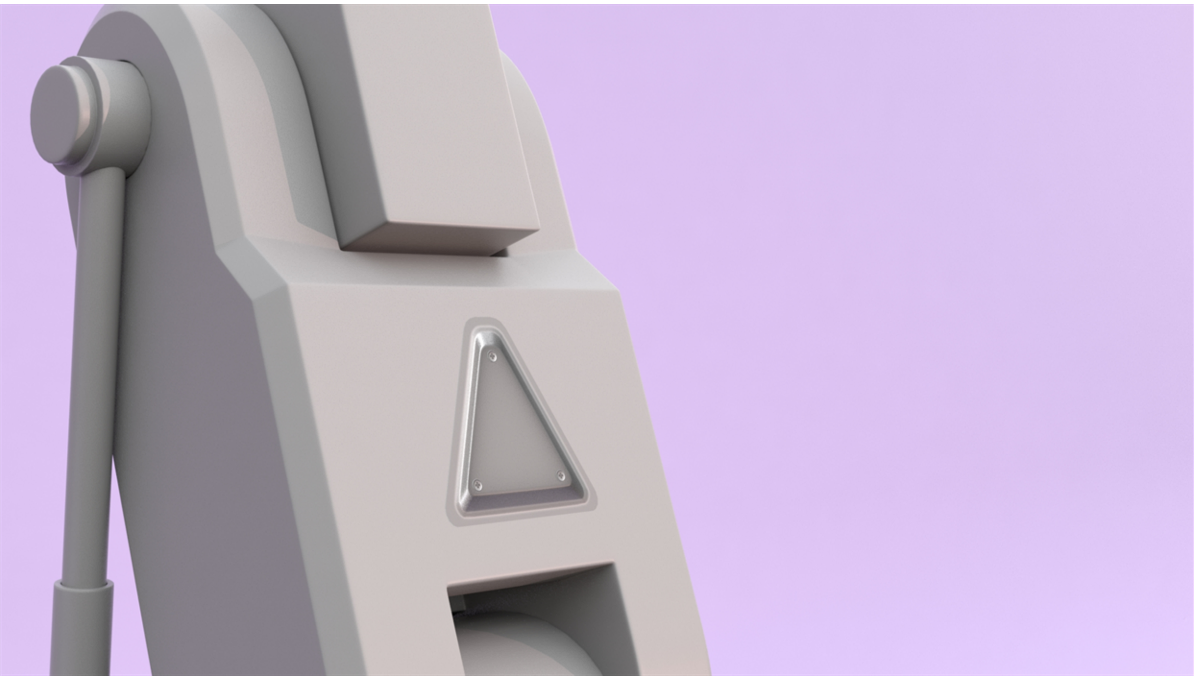
When a Decal is projected using the **Decal Planar** MeshOp, a boolean of the mesh carrying the decal is created. Once the boolean is created, the UVs of the plane carrying the Decal are smoothed over the UVs of the mesh receiving the decal.

Planar Decals can be manually set up with the **Planar Decal** MeshOperation, or can be assembled and applied to your mesh using the **Decals...** button found in the **Create** tab.



Setting Up a Decal Plane

Learn how to create a decal that can be projected onto your asset



Procedurally Project Planar Decals with The Decal Planar MeshOp

Learn how to set up and apply your decal to a mesh using the Decal Planar MeshOp

Hand Paint Planar Decals with Mesh Paint Replica

Learn how the Decals... button automatically builds a decal setup that can be manipulated with the Mesh Paint Replica tool.

Procedurally Project Planar Decals with The Decal Planar MeshOp

In addition to hand paint decals onto a mesh with **Mesh Paint Replica**, you are able to manually set up and project planar decals. While limiting you to creating a single decal at a time, this method gives granular control over the decal and how it's projected.

Note: For more information on painting multiple decals onto a mesh with **Mesh Paint Replica**, see Create and Hand Paint Decals With Mesh Paint Replica.

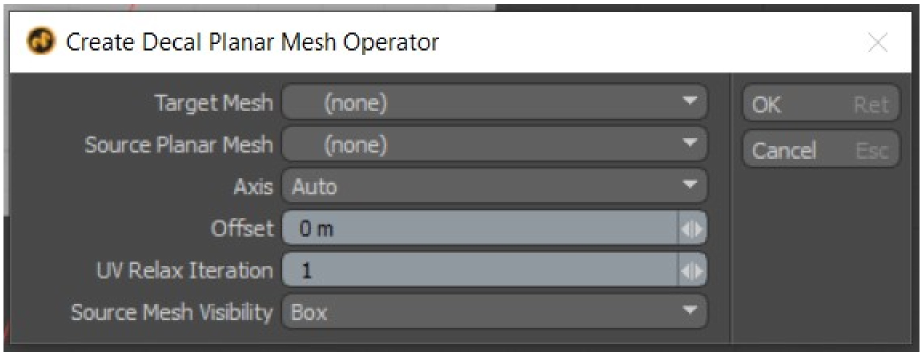
To procedurally create a planar decal:

1. Make sure you have a decal and a mesh to apply it to in your scene.

   Note: For more information on creating decals, see How to Create a Projectable Decal.

2. Create a new Mesh Item and keep it empty.
3. In the empty Mesh Item, create a **Decal Planar** MeshOp.

   The following window opens:



Create Decal Planar Mesh Operator

Target Mesh (none)
Source Planar Mesh (none)
Axis Auto
Offset 0 m
UV Relax Iteration 1
Source Mesh Visibility Box

4. In the **Target Mesh** parameter, assign the model you want the decal to be projected onto.
5. In the **Source Planar Mesh** parameter, assign the decal that you want to project onto the Target Mesh.

Note: You can use a Replicator item as a source. The local UVs, Depth and Offset per source plane are stored in special weight maps in the source mesh item. Modo automatically converts the Replicator item to internal source mesh to get the local per plane weight attributes. This allows you to use the Replicator item as a decal source without freezing it into the mesh with per source plane weight attributes.

6. Increase the **Offset** to 1 mm.

   Offset controls how far away from the mesh the decal sits. If the decal is left at 0 m, it'll be sat in the mesh and may look distorted. By placing the decal 1 mm away from the mesh, the decal is stilling slightly offset from the surface of the mesh allowing the decal to be seen in full.
7. **Axis**, **UV Relax Iteration** and **Source Mesh Visibility** can be left as is.
8. Click **OK**.

To project the decal onto your mesh:

1. Select the decal in the items list.
2. Left-click and drag the decal in the items list and hold it over the model you want to apply it to.
3. In the menu that appears, click **Align and Place**.
4. The decal is placed on the mesh.

As long as the Mesh Item containing the decal is selected, you are able to translate and scale your decal as you would were it not projected onto a mesh. When repositioning a decal, it's recommended that the decal is first selected while in item mode, and then selected using a component mode such as **Polygon Selection**. This is so that item-level transformations are left as their default, preventing any unexpected results when the decal is projected.

The Target Mesh of your decal should also have item-level transformations left as their default. This can be done by freezing your mesh, so that all transforms are at zero. For more information on freezing transformations and different selection modes in Modo, see Freezing Geometry and Selection Modes.

Decal Planar Settings

There are a number of settings available in the **Properties** tab for the **Decal Planar** MeshOp.

Control	Type	What it Does
Enable	checkbox	When check, it enables the MeshOp. When unchecked, it disables the MeshOp. **Enable** only impacts the MeshOp that enabled/disabled on. If there are

Control	Type	What it Does
		multiple Mesh Operations in a stack, they will remain active but enabling or disabling a MeshOp may impact the order of operations.
Use World Transform	checkbox	Enabled by default. Calculates transforms based on the World Position of the Mesh Operation. In most cases, this provides expected results.
Axis	button	Controls which axis the decal is projected along. • **X** - Projects decal along the X Axis of the mesh • **Y** - Projects decal along the Y Axis of the mesh • **Z** - Projects decal along the Z Axis of the mesh • **Auto** - Projects the decal along the surface normals of the mesh
Offset	text field	Determines how close to or how far away form the surface of the mesh the projected decal sits.
UV Relax Iteration	text field	Determines how many times the UVs belonging to the projected decal are relaxed. Useful for preventing image distortion along a surface.
Relax Mode	dropdown	Allows you to choose from various modes that control how the UVs of your decal are relaxed. • **Spring** - Generates a 2D dynamic simulation that makes all the edges in the UV map act like springs, each pulling on the others to find their equilibrium. This method doesn't deal with overlapping UVs very well. • **Conformal** - Relaxes the mesh using the ABF++ (Asynchronous Forward Bounding) algorithm to smooth out the UV map, as with the Unwrap tool. • **Adaptive** - An adaptive attempt to retain the original relative polygon size from the 3D mesh surface. • **Angle Based** - Uses the LSCM (Least Square Conformal Maps) algorithm to smooth out the UV map, as with the Unwrap tool.
Positive Depth	text field	Allows you to determine how far into your asset the Decal is projected.
Negative Depth	text field	Allows you to determine how far away from the surface of your asset that the projector projecting the decal is sat.
Remove Coplanar	checkbox	Simplifies decal polygons by removing coplanar edges from the decal mesh.

Control	Type	What it Does
Edges		
Include Hierarchy	checkbox	Allows a set of stacked decals to be used. Useful when layering multiple decals on top of each other.
Show Depth Handle	checkbox	Makes visible depth handles that can be used to manipulate **Positive** and **Negative Depth** parameters.

Create and Hand Paint Decals With Mesh Paint Replica

Decals created by pressing the **Decals...** button in the **Create** tab can be manipulated using the **Mesh paint Replica** tool.

Not only does **Mesh paint Replica** allow you to hand paint decals on a mesh, but it also allows you to edit the placement, scale, offset and orientation of placed decals, as well as the ability to switch between multiple decals in the scene. This makes it easier to set up and use decals.

Note: For more information on how to procedurally generate decals with **Decal Planar**, see Procedurally Project Planar Decals with The Decal Planar MeshOp.

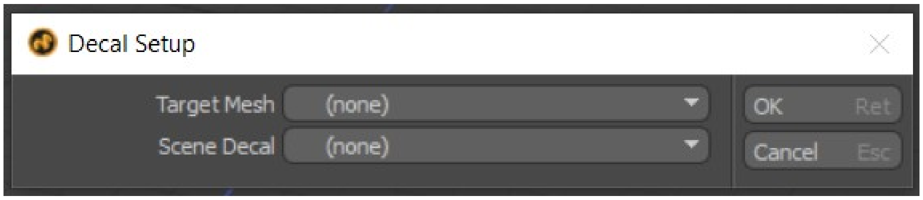
Creating a Decal

Once the **Decals...** button is pressed, a decal is created and the **Mesh paint Replica** tool becomes active/ Once active, you are able to place decals on your mesh and edit them as needed.

To create a decal with the **Decals...** button:

1. With both the decal you wish to project and the mesh you want the decal to be applied to in the scene, press the **Decals...** button found in the **Create** tab.

   Once clicked, the following window opens:

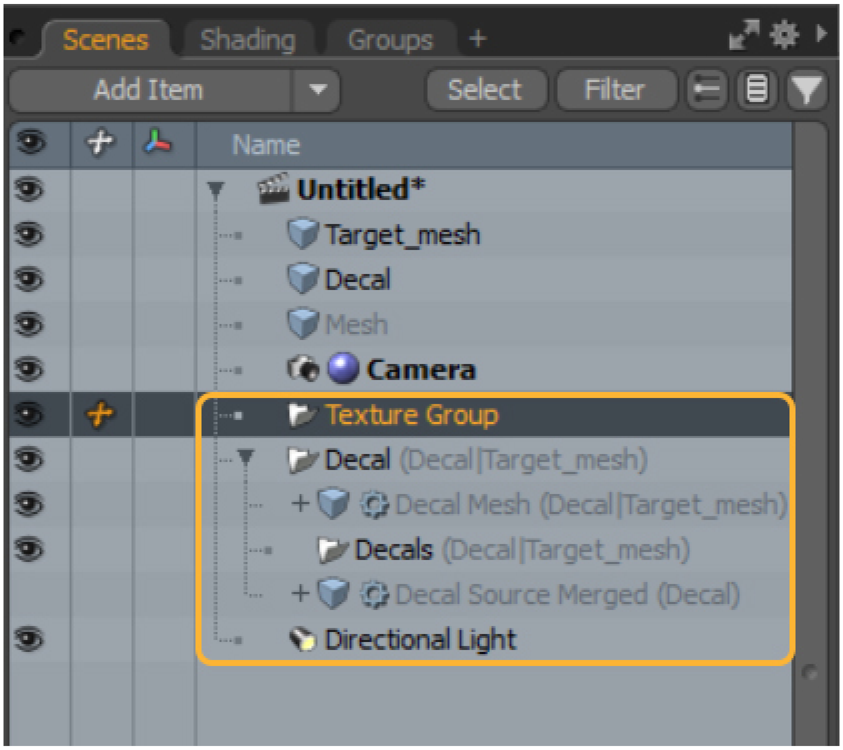

Decal Setup
OK Ret
Cancel Esc

Target Mesh (none)
Scene Decal (none)

2. In the **Target Mesh** parameter, select the mesh you want the decal to be painted on from the dropdown.
3. In the **Scene Decal** parameter, select your decal from the dropdown.
4. Click **OK**.

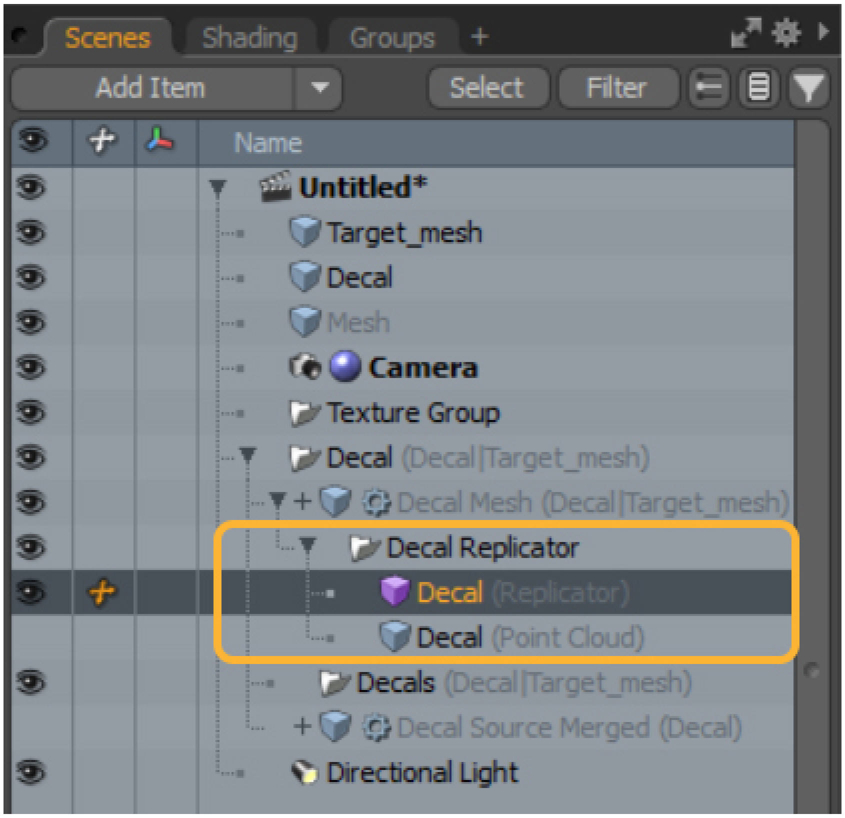
Once **OK** is clicked, **Mesh Paint Replica** is automatically activated. Once active, you can click anywhere on your mesh to place a decal. There is no limit to how many times you can place a decal on your mesh using **Mesh paint Replica**, and multiple decals can be created using this method.

Internally, decals created using this method are set up in the same way that they would be if they were created manually. Once a decal is created, a folder containing the **Decal Planar** MeshOp projecting the decal can be found in the **Items List**.


Scenes
Shading
Groups
+
Add Item
Select
Filter
Name
Untitled*
Target_mesh
Decal
Mesh
Camera
Texture Group
Decal (Decal|Target_mesh)
Decal Mesh (Decal|Target_mesh)
Decals (Decal|Target_mesh)
Decal Source Merged (Decal)
Directional Light

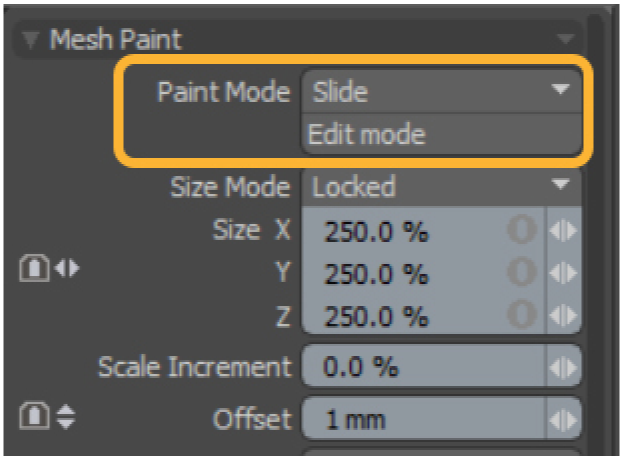
As more decals are placed on the mesh, a **Replicator** folder is created in the Tool Pipe of the **Decal Planar** MeshOp. All the decals placed by the **Mesh paint Replica** tool are created as Replicators. Much like instances, Replicators allow you to create geometry that is only seen at render time allowing you to create hundreds of decals without taxing your machine.

Note: For more information on Replicators, see Replicators.


Scenes
Shading
Groups
+
Add Item
Select
Filter
Name
Untitled*
Target_mesh
Decal
Mesh
Camera
Texture Group
Decal (Decal|Target_mesh)
Decal Mesh (Decal|Target_mesh)
Decal Replicator
Decal (Replicator)
Decal (Point Cloud)
Decals (Decal|Target_mesh)
Decal Source Merged (Decal)
Directional Light

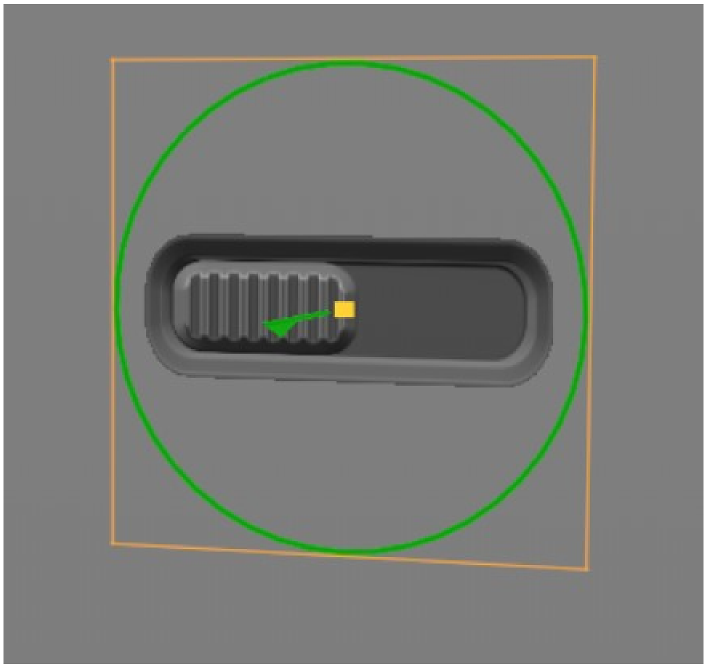
Editing Your Decal

While **Mesh Paint Replica** is active, you can continuously edit the placement, scale, offset and orientation of your decals. This can be done by clicking **Edit Mode** in the tool's **Properties** window.

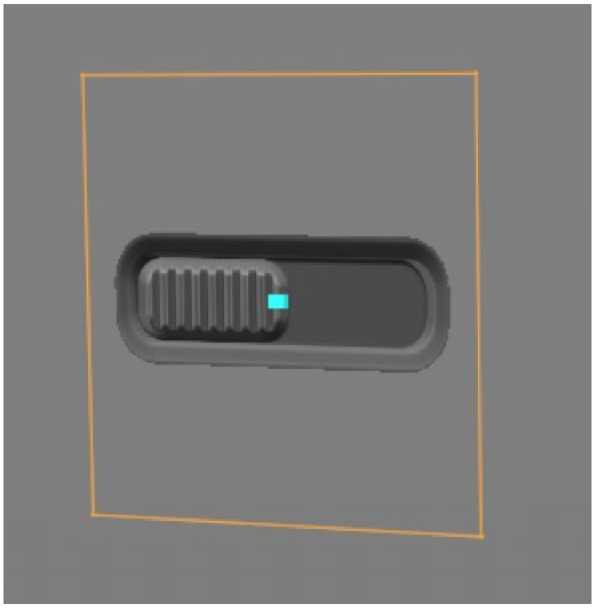


Paint Mode Slide
Edit mode
Size Mode Locked
Size X 250.0 %
Y 250.0 %
Z 250.0 %
Scale Increment 0.0 %
Offset 1 mm

Once **Edit Mode** is enabled, a handle appears on your decals. Active decals that can be moved will have a green transform handle available, while decals that can be edited but are not currently selected for editing have a blue square in the center of the decal. To select a decal for editing, click on the blue square.



Decal selected to be edited



Decal not selected to be edited

While a decal is selected, you can use the handle to Translate and Rotate the decal without needing to exit the viewport.

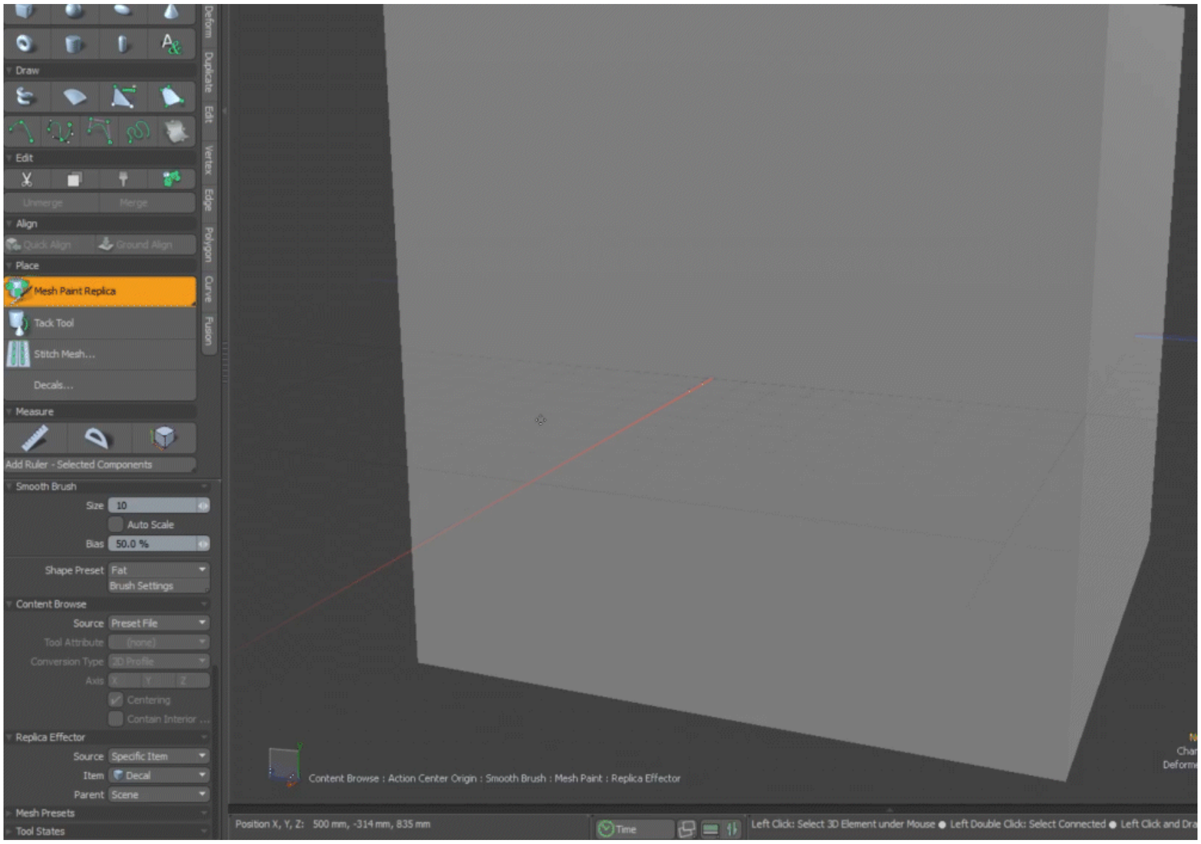
Within the **Tool Properties** for **Mesh Paint Replica**, there are a number of ways that you can edit your decal which can be found on the page for Mesh Paint. However, when working with Decals, there are a number of options that are specific to Decal workflows as outlined below:

Feature	**Properties**
Edit Mode	Clicking this button activates Edit Mode, and makes all decals placed on the mesh editable. Once clicked, the **Edit Button** is replaced by a button called **Standard Mode**. Clicking this button locks all placed decals for editing, and allows you to place more decals.
Offset	Determines how far away from the surface of the mesh the plane carrying your decal sits. This is different to the Offset parameter available in the Decal Planar MeshOp.

Switching Between Multiple Decals

There is no limit to how many decals you can create using this method, or how many decals you can place on a mesh. While using the **Mesh Paint Replica** tool, you can switch between decals by changing the **Item** dropdown found in the **Replica Effector** menu in the **Tool Properties** for **Mesh Paint Replica**.

Even after the **Item** is changed, you are able to enter **Edit Mode** and edit your decals. **Items** can be switched between whilst in **Edit Mode**, meaning you do not need to switch back to **Standard Mode** when switching between decals.


Draw
Edit
Unmerge
Merge
Align
Quick Align
Ground Align
Place
Mesh Paint Replica
Tack Tool
Stitch Mesh...
Decals...
Measure
Add Ruler - Selected Components
Deform
Duplicate
Edit
Vertex
Edge
Polygon
Curve
Fusion
Time

Smooth Brush
Size 10
Auto Scale
Bias 50.0 %
Shape Preset Fat
Brush Settings
Content Browse
Source Preset File
Tool Attribute (none)
Conversion Type 3D Profile
Axis X Y Z
Centering
Contain Interior ...
Replica Effector
Source Specific Item
Item Decal
Parent Scene
Mesh Presets
Tool States

Content Browse : Action Center Origin : Smooth Brush : Mesh Paint : Replica Effector

Position X, Y, Z: 500 mm, -314 mm, 835 mm

Left Click: Select 3D Element under Mouse ● Left Double Click: Select Connected ● Left Click and Dra

Example of a decal being created, switching over to a different decal and then entering **Edit Mode** to edit both of them individually.

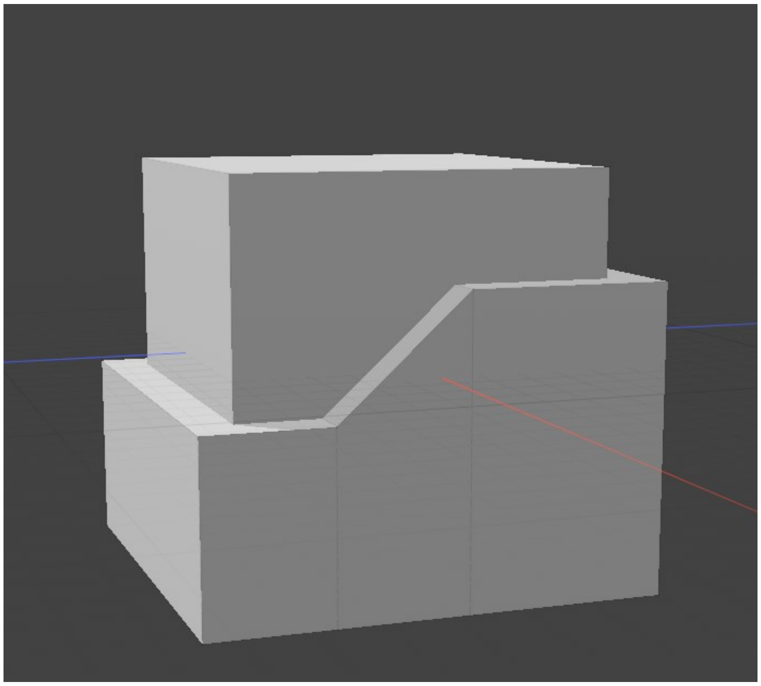
Add Seams to Models With Intersecting Geometry and Seam Decals

Seam Decals allow you to use two pieces of intersecting geometry or a curve to create a seam along a mesh. Once created, a material containing a normal map with a seam texture is applied to a strip of polygons that sits on the surface of your mesh. This strip of polygons conforms to the shape created by the intersecting geometry, or the curve that is projected onto the mesh receiving the decal.

Seam Decals are especially useful when adding panel lines to a hard surface model, or seams to clothing. Custom normal maps that have been baked in Modo or a third party application can be used with Seam Decals, allowing you to add custom seams to your assets.

To create a Seam Decal driven by intersecting meshes:

1. Create two meshes on two separate mesh items. Once created, move the mesh that you want to create the decal so that it is intersecting with the mesh that you want the decal to sit on.



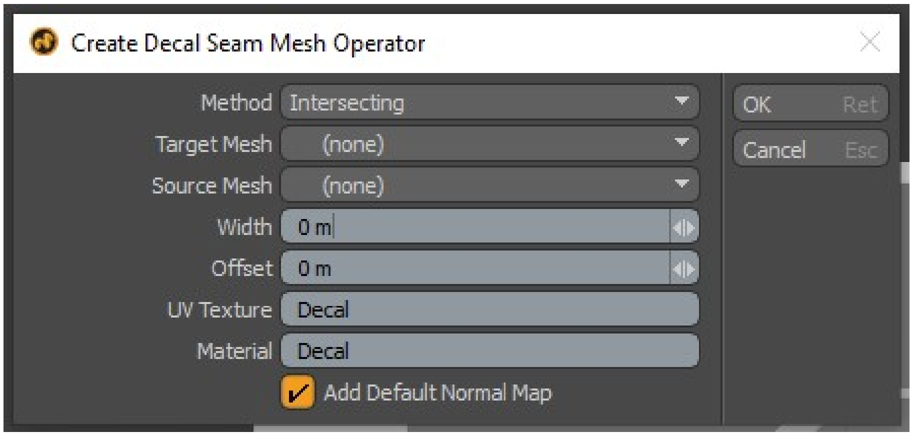
The larger mesh will drive the seam decal, while

Note: For seam decals to work, both meshes must be separate mesh items. One mesh item is treated as the mesh that the decal is placed on, while the other mesh is treated as the mesh driving the decal. Where the two mesh items intersect is where the decal is placed. For more information on mesh items, see Mesh Item.

2. Create a new Mesh Item and leave it empty.

   This Mesh Item will hold the Mesh Operation driving the Seam Decal. For the MeshOp to be added, the Mesh item it belongs to much be empty.
3. In the empty Mesh Item, create a **Decal Seam** MeshOp.

   The following window opens.



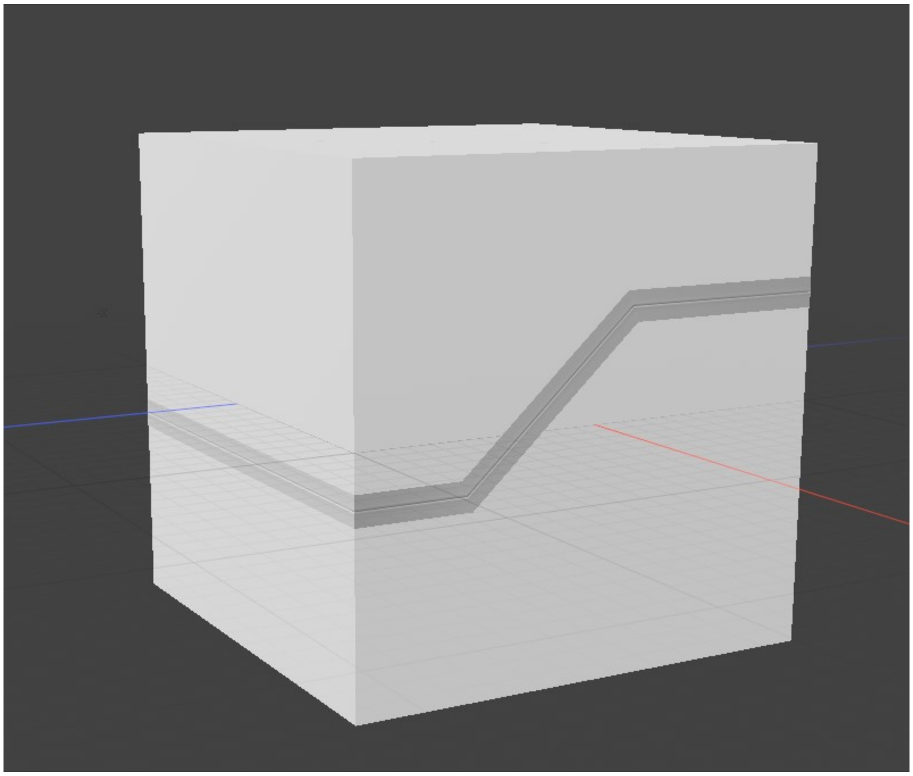
Method: Intersecting
Target Mesh: (none)
Source Mesh: (none)
Width: 0 m
Offset: 0 m
UV Texture: Decal
Material: Decal
☑ Add Default Normal Map

The Decal Seam MeshOp allows you to set the Target and Source meshes for the decal, and then generates the decals based on the information that is input in the window above.

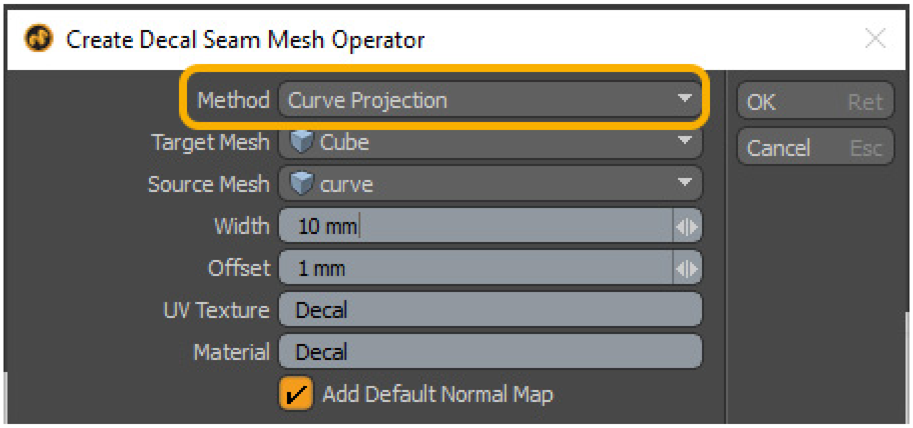
4. In the **Target Mesh** dropdown, select the Mesh Item you want the Seam Decal to be on.
5. In the **Source Mesh** dropdown, select the intersecting Mesh Item. The points where this source mesh and the target mesh meet are used to create the decal seam.
6. Adjust the **Width** and **Offset** parameters as needed.
    - **Width** allows you to adjust the thickness of your Seam Decal.
    - **Offset** controls how far away from the surface of the mesh the decal sits.
7. Leave **UV Texture** and **Material** as they are, and leave **Add Default Normal Map** ticked.
    - **UV Texture** allows you to add a custom name to the UV Map that is generated when the Seam Decal is created.
    - **Material** allows you to add a custom name to the material that is generated and applied to the seam decal when it is created.
    - **Add Default Normal Map** generates a basic normal map and applies it to the Seam Decal material that is created when the Seam Decal is created. If using a custom normal map, this option should be left un-ticked and then the normal map applied in the Decal.
8. Click **OK**.

A seam decal is added to the mesh that was assigned in the **Target Mesh** parameter when the **Decal Seam** MeshOp was created.

When a seam decal is generated, a new material called **Decal** is created in the shader tree. This material contains the normal map that is giving the illusion of new surface details being created on the asset. Additional material settings such as Specular, Diffuse and Roughness can also be adjusted as needed without impacting the material's normal map or how the seam decal sits on the mesh.



Seam Decals can also be generated with Curves. The workflow for curves is the same as the one outlined above, however the **Method** should be changed to **Curve Projection** in the **Create Decal Seam Mesh Operator** window.



Create Decal Seam Mesh Operator

Method: Curve Projection

Target Mesh: Cube

Source Mesh: curve

Width: 10 mm

Offset: 1 mm

UV Texture: Decal

Material: Decal

☑ Add Default Normal Map

Seam Decals support all linear curve types in Modo, such as Curve, Bezier, B-Spline and Polyline.

Decal Seam Settings

There are a number of settings available in the **Properties** tab for the **Decal Seam** MeshOp.

Control	Type	What it Does	Notes
Enable	checkbox	When checked, it enables the MeshOp. When unchecked, it disables the MeshOp. The results of **Enable**being turned off or on, only impacts the MeshOp it's been applied to. If there are multiple Mesh Operations in a stack, they will remain active but enabling or disabling a MeshOp may impact the order of operations.	Enabled by default.
Use World Transform	checkbox	Calculates transforms based on the World Position of the Mesh Operation.	Enabled by default.
Method	dropdown	The method used to create the decal. • **Intersecting** - seam is created by an intersecting piece of geometry • **Curve Projection** - seam is created by a projected curve	
Width	text field	Determines how wide the strip of polygons generated by the decal seam MeshOp is. The larger the value in the **Width** parameter is, the easier it is to see from a distance.	
Offset	text field	Determines how far away from the surface of the mesh the decal sits.	
UV Texture	text field	The name of the UV Texture Map that the seam decal creates.	Set to Decal by default.
Material	text field	The name of the Material that the seam decal creates.	Set to Decal by default.
UV Direction	dropdown	Changes the direction of the UV map applied.	Set to **V Direction** by default.
Curve Refinement Angle	text field	Determines the amount of spans needed to create the seam decal. Smaller angles generate more spans, while larger	Disabled unless the **Method** is set to **Curve Projection**.

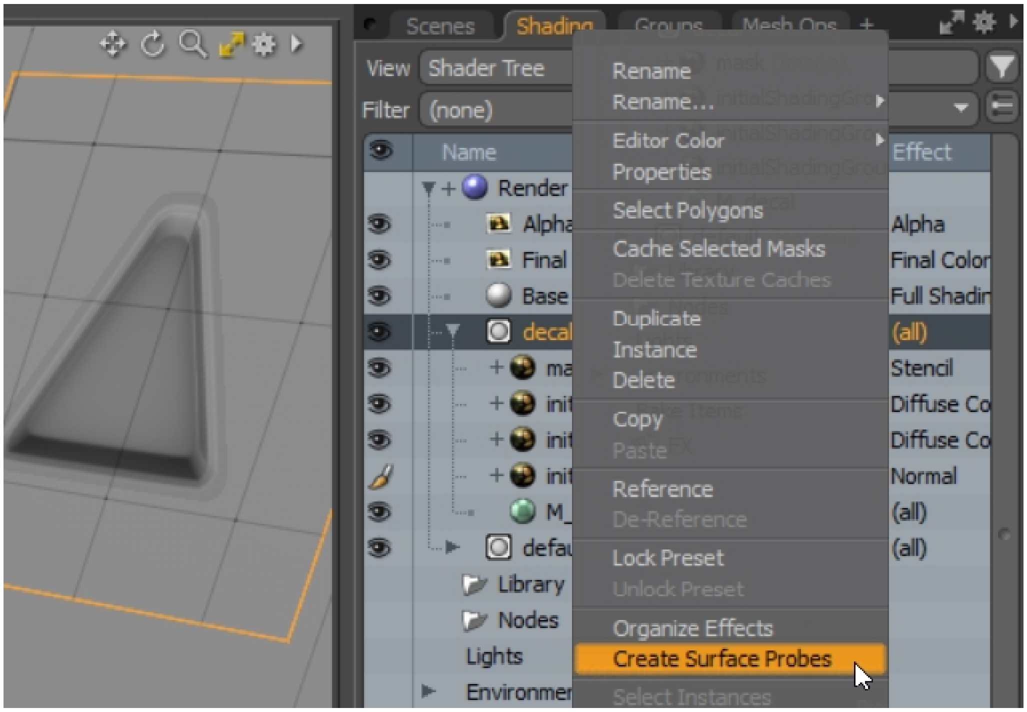
Control	Type	What it Does	Notes
		angles generate less.	Spans are the distance between two points along a curve.
Curve Interpolation	checkbox	Converts intersecting edges into a curve and smooths them.	Disabled by default.
Add Segments	text field	Increases or decreases number of segments in the quad strip.	Disabled unless **Curve Interpolation** is enabled.
Curve Smoothing Iterations	text field	Determines how many times Modo smooths the quad strip.	Disabled unless **Curve Interpolation** is enabled.
Slice Count	text field	Increases or decreases the amount of slices running through a quad strip.	Set to 0 by default.

Render and Bake Decals Using Surface Probes

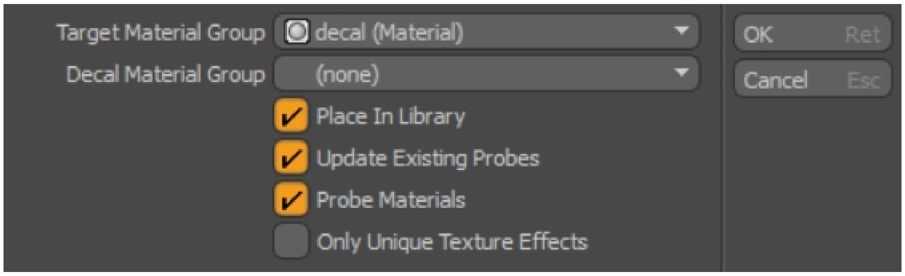
Surface Probe is a texture item available in the Shader Tree that sends probe rays in multiple directions from a target surface until it hits a decal surface. Probe rays that hit a decal surface take material and texture values from the decal and apply them to the target surface. Surface Probe textures remove the need for floating decal geometry by directly baking the decal textures and materials on to the target surface.

In this example scene, Surface Probe textures transfer material and texture values from one surface to the other. To bake a decal material on to a target material:

1. Right-click the target material or material group in the Shader Tree and select **Create Surface Probes**.


Scenes
Shading
Groups
Mesh Ops
View
Shader Tree
Filter
(none)
Name
Effect
Render
Alpha
Final
Base
decal
Library
Nodes
Lights
Environmer
Alpha
Final Color
Full Shadir
(all)
Stencil
Diffuse Co
Diffuse Co
Normal
(all)
(all)
Rename
Rename...
Editor Color
Properties
Select Polygons
Cache Selected Masks
Delete Texture Caches
Duplicate
Instance
Delete
Copy
Paste
Reference
De-Reference
Lock Preset
Unlock Preset
Organize Effects
Create Surface Probes
Select Instances

The **Create Surface Probes** dialog is displayed and the **Target Material Group** is already selected.



Target Material Group: decal (Material)

Decal Material Group: (none)

☑ Place In Library

☑ Update Existing Probes

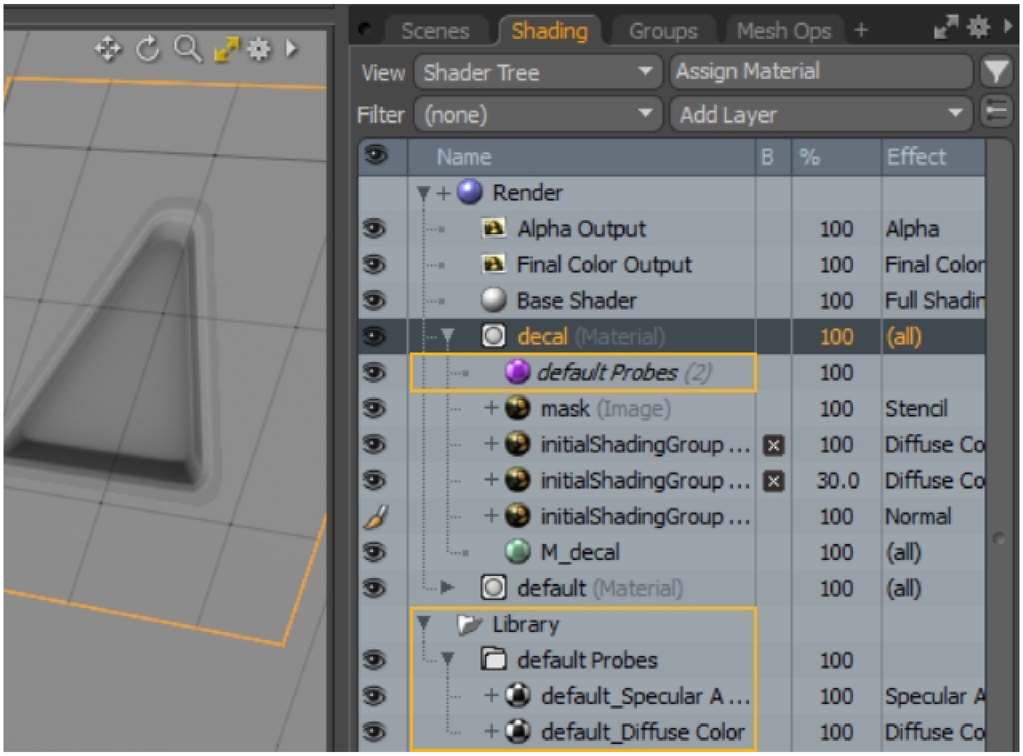
☑ Probe Materials

☐ Only Unique Texture Effects

OK Ret
Cancel Esc

2. Select the **Decal Material Group** you want to bake down on to the target material.

   New materials are flattened in to a single Shader Tree entry for clarity and copied individually into the **Library** folder by default.



View: Shader Tree

Filter: (none)

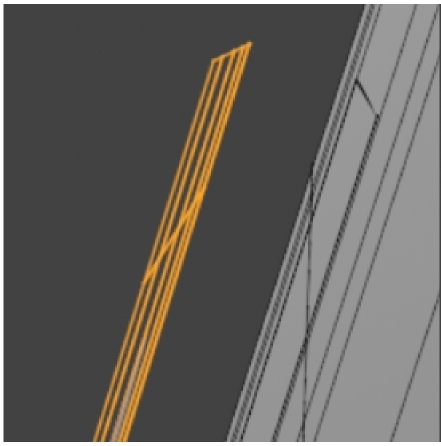
Name	B	%	Effect
Render			
Alpha Output		100	Alpha
Final Color Output		100	Final Color
Base Shader		100	Full Shadin
decal (Material)		100	(all)
default Probes (2)		100	
mask (Image)		100	Stencil
initialShadingGroup ...	☒	100	Diffuse Co
initialShadingGroup ...	☒	30.0	Diffuse Co
initialShadingGroup ...		100	Normal
M_decal		100	(all)
default (Material)		100	(all)
Library			
default Probes		100	
default_Specular A ...		100	Specular A
default_Diffuse Color		100	Diffuse Co

3. Existing surface probe textures in the probe group are replaced by new SP textures by default. Disable **Place In Library** if you want to keep the individual materials in the Shader Tree.
4. Disable **Update Existing Probes** if you only want to add new surface probe textures to the existing group.

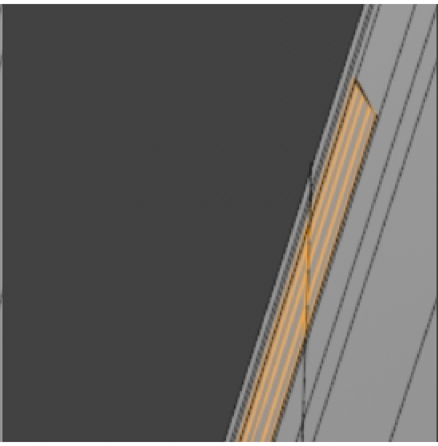
   The **Decal Material Group** is also probed by default, so that any **Properties** set in the group are taken into account.
5. Disable **Probe Materials**, if you want to ignore the material item from the decal and only transfer textures from the decal material.

   All texture effects are rendered to the target material by default.
6. Enable **Only Unique Texture Effects** if you only want to bake effects that are unique to the decal and not in the receiving material group. If you're baking the decals, use Preview Bake Mode for the best results.

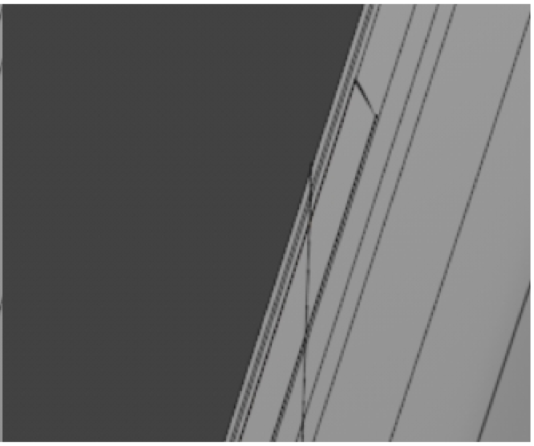
   To get a good result from baking decals down onto a surface, set the Decal Planar operator **Offset** control to the smallest value possible without the decal intersecting with the target surface.



An relatively high Offset value of 100 mm can cause artifacts in the baked texture



A low Offset value of 1 mm sits the decal as close as possible before baking



Setting the Offset to a value too close to the surface may cause parts of the decal to be occluded

Have a look at Procedurally Project Planar Decals with The Decal Planar MeshOp for more information on setting up decals.

7. Click **OK** to transfer the projected textures onto the target textures.

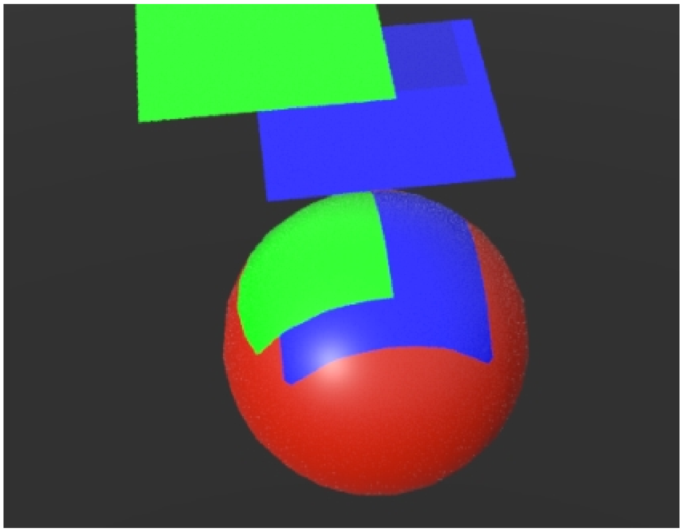
Blending Multiple Decals Together Correctly

The **Probe Blend Mode** option determines which blending mode is used when the probe rays hit more than one decal surface. Each material in the Shader Tree can have its own blend mode, but material blend defaults to **Normal**, which means the decal nearest the target renders first and the decal farthest from the target is rendered on top of everything else.

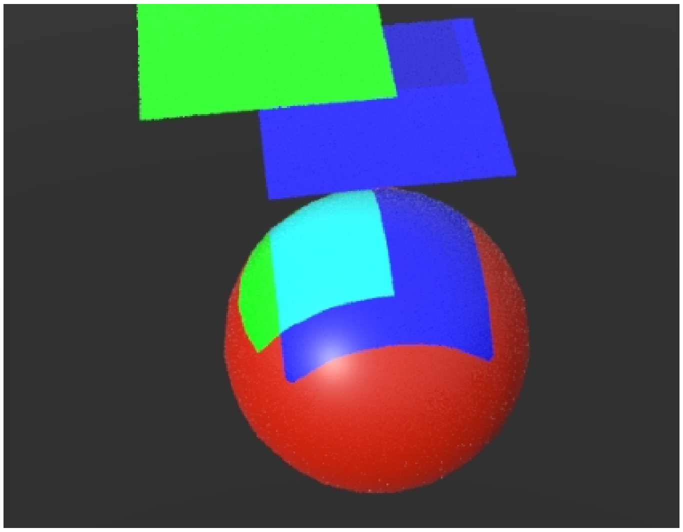
To choose the blend mode:

1. Select a material in the Shader Tree.
2. Go to the **Properties** panel and click the **Surface Probe** > **Probe Blend Mode** dropdown.
3. Select the required blend mode for this material.

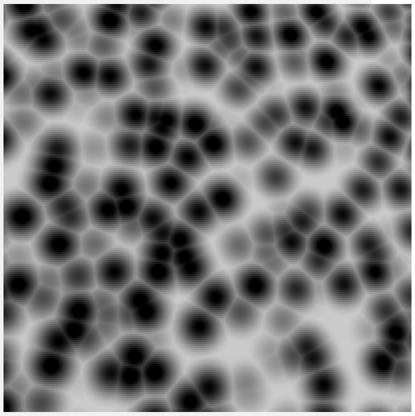
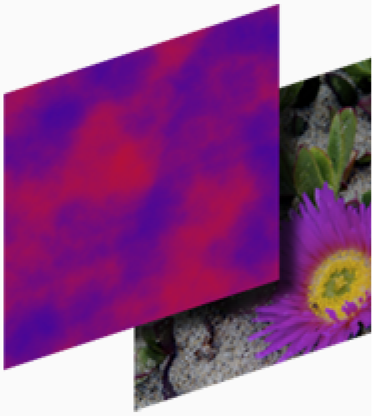
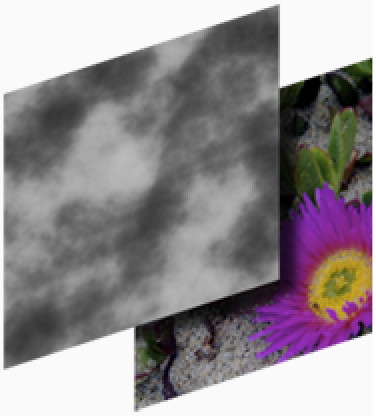
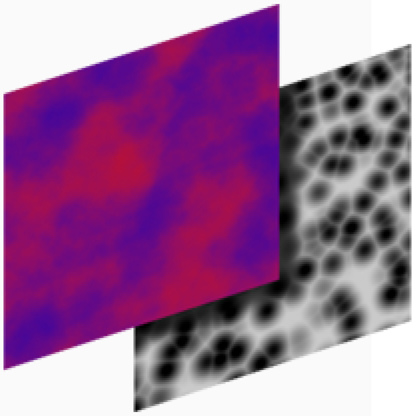
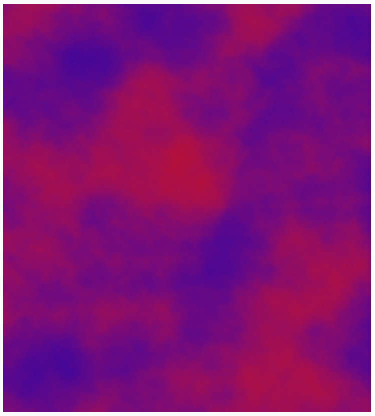
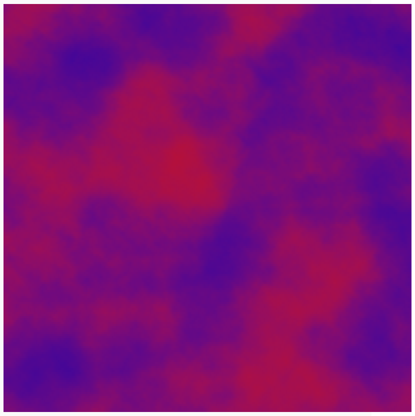
   As an example, the following image shows floating green and blue diffuse color decal surfaces applied to a red target surface in **Normal** blend mode. The decal surfaces are positioned one above the other, so the probe rays hit both and blends values between them. The topmost green decal surface renders over the lower blue decal surface and both decals render over the red target surface.

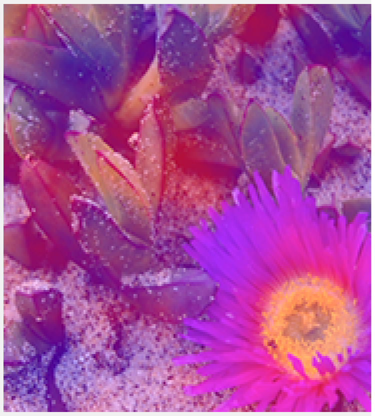
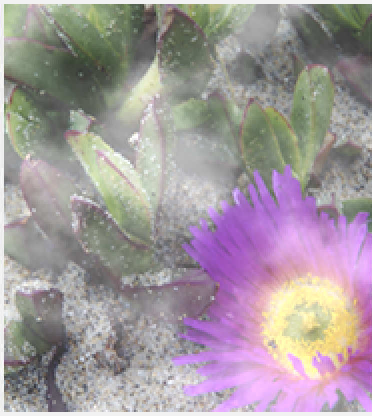
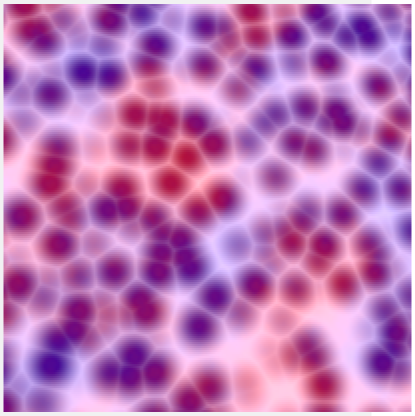
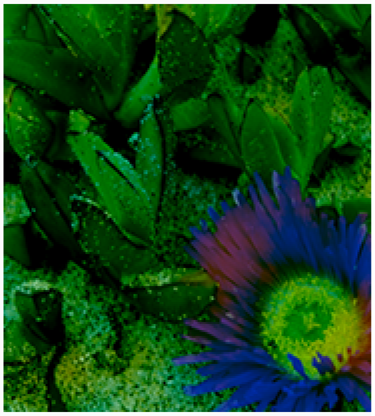
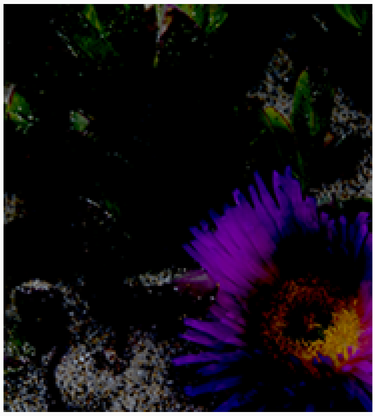
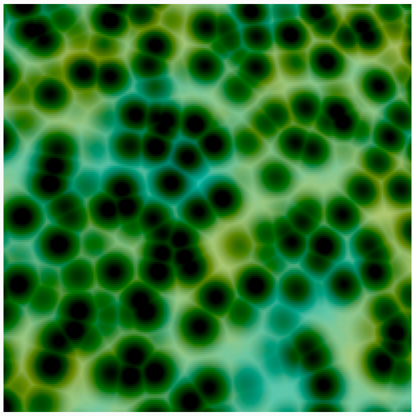
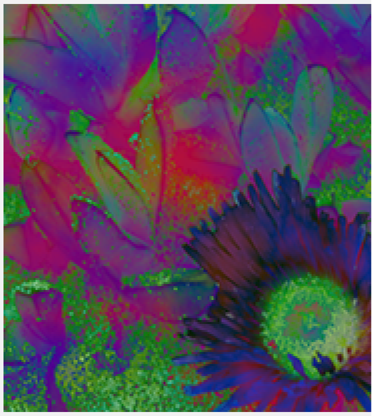
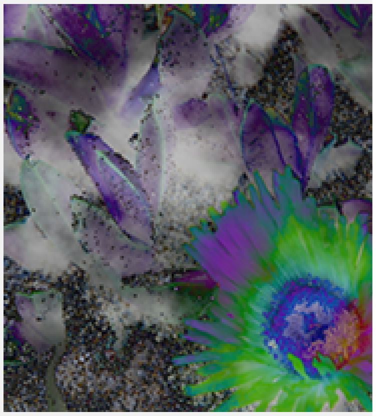
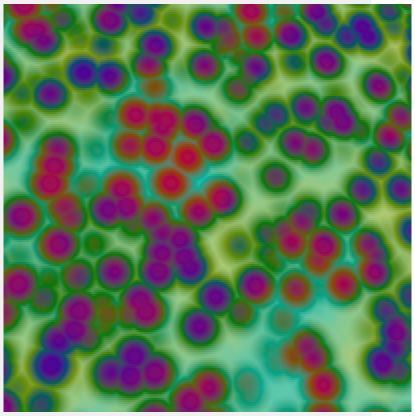


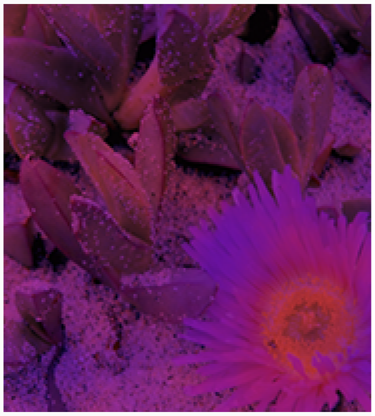
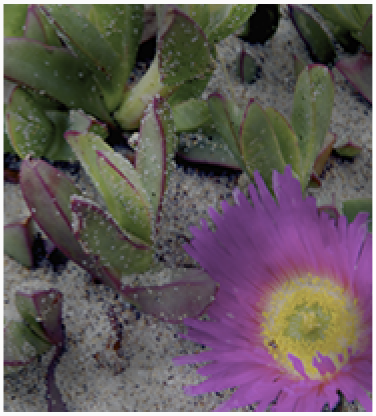
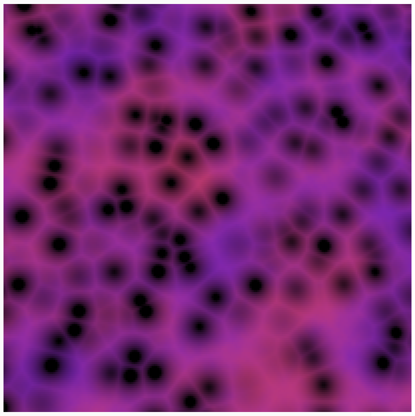
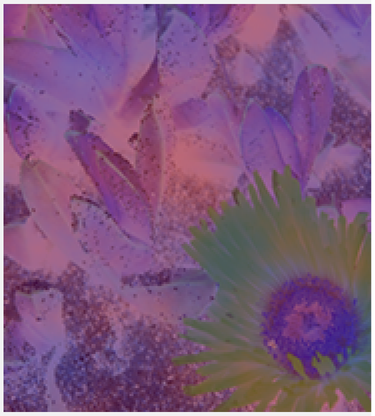
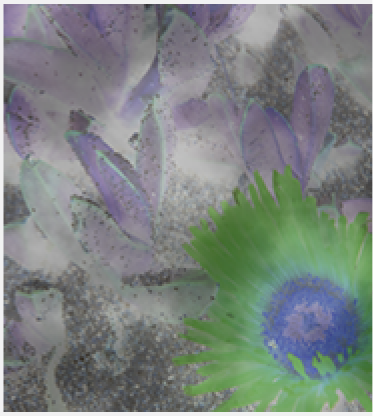
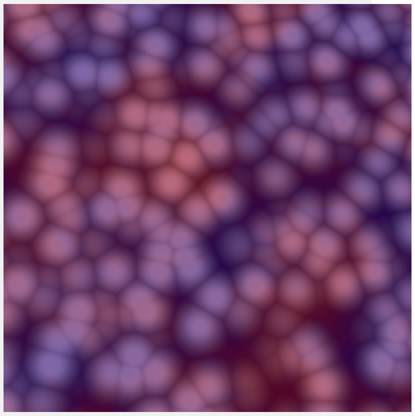
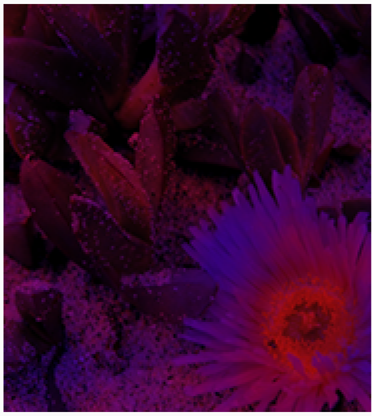
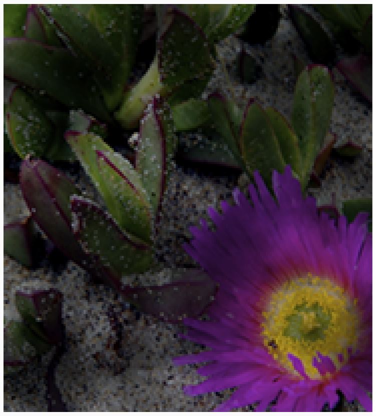
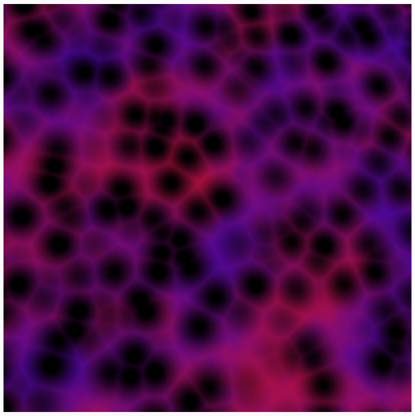
In this example with the same surfaces, but with the **Add** blend mode, the detected decal surfaces add their diffuse colors together, which renders the cyan color on the target surface.

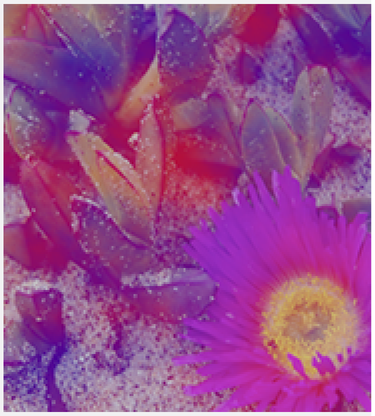
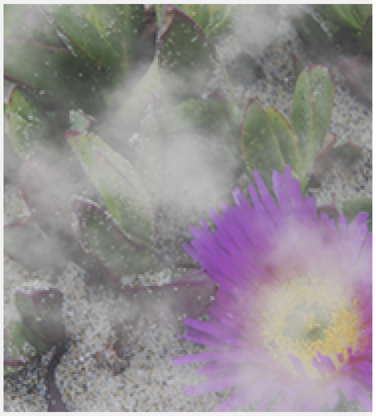
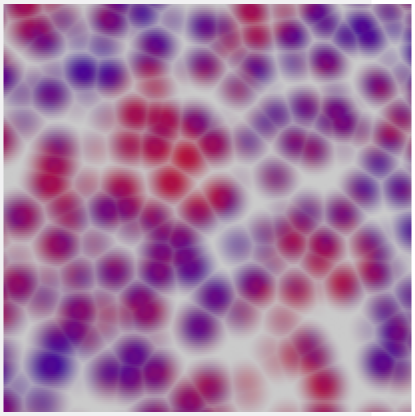
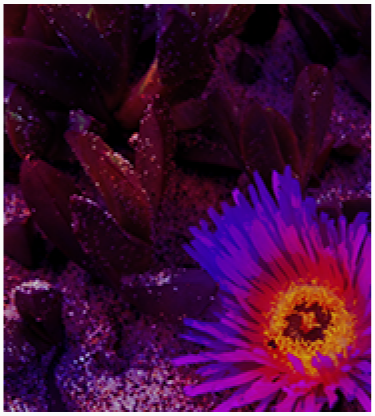
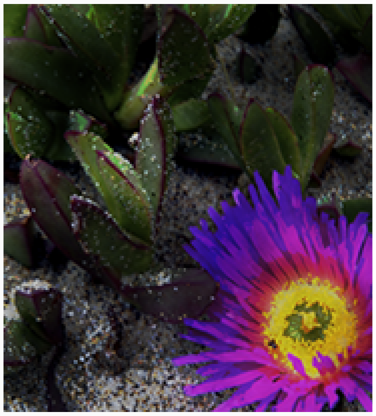
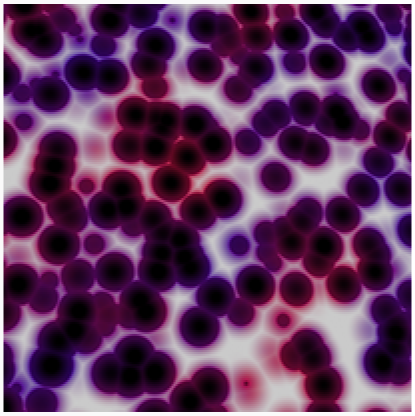
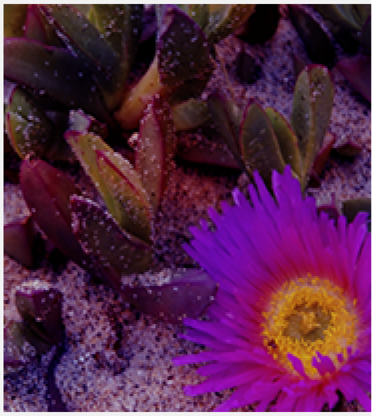
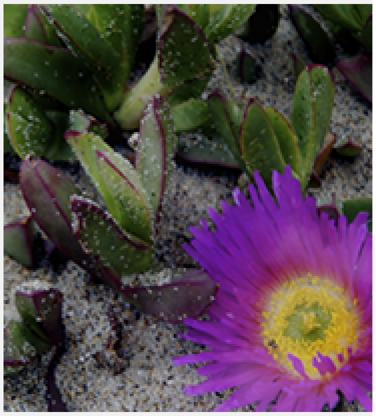
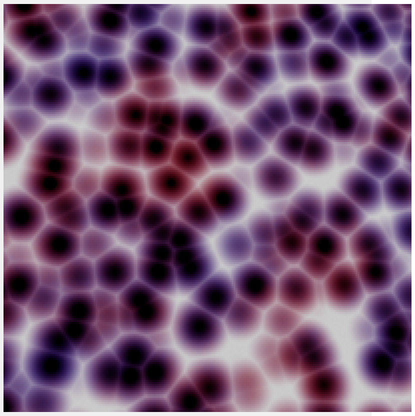


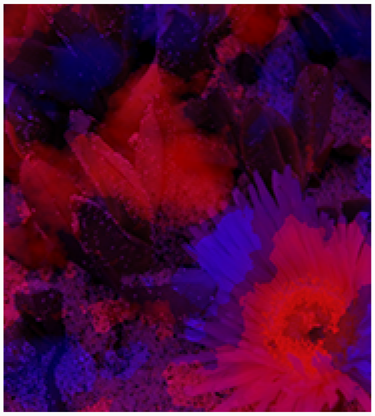
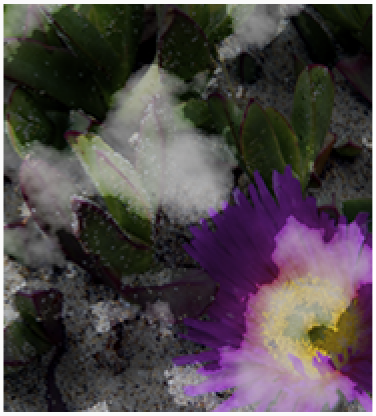
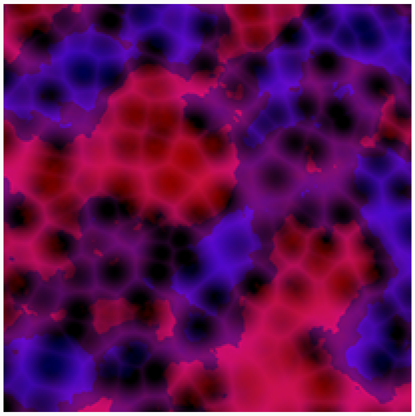
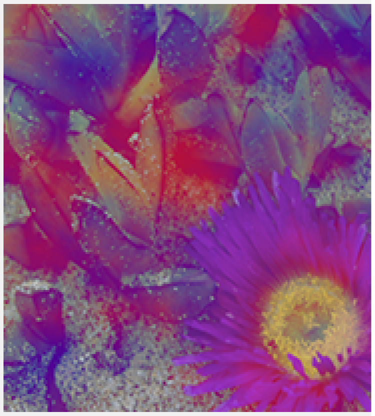
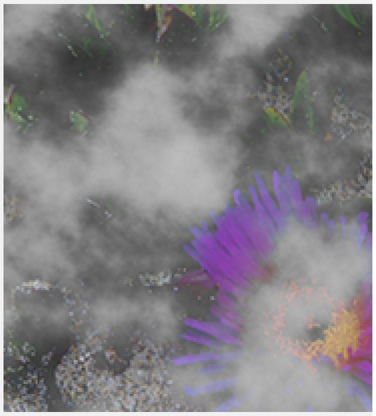
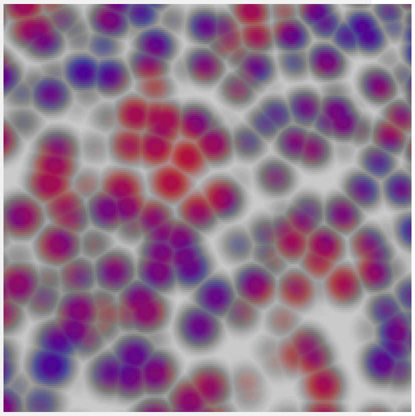
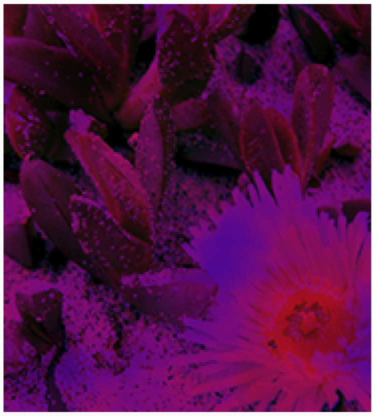
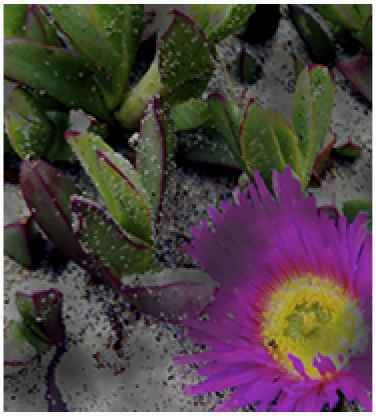
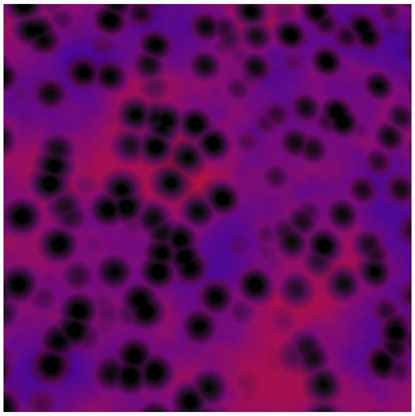
The following table show the various **Probe Blend Modes**, with samples to illustrate the effect.

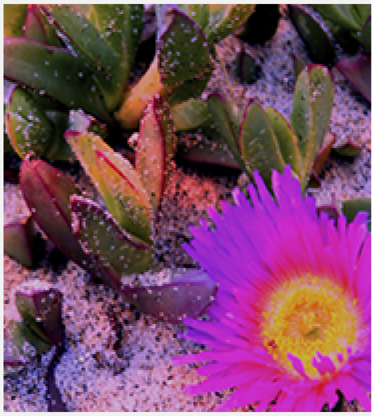
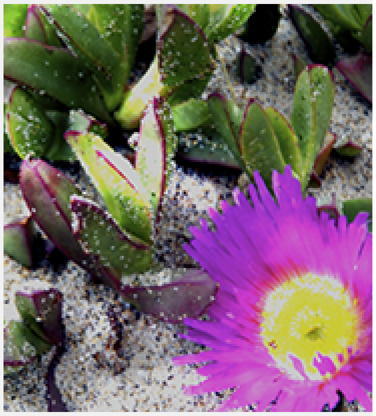
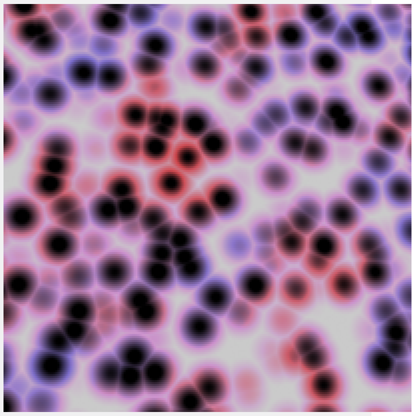
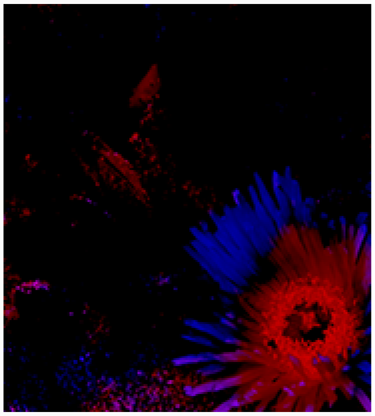
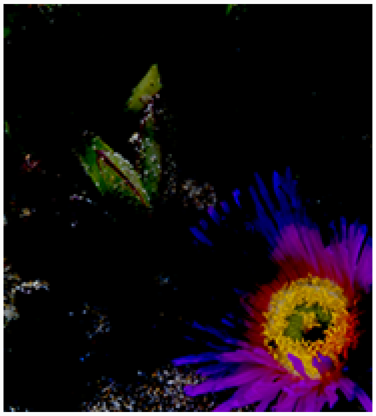
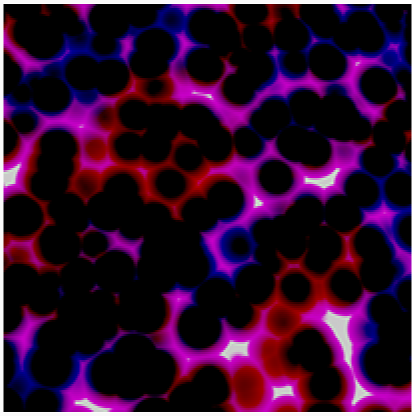
Original Images			
A two decal stack with the noise farthest away from the target surface			
Blend Mode	**Results**		
Normal			

Add			
Subtract			
Difference			

Normal Multiply			
Divide			
Multiply			

Screen			
Overlay			
Soft Light			

Hard Light			
Lighten			
Darken			

Color Dodge			
Color Burn			

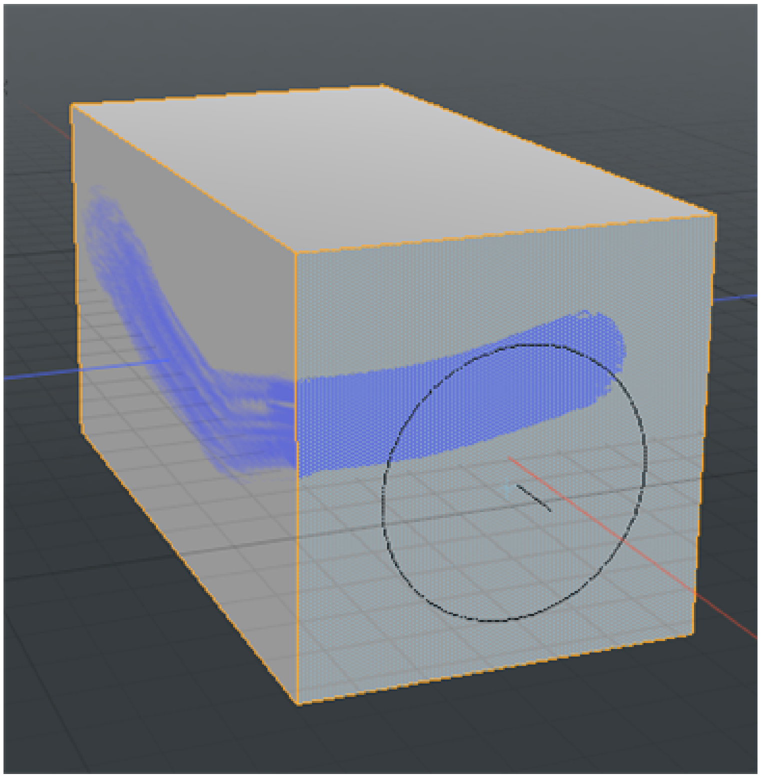
Customizing Paint Tools

Learn about brush tips, inks, and tablet nozzles.

Brush Tips

When you select a tool for sculpting and painting, it is also necessary to choose an appropriate brush tip which defines the way color or form is applied to the surface. By default, Modo auto selects the **Smooth Brush** making it easier for you to start painting or sculpting a surface. You can easily select an alternate tip from the selection available.

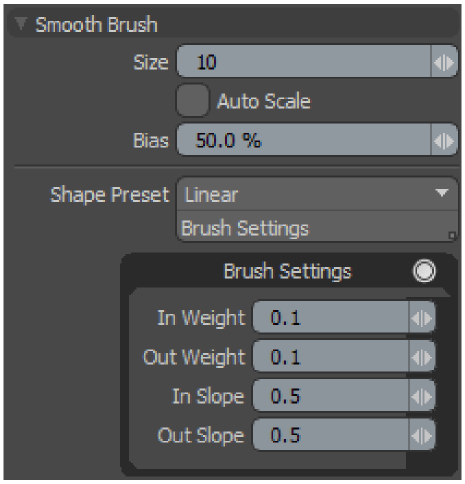
When a brush-based tool is activated, a circle is drawn under the mouse when you hover over a surface. This makes it easier to see where the sculpt or paint is applied when you left-click and drag across the surface of your mesh. To change the size of the brush tip, right-click and drag your mouse.



You can save custom tool and tip combinations using the **Save Brush Preset** button under the brush tip icons section of the toolbox. When selected, each tip has it own set of attributes covered below.

Smooth Brush

The **Smooth Brush** creates a nice feathered airbrush effect.



Smooth Brush

Size: 10

☐ Auto Scale

Bias: 50.0 %

Shape Preset: Linear

Brush Settings

Brush Settings

In Weight: 0.1

Out Weight: 0.1

In Slope: 0.5

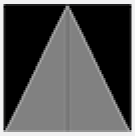
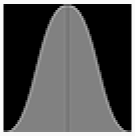
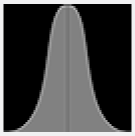
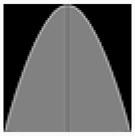
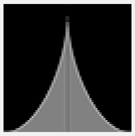
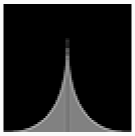
Out Slope: 0.5

Size - This value sets the radius of the brush, in pixels, from screen space. This is important to understand, as zooming the view changes the result of the brush. With the brush size being fixed, zooming in on the model decreases the relative scale of the brush, whereas zooming out increases it. You can interactively adjust the tip **Size** value by right-clicking in the viewport and dragging to adjust an on-screen display representing the brush's size.

Auto Scale - When enabled, the brush **Size** scales automatically to remain constant, regardless of the zoom percentage.

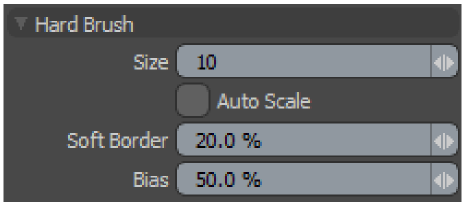
Bias - This option is like a curve adjustment to the brush that favors one side of the range over the other, acting much like a gamma curve would affect the shades of an image.

Shape Preset - This dropdown includes several preset brush options for varying effects. This setting drives the profile of the airbrush. You can also adjust the **In**/**Out Weight** and **Slope** settings to produce customized brushes.

Shape	Name/Description
	**Linear** - creates a straight ramp from the center to the edge, which results in a brush profile that looks like a cone.
	**Smooth** - creates a brush stroke that is opaque in the center and falls off smoothly to the edge of the brush with an S curve shape.
	**Bulge** - changes the bias of the curve to start out slowly and then ramp more harshly towards the middle, resulting in a brush profile that looks similar to a dome.
	**Fat** - brush starts out soft and ramps harshly at the outer edge producing a soft U-shaped brush that varies the most under light tablet input pressure.
	**Sharp** - the falloff is very harsh and steep at the center producing a fine stroke.
	**Very Sharp** - this preset is nearly identical to the **Sharp** brush shape, but with a more extreme falloff for even finer strokes.

Hard Brush

The **Hard Brush** creates a circular brush with a hard outside edge, so that the paint is consistent from the center to the outside edge.



Hard Brush

Size: 10

☐ Auto Scale

Soft Border: 20.0 %

Bias: 50.0 %

Size - This value sets the radius of the brush, in pixels, from screen space. This is important to understand, as zooming the view changes the result of the brush. With the brush size being fixed, zooming in on the model decreases the relative scale of the brush, whereas zooming out increases it. You can interactively adjust the tip **Size** value by right-clicking in the viewport and dragging to adjust an on-screen display representing the brush's size.

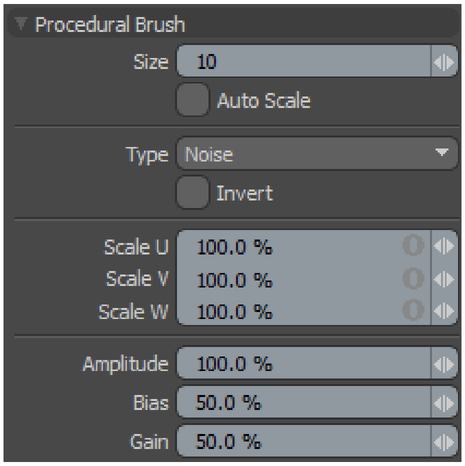
Auto Scale - When enabled, the brush **Size** scales automatically to remain constant, regardless of the zoom percentage.

Soft Border - This value sets a small number of pixels around the outside of the brush for the falloff to occur. This results in a much more pleasing brush result. The default value of 20% yields a brush stroke that has the appearance of being completely hard edged while providing the paint system a small buffer in which to anti alias the stroke. Driving the number higher can result in interesting brushes that have a solid inner area and a linear falloff to the edge of the stroke.

Bias - This option is like a curve adjustment to the brush that favors one side of the range over the other, acting much like a gamma curve would affect the shades of an image.

Procedural Brush

The **Procedural Brush** uses mathematical textures to change the amount of paint applied to the surface as the paint stroke is drawn.



Procedural Brush

Size 10

☐ Auto Scale

Type Noise

☐ Invert

Scale U 100.0 %

Scale V 100.0 %

Scale W 100.0 %

Amplitude 100.0 %

Bias 50.0 %

Gain 50.0 %

Size - This value sets the radius of the brush, in pixels, from screen space. This is important to understand, as zooming the view changes the result of the brush. With the brush size being fixed, zooming in on the model decreases the relative scale of the brush, whereas zooming out increases it. You can interactively adjust the tip **Size** value by right-clicking in the viewport and dragging to adjust an on-screen display representing the brush's size.

Auto Scale - When enabled, the brush **Size** scales automatically to remain constant, regardless of the zoom percentage.

Type - This list has choices for **Noise**, **Cellular**, and **Dots**. Each of these settings creates a different pattern for the brush based on the respective procedural texture.

Invert - This checkbox inverts the values of the procedural.

Scale U - Changes the size of the procedural pattern along the U axis in UV space.

Scale V - Changes the size of the procedural pattern along the V axis in UV space.

Scale W - Changes the size of the procedural pattern along the W axis in UV space (along normal direction of the UV plane).

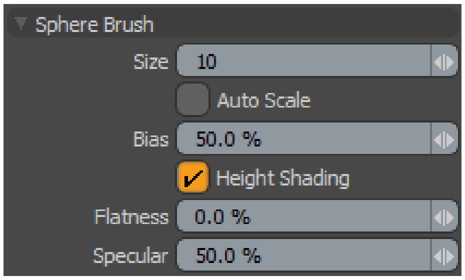
Amplitude - Changes the overall strength of the procedural

Bias - This option is like a curve adjustment to the procedural that favors one side of the range over the other, acting much like a gamma curve would affect the shades of an image.

Gain - This option, in essence, tightens the curve of the **Bias** reducing the shades of gray in the transition. 0% has the maximum number of transition values, attenuating toward 100%, where the procedural would be completely hard edged.

Sphere Brush

The **Sphere Brush** creates a gradient effect inside the brush that uses the FG color in the center of the brush, fading toward black at the edges with a smooth falloff between them. The resulting effect is that of painting a rounded brush stroke. This can be very useful when working with bump or displacement textures. There are two main modes for the **Sphere Brush**. Both are useful for color painting but one is intended mostly for displacement or bump painting.



Sphere Brush

Size 10

☐ Auto Scale

Bias 50.0 %

☑ Height Shading

Flatness 0.0 %

Specular 50.0 %

Size - This value sets the radius of the brush, in pixels, from screen space. This is important to understand, as zooming the view changes the result of the brush. With the brush size being fixed, zooming in on the model decreases the relative scale of the brush, whereas zooming out increases it. You can interactively adjust the tip **Size** value by right-clicking in the viewport and dragging to adjust an on-screen display representing the brush's size.

Auto Scale - When enabled, the brush **Size** scales automatically to remain constant, regardless of the zoom percentage.

Bias - This option is like a curve adjustment to the brush that favors one side of the range over the other, acting much like a gamma curve would affect the shades of an image.

Height Shading - When using the **Sphere Brush** on a flat image map, enabling **Height Shading** shades the brush tip the corresponding shades of gray representing the height of the sphere, producing a cylindrical shading to the brush application.

Flatness - The default value of 0% produces a fully round sphere, attenuating toward 100% producing a completely flat brush, just like the **Hard Brush**.

Specular - This option controls the amount of faux specular highlight on the brush tip's shading. At 0% there is no specular, attenuating toward 100% with full specular shading.

Image Brush and Stamp Brush

☆ Like applying a sticker, you can use the **Image Brush** to apply bitmap texture images to a surface when either painting or sculpting. Depending on the **Mode** selected, you can control the placement of those images. Using the **Image Brush** in **Stamp** mode does not use a procedural stroke, but the click and drag controls the diameter and rotation of the image inside the brush. To apply the image, release the mouse or tablet button.

When dragging the **Image Brush**, moving the tablet or mouse away from the original click location increases the size of the brush. Dragging around in a circular motion adjusts the rotation of the brush. When you first click and drag, a small handle appears. This handle is the visual cue as you edit the size and rotation of the brush. When in **Roller** mode, Modo continuously tiles the image in the direction of the stroke. **Normal** mode sprays images, similar to the behavior of other brush tips.

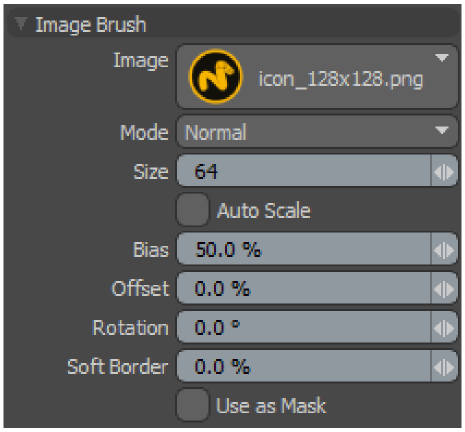


Image Brush
Image: icon_128x128.png
Mode: Normal
Size: 64
☐ Auto Scale
Bias: 50.0 %
Offset: 0.0 %
Rotation: 0.0 °
Soft Border: 0.0 %
☐ Use as Mask

Image - This button allows you to select the bitmap image to be used as the brush tip.

Mode - Options for **Normal**, **Roller**, and **Stamp**:

- **Normal** - acts as any other paint brush with the image controlling the effect.
- **Roller** - forces the brush to act similar to a roller painting tool, where the image is revealed the further you drag the brush. This mode also causes the brush to rotate with your stroke.
- **Stamp** - causes the brush to stamp down where you click. Dragging the mouse then controls the size and rotation of the brush. When you release the mouse the image brush is stamped down at the specified location, scale, and orientation.

Size - This value sets the radius of the brush, in pixels, from screen space. This is important to understand, as zooming the view changes the result of the brush. With the brush size being fixed, zooming in on the model decreases the relative scale of the brush, whereas zooming out increases it. You can interactively adjust the tip **Size** value by right-clicking in the viewport and dragging to adjust an on-screen display representing the brush's size.

Auto Scale - When enabled, the brush **Size** scales automatically to remain constant, regardless of the zoom percentage.

Bias - This option is like a curve adjustment to the brush that favors one side of the range over the other, acting much like a gamma curve would affect the shades of an image.

Offset - This control is particularly useful when working with the sculpting tools. The percentage value allows you to re-map the zero-point of the image. For instance, if your image was a grayscale ramp from mid-gray to full-white, the edges of the brush would create a step effect, since the sculpting tools would have a 50% gray input at the edges of the brush. In this case, you can set the **Offset** value to 50% which then considers the mid-gray value the zero-point and no longer causes any step effect.

Rotation - Use this control to effect the orientation of the brush.

Soft Border - This causes a smooth fade of the image to the edges of the brush. The percentage value determines how far from the edge of the brush diameter the fade begins. If you set a value of 10%, the fade begins 10% in from the outermost edge of the brush. When this value is anything other than 0%, a second ring appears on the brush to indicate where the soft border begins.

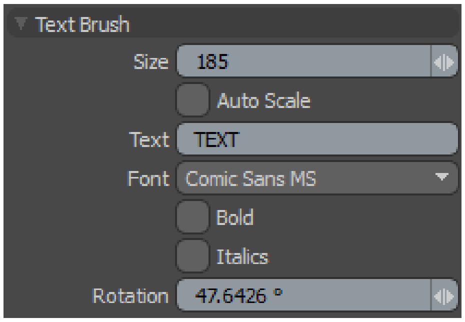
Use as Mask - This control uses the dark areas of an image as alpha.

Text Brush

The **Text Brush** provides a quick and easy way to use many of the fonts on your operating system as a brush while painting or sculpting, easily placing words, letters, and numeric call-outs on textured surfaces. The tool includes a dropdown for choosing the desired font as well as a **Text** input field to set the text string to be used as the brush.

To use the **Text** brush:

1. Enter the desired text string and choose your font from the **Font** dropdown.
2. You can also select **Bold** or **Italics** styles by enabling the checkboxes.
3. Select a painting or sculpting tool, as well as a brush, and then left-click and drag in the 3D viewport:
    - Drag to the left or right to adjust the **Size** value of the brush.
    - Drag around in a circle to adjust the **Rotation** (orientation) of the brush.
4. When satisfied with the size and orientation, release the mouse or pen to project the text onto the surface.



Text Brush

Size 185

☐ Auto Scale

Text TEXT

Font Comic Sans MS

☐ Bold

☐ Italics

Rotation 47.6426 °

Size - The size of the projected text string is determined interactively by dragging in the viewport. The **Size** value represents the radius of the circle generated by the tool when dragging, calculated in screen pixels. The **Text** tool always creates the text string from its center.

Auto Scale - When enabled, the brush **Size** scales automatically to remain constant, regardless of the zoom percentage.

Text - Enter the text to generate when you drag in the viewport. The **Text** entry field holds many characters, but the **Text Brush** tool is best suited to no more than a few words.

Font - This dropdown displays all currently active TrueType and OpenType fonts available to the operating system. Once selected, the chosen font is used by the **Text Brush**.

Bold/Italics - These checkboxes allow you to apply faux **Bold** and **Italic** type faces when fonts of that type are not directly available.

Rotation - The rotation angle of the text (like the size) is determined interactively by dragging in the viewport. The angle is calculated from the point of the initial mouse click in the viewport to the position of the cursor when you release the button, projecting the text block onto the texture.

Customizing Inks

The **Ink** functions in Modo allow you to modulate brush strokes as they are applied, and are located in the **Paint** layout's **Paint Tools** and **Sculpt Tools** toolboxes. **Image**, **Parametric**, **Random**, and **Projection** Inks can affect colors when painting or the strength of sculpting tools. All three **Ink** modes can be utilized individually, or simultaneously in combination, to create interesting effects. To use, click and hold the **Image Ink** button in the interface, select the required **Ink** mode, and then adjust the settings in the viewport or properties panel as necessary.

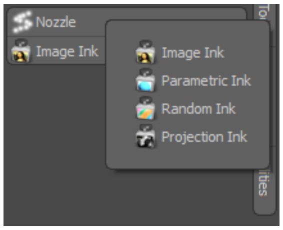


Image Ink

The **Image Ink** function allows you to place a bitmap image in the viewport (sort of like a floating window), and then using the brush to control the application, paint through the image, applying strokes to object surfaces. When **Image Ink** is first enabled, left-click in the viewport to activate the tool's widget.

If an image was selected before this action, its height fills the viewport with the image. The widget has a number of handles represented as small circles that allow you to adjust the projection plane directly in the viewport. The center handle positions it, the right handle scales the widget, and the top handle rotates it. Left-click on a handle and drag to modify.

You can also use the keyboard shortcut **E** to rotate clockwise, and **R** to rotate counter-clockwise. Scaling is done by pressing **D** to go down and **F** to go up. Interactive scaling within the viewport is achieved with the **Auto Scale** option, retaining the scale of the widget over the geometry as the viewport scale itself is modified.

Image Ink also allows you to adjust standard controls for **Scale**, **Rotate**, and **Position** as well as a **Soft Border** percentage and other options, by selecting the options found in the tool properties viewport. You can gain quick access to tool properties by pressing **K** on the keyboard to activate the properties window.

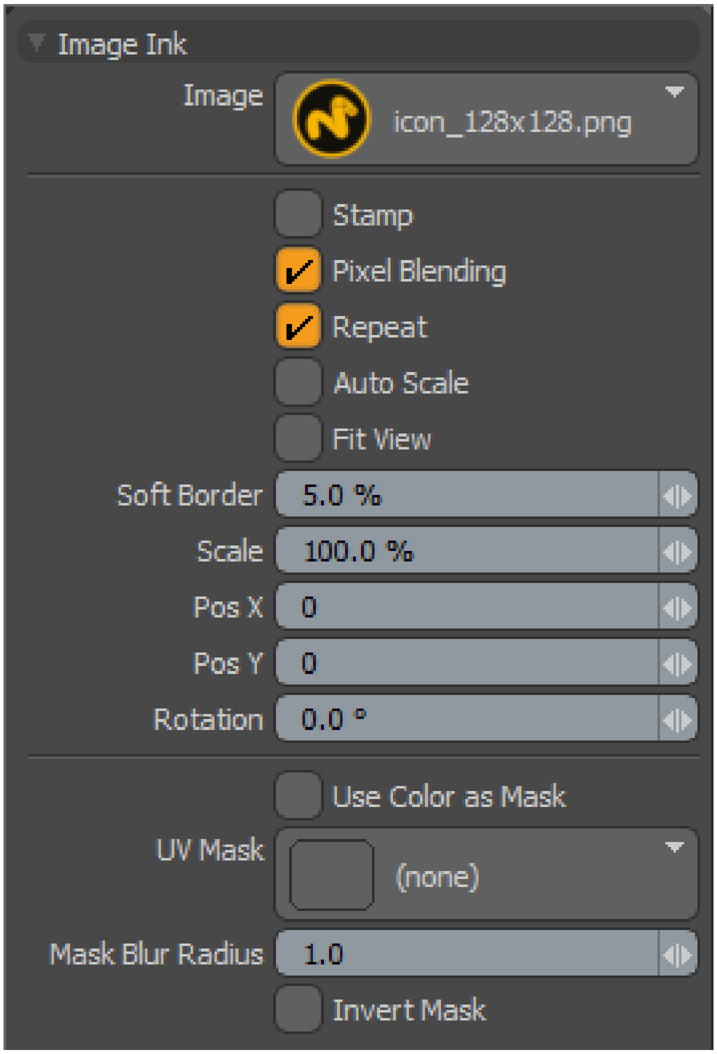


Image: icon_128x128.png

☐ Stamp

☑ Pixel Blending

☑ Repeat

☐ Auto Scale

☐ Fit View

Soft Border: 5.0 %

Scale: 100.0 %

Pos X: 0

Pos Y: 0

Rotation: 0.0 °

☐ Use Color as Mask

UV Mask: (none)

Mask Blur Radius: 1.0

☐ Invert Mask

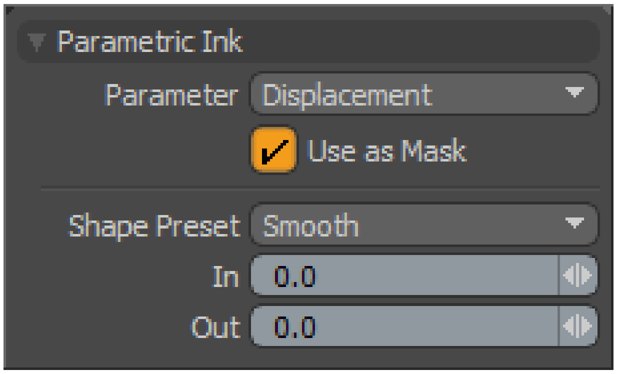
Option	Description
Image	Use the image loader to select and load images for use with the **Image Ink** function.
Stamp	When enabled, apply the entire selected **Image** in one quick stamp. This option requires the use of Projection Paint. This can be a very useful method for applying decal textures in very precise positions on a 3D mesh, as well as for stamping down texture elements such as scars. The stamp option can even be useful for quickly stamping down an image repeatedly to texture a model completely. By using some **Soft Border** and changing the rotation of the image, you can create very interesting texture effects quite quickly.
Pixel Blending	When enabled, smooth the image when it is scaled beyond its maximum native resolution, which can reduce visibly pixelated edged.

Option	Description
Repeat	When enabled, the image ink repeats continually once painting has extended beyond the image boundary.
Auto Scale	When enabled, scale the **Image Ink** frame relative to the viewport being scaled. Useful for applying a consistent texture across the surface of the model. Whenever you zoom in and out of the object, **Auto Scale** makes sure that relationship remains the same.
Fit View	When enabled, Modo uses the screen resolution for scaling the image. **Note:** When this option is enabled, but no image is selected, Modo uses the camera background image.
Soft Border	The **Soft Border** option is a percentage that creates an opacity ramp from the center of the **Image Ink** to the edges. As you paint through the image plane (Projection Mode), the **Soft Border** amount causes the edges to feather, creating a soft blend from the edge of the ink boundary. This is also very useful when working in **Stamp** mode. With a tiling image as your ink and a **Soft Border** value of approximately 50%, you can repeatedly stamp down a texture across the surface of the mesh to create very quick detail mapping.
Scale	Allows you to change the size of the image relative to the object in your viewport. Reducing the value results in the image appearing smaller relative to the model and increasing the value increases the relative scale of the image. When using **Projection Paint**, the scale handle is the square handle on the right-hand side of the image plane.
Pos X/Y	These two values represent the position of the center of the **Image Ink** frame within the viewport. Values are pixel coordinates, with 0,0 being the upper left corner. Each time the **Image Ink** function is activated, your image frame is centered within the viewport. You can also interactively position the image ink frame using the circle in the center.
Rotation	The **Rotate** value tilts the image. In the case of standard **Fast UV** painting, this is essentially rotating the image in UV space. With **Projection Painting** the image plane is rotated in screen space. The circle handle at the top of the image plane can be dragged right and left to adjust rotation interactively.

Option	Description
Use Color as Mask	When enabled, use the brightness (Luminosity) of the current **Image Ink** image as an alpha mask when painting onto the object. So instead of painting the image itself onto your object, the image becomes like a virtual silkscreen allowing the color of the brush to only go through the dark areas of the image.
UV Mask	Allows you to use a secondary image as a painting mask. The only requirement is that the masking image must use the same UV Map as the image being painted. For example, this allows you to bake an Occlusion map image and then use that as a mask to paint rust into the crevices of an old vehicle.
Mask Blur Radius	This option allows you to blur or soften the image applied as a UV Mask. The greater the value the softer the mask transition between light and dark.
Invert Mask	When enabled, invert the RGB or Gray values of the mask, reversing the application of the mask. Areas that were masked become paintable, and areas that were paintable become masked.

Parametric Ink

The **Parametric Ink** function provides a method to modulate the ink between the foreground (FG) and background (BG) values based on parameters including **Slope**, **Altitude**, **Incidence Angle**, **Pressure**, and **Tangent Pressure**. The first four parameters are attributes of the surface on which you are painting and the last two are based on tablet input device feedback.



Parametric Ink

Parameter Displacement

☑ Use as Mask

Shape Preset Smooth

In 0.0

Out 0.0

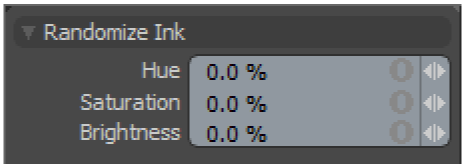
Option	Description
Parameter	The various parameter options allow you to choose how **Parametric Ink** modulates the painted brush stroke: • **Displacement** - Allows you to modulate between the foreground and background

Option	Description
	color based on the amount of displacement applied to a surface, providing you a means to paint in the cavities of surfaces. **Note:** When working with procedural texture displacement, you need to convert the procedural to an **Image Map**, using Modo's baking functions, to produce results with this option. • **Slope** - Changes the ink from the FG value to the BG value as the geometric normals change. If we look at a sphere in **World Coordinates**, the top of the sphere has normals that point directly upward, the bottom points directly down, and at the equator normals point straight out. These normals range from 0 to 90. • **Altitude** - Based on the bounding area of the mesh, the colors change using the FG color at the very bottom of the model and the BG color as you reach the top of the mesh. • **Incidence Angle** - This is the angle of the surface of the model as it relates to the viewport in which you are painting. Polygons that face the view shade using the FG color, where as polygon normals that are perpendicular to the view shade with the BG ink color. • **Pressure** - When painting with a tablet, 100% pressure uses the FG color for the ink. As you use less pressure, it fades to the BG color. • **Tangent Pressure** - This allows you to change between FG and BG colors by rotating the tablet stylus.
Use as Mask	When this option is enabled, a mask is generated and the **Parameter** control modulates the mask amount between masked and unmasked.
Shape Preset	Similar to the **Smooth Brush** option, **Shape Preset** allows you to fine tune the application of strokes when using the **Parametric Ink** function. You can select from the various presets, or adjust the **In** and **Out** values to further customize the brush shape.

Random Ink

The **Random Ink** option allows the paint tools ink color to randomly vary as you paint on the mesh. There are three input fields: **Hue**, **Saturation**, and **Brightness**. The default values of 0% result in no random variation to the ink. By increasing any of the values you can randomize the ink in one or more inputs. Setting any of these values to 100% causes the ink to randomly swing plus or minus 100% from its original value.

Setting **Brightness** and **Saturation** 10-30% can be very useful when creating noise maps to rough up a surface.



Randomize Ink

Hue 0.0 %

Saturation 0.0 %

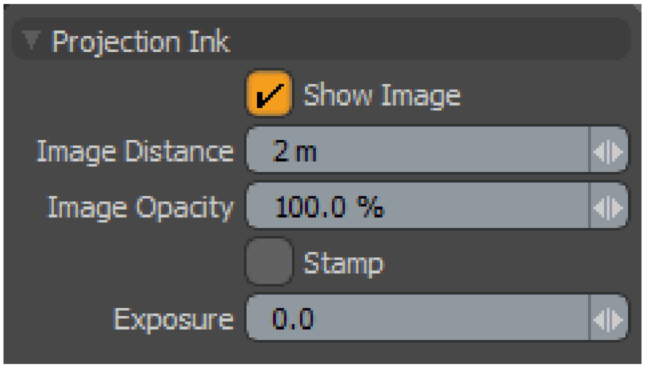
Brightness 0.0 %

Option	Description
Hue	Modifying this value results in variations of color for the individual paint dabs when stamped onto the surface. 0% effectively disables this setting, ramping toward 100% with the maximum amount of **Hue** variation in the stroke.
Saturation	Modifying this value results in variations of color intensity for the individual paint dabs when stamped onto the surface. 0% effectively disables this setting, ramping toward 100% with the maximum amount of **Saturation** variation in the stroke.
Brightness	Modifying this value results in variations of luminosity for the individual paint dabs when stamped onto the surface. 0% effectively disables this setting, ramping toward 100% with the maximum amount of **Brightness** variation in the stroke.

Projection Ink

The **Projection Ink** lets you paint using a camera's backdrop image and projection. If you have a camera and a backdrop image set up, you can paint textures from the image into the scene as if they were projected from the camera, similarly to the Projection Texture.

Note: For more information on the Projection Texture, see Projection Texture.



Projection Ink

☑ Show Image

Image Distance 2 m

Image Opacity 100.0 %

☐ Stamp

Exposure 0.0

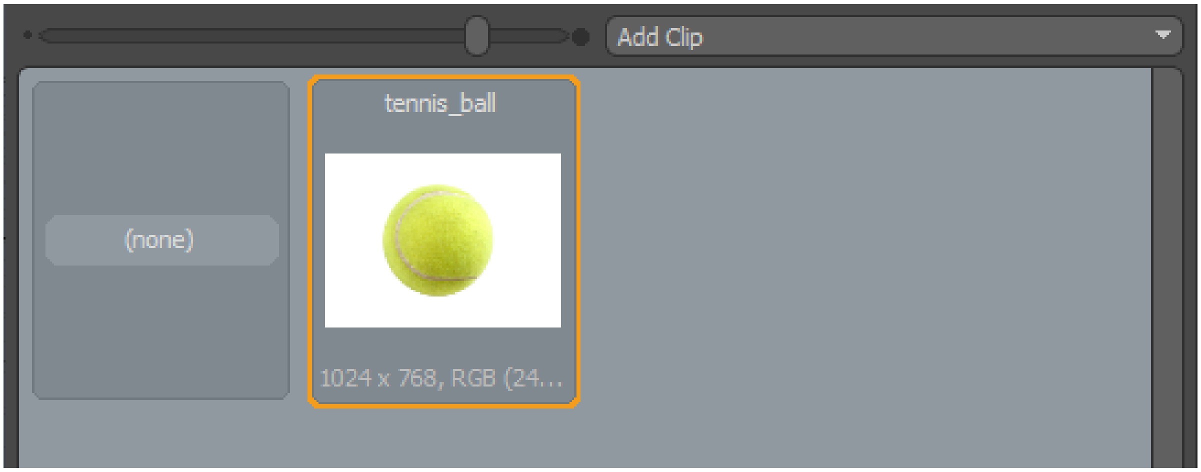
Using the Projection Ink

1. Set up the required number of cameras matched to background images.

> **Note:** For information on camera matching, see Camera Matching.

2. Go to the **Paint** layout and select your mesh.
3. In the **Item List**, select the camera whose background image and projection you want to paint with.
4. If no background image is assigned yet, go to the camera's **Properties** panel, and in the **Background Image** section, click the **Background Image** field.

   The **Clip Browser** window opens.



5. Select the desired projection image by clicking the target image. If it's not already loaded, click the **Add Clip** button, and click **(load image)**.
6. Pick the **Airbrush** or **Paintbrush** tools.
7. In the **Paint Tools** tab, click and hold **Image Ink** to open a dropdown, and select **Projection Ink**, then start painting on your mesh.

To paint with a different camera's image, just select that camera.

> **Note:** When you select a new camera, you need to select the brush again. If you have multiple cameras, it may be useful to use the Pick Walking function in Modo, found in the **Properties** tab's **Assembly** sub-tab. For more information on using this function, see Assembly Tab.

You can set the following properties for **Projection Ink**:

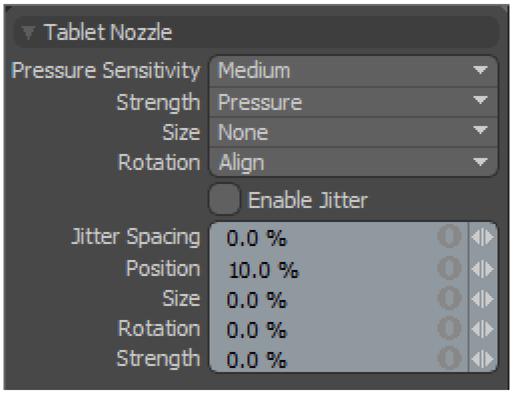
Option	Description
Show Image	When enabled, the projected image is displayed in the viewport at the specified size and distance from the camera with the specified opacity.
Image Distance	Set the distance of the image from the camera.
Image Opacity	Set the transparency of the projected image.
Stamp	Fill the visible selected polygons with the projected image.
Exposure	Adjust to compensate for differences in lighting between different regions and/or images.

Tablet Nozzles

If you had a virtual microscope in Modo that would allow you to look at how brush strokes are created, they aren't actually continuous lines, but a series of tiny dabs sprayed across the canvas. The **Tablet Nozzle** option gives you greater control over these dabs, as well as over a tools pressure sensitivity when using a tablet as an input device.

Note: Make sure to set the input preference to **Tablet** if you wish to work this way. The setting can be found in the menu bar under **System** > **Preferences** in the **Input** > **Remapping** > **Input Device** option field.

The **Tablet Nozzle** also provides options to vary the positions and scales of the paint dabs when stamped down, providing some randomization to that spray of dabs from your brush. There are five parameters: **Spacing**, **Position**, **Size**, **Rotation**, and **Strength**.



Tablet Nozzle

Pressure Sensitivity Medium

Strength Pressure

Size None

Rotation Align

☐ Enable Jitter

Jitter Spacing 0.0 %

Position 10.0 %

Size 0.0 %

Rotation 0.0 %

Strength 0.0 %

Tablet Nozzle

Pressure Sensitivity - This options controls the brushes sensitivity to changes in tablet pressure when painting strokes on the canvas, the **Low** option is less sensitive to variation of pressure, while **Medium** and **High** are increasingly more sensitive.

Strength - The **Strength** option defines what action controls the strength (and therefore density) of the stroke:

- **None** - disables strength modulation.
- **Pressure** - controls the density of the stroke based on the pens pressure input value.
- **Speed** - controls the density of the stroke based on how fast or slow the cursor moves over the surface.
- **Both** - combines the pressure sensitivity of the tables input with the speed of the cursor to control the final strokes strength.

Size - The **Size** option defines what action controls the scaling of the stroke dabs providing you the ability to vary strokes width:

- **None** - disables scaling of the tip.
- **Pressure** - controls the scaling of the stroke based on the pens pressure input value.
- **Speed** - controls the scaling of the stroke based on how fast or slow the cursor moves over the surface.
- **Both** - option combines the pressure sensitivity of the tables input with the speed of the cursor to control the final strokes size.

Rotation - The **Rotation** option controls the facing direction of paint dabs when placed on the canvas:

- **None** - disables rotation of the dabs, aligning them vertically to the viewport when brushed to the canvas.
- **Align** - aligns the dab to the vector direction the stroke is traveling, meaning each dab is rotated to flow along the direction the brush stroke is moving.

Enable Jitter - When this option is enabled, you can adjust the values for the various options to control randomization in the brush dabs.

Jitter Spacing - Modo brush dabs spray with tiny but consistent spaces between each dab. This value randomizes that spacing. Higher values cause the dabs to clump together leaving noticeable gaps, while smaller values vary the positions only slightly.

Position - This value randomize the placement of the dabs away from the brushstroke. Higher values cause the dabs to splatter around the stroke, while smaller values vary the positions only slightly creating uneven natural-looking strokes.

Size - This value randomizes the size of each dab as it is placed on the canvas. Higher values create blobby clumped strokes while smaller values reduce the clumping, making strokes with randomly varying thickness.

Rotation - This value randomly rotates each dab. Higher values randomize the rotation of each dab to a higher degree, and obviously lower values vary the rotation in smaller increments. While not too useful for the perfectly round dabs of the smooth and hard brushes, when used with image maps and the preset brush, one can quickly add an overall texture to a surface with only one image, avoiding obvious visual repetition.

Strength - This value randomizes the strength of each dab, essentially varying the pressure applied to each one. This results in different degrees of opacity for each dab. Higher values vary the pressure to a higher degree, while lower values vary the pressure in smaller amounts.

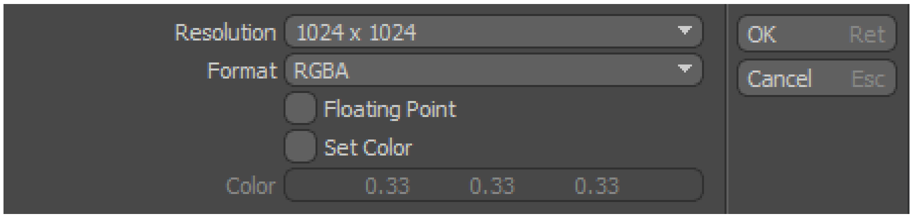
Try out different combinations and you can create dramatic variations of simple brushes with just a few settings easily creating special effect brushes that complement the many already present in Modo.

Adding Images

Bitmap images are an essential component of texturing, painting and image-based sculpting. To simplify this common requirement, saving you a trip to an image editing application, Modo allows you to create *blank* images directly in Modo, easily and quickly for a variety of applications using the **New Image** command found in all image browsing palettes and in the image selector drop downs of many panel options. Additionally, you can use the **New Image** command in the menu bar under **File** > **New Image** menu.

Once an image is created, it must be properly placed in the **Shader Tree**, so that it is applied to the surface you wish to affect. It also may be necessary to set its layer **Effect** to the proper option, such as **Displacement** for image-based displacement. For more information on working with layers in the **Shader Tree** and assigning their **Effect**, see the Shader Tree topic. Modo also has a variety of utilities to make it easier to add specific types of new images to the **Shader Tree**, which are covered in Add Image Sequence.

When you click **New Image**, you are presented with an OS-specific dialog window where you can specify the saved file's name, location and format. Once that requirement is filled, the following Modo panel opens up to designate specifics of the image itself.


Resolution 1024 x 1024
Format RGBA
☐ Floating Point
☐ Set Color
Color 0.33 0.33 0.33

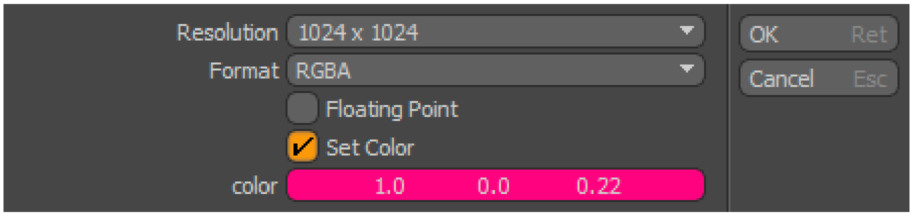
- **Resolution**: When painting in 3D, the images are displayed via OpenGL. Performance of GL texture rendering is superior when working with square images. As a result the choices for resolution are based on standard GL texture sizes (64x64, 128x128, 256x256, 512x512, 1024x1024, 2048x2048, 4096x4096 and 8192x8192)

- **Format**: The created image can be gray, RGB color or RGBA color with an alpha channel. Gray images are typically smaller in file size and are very useful for textures that are used for displacement or bump maps. Typically, the best choice for color textures is RGBA so that the new image texture does not completely obscure the materials and textures layered beneath it. RGB images are opaque.
- **Floating Point**: Also known as HDRI (High Dynamic Range Images), floating point images give a much broader range of potential color values. Whereas an integer (default) image defines colors in 8 bits per channel (0 to 255) a floating point image can use values many times larger. This can be incredibly useful when using images as light sources or to drive finely detailed displacement maps.
- **Set Color**: This toggle defaults to off so that new images are completely "blank" meaning they have no color or alpha pixels. Turning this option on creates an image with a opaque alpha so all pixels are visible and a base color for the image. The color is set by the color component under this toggle option.
- **Color**: The color swatch is only applicable when the Set Color toggle is active. This color value is used to fill the new image.

Tip: To save an image as 16bit, you must enable the **Floating Point** option or images default to 8-bit regardless of the chosen image format.

Add Image Sequence

You can add image sequences as easily as adding still images. Sequences are simply a series of still frames stored as consecutively numbered image files. These can be useful for baking out animated textures, or painting animated sequences. When applied to a surface, the sequence frame number is equal to the rendered frame number. When frames go missing, the last plausible frame is repeated until another frame is found. To create a new sequence you can use the menu bar command **File** > **New Image Sequence** or use the **New Sequence** option in the **Clips** browser. When invoked, an OS-specific dialog opens where you can define the name, location, and format of the files. Then click **OK** for the Modo dialog where you define the resolution and other aspects of the frames themselves.


Resolution 1024 x 1024
Format RGBA
☐ Floating Point
☑ Set Color
color 1.0 0.0 0.22

- **Resolution**: When painting in 3D, the images are displayed via OpenGL. Performance of GL texture rendering is superior when working with square images. As a result the choices for resolution are based on standard GL texture sizes (64x64, 128x128, 256x256, 512x512, 1024x1024, 2048x2048, 4096x4096 and 8192x8192)

- **Format**: The created image can be gray, RGB color or RGBA color with an alpha channel. Gray images are typically smaller in file size and are very useful for textures used for displacement or bump maps. Typically, the best choice for color textures is RGBA so that the new image texture does not completely obscure the materials and textures layered beneath it. RGB images are opaque.
- **Floating Point**: Also known as HDRI (High Dynamic Range Images), floating point images give a much broader range of potential color values. Where an integer (default) image defines colors in 8 bits per channel (0 to 255), a floating point image can use values many times larger. This can be incredibly useful when using images as light sources or to drive finely detailed displacement maps.
- **Set Color**: This toggle is disabled by default, so that new images are completely blank, meaning they have no color or alpha pixels. Turning this option on creates an image with a opaque alpha so all pixels are visible and a base color for the image. The color is set by the color component under this toggle option.
- **Color**: The color swatch is only applicable when the **Set Color** toggle is active. This color value is used to fill the new image.

Saving Images

Once you have modified your images or sequences in Modo in any way, through baking, sculpting or painting, the images themselves need to be saved. This can be done several ways. Typically, you can select a particular image in the **Images** viewport, right-click to open the context menu and then select the **Save** option. You can also select the particular image they wish to save and use the menu bar commands **File** > **Save Image/Save Sequence**. If you want to simply save all changes, use the menu bar command **File** > **Save All Images**. When closing a Modo session, you are prompted by a dialog to save any unsaved images if not saved before closing the application.

Add Texture Commands

The **Add Texture** commands appear in the main **Paint** interface tab, under the **Utilities** sub-tab of the toolbox, with some additional options available under the **Hair Tools** sub-tab. You can use these as shortcuts to quickly add the appropriate image type to the **Shader Tree** for the most common image mapping options.

Add Color Texture

This command automatically creates a new blank image map texture layer in the **Shader Tree**, set as **Diffuse Color** for the **Effect**. This is the necessary option for painting a surface with the paint tools. The image is added to whatever Material Group is currently selected in the **Shader Tree**, regardless of the current foreground item selections. To ensure proper **Shader Tree** placement, you should first select the Material

Group Item where you want the texture layer to reside. It may also be necessary to adjust the projection method in the Texture Locator. The image format defaults to whatever is specified by the **Default Image Format** defined by your **Preferences**, but can be changed in the **New Image** dialog window that opens when the command is invoked.

Add Displacement Texture

This command automatically creates a new blank OpenEXR format image map texture layer in the **Shader Tree**, set as **Vector Displacement** for the **Effect**. This is the most flexible option for image based sculpting with the Sculpting tools. The image is added to whatever Material Group is currently selected in the Shader Tree, regardless of the current foreground item selections. To ensure proper **Shader Tree** placement, first select the Material Group Item where you want the texture layer to reside. It may also be necessary to adjust the projection method in the Texture Locator. For best results, accept the default values of the **New Still** pop-up window, only adjusting the resolution as necessary.

Add Mask Texture

This command automatically creates a new blank image map texture layer in the **Shader Tree**, set as **Layer Mask** for the **Effect**. Mask textures are useful in limiting the effect of another texture layer has on a shaded surface. The image is added to whatever Material Group is currently selected in the **Shader Tree**, regardless of the current foreground item selections. To ensure proper **Shader Tree** placement, you should first select the Material Group Item where they want the texture layer to reside. This layer's position in the **Shader Tree** is important, as a layer mask only masks the texture layer directly above it in the tree. It may also be necessary to adjust the projection method in the Texture Locator. The image format defaults to whatever is specified by the **Default Image Format** defined by your **Preferences**, but can be changed in the **New Image** dialog that opens when the command is invoked.

Add Fur Texture

There are four Add Fur commands that automatically create new, blank OpenEXR format image map texture layers in the **Shader Tree**. The Add Fur Texture option, once added to the **Shader Tree**, is set as **Fur Vector** for the **Effect**. The **Fur Direction** and **Fur Length** options are set to **Fur Direction** and **Fur Length** as their **Effect**, respectively. These options provide the most flexible method of styling fur material layers using the styling tools. The images are added to whatever Material Group is currently selected in the **Shader Tree**, regardless of the current foreground item selections. To ensure proper **Shader Tree** placement, first select the Material Group Item where you want the texture layer to reside, such as the group with the fur material item. It may also be necessary to adjust the projection method in the Texture Locator. For best results, accept the default values of the **New Still** pop-up window, only adjusting the resolution as necessary.

Image Adjustment Commands

The image adjustment commands are located in the menu bar under the **Texture** > **Image Processing** menu. If the commands are inactive, this indicates that no image is currently selected. Use the **Shader Tree** or **Images** viewport tab (or **Clip List**) to select an image and enable the commands for use.

Invert Image

This command inverts the RGB color values of the currently selected image map producing a negative of the original.

Clear Image

This option deletes all the pixels in a given image making it effectively blank, while retaining the image size, resolution and filename.

Expand UV Seams

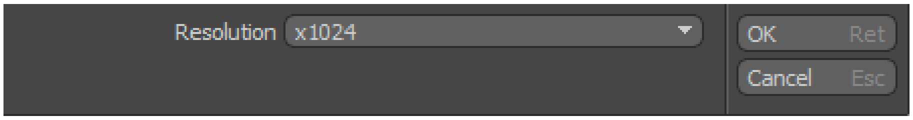
This command analyzes the currently selected image map and the UV map to which it is associated and expands the pixels outward beyond the UV map boundary. This can be very useful for repairing texture boundaries when aggressive texture antialiasing or displacements can cause seams to become visible. The amount of expansion is based on the Bake UV Border Size preference setting. A recent render with a corresponding Alpha channel is required for Modo to know which edge seams to expand.

Convert to Vector Displacement

This command converts a grayscale height-based displacement map into a vector displacement image. This is useful when importing sculpt data from other applications, as they are only capable of generating grayscale height-fields. To take advantage of Modo's vector displacement capability, either sculpt on a separate vector displacement image layer or convert the height-field to vector displacement using this command.

Set Resolution

This command changes the currently selected image to a custom resolution. This is very useful if you are working on a texture or sculpt and decide you need more detail, as you can increase the image resolution with this command. The command is based on power of 2 images (64, 128, 256, 512, 1024, 2048, 4096) as most OpenGL hardware is optimized for square aspect image maps.



Resolution x1024

Flatten Layers

This command works on a group of selected image texture layers in the Shader Tree. All images in use must be of the same type and resolution. The command flattens the layers into a single image texture layer using a single image map that is the composite of the original images selected. Before running the command, select the layers to flatten by holding **Ctrl** and left-clicking each layer in the Shader Tree you wish to combine.

Using Hair Tools

Much like the sculpt and painting tools, Modo offers a variety of tools for styling fur and its associated hair guides. Located in the **Paint** layout, in the **Hair Tools** sub-tab of the toolbox, giving you everything needed for combing and sculpting fur into the proper do. The tools all work double duty styling both guides, and when the **Edit Fur Map** option is enabled in the tool's properties, image maps can be used to control the styling of a regular fur layer as well.

Styling Tools

Hair Styling		
	Blow	The **Blow** styling brush pushes the fur away from the brush, like a hair dryer would.

Hair Styling		
	Comb	The **Comb** brush works much like a virtual comb, dragging fibers in the direction of the cursor.
	Curl	The **Curl** brush adds swirls to fibers rotating them based on the size of the brush tip.
	Stretch	The **Stretch** brush scales the individual fibers, reducing their overall length.
	Smooth	The **Smooth** brush straightens and smooths bent and kinked fibers.
	Move	The **Move** brush grabs the fibers like a magnet and moves them in the direction of the mouse pointers movement.
	Separate	The **Separate** brush moves adjoining fibers or guides apart from each other with a falloff based on the brush shape and strength.
	Root Puff	The **Root Puff** brush rotates the root of the fibers or guides toward the direction of the surface normal.

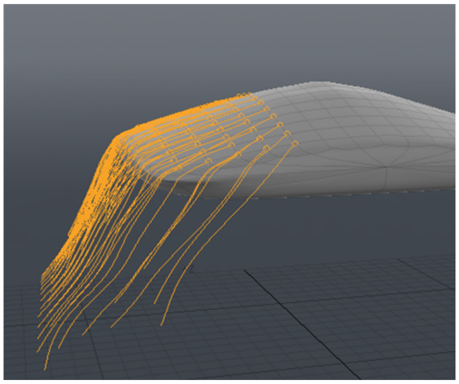
Brushes

Select a brush style to use with the styling tools. For a full description of each style, see Customizing Paint Tools.

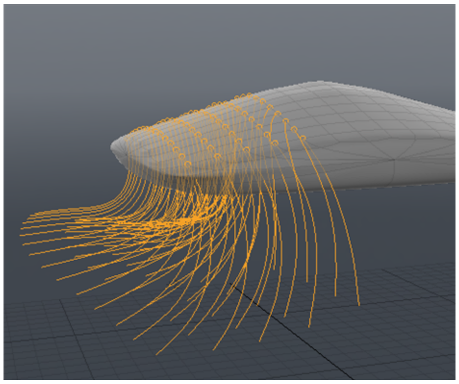
Toggles

The panel provides the following controls that affect hair styling:

- **Collide with Base Surface** - Check this if you want the hair to collide with the surface (from which the hair originates) when styling the hair. This ensures the hair remains on the surface and does not pass through it.



Hair guides combed down with **Collide with Base Surface** on.
Note how the guides flow over the surface.



Hair guides combed down with **Collide with Base Surface** off.
Note how the guides pass through the surface

- **Deferred Fur Update** - When editing guides using tools in the **Guides** section, check this to only update the preview fibers when you drop the tool, not during editing. If unchecked the preview fibers update with the guides. By checking this you can focus on adjusting the guides accurately without extraneous detail.

Note: For more information on customizing the preview fibers, see GL Display.

Guides

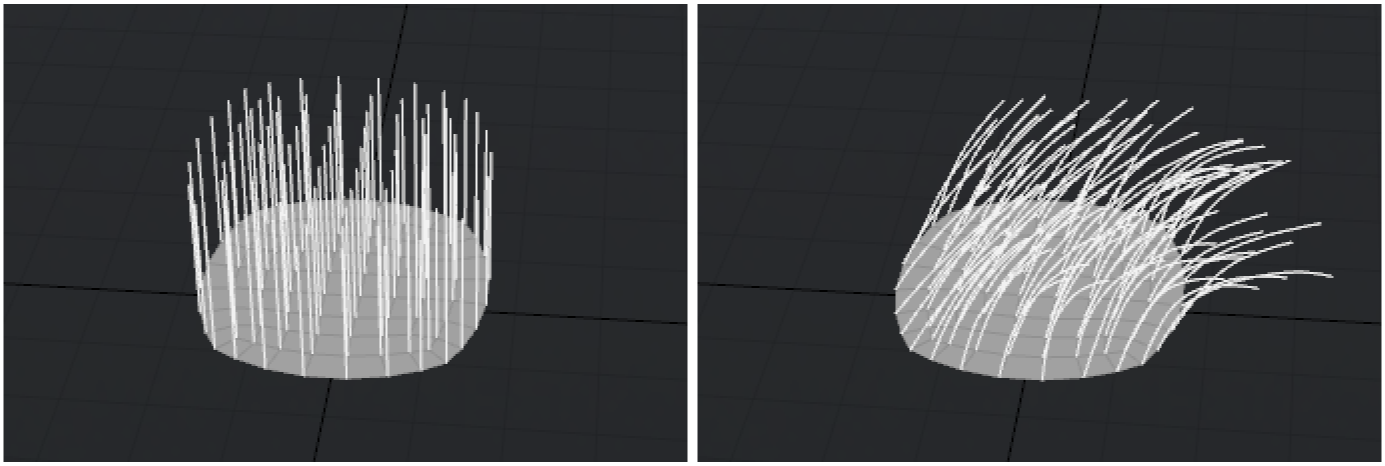
For a description of how to use these tools, see Hair Guides.

Fur Image Maps

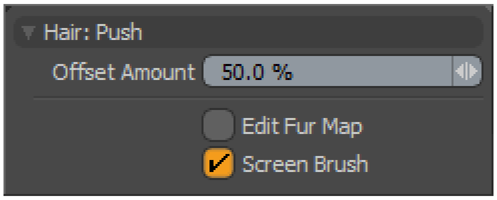
See Controlling Fibers with Vector Maps for more information.

Blow

The **Blow** styling brush pushes the fur away from the brush, like a hair dryer would. The direction of the blowing action moves away from your viewpoint into the current viewport. The Fur brush works with the Customizing Paint Tools to further customize how the Blow brush styles the fur. The **Blow** tool can be found in the **Hair Tools** sub-tab of the **Paint** layout's toolbox. When selected, the following attributes display in the **Properties** panel.



On the left, a group of guides, on the right the **Blow** tool applied



Hair: Push

Offset Amount 50.0 %

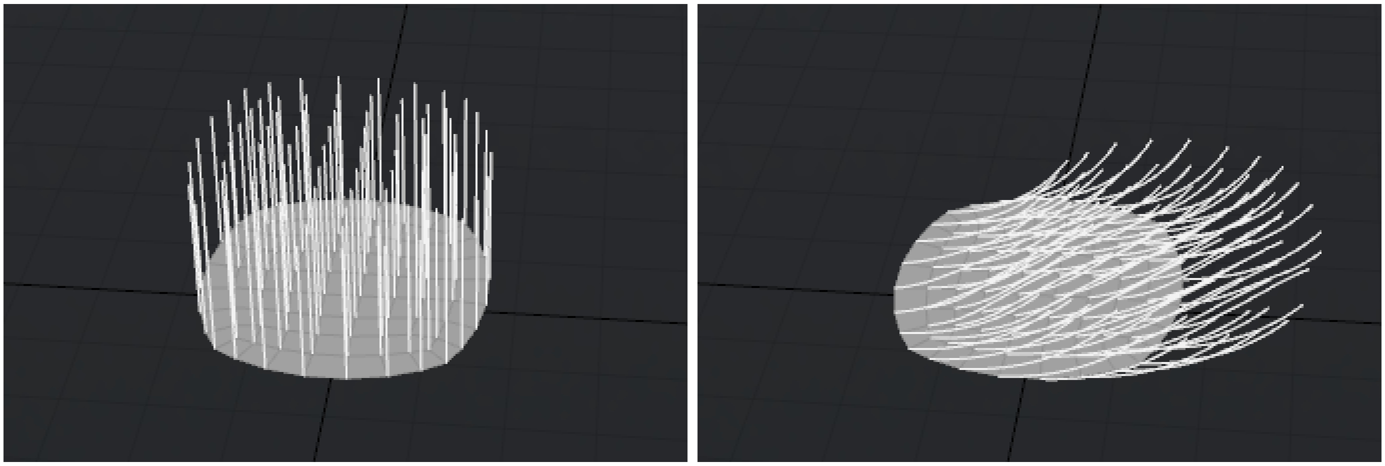
☐ Edit Fur Map

☑ Screen Brush

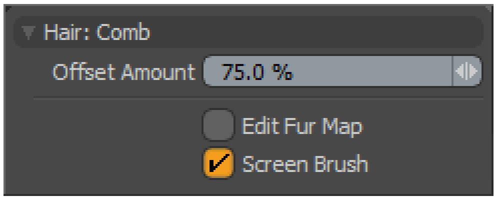
Option	Description
Offset Amount	The **Offset Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Edit Fur Map	You can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to **Fur Vector**. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, you need to enable the **Edit Fur Map** option.
Screen Brush	The **Screen Brush** option, enabled by default, affects all vertices under the brush tips circle, regardless of their distance. When the option is disabled, only vertices that are within the 3D sphere of influence around the brushes tip are affected.

Comb

The **Comb** brush works much like a virtual comb, dragging fibers in the direction of the cursor. Similar to the Blow tool, but the comb brush exerts its greatest influence toward the roots of the fibers. The Fur brush can be combined with the Customizing Paint Tools to further customize how the brush styles the fur. The **Comb** tool can be found in the **Hair Tools** sub-tab of the toolbox, found in the **Paint** tab of the interface. When selected, the following attributes display in the **Properties** panel.



On the left, a group of guides, on the right the **Comb** tool applied.



Hair: Comb

Offset Amount 75.0 %

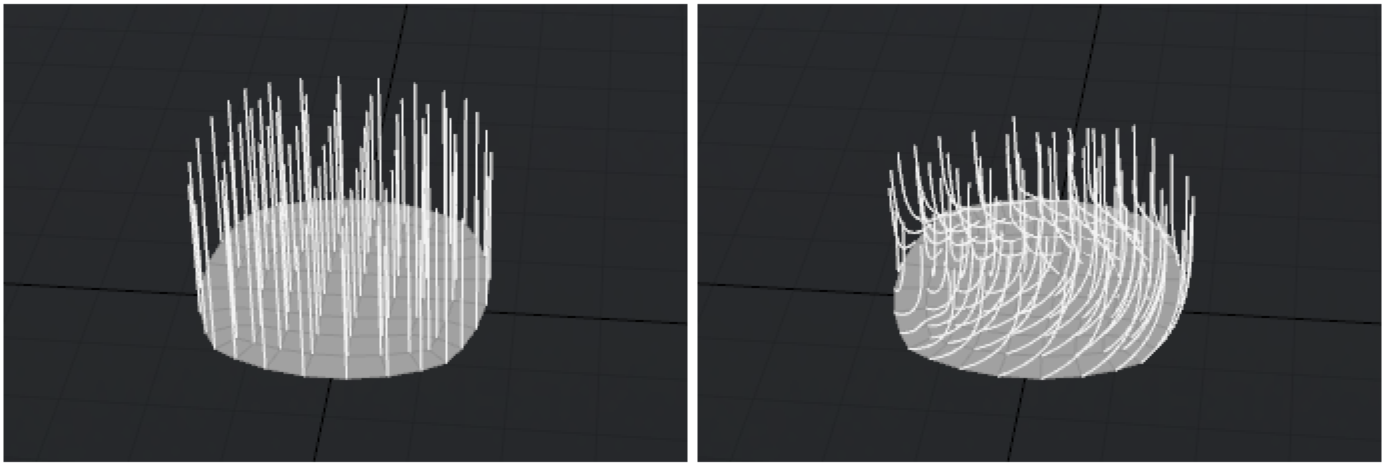
☐ Edit Fur Map

☑ Screen Brush

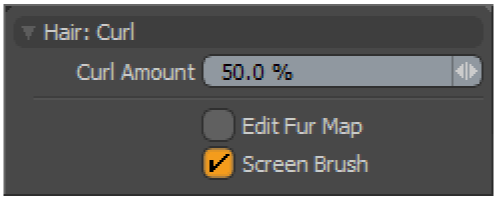
Option	Description
Offset Amount	The **Offset Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Edit Fur Map	You can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to **Fur Vector**. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, you need to enable the **Edit Fur Map** option.
Screen Brush	The **Screen Brush** option, enabled by default, affects all vertices under the brush tips circle, regardless of their distance. When the option is disabled, only vertices that are within the 3D sphere of influence around the brushes tip are affected.

Curl

The **Curl** brush adds swirls to fibers rotating them based on the size of the brush tip. The effect if similar to theSpin sculpting brush with the curl effect projecting straight out from the brush tip through the viewport, so you can easily control its application by rotating the guides orientation within the viewport. The Fur brush can be combined with the Customizing Paint Tools to further customize how the brush styles the fur. The **Curl** tool can be found in the **Hair Tools** sub-tab of the toolbox, found in the **Paint** tab of the interface. When selected, the following attributes display in the **Properties** panel.



On the left, a group of guides, on the right the **Curl** tool applied.



Hair: Curl

Curl Amount 50.0 %

☐ Edit Fur Map

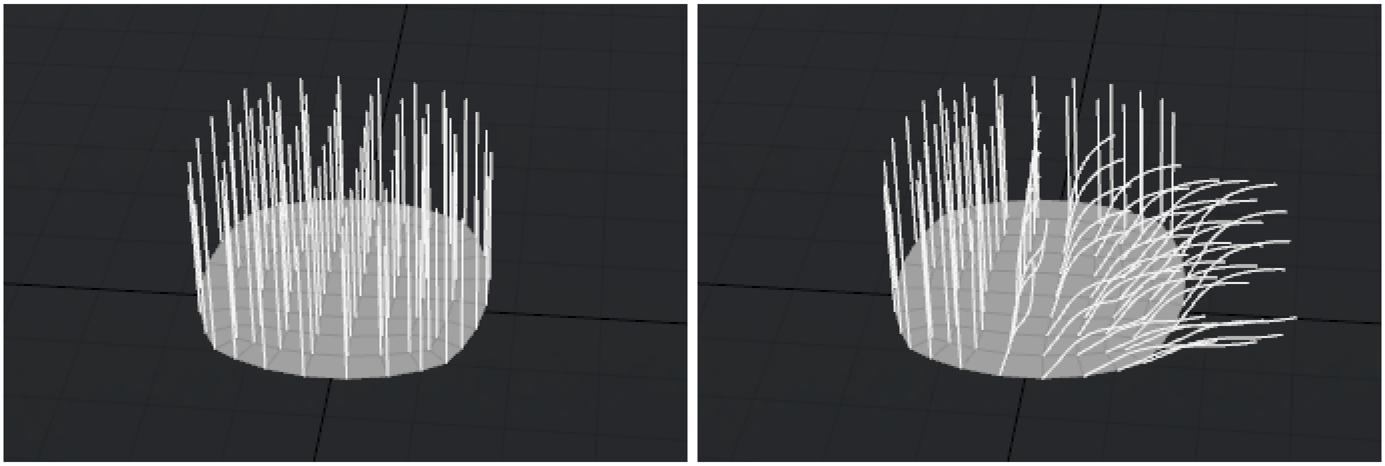
☑ Screen Brush

Option	Description
Curl Amount	The **Curl Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Edit Fur Map	You can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to **Fur Vector**. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, you need to enable the **Edit Fur**

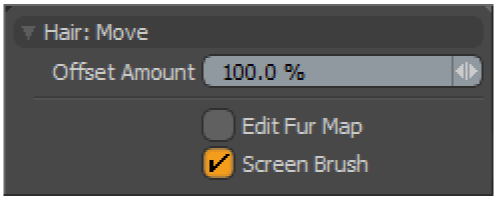
Option	Description
	Map option.
Screen Brush	The **Screen Brush** option, enabled by default, affects all vertices under the brush tips circle, regardless of their distance. When the option is disabled, only vertices that are within the 3D sphere of influence around the brushes tip are affected.

Move

The **Move** brush grabs the fibers like a magnet and moves them in the direction of the mouse pointers movement. The tool exerts its greatest influence over the tips of the fibers with an overall gradual falloff based on the brush's size and shape. The Fur brush can be combined with the Customizing Paint Tools to further customize how the brush styles the fur. The **Move** tool can be found in the **Hair Tools** sub-tab of the toolbox, found in the **Paint** tab of the interface. When selected, the following attributes display in the **Properties** panel.



On the left, a group of guides, on the right the **Move** tool applied.



Hair: Move

Offset Amount 100.0 %

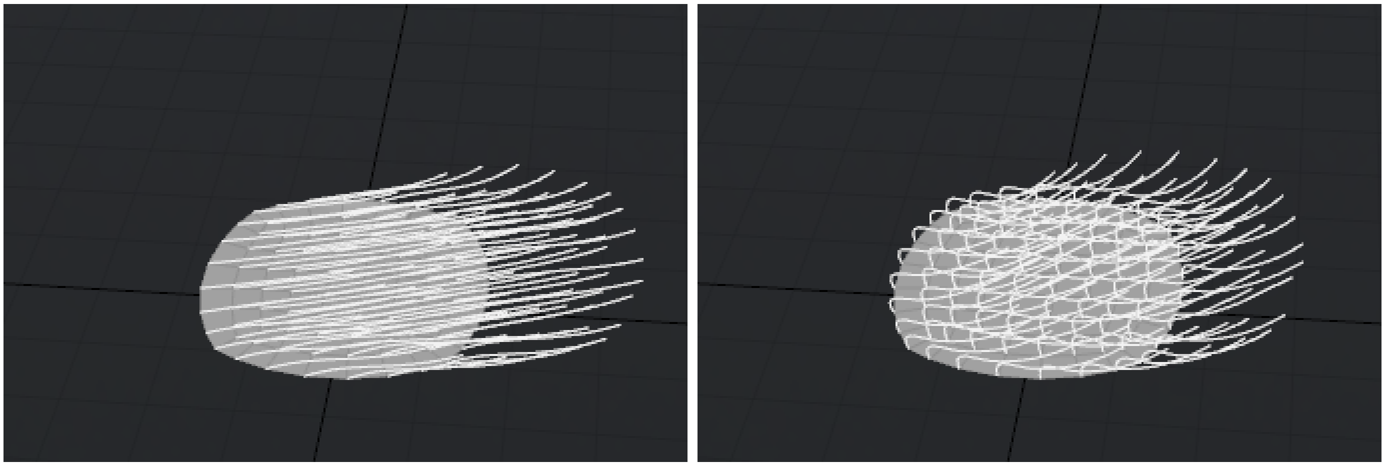
☐ Edit Fur Map

☑ Screen Brush

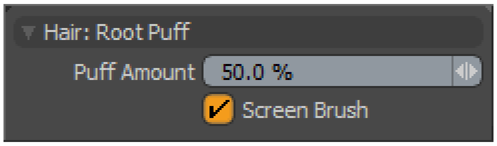
Option	Description
Offset Amount	The **Offset Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Edit Fur Map	You can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to **Fur Vector**. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, you need to enable the **Edit Fur Map** option.
Screen Brush	The **Screen Brush** option, enabled by default, affects all vertices under the brush tips circle, regardless of their distance. When the option is disabled, only vertices that are within the 3D sphere of influence around the brushes tip are affected.

Root Puff

The **Root Puf** brush rotates the root of the fibers or guides toward the direction of the surface normal. Especially useful when fibers have been combed to the point where the roots are rotated parallel to the surface. The Fur brush can be combined with the Customizing Paint Tools to further customize how the brush styles the fur. The **Root Puff** tool can be found in the **Hair Tools** sub-tab of the toolbox, found in the **Paint** tab of the interface. When selected, the following attributes display in the **Properties** panel.



On the left, a group of combed guides, on the right the **Root Puff** tool applied.



Hair: Root Puff

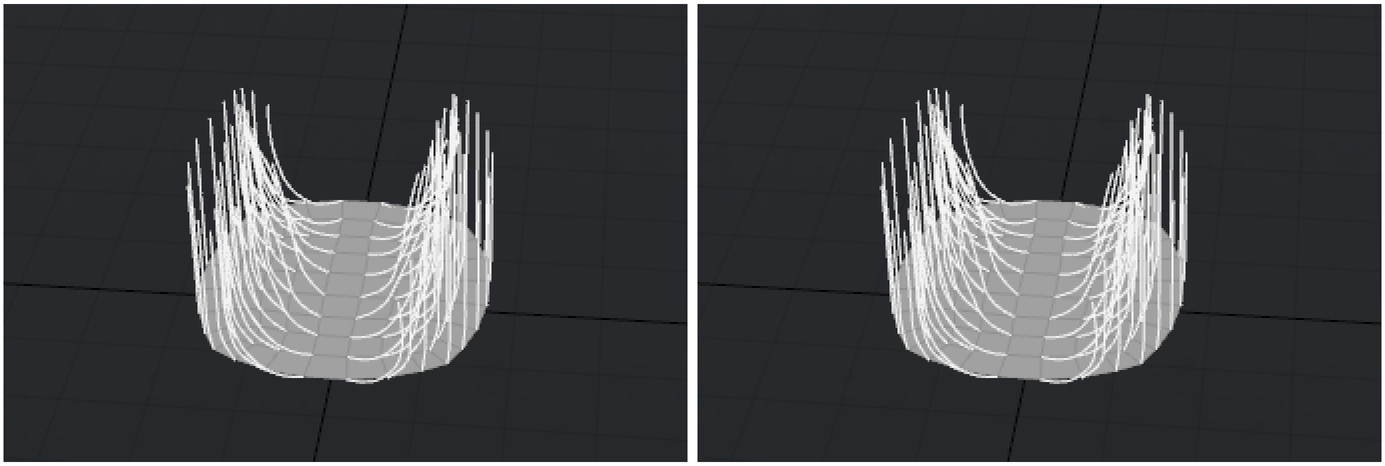
Puff Amount 50.0 %

☑ Screen Brush

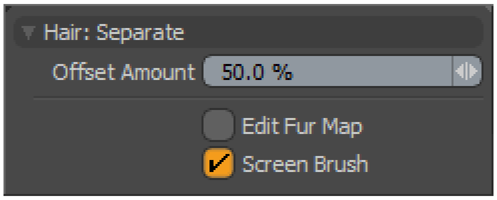
Option	Description
Puff Amount	The **Puff Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Screen Brush	The **Screen Brush** option, enabled by default, affects all vertices under the brush tips circle, regardless of their distance. When the option is disabled, only vertices that are within the 3D sphere of influence around the brushes tip are affected.

Separate

The **Separate** brush moves adjoining fibers or guides apart from each other with a falloff based on the brush shape and strength. By holding down the **Crtl** key, you can reverse the separate function and gather the fibers closer together. The Fur brush can be combined with the Customizing Paint Tools to further customize how the brush styles the fur. The **Separate** tool can be found in the **Hair Tools** sub-tab of the toolbox, found in the **Paint** tab of the interface. When selected, the following attributes display in the **Properties** panel.



On the left, guides are separated by the brush, on the right the guides are gathered.



Hair: Separate

Offset Amount 50.0 %

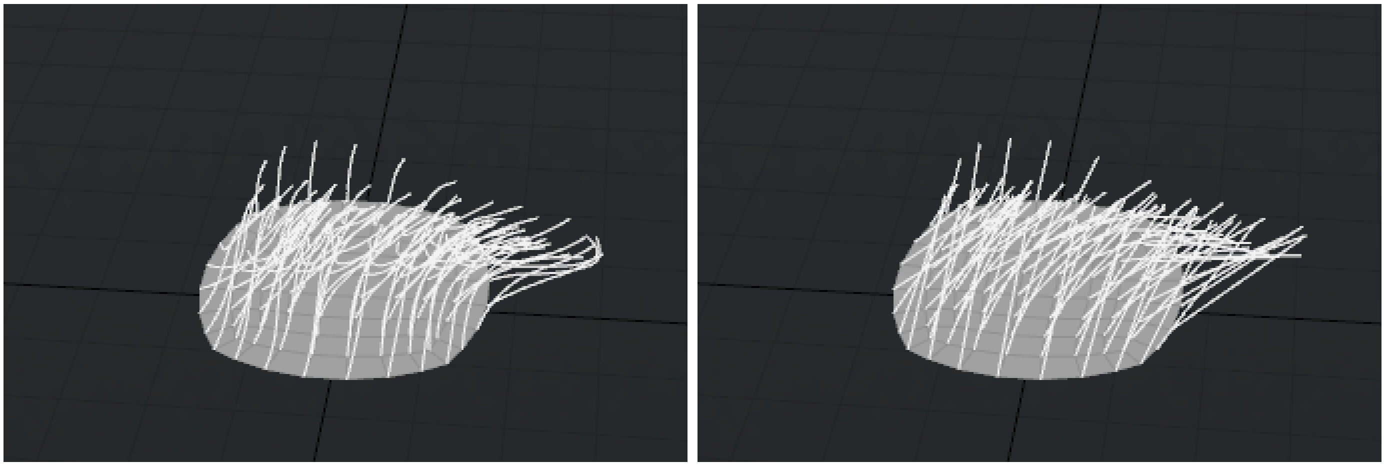
☐ Edit Fur Map

☑ Screen Brush

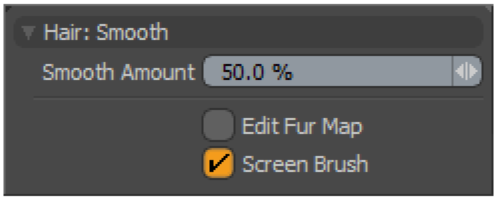
Option	Description
Offset Amount	The **Offset Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Edit Fur Map	You can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to **Fur Vector**. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, you need to enable the **Edit Fur Map** option.

Smooth

The **Smooth** brush straightens and smooths bent and kinked fibers. The effect is similar to aligning all points into a straight line between the root and tip of each individual fiber. The Fur brush can be combined with the Customizing Paint Tools to further customize how the brush styles the fur. The **Smooth** tool can be found in the **Hair Tools** sub-tab of the toolbox, found in the **Paint** tab of the interface. When selected, the following attributes display in the **Properties** panel.



On the left, a group of combed guides, on the right the **Smooth** tool applied.



Hair: Smooth

Smooth Amount 50.0 %

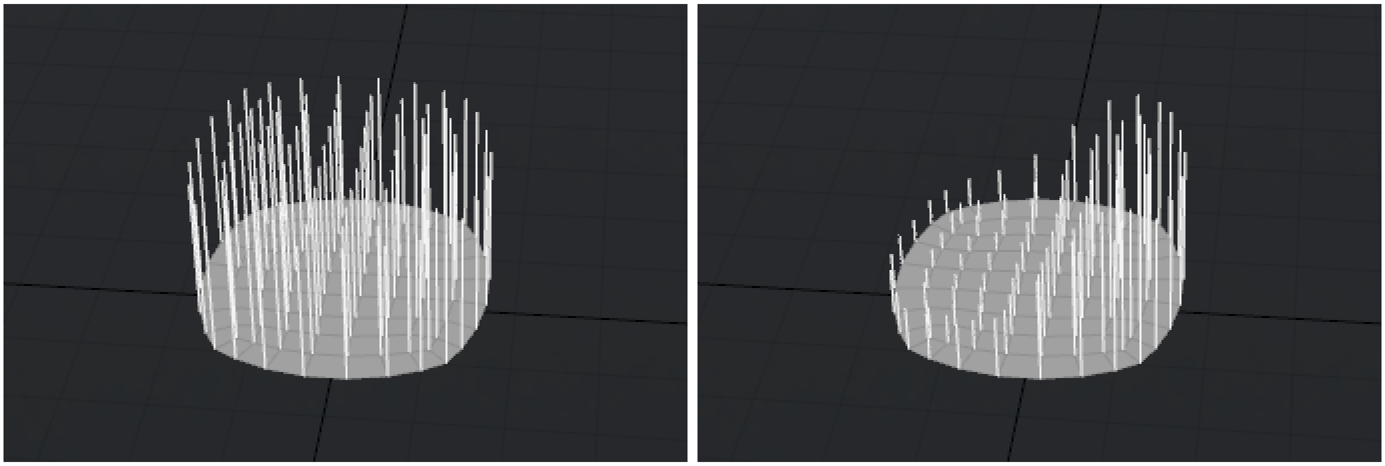
☐ Edit Fur Map

☑ Screen Brush

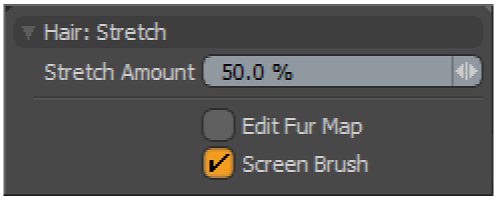
Option	Description
Smooth Amount	The **Smooth Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Edit Fur Map	You can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to **Fur Vector**. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, you need to enable the **Edit Fur Map** option.
Screen Brush	The **Screen Brush** option, enabled by default, affects all vertices under the brush tips circle, regardless of their distance. When the option is disabled, only vertices that are within the 3D sphere of influence around the brushes tip are affected.

Stretch

The **Stretch** brush scales the individual fibers, reducing their overall length. By holding down the **Ctrl** key, you can increase a fibers size growing the overall length. The Fur brush can be combined with the Customizing Paint Tools to further customize how the brush styles the fur. The **Stretch** tool can be found in the **Hair Tools** sub-tab of the toolbox, found in the **Paint** tab of the interface. When selected, the following attributes display in the **Properties** panel.



On the left, a group of guides, on the right the **Stretch** tool applied.



Hair: Stretch

Stretch Amount 50.0 %

☐ Edit Fur Map

☑ Screen Brush

Option	Description
Stretch Amount	The **Stretch Amount** value determines the strength of the brush stroke. Higher values have a greater influence over the fibers than lower values.
Edit Fur Map	You can style fur by way of a Fur Vector Map, providing a means to control combing direction and length with an image map, producing very detailed results. All that is necessary is a Fur Layer in the Shader Tree and a Floating Point image (such as EXR) with the layer effect set to **Fur Vector**. Generally, the fur styling tools are meant to edit Guides, in order to edit the associated fur map, you need to enable the **Edit Fur Map** option.
Screen Brush	The **Screen Brush** option, enabled by default, affects all vertices under the brush tips circle, regardless of their distance. When the option is disabled, only vertices that are within the 3D sphere of influence around the brushes tip are affected.

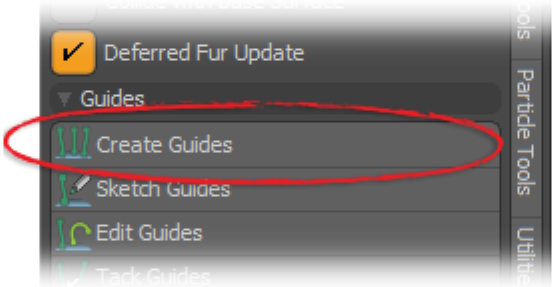
Hair Guides

Much like the sculpt and painting tools, Modo offers a variety of tools for styling fur and for creating and editing guides used in longer hairstyles. Located in the **Paint** layout, the **Hair Tools** sub-tab (along the edge of the toolbox) gives you everything needed for combing and sculpting fur into the proper do. With the Fur Vector Maps (see Effect - Texture Item) and the settings of the Fur Material item, a lot of control is afforded for getting specific looks, but for longer hair styles, guides are necessary to really control the direction and length of the fur. Before any guides can be styled and combed though, they must be created. In this regard, Modo offers several tools specific to this task.

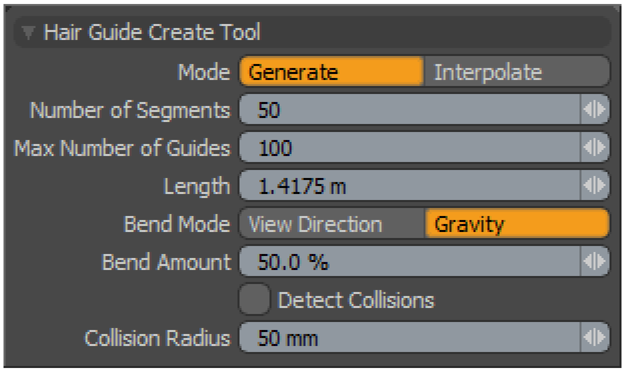
Initial guides can be easily generated using the **Create Guides** tool. This tool automatically generates the polyline segment type guides necessary for Fur, by virtually growing them from selected polygons. Simply click on the appropriate button to enable any of the following functions.

Create Guides

When using the **Create Guides** tool, the generated curves inherit the material tag from the selected surface, so it is best to name the target Fur surface with a proper material before activating it.



If the polygons aren't already selected, highlight the polygons that the fur should grow from and keep them highlighted. Click the **Create Guides** button in the toolbox to activate the tool. Next, clicking in the 3D viewport initiates the interactive growing process, spawning one guide per vertex for all selected polygons up to the **Max Number of Guides** value. Clicking and hauling left or right grows or shrinks the guides in length. You can also manually set the length directly in the tool properties. When the tool is active, there are additional attributes you can use to adjust in the tool's **Properties** panel.



Hair Guide Create Tool

Mode	Generate	Interpolate
Number of Segments	50	
Max Number of Guides	100	
Length	1.4175 m	
Bend Mode	View Direction	Gravity
Bend Amount	50.0 %	
	☐ Detect Collisions	
Collision Radius	50 mm	

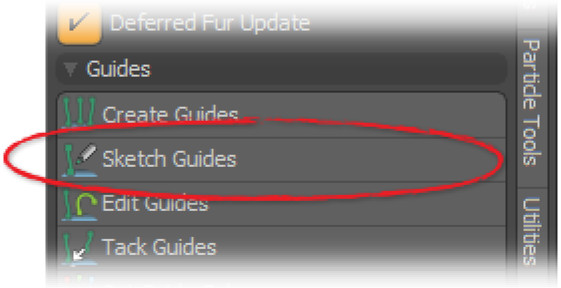
Hair Guide Create Tool

Option	Description
Mode	Hair Guides can be created in two ways, either by growing from selected polygons in the **Generate** mode, or using **Interpolate** to generate new guides from existing ones.
Generate	Guide Curves are created parametrically with the various settings of the tool. Clicking and hauling in the viewport interactively adjusts the **Length** value, other values are adjusted numerically in each input field.
Interpolate	Guides are created one at a time on the target surface where you click. Interpolation generates a new guide curve based on existing curves. Clicking on a surface generates the curve as an interpolation (in-between morph) of nearby guides. Clicking and dragging moves the curve over the surface (note how its shape changes to match those of nearby curves as it slides around). **Shift**+clicking adds new guides. When in interpolation mode, no numeric inputs are available.
Number of Segments	This setting is independent of the fur segments, specifying the number of segments used to create the guides themselves. Increasing the number of guides can allow for more detailed control, but watch out as manipulating them with too many segments can become slow and may produce less than satisfactory results. Add just as many as are needed.
Max Number of Guides	Generally, with this tool, a guide is created at each vertex position on the target selection/surface. If the surface has a lot of polygons, it may be desirable to control the number of guides created using the **Max Number of Guides**. If the target has less than the max number of vertex points, only that many guides are created.
Length	You can use this setting to manually set the guide length to a specific value. Clicking and hauling in the viewport interactively adjusts this value.
Bend Mode	With **Bend Mode** you can choose how the generated guides bend based on either the **View Direction** or **Gravity**. With **View Direction**, the guides grow straight from the normal direction of the polygon, but bend based on the rotation of the currently active viewport, bending away from the camera. This is cool because its possible to select a few polygons from the fur surface and create the guides with the view rotated in one direction. Then select some other polygons and create the guides with a different rotation, which can help when devising guides for hair parting seams. **Gravity** bend all guides toward the negative **Y** direction.
Bend Amount	The **Bend Amount** controls how far a guide bends when it is created. A setting of 0%

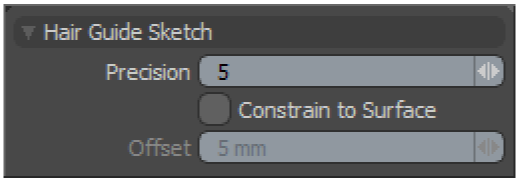
Option	Description
	would grow guides perfectly straight in the direction of the vertex normal at its root, increasing this value increases the intensity of the bend in the direction defined by the **Bend Mode**.
Detect Collisions	When creating the guides, you might find that some are intersecting with surrounding surfaces. Enabling **Detect Collisions** avoids this situation by detecting surrounding surfaces realistically drooping and bending the guides around them. Best example is generating guides for long hair, you don't want the hair to intersect the shoulders, so enable this setting to drape the guides over the shoulder for a more realistic look.
Collision Radius	You still might find that fur generated from guides intersects with a surface. When **Detect Collisions** is enabled, the collision radius allows you to offset the guides from the surface by the specified amount to avoid the intersection. Quite often guides are used in **Range** mode, generating fur fibers within a certain range around the guide. It may be helpful to set the collision radius to half the range (as range is a diameter) to effectively avoid fibers intersecting the surfaces as well.

Sketch Guides

With the **Sketch Guides** tool you can draw out guides directly in the 3D viewport with a pen tool.



Clicking on the target surface, the guide is planted on the surface and is drawn in a plane perpendicular to the view direction, going through the surface normal.



Precision 5

☐ Constrain to Surface

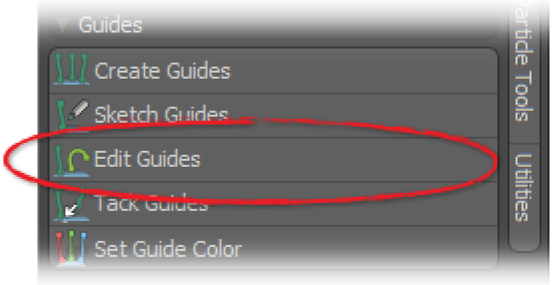
Offset 5 mm

Hair Guide Sketch

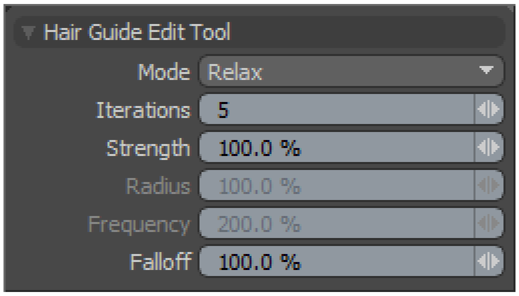
Option	Description
Precision	This value sets how finely Modo interpolates the drawn stroke, lower values produce strokes with greater numbers of points, higher values produce strokes with lower numbers of points. Think of it roughly like the distance in pixel the pointer travels before creating a new point.
Constrain to Surface	The **Constrain to Surface** option, when enabled, draws Hair Guides constrained directly over the target surface (like the background constraint does), the **Offset** value can be used to push the drawn guides away from the surface when drawn.
Offset	The **Offset** value works in conjunction with the **Constrain to Surface** option to push the drawn curves away form the surface while drawn (the first, root vertex of the guide always plants itself directly against the target surface still).

Edit Guides

Once Guides have been generated, the **Edit Guides** tool can be used to easily modify guides over their length, creating effects that would be difficult if not impossible to style manually.



Different controls are applied depending on the selected **Mode**.

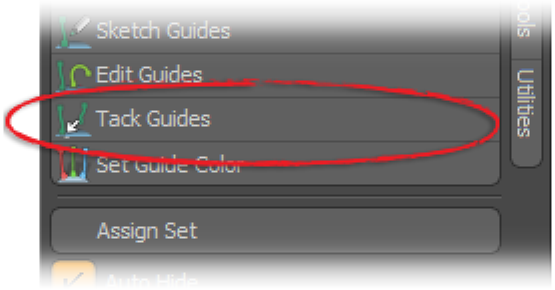


Mode Relax
Iterations 5
Strength 100.0 %
Radius 100.0 %
Frequency 200.0 %
Falloff 100.0 %

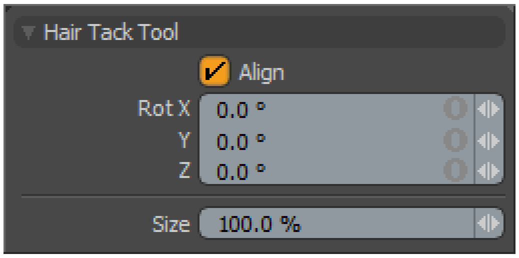
Hair Guide Edit Tool

Option	Description
Mode	The **Mode** options determine what form the edits takes on the target hair guides. When the tool is invoked, it modifies all visible curves in the foreground layer(s). To target only specific curves, either select them prior to activating the tool or hide unwanted hair guides. Managing multiple groups of hair guides can be done withManaging Hair Guides controls. In general moving the mouse left and right control the **Strength** of the effect and up and down movements controls the **Falloff** amount. For **Curls** and **Waves**, holding the **Ctrl** key gives control to **Frequency** and **Radius** values, where **Frequency** controls the number of curls and **Radius** the amplitude. There are six types of **Edit** modes available: • **Relax** - Relaxes the guide. • **Straighten** - Transforms the guide back to its straight (linear) shape. • **Bend** - Bends the guide like it is affected by gravity. • **Size** - Changes the size of the entire guide while preserving the shape. • **Curls** - Creates curls along the guide. • **Waves** - Another form of curling with more amplitude on one axis than the other.
Iterations	The **Iterations** value determines the number of times the edit function is applied to the target. Works with **Smooth**, **Straighten**, and **Bend**, and works in conjunction with the **Strength** option to control the overall effectiveness of the selected edit.
Strength	Controls the amount of the selected edit that affects the target hair guides. Clicking and dragging the cursor left and right on the screen interactively adjust this value.
Radius	The **Radius** option controls the amplitude (size) of the resulting curls for both **Curls** and **Waves**. Interactively editable by holding the **Ctrl** key while editing, dragging the cursor left and right.
Frequency	The **Frequency** option controls the number of curls along the length of the guide. Interactively editable by holding the **Ctrl** key while editing and dragging the cursor up and down.
Falloff	The **Falloff** value controls the root falloff, or to what degree the root of the Hair Guide is affected by the edit. Low values affect more of the guide along it length while higher values affect more of the guide. Clicking and dragging the cursor up and down on the screen interactively adjust this value.

Tack Guides



The **Tack Guides** tool can be used to position and rotate individually selected hair guides interactively. It works by you first selecting a target guide. This is done using the **Polygons** selection mode and clicking and dragging over the target guide to select, highlighting it orange. Next activate the **Tack Guides** tool in the **Hair Tools** tab of the toolbox and click on the target surface to move the guide to the new position, this moves the guides origin to the intersection of the surface and the mouse's pointer, moving interactively while the mouse is moved. With the **Align** option enabled the guide remains oriented in the same angle relative to the underlying surface normal. Pressing the **Ctrl** key interactively rotates the guide around the surface normal.


Hair Tack Tool

☑ Align

Rot X 0.0 °

Y 0.0 °

Z 0.0 °

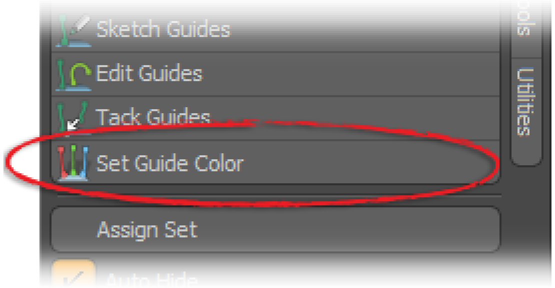
Size 100.0 %

Hair Tack Tool

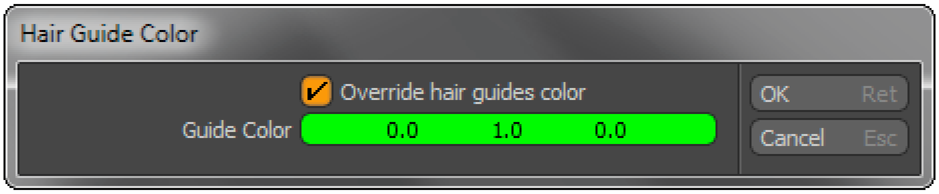
Option	Description
Align	When the **Align** option is enabled, the guide retains its orientation angle relative to the underlying surface normal. When disabled, the guide retains its initial orientation.
Rot X/Y/Z	The Rotation input field allow you to numerically enter specific rotation values for the edited guide. You can also interactively rotate the guide by pressing the **Ctrl** key and clicking and dragging in the 3D viewport.
Size	The **Size** option controls the scaling of the edited hair guide.

Set Guide Color

When styling fur, depending on the source surface's coloration, it can be difficult to see the guides clearly. The **Set Guide Color** function allows you to override the default guide color (temporarily if you wish) and set the guide's color to any arbitrary value desired.


Sketch Guides
Edit Guides
Tack Guides
Set Guide Color
Assign Set
Auto Hide
ols
Utilities

To apply, simply press the **Set Guide Color** button, located below the **Hair Tools** sub-tab of the **Paint** toolbox to open the **Hair Guide Color** dialog.


Hair Guide Color

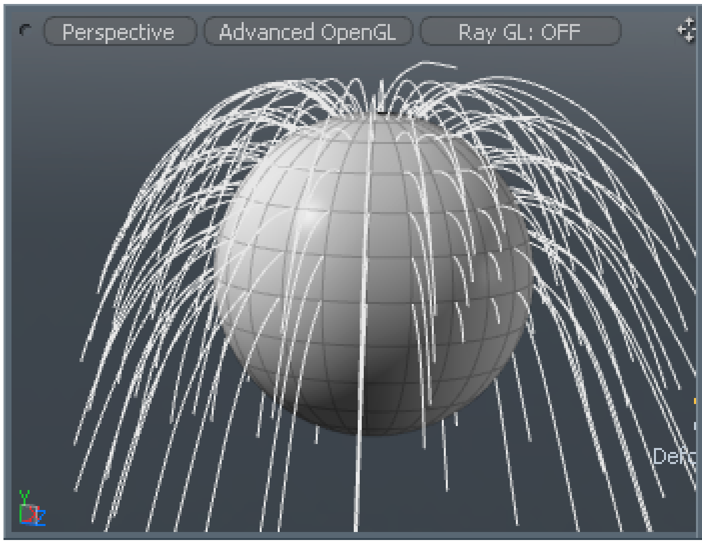
☑ Override hair guides color

Guide Color 0.0 1.0 0.0

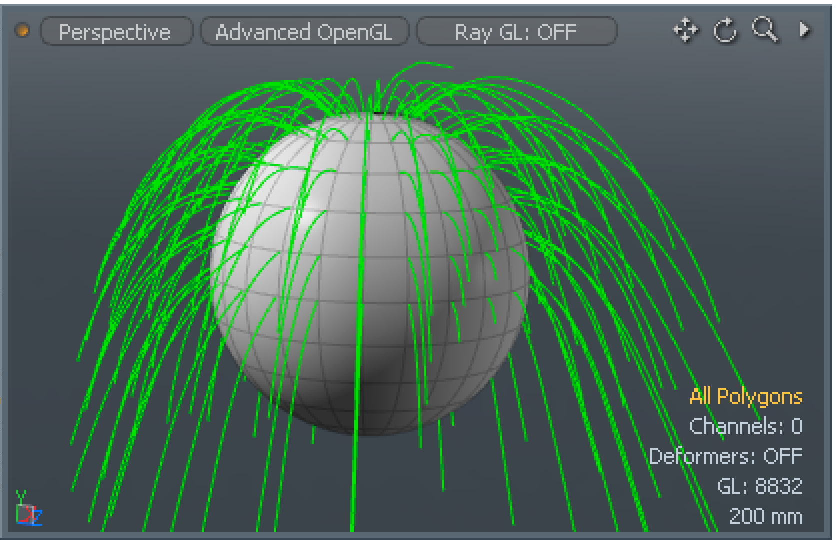
OK Ret
Cancel Esc

Enable the **Override hair guides color** option, define a color and press **OK**. All guides use this color for display. To disable, open the **Hair Guide Color** dialog again and disable the **Override** option and press **OK**. As the color override is system wide, this can also be useful when working with guides for other means as well.

Note: The color display is only visible while in the Shaded and Advanced OpenGL display modes.


Perspective
Advanced OpenGL
Ray GL: OFF
Def

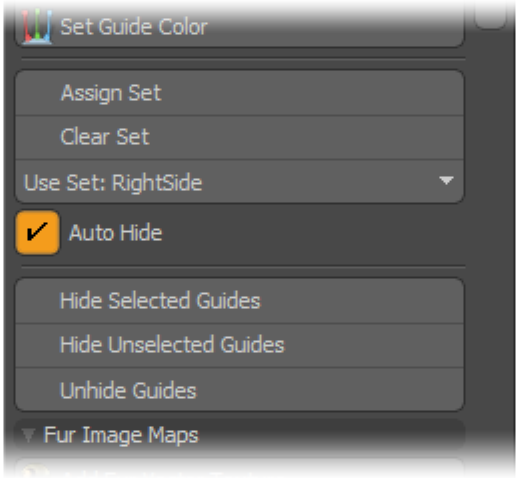
Hair Guide set at default color.


Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 8832
200 mm

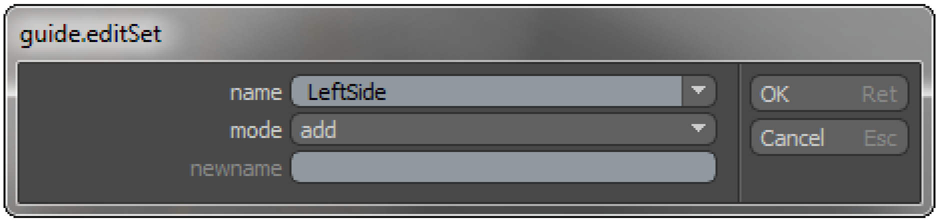
Hair Guide Color Override enabled.

Managing Hair Guides

When working with many multiples of Hair Guides, it can be handy to manage these guides with Hair Guide Selection Sets. This allows you to only work on specific guides, with the additional benefit of increasing performance for guide heavy setups. Take for instance all the hair on one side of a part would be styled in one way, while the hair on the other is styled a different way. Sets are simple way to group like guides together and easily manage their selection state and visibility for both sides of the part. The Selection Set controls are found directly under the Hair Guide edit and creation tools on the Hair Tools sub-tab of the painting toolbox.


Set Guide Color
Assign Set
Clear Set
Use Set: RightSide
Auto Hide
Hide Selected Guides
Hide Unselected Guides
Unhide Guides
Fur Image Maps

Creating a set starts with selecting the target Hair Guides. Guides are selected in the **Polygons** selection mode. You can either lasso select around multiple guides at once, or paint select them by clicking and dragging the cursor over the guides to select. The default selection helpers work the same for hair guides as well, **Ctrl** to remove items from a selection and **Shift** to add to it. Once selected, click the **Assign Set** button to open the selection set assignment dialog.


guide.editSet

name: LeftSide

mode: add

newname:

With the dialog open, you need only to specify a **Name** and press **OK** to create the set. There are also options to **Remove** items from the set and to **Rename** the set to a New Name. Once the first set is created, additional buttons appear in the interface to clear an existing set and another to set the current set. Click the **Use Set** button to open the small pop-up menu to select from all available sets. Click the target set to select all the associated guides. Now when editing is applied, only the selected guides are affected. When the **Auto Hide** toggle is enabled, only the currently selected set is visible in the 3D viewport, hiding all other guides.

You can also manually control visibility for any given guide set with the visibility buttons, for selected hair guides there are controls to **Hide Selected Guides**, **Hide Unselected Guides**, and to **Unhide Guides** which makes all hidden guides visible. It should be noted that the visibility toggle buttons do not strictly require **Sets**, but merely work on selection, with the sets as an enhancement to the overall workflow.

Using Particle Tools

Particles are single vertices that provide important functions. They may act as point sources for Replicators and Texture Replicators, as well as for Volumes, Blobs and Sprites. With the particle tools you can create and edit particle point clouds by using the familiar brush-based workflow available with other functions in Modo. Additionally, you can edit Vertex Maps by using the Size and Dissolve tools to directly modify associated particle Vertex Maps.

Particle Creation

Particle Paint Tool

Particle Editing

Particle Paint Smooth

Particle Paint Pinch

Particle Paint Flatten

Particle Paint Smudge

Particle Paint Tangent Pinch

Particle Paint Size

Particle Paint Move

Particle Paint Spin

Particle Paint Dissolve

Particle Paint Dissolve

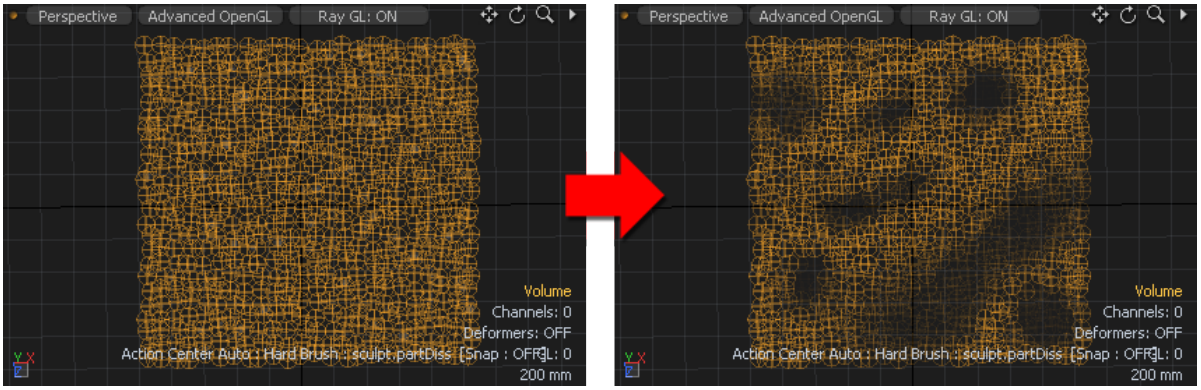
The particle tools (Size and Dissolve) modify particle-related attributes to give you control over settings that would be difficult to control otherwise. You can use the Particle Dissolve function, when applied as a particle dissolve Vertex Map, to directly control the opacity of Volumes and Sprites on individual particles. In this case, particles can be any number of things: a mesh layer with single point vertices or a collection of single point polygons (also called point clouds). There are several ways you create a point cloud, but the most direct way is to use the Particle Paint Tool.

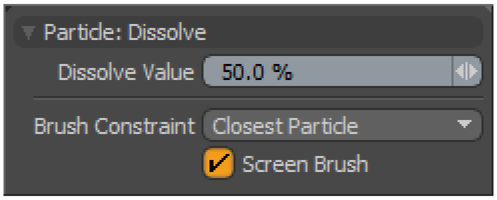
Note: You cannot use particle Vertex Maps to control procedural particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. To use the Particle Dissolve tool, first select the target point cloud Mesh Item layer in the **Items** list. You also need to create and select a particle dissolve map in the Vertex Map list viewport. See the Working with Vertex Maps topic for information on creating Vertex Maps. Preferably, apply the volume or sprite items to the point cloud before using the Dissolve tool because the items' viewport representations provide feedback when you apply the Size and Dissolve functions.

After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. Pressing **Ctrl** negates the effects of the tool to add opacity instead of removing it.


Perspective
Advanced OpenGL
Ray GL: ON
Volume
Channels: 0
Deformers: OFF
Action Center Auto : Hard Brush : sculpt.partDiss [Snap : OFF]L: 0
200 mm
Perspective
Advanced OpenGL
Ray GL: ON
Volume
Channels: 0
Deformers: OFF
Action Center Auto : Hard Brush : sculpt.partDiss [Snap : OFF]L: 0
200 mm


Particle: Dissolve

Dissolve Value: 50.0 %

Brush Constraint: Closest Particle

☑ Screen Brush

Particle: Dissolve	
Dissolve Value	Controls the strength of the dissolve amount applied to the particles. Lower values affect the transparency more slowly than higher values do.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Screen Brush	When enabled, flattens the particle volume so that all particles through the

Particle: Dissolve	
	volume are affected by the brush regardless of how close or distant they are. When particle painting with this disabled, the volume of the brush tip in 3D space is a fixed size, and it only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

Tip: If you apply any **Shader Tree** layers defined as a **Particle Dissolve** effect to the particle item, these values supersede those of the Particle Dissolve Vertex Map.

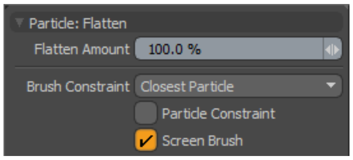
Particle Paint Flatten

The particle tools (Smooth, Smudge, Move, Pinch, Tangent Pinch, Spin, and Flatten) modify particle positions within a point cloud. Much like the sculpt tools shape surfaces, the particle tools control, position, and direct particles, even individually, to produce a desired result. Particles can be any number of things: a mesh layer with single-point vertices or a collection of single point polygons (also called point clouds). There are several ways to create a point cloud, but the most direct way is by using the Particle Paint Tool.

Note: You cannot use the particle tools to modify procedurally generated particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. The Flatten tool gathers particles toward the brush's center. The speed at which the particles move is dependent on the **Flatten Amount**. Pressing **Ctrl** negates the effects of the tool by spreading the particles outward (away from the brush's center).



Particle: Flatten	
Flatten Amount	Controls the strength of the flattening, which is the speed at which the affected particles move inward or outward.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry. **Tip:** When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.
Screen Brush	When enabled, flattens the particle volume so that all particles through the volume are affected by the brush, regardless of how close or distant they are. When disabled, the volume of the brush tip in 3D space is a fixed size and only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

Particle Paint Move

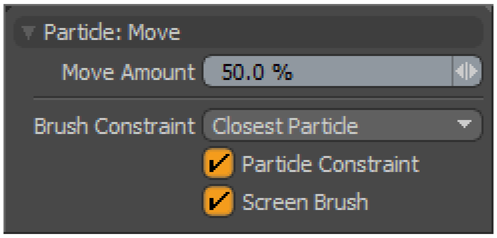
The particle tools (Smooth, Smudge, Move, Pinch, Tangent Pinch, Spin, and Flatten) modify particle positions within a point cloud. Much like the sculpt tools shape surfaces, the particle tools control, position, and direct particles, even individually, to produce a desired result. Particles can be any number of things: a mesh layer with single-point vertices or a collection of single point polygons (also called point clouds). There are several ways to create a point cloud, but the most direct way is by using the Particle Paint Tool.

Note: You cannot use the particle tools to modify procedurally generated particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. The Move tool drags the particles the distance that the brush travels. The **Move Amount** determines how much the particles are affected. Pressing **Ctrl** negates the effects of the tool by moving the particles in the opposite direction of the pointer.

Particle Move



Particle: Move

Move Amount 50.0 %

Brush Constraint Closest Particle

☑ Particle Constraint

☑ Screen Brush

Particle Move	
Move Amount	Controls the strength of the brush's influence in moving particles. A value of **100%** moves the particles the same distance the pointer travels; with a value of **0%**, the particles wouldn't move at all.

Particle Move	
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry. **Tip:** When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.
Screen Brush	When enabled, flattens the particle volume so that all particles through the volume are affected by the brush, regardless of how close or distant they are. When disabled, the volume of the brush tip in 3D space is a fixed size and only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

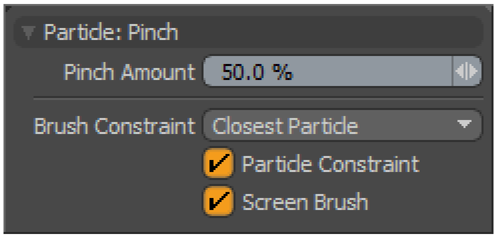
Particle Paint Pinch

The particle tools (Smooth, Smudge, Move, Pinch, Tangent Pinch, Spin, and Flatten) modify particle positions within a point cloud. Much like the sculpt tools shape surfaces, the particle tools control, position, and direct particles, even individually, to produce a desired result. Particles can be any number of things: a mesh layer with single-point vertices or a collection of single point polygons (also called point clouds). There are several ways to create a point cloud, but the most direct way is by using the Particle Paint Tool.

Note: You cannot use the particle tools to modify procedurally generated particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. The Pinch tool gathers particles toward the brush's center. The **Pinch Amount** determines the speed at which they move. Pressing **Ctrl** negates the effects of the tool by spreading the particles outward (away from the brush's center).



Particle: Pinch

Pinch Amount: 50.0 %

Brush Constraint: Closest Particle

☑ Particle Constraint

☑ Screen Brush

Particle: Pinch	
Pinch Amount	Controls the strength of the pinching, which is the speed at which the affected particles move inward or outward.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry.

Particle: Pinch	
	Tip: When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.
Screen Brush	When enabled, flattens the particle volume so that all particles through the volume are affected by the brush, regardless of how close or distant they are. When disabled, the volume of the brush tip in 3D space is a fixed size and only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

Particle Paint Size

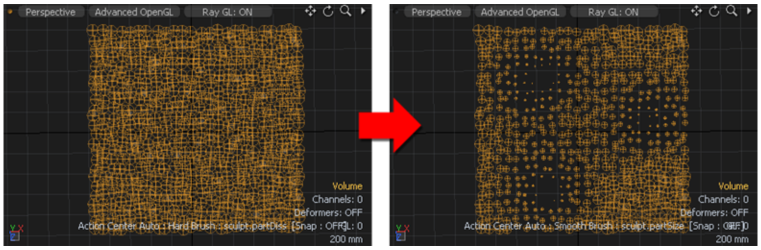
The particle tools (Size and Dissolve) modify particle-related attributes to give you control over settings that would be difficult to control otherwise. You can use the Particle Size function, when applied as a particle size Vertex Map, to directly control the scale of volumes and sprites on individual particles. In this case, particles can be any number of things: a mesh layer with single point vertices or a collection of single point polygons (also called point clouds). There are several ways you create a point cloud, but the most direct way is to use the Particle Paint Tool.

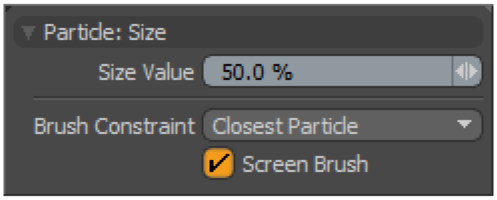
Note: You cannot use particle Vertex Maps to control procedural particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. To use the Particle Size tool, first select the target point cloud Mesh Item layer in the **Items** list. You also need to create and select a particle size map in the Vertex Map list viewport. See the Working with Vertex Maps topic for information on creating Vertex Maps. Preferably, apply the volume or sprite items to the point cloud before using the Size tool because the items' viewport representations provide feedback when you apply the Size and Dissolve functions.

After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. Pressing **Ctrl** negates the effects of the tool by decreasing the particle size instead of increasing it.





Particle: Size
Size Value 50.0 %
Brush Constraint Closest Particle
☑ Screen Brush

Particle: Size	
Size Value	Controls the strength of scaling amount applied to the particles. Lower values scale the particles more slowly than higher values do.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry.

Particle: Size	
	Tip: When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.
Screen Brush	When enabled, flattens the particle volume so that all particles through the volume are affected by the brush regardless of how close or distant they are. When particle painting with this disabled, the volume of the brush tip in 3D space is a fixed size, and it only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

Tip: If you apply any **Shader Tree** layers defined as a **Particle Size** effect to the particle item, these values supersede those of the Particle Size Vertex Map.

Particle Paint Smooth

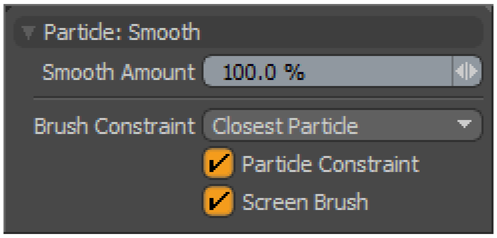
The particle tools (Smooth, Smudge, Move, Pinch, Tangent Pinch, Spin, and Flatten) modify particle positions within a point cloud. Much like the sculpt tools shape surfaces, the particle tools control, position, and direct particles, even individually, to produce a desired result. Particles can be any number of things: a mesh layer with single-point vertices or a collection of single point polygons (also called point clouds). There are several ways to create a point cloud, but the most direct way is by using the Particle Paint Tool.

Note: You cannot use the particle tools to modify procedurally generated particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth**

Brush as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. The Smooth tool gathers particles toward the brush's center. The **Smooth Amount** determines the speed at which they move. Pressing **Ctrl** negates the effects of the tool by spreading the particles outward (away from its center).



Smooth Amount 100.0 %

Brush Constraint Closest Particle

☑ Particle Constraint

☑ Screen Brush

Particle: Smooth	
Smooth Amount	Controls the strength of the smoothing, which is the speed at which the affected particles move inward or outward.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry. **Tip:** When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.
Screen Brush	When enabled, flattens the particle volume so that all particles through the volume are affected by the brush, regardless of how close or distant they

Particle: Smooth	
	are. When disabled, the volume of the brush tip in 3D space is a fixed size and only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

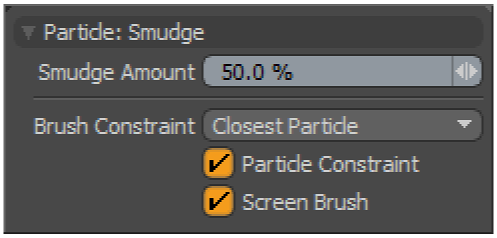
Particle Paint Smudge

The particle tools (Smooth, Smudge, Move, Pinch, Tangent Pinch, Spin, and Flatten) modify particle positions within a point cloud. Much like the sculpt tools shape surfaces, the particle tools control, position, and direct particles, even individually, to produce a desired result. Particles can be any number of things: a mesh layer with single-point vertices or a collection of single point polygons (also called point clouds). There are several ways to create a point cloud, but the most direct way is by using the Particle Paint Tool.

Note: You cannot use the particle tools to modify procedurally generated particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. The Smudge tool drags particles in the direction of the brush as if they are smeared. The **Smudge Amount** determines the distance that the particles are dragged. Pressing **Ctrl** negates the effects of the tool by dragging the particles in the opposite direction of the brush's movement.


Particle: Smudge
Smudge Amount 50.0 %
Brush Constraint Closest Particle
☑ Particle Constraint
☑ Screen Brush

Particle: Smudge	
Smudge Amount	Controls the strength of the smearing, which is the distance the affected particles move.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry. **Tip:** When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.
Screen Brush	When enabled, flattens the particle volume so that all particles through the volume are affected by the brush, regardless of how close or distant they are. When disabled, the volume of the brush tip in 3D space is a fixed size and only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

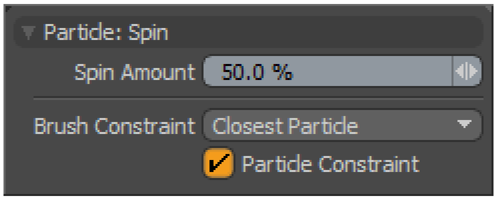
Particle Paint Spin

The particle tools (Smooth, Smudge, Move, Pinch, Tangent Pinch, Spin, and Flatten) modify particle positions within a point cloud. Much like the sculpt tools shape surfaces, the particle tools control, position, and direct particles, even individually, to produce a desired result. Particles can be any number of things: a mesh layer with single-point vertices or a collection of single point polygons (also called point clouds). There are several ways to create a point cloud, but the most direct way is by using the Particle Paint Tool.

Note: You cannot use the particle tools to modify procedurally generated particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. The Spin tool rotates the particles around the brush's center. Pressing **Ctrl** negates the effects of the tool by rotating the particles in the opposite direction.


Particle: Spin
Spin Amount 50.0 %
Brush Constraint Closest Particle
☑ Particle Constraint

Particle: Spin	
Spin Amount	Controls the strength of the rotation, which is the speed at which the affected particles move inward or outward.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it

Particle: Spin	
	moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry. **Tip:** When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.

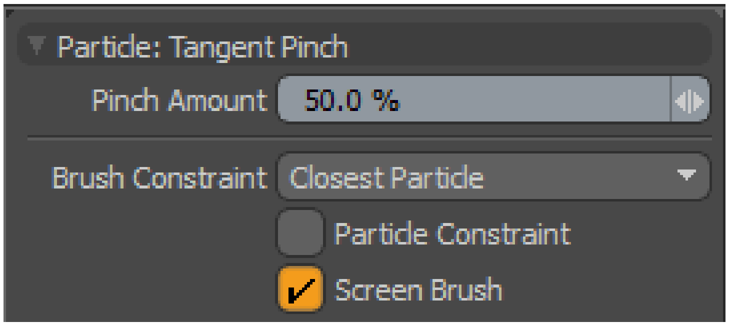
Particle Paint Tangent Pinch

The particle tools (Smooth, Smudge, Move, Pinch, Tangent Pinch, Spin, and Flatten) modify particle positions within a point cloud. Much like the sculpt tools shape surfaces, the particle tools control, position, and direct particles, even individually, to produce a desired result. Particles can be any number of things: a mesh layer with single-point vertices or a collection of single point polygons (also called point clouds). There are several ways to create a point cloud, but the most direct way is by using the Particle Paint Tool.

Note: You cannot use the particle tools to modify procedurally generated particles such as those of a Particle Cloud or a Particle Generator.

Using the Tool

The particle tools are in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. After you select the tool, its attributes appear in the **Properties** viewport. When you first select the tool, it has **Smooth Brush** as the default brush tip. You can select an alternate brush tip before proceeding or right-click in the 3D viewport to scale the brush size. Drag in the viewport to affect the particles. The Tangent Pinch tool gathers particles toward the brush's center. The **Tangent Pinch Amount** determines the speed at which they move. Pressing **Ctrl** negates the effects of the tool by spreading the particles outward (away from the brush's center).



Particle: Tangent Pinch

Pinch Amount 50.0 %

Brush Constraint Closest Particle

☐ Particle Constraint

☑ Screen Brush

Particle: Tangent Pinch	
Pinch Amount	Controls the strength of the pinching, which is essentially the speed at which the affected particles move inward or outward.
Brush Constraint	Controls what area within the volume is affected by the brush when you paint into a volume of particles from a 2D viewport. (You can only view the 3D elements in the flat 2D space of the screen.) • **Closest Particle** - Affects the particle(s) closest to the viewport camera (closest to you). • **Grid** - Affects the particle(s) closest to the Work Plane grid. • **Center Particle** - Affects the particle(s) closest to the center of the volume under the brush.
Particle Constraint	When enabled, constrains each individual vertex against any foreground polygons to ensure the vertex always rests on a polygon's surface as it moves. Works in the same manner as the **Constrain to Background** - Point mode, but this is not dependent upon background geometry. **Tip:** When working with the **Particle Constraint** option, the constraining geometry doesn't have to be in the same layer as the point cloud. You can press **Ctrl** as you select additional item layers to bring them into the foreground. The edited particles conform to the foreground polygon's surface but remain within their initial layer.
Screen Brush	When enabled, flattens the particle volume so that all particles through the volume are affected by the brush, regardless of how close or distant they are. When disabled, the volume of the brush tip in 3D space is a fixed size and only affects the particles within this volume (as defined by the **Brush Constraint** option). Particles that are further away in the volume are not affected even though the brush is passing over them.

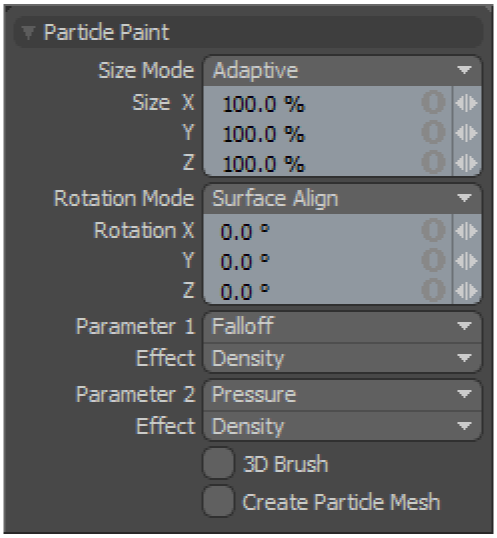
Particle Paint Tool

Similar to the Painting tool, you use the Particle Paint tool to spray particles onto a surface in a controllable manner. Particles are single vertices that can control a number of other functions in Modo, such as point sources for geometry Replicators and the Texture Replicator, as well as point clouds for Blobs, Volumes, and Sprites.

Once you create the particles, you can continue editing them by using the other particle tools. When you use these tools with the **Particle Constraint** option, you have control over the placement of individual particles. Also, you can take advantage of the additional controls available with Particle Vertex Maps (such as the transform map), which provide extensive controls for generated particles.

Using the Tool

The Particle Paint tool is in the **Paint** interface tab under the **Particle Tools** sub-tab of the toolbox. Particles only spray onto polygon surfaces, so you need to select the appropriate target layer to spray onto before activating the tool. Upon selecting the tool, its attributes appear in the **Properties** viewport. When first selected, the tool defaults to **Smooth Brush** as the brush tip. You can select an alternate brush tip before proceeding, or you can right-click in the 3D viewport to scale the brush size. After specifying the brush tip, drag in the viewport to create particles. You can add more control by setting the **Parameter 1** and **Parameter 2** options.


Particle Paint
Size Mode Adaptive
Size X 100.0 %
Y 100.0 %
Z 100.0 %
Rotation Mode Surface Align
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Parameter 1 Falloff
Effect Density
Parameter 2 Pressure
Effect Density
☐ 3D Brush
☐ Create Particle Mesh

Particle Paint	
Size Mode	Determines the scaling for the associated particle size Vertex Map. When creating particles, Modo generates a particle size Vertex Map automatically to store additional information for controlling any scaling of eventual clones or volume elements. • **Uniform** - Results in identically-sized elements that are determined by the **Size X, Y, Z** values. • **Adaptive** - Scales elements larger or smaller based on the size of the brush tip used. • **Random** - Scales the elements randomly up to the maximum scaling amount that is specified by the **Scale X, Y, Z** values. • **Adaptive+Random** - Scales the elements randomly in relation to the size of the brush tip with the maximum scaling amount specified by the **Scale X, Y, Z** values.
Rotation Mode	Determines the rotation for the associated particle size Vertex Map. When creating particles, Modo generates a particle size Vertex Map automatically to store additional information for controlling any rotation of eventual clones or volume elements. • **Uniform** - Orients elements identically by the **Rotation X, Y, Z** values. • **Surface Align** - Orient elements to align with the surface polygon's normal direction. • **Random** - Rotates each element randomly up to the maximum rotation amount specified by the **Rotation X, Y, Z** values. • **Surface Align+Random** - Orients each element to the surface's normal direction first, and then offsets the orientation with an additional random rotation up to the maximum amount specified by the **Rotation X, Y, Z** values. • **Surface Align+Tablet** - Orients each element to the surface's normal direction first, and then offsets the rotation based on the angle of the stylus on the tablet. • **Surface Align+Tablet (No Pitch)** - Orients each element to the surface's normal direction first (ignoring the sloping angle, or pitch), and then offsets the rotation based on the angle of the stylus on the tablet. • **Screen Align+Tablet** -

Particle Paint	
	• **Screen Align+Tablet (No Pitch)** -
Parameter 1 / Parameter 2	Provide two additional controls for creating particles. First, select the controlling input from the **Parameter** list, and then select the result from the **Effect** list. Usually, these options combine with the brush tip settings, but some options use a stylus-and-tablet input combination to control elements. • **Falloff** - Allows you to specify one of Modo falloffs to determine the effect. Because the brush size supersedes the falloff, define the falloff before activating the tool. • **Pressure** - Uses the pressure of the stylus against the tablet's surface to control the effect. • **Slope** - Uses the downward slope of the target polygon's surface to control the effect. • **Altitude** - • **1.0 Slope** - • **1.0 Altitude** -
Effect	Specifies the results that are controlled by the selected **Parameter**. • **(none)** - Disables the **Parameter** and **Effect** options. • **Density** - Controls the number of sprayed particles as determined by the specified **Parameter**. • **Size** - Controls the scale of sprayed particles as determined by the specified **Parameter**. • **Rotation**- Controls the orientation of sprayed particles as determined by the specified **Parameter**. • **Size+Density**- Controls the number and scale of sprayed particles as determined by the specified **Parameter**.
3D Brush	When enabled, creates particles within the 3D volume of the brush tip. This adds depth to the particles, so they aren't placed directly against the constraining background surface.
Create Particle Mesh	When enabled, creates an additional Mesh Item layer as a child item of the target layer. This mesh layer contains all the particles and eliminates the need to copy and paste the vertices into a new mesh layer manually when used as a point source.

Creating a Layout

Once elements are modeled and textured, transforming a series of models into a scene requires some amount of layout - the placement and preparation for final rendering. These steps can also be useful in preparation for animation. Modo provides a useful **Layout** interface, with access to item transformation and duplication tools and a **Preset Browser** to speed up layout tasks. Learn more about the layout workflow, transforming, and duplicating items.

Working with Items

Modo scenes are basically collections of items: cameras, lights, meshes, locators, and so on. These items are collected and cataloged either in the **Item List** or in the **Shader Tree**, where each layer represents an entire object unto itself. You can tell Modo what particular item you wish to edit by selecting them. When selected, their associated attributes appear in a Properties Panel panel where you can manipulate their values directly, or easily manipulate items directly in the 3D Viewports.

Creating Items

Modo starts with a number of basic item layers. You have a camera that is necessary for rendering and an empty mesh layer for building geometry. You can add additional item layers to the scene in a number of ways. You can do most of your scene item management from the Items (Scenes) List.

1. Click on the **Add Item** button in the upper right area of the viewport to open the **Preset Browser**.
2. Click on the appropriate item to select it and add to the list.

   New items are added directly above the current selection.

You can also add items from the menu bar: in the **Item** menu, there are a variety of commands that add the appropriate item.
Create Camera: This command adds an additional camera item with default settings at the origin of the scene. If an item exists with an identical name, Modo adds an appropriate number to the new camera to differentiate it from the existing camera. Cameras are necessary for setting point of view and render settings specific to the frame. For more information on the camera, item, see Camera Items.
Create Light: This menu opens options for adding any of the variety of direct light sources Modo supports, including area, cylinder, directional, dome, photometric, point and spot light types. If an existing item in the

scene has an identical name to the added item, Modo adds an appropriate number to the new light's name to differentiate it from existing lights.

Create Locator: This command adds a locator item, also sometimes referred to as a null object. Locators are useful as reference objects, and can aid in the creation of hierarchies. Locators do not contribute to the rendered image but do have a 3D viewport representation useful for applying transforms. For more information on locators, see Locator Item.

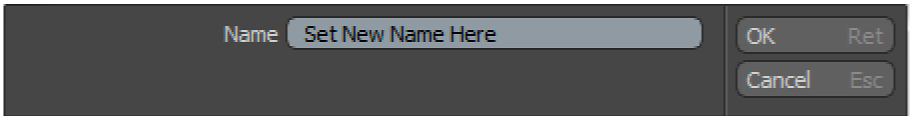
Create Texture Locator: This command adds a texture locator item. You generally don't add a texture locator manually, as they are added automatically whenever an image map or procedural texture layer is added. Texture locators define how textures are applied to surfaces at render time. For more information, see Texture Locator.

Create Backdrop: This command creates a backdrop item. Backdrop items allow you to display images directly in the 3D Viewports, useful as modeling reference. These images are not included in final rendered images. For more information, see Modeling with Images.

Create Environment: This command adds an additional environment item to the **Shader Tree**. Environments represent the background areas of the rendered scene. For more information, see Environment Item.

Primitive: This menu allows you to choose from and add a variety of unit primitives, basic shaped geometry scaled approximately to 1 cubic meter in size. These objects can act as stand-in proxy geometry, the basis for more complex objects, or as final geometry requiring only simple forms. For more information on these tools, see Creating Geometry. You can also add unit primitives by holding **Ctrl** and clicking on the appropriate tool in the **Modeling** toolbox. Press **Shift** to add the unit primitives as a new Mesh Item layer.

Renaming Items



Name Set New Name Here

To rename an item:

1. Select the target layer, then right-click directly on the item.

   This opens a context menu.

2. Selecting the **Rename** option form the menu to open a pop-up dialog.
3. Enter the new name and click **OK**.

You can also edit the name by clicking on an item, then clicking on it again. The name becomes editable and you can enter the new name. User can also select the target item in either the **Item List** or the **Shader Tree** and select the menu bar command **Item** > **Rename**.

Duplicating Items

You can clone items in a variety of ways, depending on the desired outcome and number of clones. The easiest route is the direct duplicate.

1. First, select the target layer and right-click directly on the layer.

   A context menu displays.
2. Choose the **Duplicate** command from context menu.

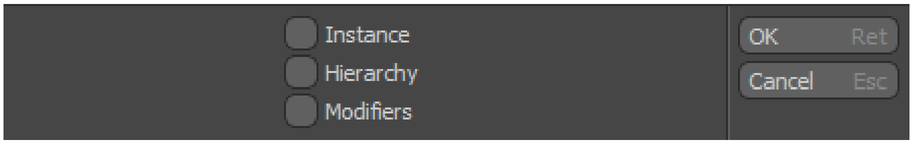
   A copy of the item is created with a number value appended to it, differentiating it from its source.

You can also select the target item and press the **Ctrl**+**D** keyboard shortcut to quickly duplicate items. The duplicate item is an identical copy of the source item, inheriting all its attributes. You can then modify or transform the duplicate to make it a unique object.

When you need a greater number of duplicates, use any of the Clone and Array commands found in the **Duplicate** tab of the Modo toolbox. The same options are also available in the menu bar under **Geometry** > **Duplicate**. For more information, see Duplicating Items.

Replicators are also an excellent way to generate many duplicates of an item providing a procedural means to clone objects over a surfacing by specifying another item that acts as a point source. For more information on using Replicators, see Replicator Item.

Modo also allows you to create instances of items. Instances are virtual duplicates that require no additional memory as they are only references back to their source, and update automatically when any component edits are applied to the source geometry. You can position, scale and rotate instances as if they were their own item, but you cannot make direct component level edits to them. Many of the Clone and Array tools provide options to generate instances instead of geometric clones, greatly reducing scene overhead.
Duplicate Options: This command provides you with a few options for the duplicated items. Right-click an item and choose **Duplicate...** to open this menu, or in the menu bar, click **Item** > **Duplicate Options**.



Instance: Enable this checkbox to generate instances instead of duplicates.

Hierarchy: Enable this checkbox to duplicate all child items of the source, retaining their hierarchy.

Modifiers: Enable this checkbox to duplicate any related modifiers associated with the source.

Converting Items

You can convert any type of item into any other type of item from the menu bar by clicking **Item** > **Change Type**, and then selecting from the various item options. This command converts the current selection into the type of item selected.

Another way to do this, is to right-click on a layer opening the context menu and select the **Change Type** option. This opens another menu from which you can choose the item type you wish to convert to. The standard transform properties of the item (position, scale and rotation) are retained, but in unrelated items, the converted object is based on default item values.

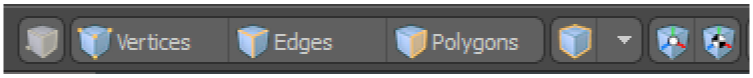
Changing a Mesh Item into a light can be useful from time to time, but the more important function lies in converting heavy Mesh Items into static mesh. Since a converted Mesh Item is frozen and triangulated, it is not always desirable, but in cases where items are too large to manage otherwise, it is a good option to have available. For more information on static meshes, see Static Mesh Item.

Removing Items

To remove an item, right-click on it in the **Item List** or the **Shader Tree** and select the **Delete** command from the context menu. You can also select the target items and in the menu bar, click **Edit** > **Delete**. Finally, you can remove selected items by simply pressing **Delete** on the keyboard.

Understanding Items vs. Components

An areas that is often confusing to new users of Modo is the various selection modes, which is a crucial concept to understand. When in their respective selection modes, only selections of that type can be made, keeping in mind that when nothing specific is selected, then everything in a layer is selected.



The first button is Auto Select used to automatically change the component mode. Followed by **Vertices**, **Edges** and **Polygons**, which are called the component modes. Click any of the interface buttons to enter these modes and press the space bar to toggle through them sequentially. It can also be handy to know the mode keyboard shortcuts: pressing **1**, **2** or **3** respectively enters **Vertices**, **Edges** or **Polygons** modes. The active component button is displayed in orange. You can enter the **Items** mode automatically by deselecting the active component mode button directly within the interface. Selected objects in the 3D viewport are displayed using a full shader to assist you in quickly identifying that you are in **Items** mode. Alternatively, pressing **5** enters **Items** mode. These aren't the only modes, but they are the most important ones.

You can transform (move, scale or rotate) geometry in any of these modes. When applying these kinds of changes, the big difference between what happens with each lies in the layer's center point. When applying a transform in a component mode, you edit the raw geometry that makes up any model. These are called component level transforms. Regardless of the position of the geometry in the Mesh Item layer, the center position for that item in the **Properties** panel always remains at the World origin- 0,0,0.

When applying a transform in Items mode, you apply an item level transform. Moving the geometry in 3D space, the values in the item's **Properties** panel change from 0,0,0, moving away from the World origin. You can no longer affect individual vertices, edges, or polygons, as the layer as a whole is being modified. This is how objects are animated (and positioned for layout) in Modo, by applying item level transforms. For more information in the **Mesh Item Properties** panel, see Mesh Item. To sum up, modifications made in the Vertex, Edge and Polygon modes happen around the layers center point, while transforms applied while in **Items mode** move the entire item, including its center point.

It can become confusing as the same tools for move, scale and rotate are used for transforming items in component and **Items mode**, but if you keep an eye out for which mode you are in, usually based on what is being selected, it soon become second nature.

Note: The component modes only apply specifically to Mesh Item layers, other layers such as lights, texture locators, replicators or cameras can only be edited while in **Items mode**.

Other Modes

Besides component and Items modes, there are three other selection modes: Center, Pivot and Material modes. While in the prior two types, you can change the position of a center or pivot relative to the geometry or item, essentially modifying the layers origin position. Because of Specifying Action Centers and Falloffs , the positions of centers and pivots aren't important until an item needs to be animated. Then it becomes very important because the center represents where the animation transforms originate from. Rotating a tire when the center is positioned below the tire would not produce the desired results. But when the tire is already in the proper position relative to a car body, its much easier to just transform the center itself moving it to the proper position at the axle so the wheel can spin smoothly. The **Animate** section has a number of setup tools that can make it much easier to modify geometry and centers relative to each other. In order to manipulate a center or a pivot, you must first select the item while in the particular mode, by clicking on the **Center Selection** or **Pivot Selection** icons in the 3D viewport and then transforming them. This extra step makes it more difficult to accidentally change their respective positions.

The final selection mode is Material mode. While you can select and transform geometry based on Material assignments, it is mostly useful for easily selecting geometry based on particular tag types. You can change precisely what type of tag the mode specifically selects by adjusting the settings in the menu bar, in **Select** > **Polygon Tag Type**. In Materials mode (the default), the selection function doesn't only highlight all the geometry in the scene with a specific material tag, but it also highlights the associated Shader Tree layers making it easier to pinpoint a particular layer in a complex tree. In the **Part** and **Smoothing Group** options, either Part tags or Smoothing Group assignments are selected respectively. For more information, see Polygon Set Material/Part/Smoothing.

Modo provides you with the flexibility to work in the most comfortable way. For example, you can Radial Sweep a tire model around the origin and position it in the wheel well, or you can Radial Sweep the tire in position and adjust the center point after the fact, either way works.

Tip: You should always be aware of an object's center and pivot position, so when issues arise, such as when working with Replicators, it is easy to spot the fix.

Adding Layer Items

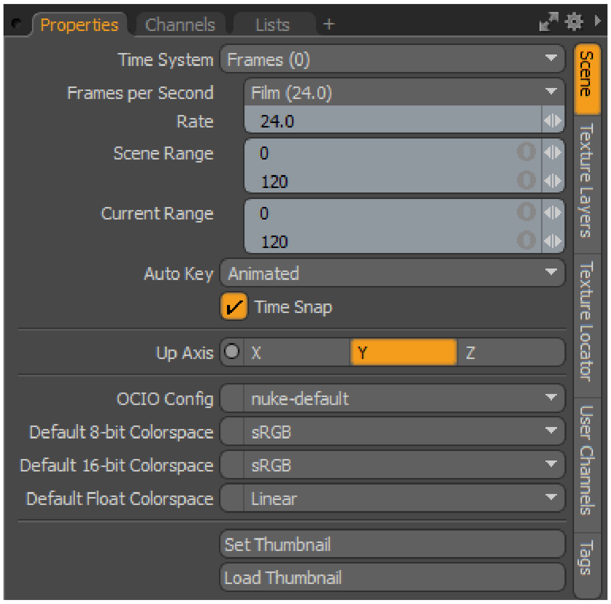
Modo scenes are basically collections of items: cameras, lights, meshes, locators, and so on. They are also referred to as layers. You can tell Modo what particular item you wish to edit by selecting them. When selected, their associated attributes appear in a Properties Panel panel where you can manipulate their values directly. For information on a specific item, click a link.

Icon	Name	Description
🎬	Scene Item	The topmost layer of a Modo file.
⁘	Fractal Gasket	A plug-in item type that generates various geometric arrays of objects.
⚙	Gear Item	A plug-in item type that generates accurate procedural gear and cog shaped geometry.

Icon	Name	Description
	Ground Plane	An item that generates a flat plane.
	Group Locator Item	A locator that does not draw anything in the 3D viewport, eliminating scene clutter.
	Instance Item	A virtual duplicate of a Mesh Item.
	Locator Item	Items serving various functions in animation, such as targets, controllers, parents, and manipulators.
	Mesh Item	The model itself, consisting of vertices, edges, and polygons.
	Replicator Item	Replicators generate render time duplicates of geometry in a scene.
	Static Mesh Item	A non-editable object.

Scene Item

The Scene Item is the topmost layer of any file within Modo: all items within the scene are children of the scene itself. A scene item is always named by the file it represents. To change the name of a scene, you need to use the **Save As** command to assign a new file name to the scene. There are a number of attributes that are associated to scenes. You can select the scene item in the **Item List** and access them in the **Properties** viewport.



Time System: Frames (0)
Frames per Second: Film (24.0)
Rate: 24.0
Scene Range: 0 / 120
Current Range: 0 / 120
Auto Key: Animated
☑ Time Snap
Up Axis: X [✓ Y] Z
OCIO Config: nuke-default
Default 8-bit Colorspace: sRGB
Default 16-bit Colorspace: sRGB
Default Float Colorspace: Linear

Option	Description
Time System	Allows you to specify which standard is used to display individual frame numbers. • **Seconds**: Indicates frames as a passage of time, using the **0.0s** format. • **Frames**: Indicates frames as sequential numbers, using standard numbering format. • **SMPTE**: Specifies frames utilizing a shorthand method **00:00:00:00**; representing hours:minutes:seconds:frames. • **Film Code**: Specifies frames by the length of the film in feet (linear measurement), using **0.0 + 0** format.
Frames per Second	You can specify a frame rate by standard definitions, such as **Film (24 FPS)**, **PAL (25 FPS)**, and **NTSC (29.97 FPS)**. Frames per Second, usually referred to as FPS, is the number of still images that play back in sequence in one second of time.
Rate	The actual frame rate value is displayed here, change this value for specific nonstandard frame rates.

Option	Description
Scene Range	This value specifies the total length for animated scenes.
Current Range	Allows you to temporarily crop out the rest of the scene so you can concentrate on one section. This is useful when you are fine-tuning a single area during animation.
Auto Key	Automatically creates keyframes when items are moved in a scene. The different settings specify how keyframes are created. • **All**: Generates keyframe data for all channels associated with the modified item. • **Animated**: Only keyframes those values that have been edited or modified. • **None**: Disables the auto-keyframe functionality.
Time Snap	Snaps the current time indicator and keyframes to whole frame values.

Note: These setting can also be modified in the Timeline Options dialog. For more information, see Working with the Timeline.

Up Axis	By default, the Y axis represents up in Modo's virtual world. The up axis indicator allows you to specify an alternative axis.
OCIO Config	This option controls which configuration type can be used to define the various colorspaces in the scene (as different config files contain different combinations of possible colorspaces). The initial values in this dialog are defined in **System** > **Preferences** > **Rendering** > **Color Management** for Windows and **Menu** > **Preferences** > **Rendering** > **Color Management** for Mac OS.
Default Colorspace	For the various bit depths of images (8, 16 and 32bit float), this determines the scene-wide working colorspace for all images (per bit-depth). As images are imported into a scene for various purposes (background, textures, and so on) color management is automatically applied to ensure consistent color throughout the workflow. Within each texture layer, under the **Image Still** sub-tab, there is a **Colorspace** setting. By default, this option is set to **Default**, which means that the setting itself defers to the scene-wide setting defined here in the scene item. You can also change from the **Default** value within the texture item to another colorspace, which then overrides the scene-wide designation. For more information on color management, see Color Management.
Set Thumbnail	Sets a thumbnail image to use in the current scene. You can use images from the

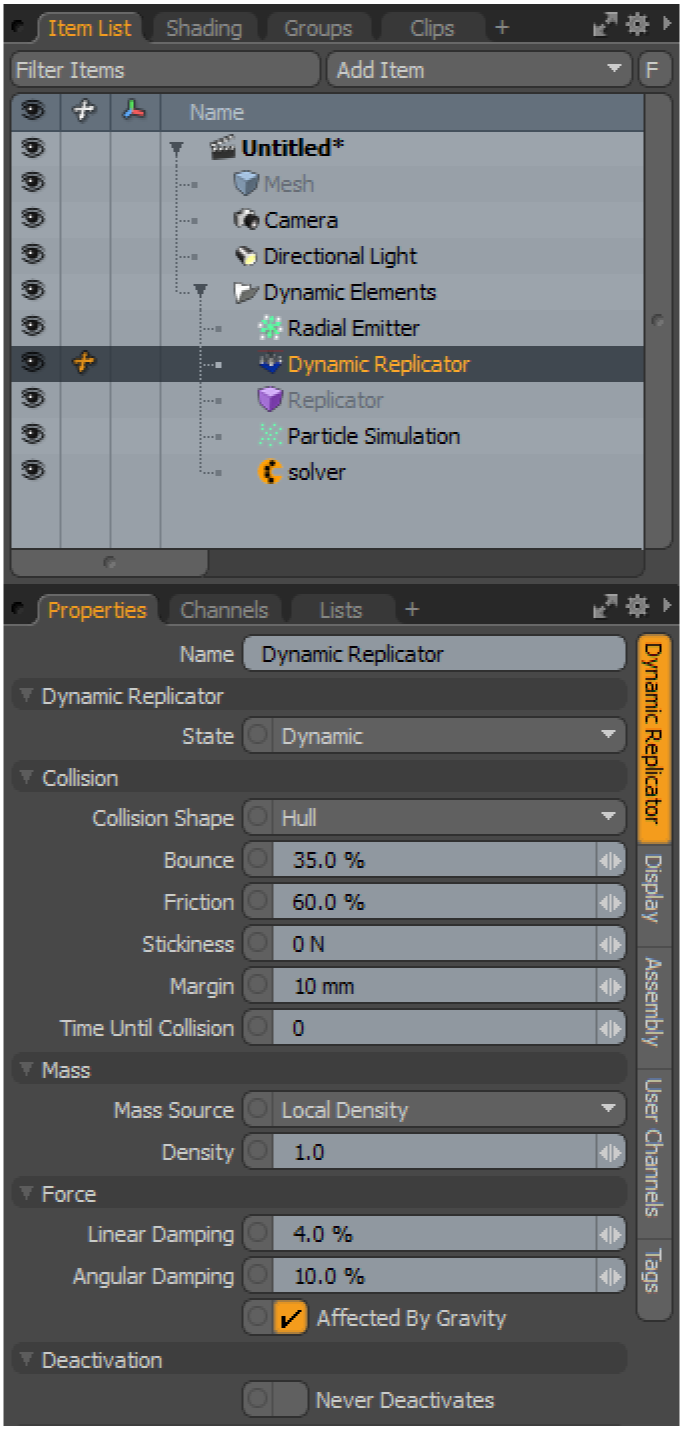
Option	Description
	Preview Display (render buffer), the Render Display, or the current 3D View as the source for the thumbnail image. **Note:** Consider the size of the 3D View or any external images before setting them to use as thumbnails. Large images and 3D Views can greatly increase the **LXO** file size. **Note:** Setting the thumbnail for a scene causes the Preset Browser to refresh and your changes are not immediately visible. You must save your scene after adding a thumbnail image. The following option is available: • **Image Source** - Select one of the following. • **3D View** - Takes a snapshot of the current 3D View and embeds that into the **LXO** file. Your scene updates every time the file is saved. If no 3D View is available, Modo uses whatever thumbnail already exists. If no 3D View is available and no previous thumbnail exists, the standard **LXO** icon is used. • **Preview** - Takes a snapshot of the current rendering done in the Preview Viewport and embeds that into the **LXO** file.. • **Render** - Takes a snapshot of the current rendering done in the Render Display. When the file is saved, Modo uses the most recent render. The render buffer image is constrained to fit within 256 x 256 pixels.
Load Thumbnail	Loads a thumbnail image from a file.

Dynamic Replicator

The Dynamic Replicator, when connected to a Particle Simulation and a Replicator Item, allows the spawned replicas to act like Active Rigid Bodies. This can be useful for a number of cases, such as filling up a large container of objects, or in other cases where large numbers of dynamic items are required.

Usage

To create a Dynamic Replicator, you first need to create a particle simulation of some sort. You can use any of the various Particle Emitters available. Next, a Replicator needs to be created, this can be added in the **Items** list using the **Add Items** function under the **Particles** > **Replicator** menu. Once added, the particle simulation is designated as the **Particle Source** in the Replicator Item's **Properties** panel, and the target Mesh Item (or a group of items, for variations) designated as the **Prototype** option. Once defined, select the Replicator Item in the **Items** list and then in the **Dynamics** toolbox, tagged as **Active Rigid Body** by clicking the toolbox button of the same name. This adds a solver item (if not already present), and the Dynamic Replicator Item to the scene. Selecting the Dynamic Replicator in the **Items** list shows the following attributes in the **Properties** viewport.


Item List
Shading
Groups
Clips
Filter Items
Add Item
F
Name
Untitled*
Mesh
Camera
Directional Light
Dynamic Elements
Radial Emitter
Dynamic Replicator
Replicator
Particle Simulation
solver
Properties
Channels
Lists
Dynamic Replicator
Display
Assembly
User Channels
Tags

Name: Dynamic Replicator

Dynamic Replicator

State: Dynamic

Collision

Collision Shape: Hull

Bounce: 35.0 %

Friction: 60.0 %

Stickiness: 0 N

Margin: 10 mm

Time Until Collision: 0

Mass

Mass Source: Local Density

Density: 1.0

Force

Linear Damping: 4.0 %

Angular Damping: 10.0 %

☑ Affected By Gravity

Deactivation

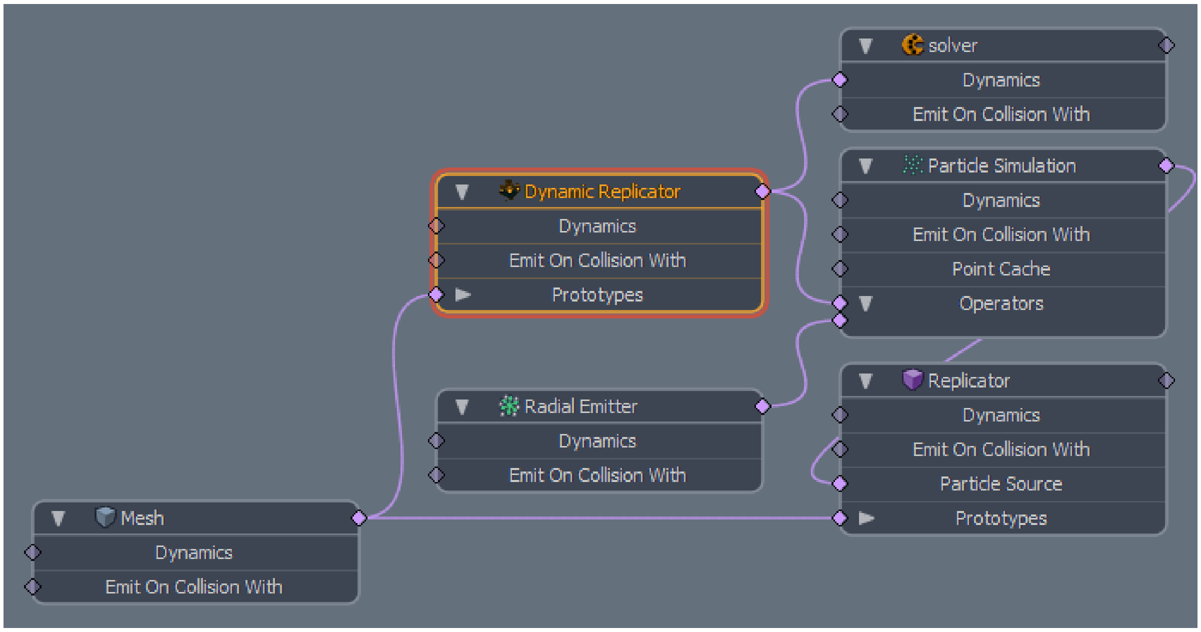
☐ Never Deactivates

Option	Description
Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.

Option	Description
Dynamic Replicator	
State	Determines how the spawned dynamic items are evaluated for collisions. **Dynamic** means that they are treated like Active Rigid Bodies, with inter-object collisions as well as collisions between other dynamic items. **Kinematic** means they are treated like Kinematic Rigid Bodies, no inter-object collisions between elements within the same Replicator, but they collide with other dynamic objects in the scene, acting on them much like a force does.
Collision	
Collision Shape	Defines the shape used to calculate collision contacts between both replicated elements and other dynamic objects within the scene. It is best to choose the simplest shape that produces satisfactory results. The default collision shape is set to **Hull** which generally balances accuracy with calculation speed. • **Box**: A rectangular box defined by the bounding box of the geometry's maximum extents. • **Sphere**: A sphere sized to encompass all the geometry of the object. • **Hull**: The default collision option, creates the smallest convex volume that encompasses all the points in this mesh. To visualize it, imagine the mesh enveloped in shrink-wrap. The **Hull** collision type does not support holes or indentations in the object. While only approximating the item's shape, it provides very fast collision detection on complex shaped objects. • **Mesh**: Uses the actual mesh of the object for calculating collisions. For subdivision surfaces models, the actual frozen Sub-D cage is used. Avoid this option on Dynamic Rigid Bodies (it is fine for Static and Kinematic). When used otherwise, performance, as well as simulation stability can suffer. • **Convex Decomposition**: Creates multiple Hulls (described above) and connects them together. This works to better approximate the actual shape of the geometry but can be costly to initially compute on complex shapes. • **Plane**: Generates a infinite ground plane collision shape originating from the center of the associated geometry's bounding box, not on the item's actual center position.

Option	Description
Bounce	This is the collision response when this object hits another dynamic object. A value of **0%** means the object doesn't bounce or inherit energy from other dynamic objects it contacts. A value of **100%** means the object inherits all of the energy from the contact object. Values above 100% produce rubber ball-like effects, where objects impart more energy from contacts.
Friction	The amount an object slides against another object. A value of **0.0** offers no resistance, while a value of **1.0** keeps the object from sliding entirely.
Stickiness	When two dynamic elements collide, both with **Stickiness** values defined, they attach to each other at the collision point. The higher the **Stickiness** value, the stronger the bond between the elements. Lower values allow stuck elements to be broken more easily.
Margin	This value helps the bullet engine to determine collisions, as well as improve its performance and reliability. It defines a gap around the object used to detect a collision contact. In most cases it should be kept at its default value. When used with the **Mesh** and **Convex Decomposition** settings, this can offset items away from each other, so only a small amount is necessary. **Note:** A setting of **0** (zero) may produce errors in collision detection.
Time Until Collision	This option determines the point in time from when the Replicator Item is created until when collisions are actually calculated for the object. This allows the object to spawn without causing immediate collision issues. Time is defined by frames.
Force	
Linear Damping	This is a force that acts upon the motion of objects, acting to slow them down over time. A small value here can benefit object stability, since it helps objects come to rest.
Angular Damping	Same as **Linear Damping**, but applies to the rotation of the object.

Option	Description
Affected By Gravity	Toggle gravity's global influence on or off for specific items.
Mass	
Mass Source	When the **Mass Source** is set to **World Density**, the mass of an object is calculated by multiplying the calculated volume of the dynamic item by the **Global Density** value defined on the Dynamics Solver item. You can override that value by setting the **Mass Source** to **Local Density** or **Local Mass** and expressly setting a value.
Density	The measure of the amount of matter in any given volume. If the **Mass Source** option is set to **Local Density**, this is the density value that is used. The mass of the object is determined by multiplying the computed volume by this value.
Deactivation	
Never Deactivates	When this option is enabled, the item never gets deactivated due to no motion. When disabled, the item is disabled when it meets the defined linear and angular thresholds defined in the solver item (or overridden here).


solver
Dynamics
Emit On Collision With
Particle Simulation
Dynamics
Emit On Collision With
Point Cache
Operators
Dynamic Replicator
Dynamics
Emit On Collision With
Prototypes
Replicator
Dynamics
Emit On Collision With
Particle Source
Prototypes
Radial Emitter
Dynamics
Emit On Collision With
Mesh
Dynamics
Emit On Collision With

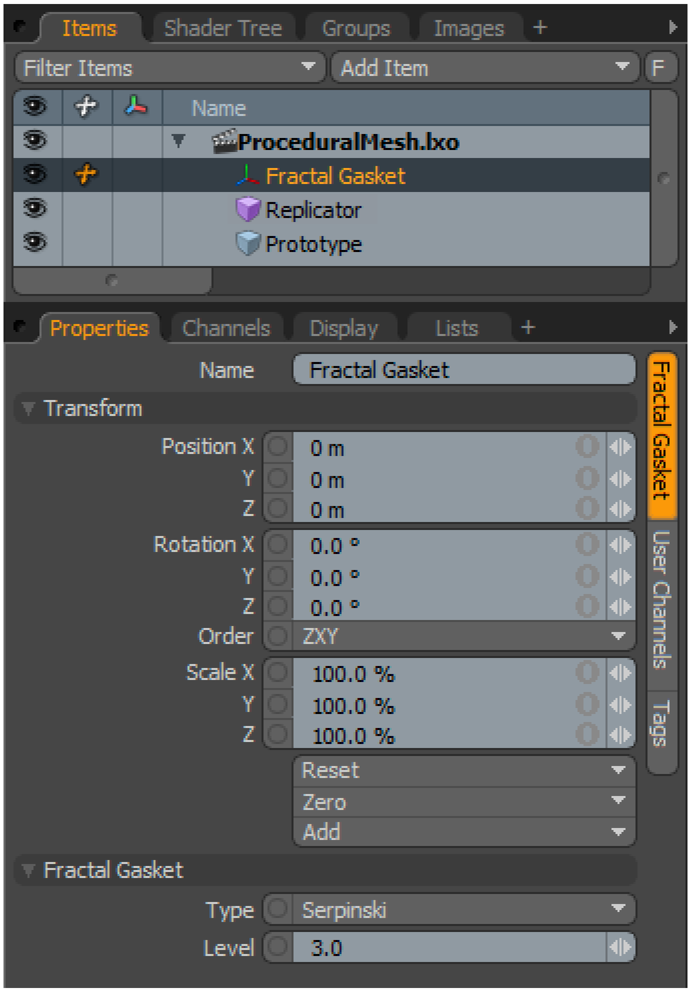
When following the procedure outlined in the Usage section, the Dynamic Replicator Item is added to the scene automatically when the Replicator Item is tagged as an Active Rigid Body. If you prefer to work in a more Schematic-centric workflow, you can manually add the Dynamic Replicator Item to the **Schematic** viewport.

1. Click the **Items** list's **Add Item** button.
2. Select **Other** > **Dynamic Replicator** from the menu.
3. Then drag and drop, the item to the **Schematic** viewport.
4. A Dynamics Solver also needs to be manually added, if not already present. In the menu bar, click **Dynamics** > **Add Solver**. Add this item also to the **Schematic** viewport.
5. In the **Schematic** viewport, connect the Dynamic Replicator to the Solver under its **Dynamics** input, and connect it to the particle simulation as an **Operator**.

Lastly, the item or group of items needs to be connected as the **Prototype** on the Dynamic Replicator itself. This completes the necessary connections for a basic particle simulation. For more information on working with nodes and the **Schematic** viewport, see Schematic Viewport.

Fractal Gasket

The Fractal Gasket item is a plug-in item type that generates various geometric arrays of objects. Used in combination with a Replicator Item, you would add the Fractal Gasket item and assign it as the point source within the Replicator's properties. You can then adjust the options of the Fractal Gasket item to change the array types and their level of complexity. It can also be used in combination with blobs, volumes and sprites. For more information on these, see Blobs, Volumes and Sprites.


Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
ProceduralMesh.lxo
Fractal Gasket
Replicator
Prototype
Properties
Channels
Display
Lists
Fractal Gasket
User Channels
Tags
Transform
Reset
Zero
Add
Fractal Gasket

Name: Fractal Gasket

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

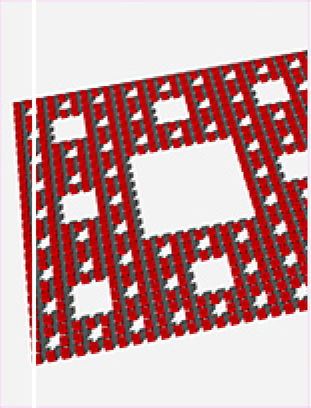
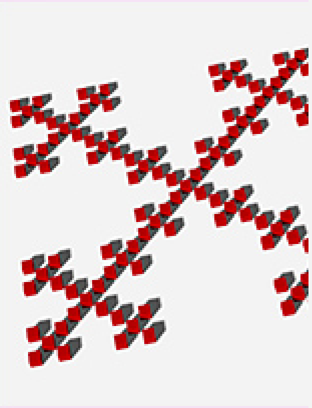
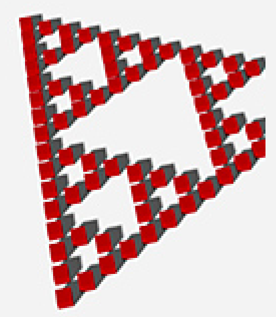
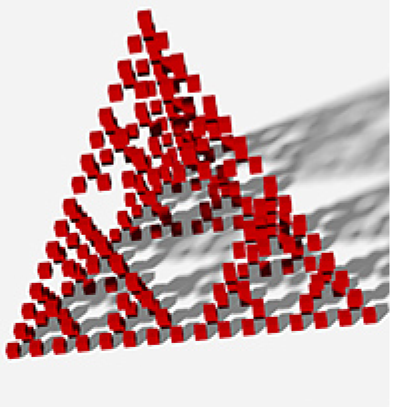
Order: ZXY

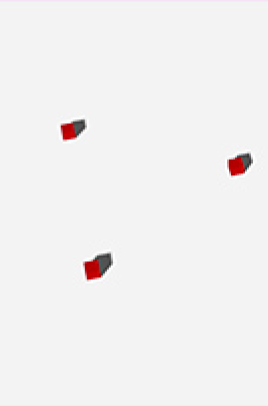
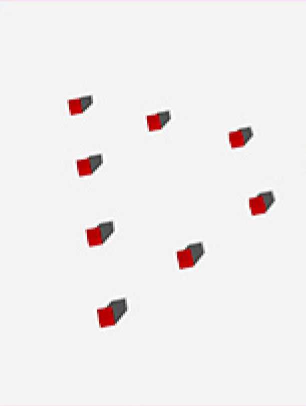
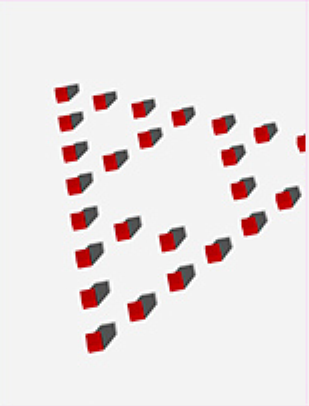
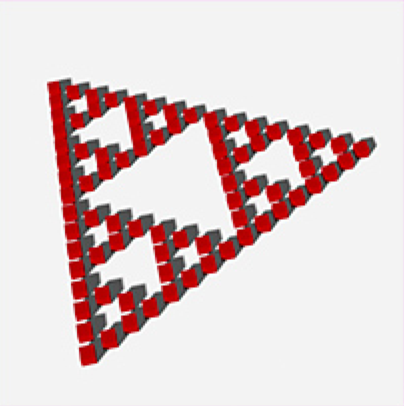
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Type: Serpinski

Level: 3.0

Option	Description
Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space. By default, position transforms originate from the center position.
Rotation	An Item transform that allows you to numerically set the rotation of the item. By default, rotation transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale

Option	Description
	transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and item position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. You can simply transform the target item with one of the various transform tools, or by editing the value input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the channel list while keeping the default **0,0,0** values. A necessary step for referencing, in order to override the channels, they must first exist.
Fractal Gasket	
Type	This option determines the array type. There are four options to choose from. The first three, **Square Hole**, **Square Flake** and **Serpinski** generate flat two-dimensional arrays, while the **Tetrahedron** option generates a three-dimensional array.  Square Hole  Square Flake  Serpinski  Tetrahedron
Level	Determines the complexity of the array with each additional level adding an exponential amount of detail. Each new level increases the **Square Hole** type by a factor of 8, **Square Flake** by a factor of 5, **Serpinski** by a factor of 3 and **Tetrahedron** by a factor of 4. Therefore, care should

Option	Description
	be given to not go overboard when increasing levels, especially with the more complex types.
	 Serpinski Level 2  Serpinski Level 3  Serpinski Level 4  Serpinski Level 5

Tip: To easily change the size of items in a Fractal Gasket Array, you can set the Replicator to **Point Mode** and then scale the size of the prototype object in one of the component modes.

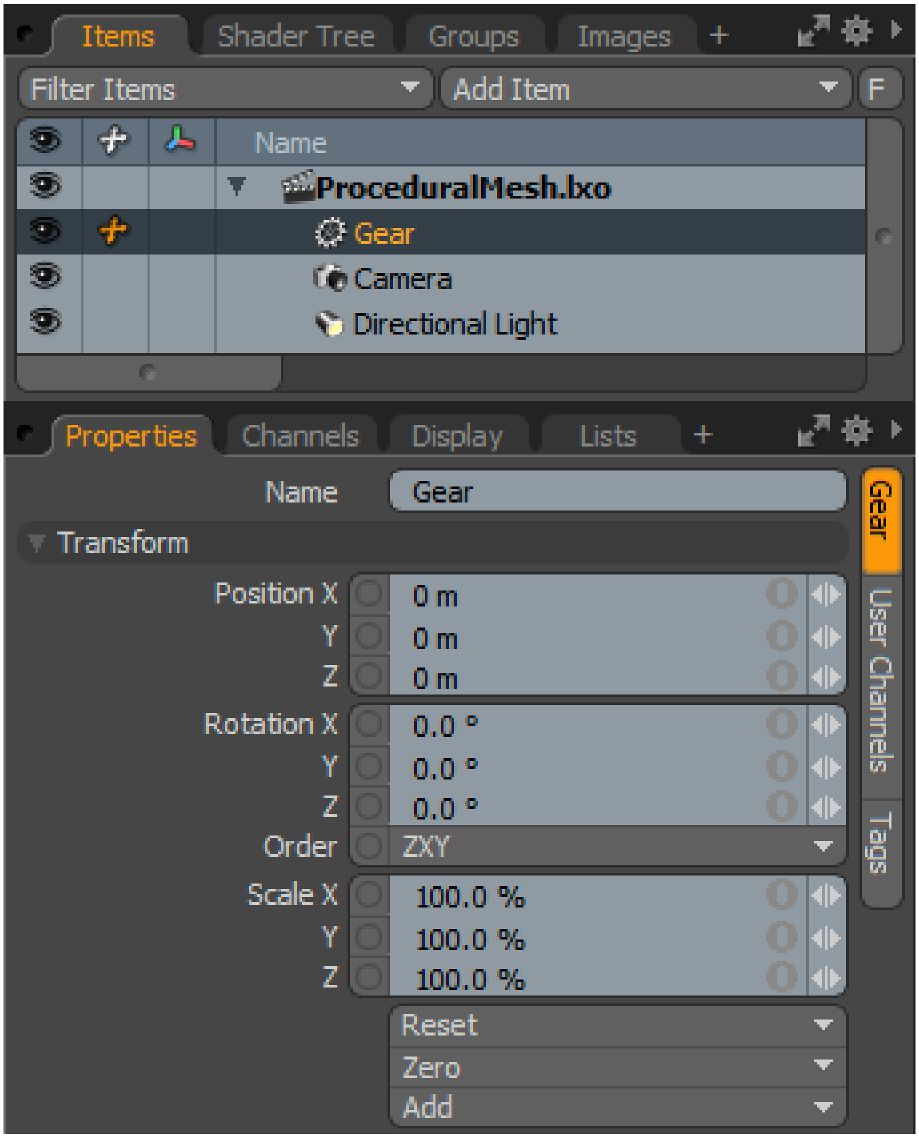
Gear Item

The gear item is a plug-in item type that generates accurate procedural gear and cog-shaped geometry with ease. Since it is procedural, the many channels can be animated, changing the design and look of the gear over time.

To add a gear item to the scene, click **Add Item** button within the **Items** list. You can also convert the procedural gear to a Mesh Item by right-clicking on the item layer in the **Items** list and selecting the **Change Type** > **Mesh** option from the pop-up menu. Select the item to display its associated attributes in the **Properties** panel. Adjust the options to customize the look and level of complexity.

Gear Item Properties

Transform

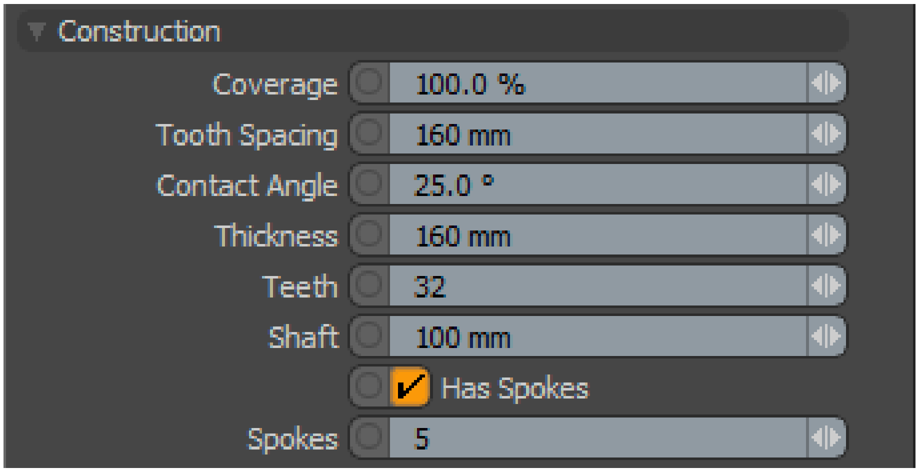

Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
ProceduralMesh.lxo
Gear
Camera
Directional Light
Properties
Channels
Display
Lists
Transform
Reset
Zero
Add
Gear
User Channels
Tags

Name: Gear
Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.
Transform	

Position	An item transform that allows you to numerically position the item in XYZ space. By default, position transforms originate from the center position.
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, rotation transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)**, returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and item position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. You can simply transform the target item with one of the various transform tools, or by editing the value input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the channel list while keeping the default **0,0,0** values. A necessary step for referencing, in order to override the channels, they must first exist.

Construction



Construction

Coverage 100.0 %

Tooth Spacing 160 mm

Contact Angle 25.0 °

Thickness 160 mm

Teeth 32

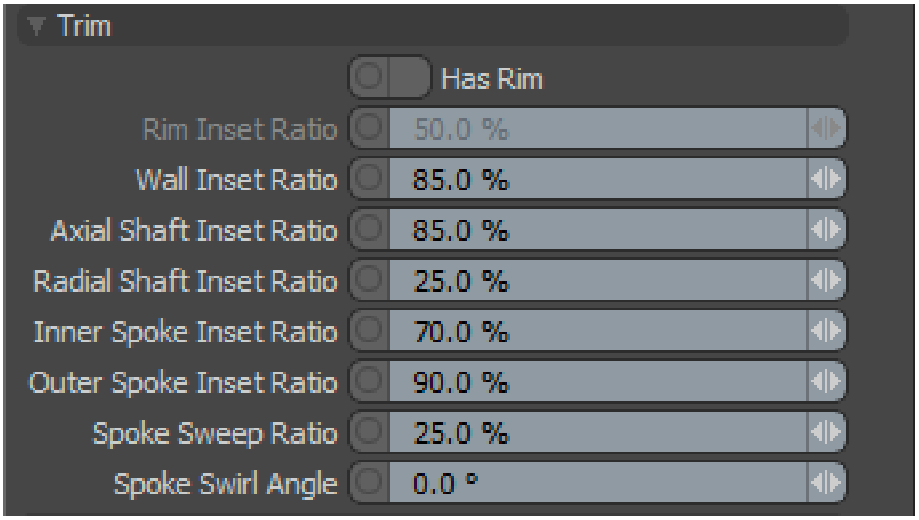
Shaft 100 mm

☑ Has Spokes

Spokes 5

Construction	
Coverage	Percentage of constructed shape.
Tooth Spacing	Space between each tooth.
Contact Angle	Angle of each tooth (up to 45 degrees).
Thickness	Thickness of the gear.
Teeth	Number of teeth for the gear.
Shaft	Size of the shaft.
Has Spokes	Determines whether the interior of the gear has spokes.
Spokes	Number of spokes in the interior of the gear.

Trim



Has Rim

Rim Inset Ratio 50.0 %

Wall Inset Ratio 85.0 %

Axial Shaft Inset Ratio 85.0 %

Radial Shaft Inset Ratio 25.0 %

Inner Spoke Inset Ratio 70.0 %

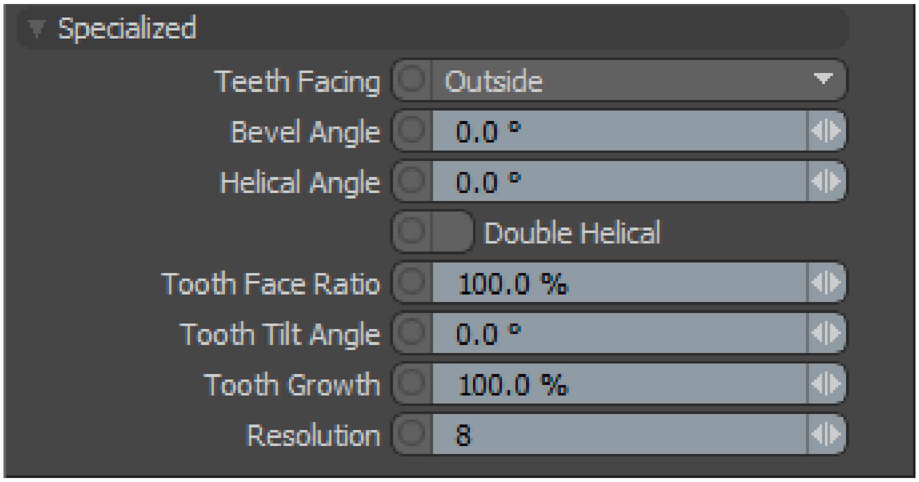
Outer Spoke Inset Ratio 90.0 %

Spoke Sweep Ratio 25.0 %

Spoke Swirl Angle 0.0 °

Trim	
Has Rim	If the interior of the gear has a rim.
Rim Inset Ratio	Ratio of rim thickness to total thickness.
Wall Inset Ratio	Ratio of wall offset amount.
Axial Shift Ratio	Ratio of gear wall to gear radius.
Radial Shaft Inset Ratio	Ratio of shaft rim to gear radius.
Inner Spoke Inset Ratio	Ratio of inner spoke thickness to total thickness.
Outer Spoke Inset Ratio	Ratio of outer spoke thickness to total thickness.
Spoke Sweep Ratio	Ratio of spoke sweep to spoke spacing.
Spoke Swirl Angle	The angle by which the spokes should be swirled.

Specialized



Specialized

Teeth Facing: Outside

Bevel Angle: 0.0 °

Helical Angle: 0.0 °

☐ Double Helical

Tooth Face Ratio: 100.0 %

Tooth Tilt Angle: 0.0 °

Tooth Growth: 100.0 %

Resolution: 8

Specialized	
Teeth Facing	The direction in which the teeth are facing.
Bevel Angle	Angle of tooth bevel.
Helical Angle	Angle of tooth helix.
Double Helical	Double helical (herringbone) pattern.
Tooth Face Ratio	Ratio of tooth face to thickness.
Tooth Tilt Angle	Angle of tool tilt along the rotational axis.
Tooth Growth	Amount of tooth growth.
Resolution	Overall geometry resolution, higher values produce smoother shading, but create more polygons at render time.

Ground Plane

The ground plane is a procedurally created item that generates a flat plane based on a user-defined size. In order to apply surfacing to a ground plane (since there is no actual geometry that you can select and tag), you need to employ an item mask.

Note: Since the ground plane is a volumetric effect, it does not render as quickly as geometry would.

Adding a Ground Plane

To add a ground plane item, go to the **Items** list on the right side of the screen and click **Add Item**, then select **Volumes** > **Ground Plane** option.

In order to apply any surfacing to the plane, you need to create an item mask. Right-click on the item layer in the **Items** list to open the contextual menu, and select **Create Item Mask**. This creates a Material Group Item with the ground plane item specified in its grouping options. Now any textures added into this group apply exclusively to the ground plane (the sample above has a checkerboard texture applied). For more information on adding and working with Shader Tree layers, see Shader Tree.

Transform Properties

Name	This displays the current item's name. You may change it by clicking the current name and typing a new name.
Position XYZ	Specify to position the item numerically in XYZ space. Position transforms originate from the Center position.
Rotation XYZ	Specify to set the rotation of the item numerically. Rotation transforms originate from the Center position.
Order	Sets the order that Modo applies rotations to the falloff item. By changing the order that Modo applies rotations you can sometimes reduce or eliminate gimbal lock.
Scale XYZ	Sets the size of the item numerically. Scale transforms originate from the Center position.
Reset	Resets the selected transform values to (0,0,0) and returns the items back to their default state.
Zero	Resets the selected transform property value to **0** and leaves the Center position and Item position intact.
Add	Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for Referencing. To be able to override a channel, it must first exist.)

Note: Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function (as given here), you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Ground Plane Properties

Size	Determines the overall dimensions of the plane as a square (width and depth are the same).
Ground	This option toggles the rendering of the ground plane itself. When enabled, Modo draws the plane, when disabled, it doesn't.
Clipping	When this option is enabled, any geometry above the plane is clipped (doesn't render). The clipped areas take on the surfacing of the plane itself.
Fog	Renders a Ground Fog effect when enabled.
Fog Height	Determines the maximum level for the fog effect offset the height distance from the plane. The fog is thickest at the ground level, based on the **Density** value, attenuating up to the **Fog Height** value.
Near Clip	Determines the closest point to the camera that the fog effect renders.
Far Clip	Determines the furthest point from the camera that the fog effect renders.
Density	Determines the overall thickness or opacity of the fog effect.
Sample Rate	Determines the quality of the sampling of the fog effect, high values produce cleaner results at the expense of longer render times.

Note: Because of its procedural nature, neither bump maps, nor displacement maps render properly on a ground plane. For those you need to use an actual geometry plane.

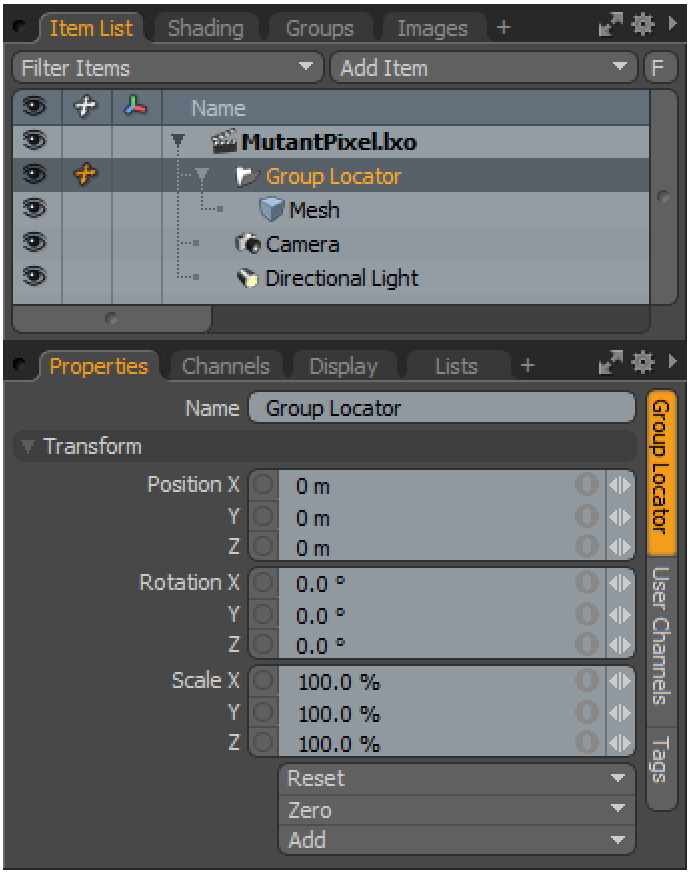
Group Locator Item

As a project becomes more complex, you may find that the flat listing of layers in the **Items** list becomes unwieldy to work with. Parenting items to a group locator helps to organize the project. Different from an actual group, group locators are simply a special kind of Locator Item that does not draw anything in the 3D Viewports, eliminating additional scene clutter. Group locators can be easily collapsed within the **Items** list, which keeps the viewport clean and manageable. Group locators have an icon that looks like a folder in the **Items** list to distinguish them from regular locator items. In all other aspects they act the same as a normal locator, for example, they can be moved and rotated in the 3D Viewports.

To create a group locator, in the **Items** list, click **Add Items** and choose **Locators** > **Group Locator** from the dropdown menu. Once created, you can begin to drag and drop items into a hierarchy, organizing the list.

You can parent several items to a group locator by selecting all the target layers in the **Items** list.

1. Hold **Ctrl** and click on the items you want to select.
2. Once selected, press **Ctrl**+**G** to create a new group locator, with all selected items parented in one step.
   Then you can click the small arrow (►) preceding the layer to toggle the visibility of the child items.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Group Locator
Mesh
Camera
Directional Light
Properties
Channels
Display
Lists
+
Transform
Reset
Zero
Add
Group Locator
User Channels
Tags

Name: Group Locator

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Option	Description
Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space.
Rotation	An item transform that allows you to numerically set the rotation of the item.
Scale	An item transform that allows you to numerically set the size of the item.
Reset	Resets the selected transform values to **(0,0,0)**, returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and item position intact by adding a negative item transform to the channels list.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. You can simply transform the target item with one of the various transform tools, or by editing the value input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the channel list while keeping the default **0,0,0** values. A necessary step for referencing, in order to override the channels, they must first exist.

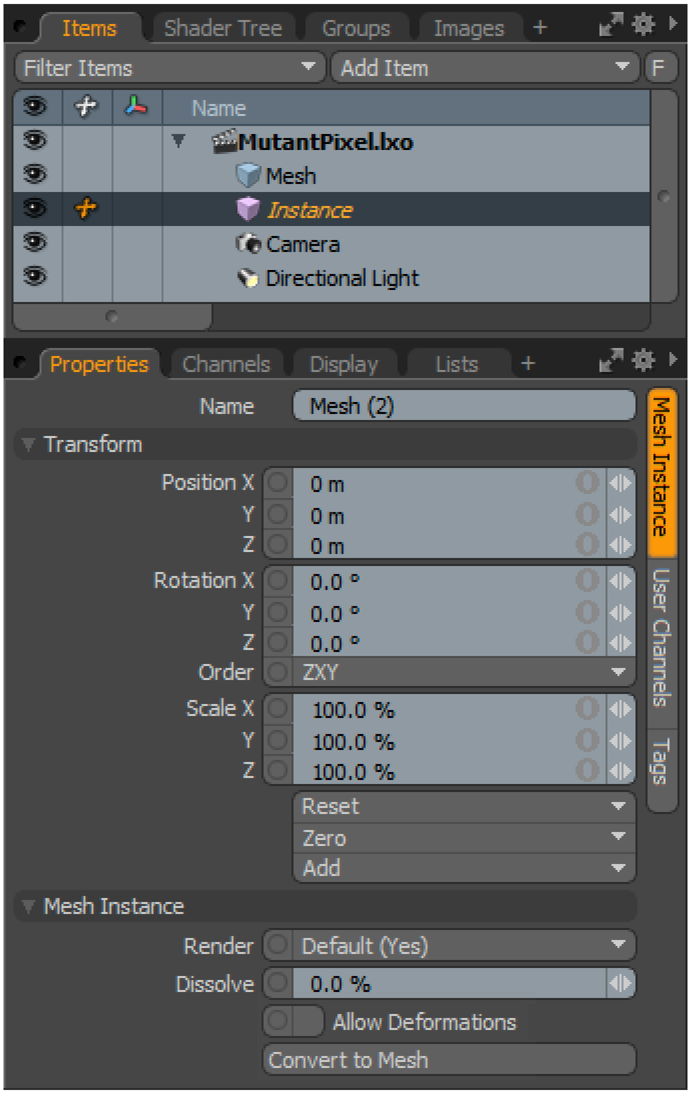
Instance Item

Instances are virtual duplicates of a Mesh Item in Modo, this means that they do not add any additional geometry to the scene. They are merely containers that reference the original item at render time. As such, instance meshes can only be edited at the item level. Component level edits must be applied to the original item, where changes propagate to all instances.

You can create instances in several ways. When in **Items mode**, many of the duplication tools allow for instance creation. The simplest method is to right-click on a Mesh Item in the **Items** list and choose **Instance** from the dropdown menu. This adds a new instance item to the bottom of the **Items** list.

Tip: For more information on using instances, see Instances.

The link between instances and the original always remains live. If you wish to convert an instance mesh into a standard Mesh Item, you can right-click the instance in the **Items** list and choose **Change Item** > **Type** > **Mesh** or simply use the **Convert to Mesh** button in the item's properties.


Items
Shader Tree
Groups
Images
+
Filter Items
Add Item
F
Name
Properties
Channels
Display
Lists
+
Mesh Instance
User Channels
Tags
Reset
Zero
Add
Convert to Mesh

MutantPixel.lxo
Mesh
Instance
Camera
Directional Light

Name: Mesh (2)

Transform

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Mesh Instance

Render: Default (Yes)
Dissolve: 0.0 %
○ Allow Deformations

Option	Description
Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space. By default, position transforms originate from the center position.
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, rotation transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)**, returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and item position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. You can simply transform the target item with one of the various transform tools, or by editing the value input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the channel list while keeping the default **0,0,0** values. A necessary step for referencing, in order to override the channels, they must first exist.
Mesh Instance	

Option	Description
Render	This dropdown menu allows you to select from three options. When set to **Default**, you can enable/disable Mesh Items using the visibility column (👁) of the **Items** list. When the mesh layer is visible, it contributes to the final rendered scene and when invisible, it doesn't. In some cases you may prefer to fix this state, setting the mesh as **On** (enabled) or **Off** (disabled) regardless of visibility. This is useful for workflows that auto-toggle visibility: you don't have to manually enable mesh layers for test renders.
Dissolve	Controls the visibility of an item layer. At 0% the item renders normally, utilizing the Shader Tree settings. Values above 0% fade the visibility of the item layer ramping toward 100%, where the item becomes completely invisible.
Allow Deformations	When this option is enabled, you can apply deformers to the instance, independent of the source mesh. When disabled, the instance inherits any deformations applied to the source.
Convert to Mesh	Converts a mesh instance into a fully editable mesh, removing the link between the instance and the source. **Note:** Converting instances to meshes increases file size.

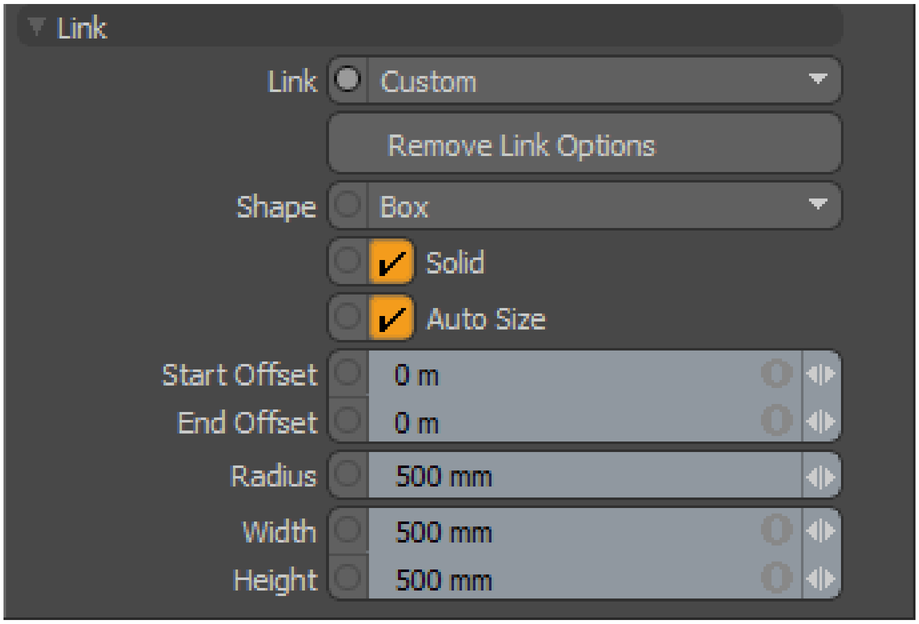
While instance items derive all mesh information from the original, you can create custom material and texture settings for an instanced item utilizing item masks, which can in turn be specified in the Material Group item.

Locator Item

Locators are like the Swiss army knife of 3D; playing a key role in animation, it serves many important functions: targets, controllers, parents, manipulators, selectors, and even joints. Since locators are just items, they have their own set of properties, the most important being their own transforms, and as such, they can be positioned and animated independently of Mesh Items in a scene. This might not seem so exciting by itself, as locators don't even render, but when working with constraints, modifiers, and influences, you quickly discover how useful they really are, serving as the virtual backbone of any rigging setup. As such, there are some special properties associated specifically with locators. Additional draw properties specific to

Locators are available in the **Display** sub-tab. Pick Walking and Selection Redirect settings are available via the **Assembly** viewport.

You can add a Locator to a scene using the **Add Item** button, found in the top-right corner of the Items (Scenes) List viewport. Additionally, you can use the menu bar command **Item** > **Create Locator**. Once selected, the locator's attributes become available in the **Properties** viewport.



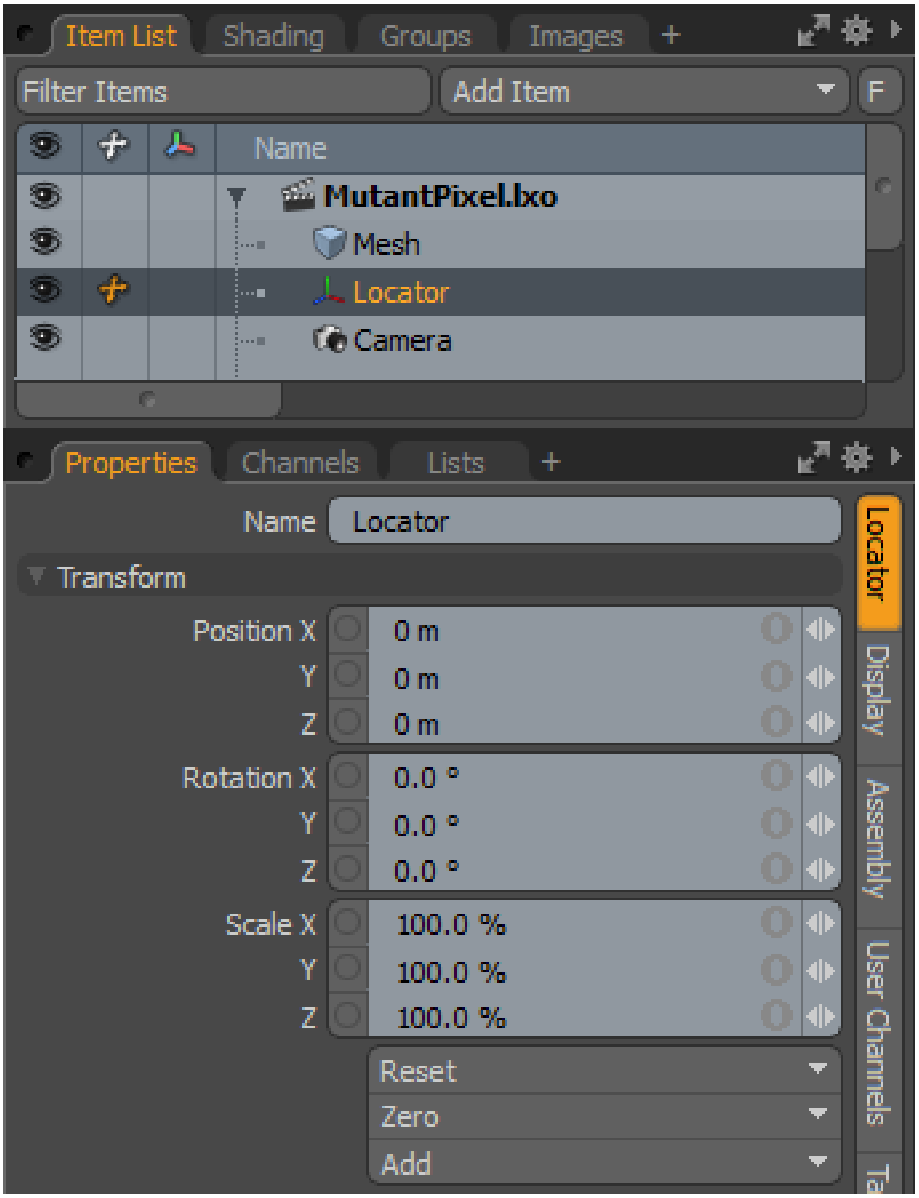
Link: Custom
Shape: Box
☑ Solid
☑ Auto Size
Start Offset: 0 m
End Offset: 0 m
Radius: 500 mm
Width: 500 mm
Height: 500 mm

Tip: To improve your workflow while working with User Channels, use the **Channel Haul** tool to toggle the visibility of all channels available for a selected item. For more information, see Displaying All Available User Channels in the 3D Viewport

Locator Options

Name: This data field displays the current item name. You can change it by clicking into the field and typing the new name.

Locator: Transform Properties


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Locator
Camera
Properties
Channels
Lists
+
Locator
Display
Assembly
User Channels
Ta
Reset
Zero
Add

Name: Locator

Transform

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Position: An item transform that allows you to numerically position the item in XYZ space.

Rotation: An item transform that allows you to numerically set the rotation of the item.

Scale: An item transform that allows you to numerically set the size of the item.

Reset: Resets the selected transform values to **(0,0,0)**, returning the items to their default state.

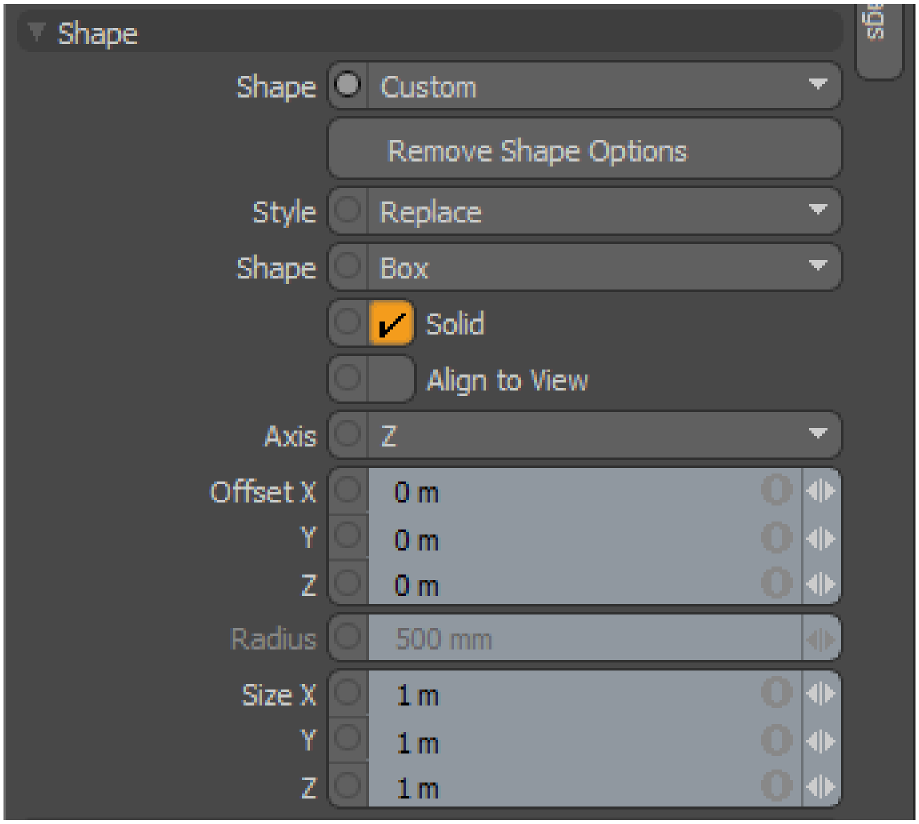
Zero: Resets the chosen transform property values to **0**, leaving the center position and mesh position intact. This is done by adding an additional transform item, with an inverted version of the current values, to the

Mesh Item's channels. This is useful to allow a joint, for instance, to have a base value of 0,0,0, whilst still being located away from the World Origin.

Add: This option adds additional transform items if such items already exist, or creates new ones, if not. Transform items are the channel groups that store the actual transform values, controlling any item's position, rotation, and/or scale. You can add as many transform items as you wish for each transform property. Adding additional transform items produces an additive effect whereby transform groups are evaluated in succession. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

By default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are created on an as-needed basis, reducing scene overhead. There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport. Because of this, you may need to specifically create item transforms when referencing, because in order to override the channels in the master scene, they must first exist.

Locator: Shape Properties


Shape
Shape: Custom
Remove Shape Options
Style: Replace
Shape: Box
☑ Solid
☐ Align to View
Axis: Z
Offset X: 0 m
Y: 0 m
Z: 0 m
Radius: 500 mm
Size X: 1 m
Y: 1 m
Z: 1 m

Shape: Choose between the **Default** cross style, or choose **Custom** to enable additional properties for locator display customization.

Remove Shape Options: This button removes the additional **Shape** options, reverting to the **Default** cross representation.

Style: Controls whether the **Shape** options draw in addition to the default value (**Add**) or if they replace its display (**Replace**).

Shape: Choose from multiple primitive shape options for the locator's representation.

Solid: Toggles between a filled or wireframe representation of the shape.

Align to View: Forces the representation to align with the viewport's camera.

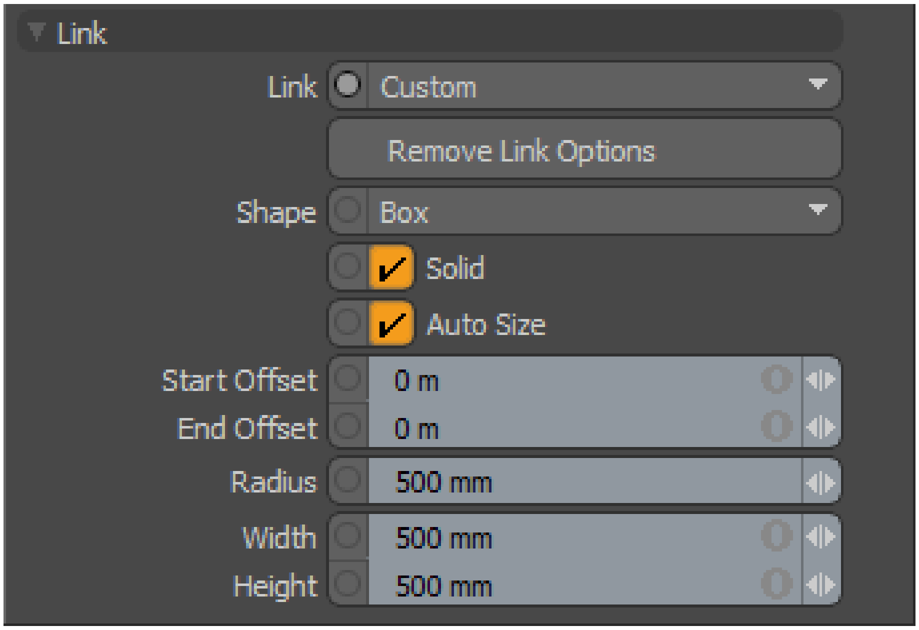
Axis: Chooses the axis for display of the shape representation.

Offset: Adjusts the offset values to position the locator representation.

Radius: Adjusts the size of circular locator representations.

Size: Adjusts the size of rectangular locator representations.

Locator: Link Properties



Link: Custom

Shape: Box

☑ Solid

☑ Auto Size

Start Offset: 0 m

End Offset: 0 m

Radius: 500 mm

Width: 500 mm

Height: 500 mm

Modo can automatically draw a non-rendering element between an item and its child (links always draw to child items in hierarchies). Links are useful in creating visual representation of an item's associations in complex hierarchies.

Link: Choose between the default value **None**, or a simple **Line**, or choose **Custom** to enable additional properties for link display customization.

Remove Link Options: This button removes the additional **Link** options reverting to the default **None**.

Shape: Choose from multiple primitive shape options for the link representation.

Solid: Toggles between a filled or wireframe representation of the shape.

Auto Size: Dynamically scales the link shape depending on the distance between the two items.

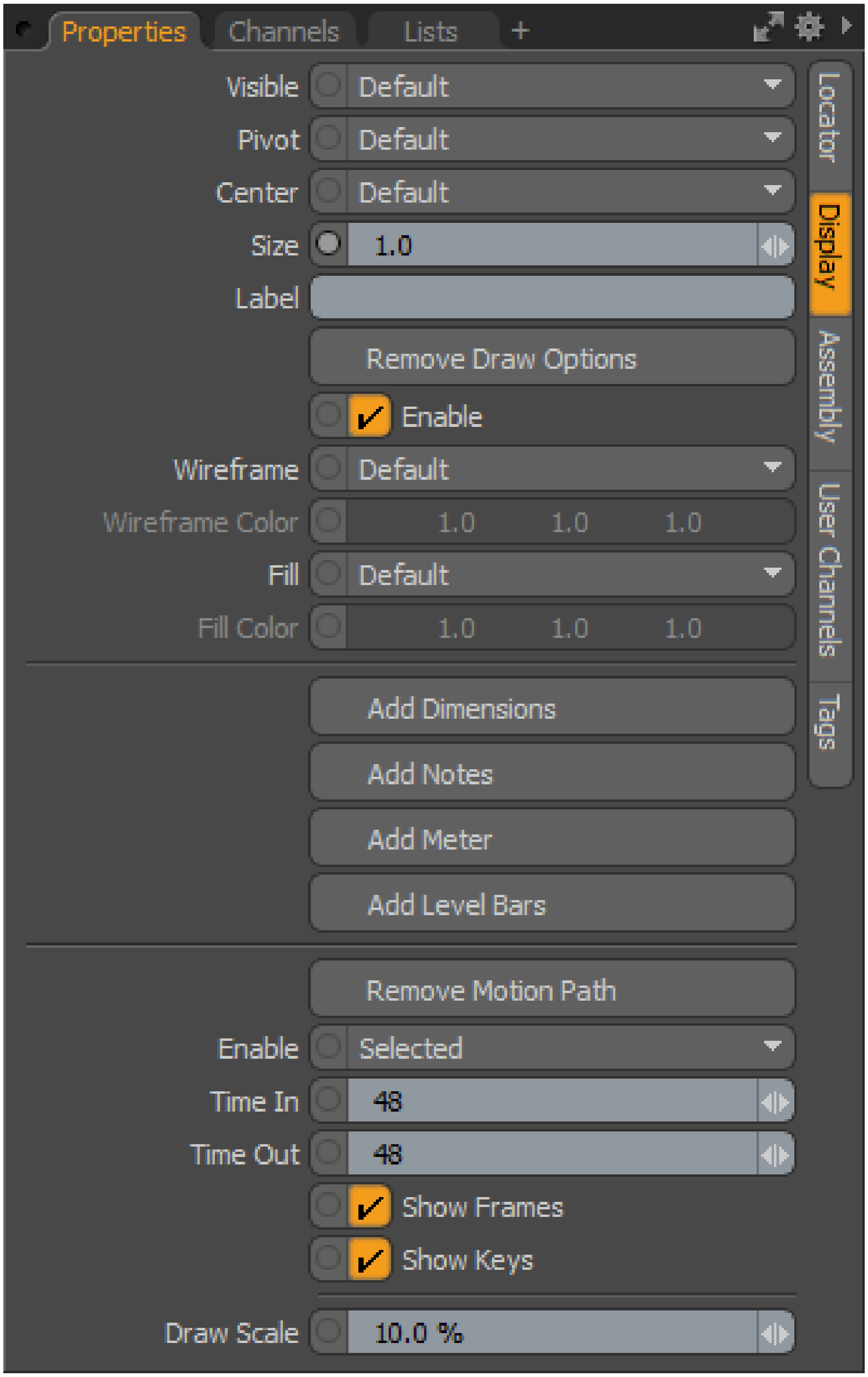
Start Offset: Offsets the start position of the link representation.

End Offset: Offsets the end position of the link representation.

Radius: Adjusts the size of circular link representations.

Width/Height: Adjust the size of rectangular link representations.

Display options


Properties
Channels
Lists
+
Locator
Display
Assembly
User Channels
Tags
Remove Draw Options
Add Dimensions
Add Notes
Add Meter
Add Level Bars
Remove Motion Path

Visible: Default
Pivot: Default
Center: Default
Size: 1.0
Label:
☑ Enable
Wireframe: Default
Wireframe Color: 1.0 1.0 1.0
Fill: Default
Fill Color: 1.0 1.0 1.0
Enable: Selected
Time In: 48
Time Out: 48
☑ Show Frames
☑ Show Keys
Draw Scale: 10.0 %

Visible: Sets visibility for the 3D item itself, **Default** respects viewport settings, **Yes** sets the item as visible always, **No** disables visibility entirely.

Pivot: Sets visibility for the 3D item pivot, **Default** respects viewport settings, **Yes** sets the pivot as always visible, **No** disables pivot visibility entirely.

Center: Sets visibility for the 3D item center, **Default** respects viewport settings, **Yes** sets the center as always visible, **No** disables center visibility entirely.

Size: Multiplier that adjusts the display size of the locator.

Label: Typing in text for the **Label** displays the text positioned next to the locator in the viewport for easy identification. It also adds two additional properties to control the display of the label in the viewport - **Show Label** and **Label Offset**.

Show Label: This toggle temporarily enables or disables the display of the label in the viewport without losing settings.

Label Offset: This value specifies the distance away from the locator where the label appears.

Add/Remove Draw Options: The **Draw Options** control the look of the locator in 3D Viewports, used mostly when **Custom Locator Shapes** are enabled.

Enable: This checkbox toggles the display of the **Draw Options** without losing values.

Wireframe: Adjusts draw color of locator in the 3D viewport.

Wireframe Color: When **Wireframe** is set to **Custom**, the **Wireframe Color** sets the color of the wireframe part of the locator.

Fill: If Custom Draw shape is set to **Solid**, this value controls the color of the fill itself.

Fill Color: When **Fill Color** is set to **Custom**, this value specifies the RGB color of the fill of the **Solid** in the 3D viewport.

Add/Remove Dimensions: When this option is enabled, dimension values for the overall bounding box size appears in the viewport around the Mesh Item. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.

Add/Remove Notes: When this option is enabled, you can define multiple lines of text (up to 9) that show up when the item is selected. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.

Add/Remove Meter: When this option is enabled, you can display an on-screen analog style meter graph (looks like a speedometer) that can be rigged to show individual values in the 3D viewport. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.

Add/Remove Level Bars: When this option is enabled, you can display an on-screen equalizer-type graph for visualizing multiple on-screen values. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.

Note: For more details on display customization, see Additional Draw Options.

Add/Remove Motion Path: To display the motion path for specific items within a scene, you can select the **Add Motion Path** option, revealing additional properties controlling the display of the path in the 3D GL viewport.

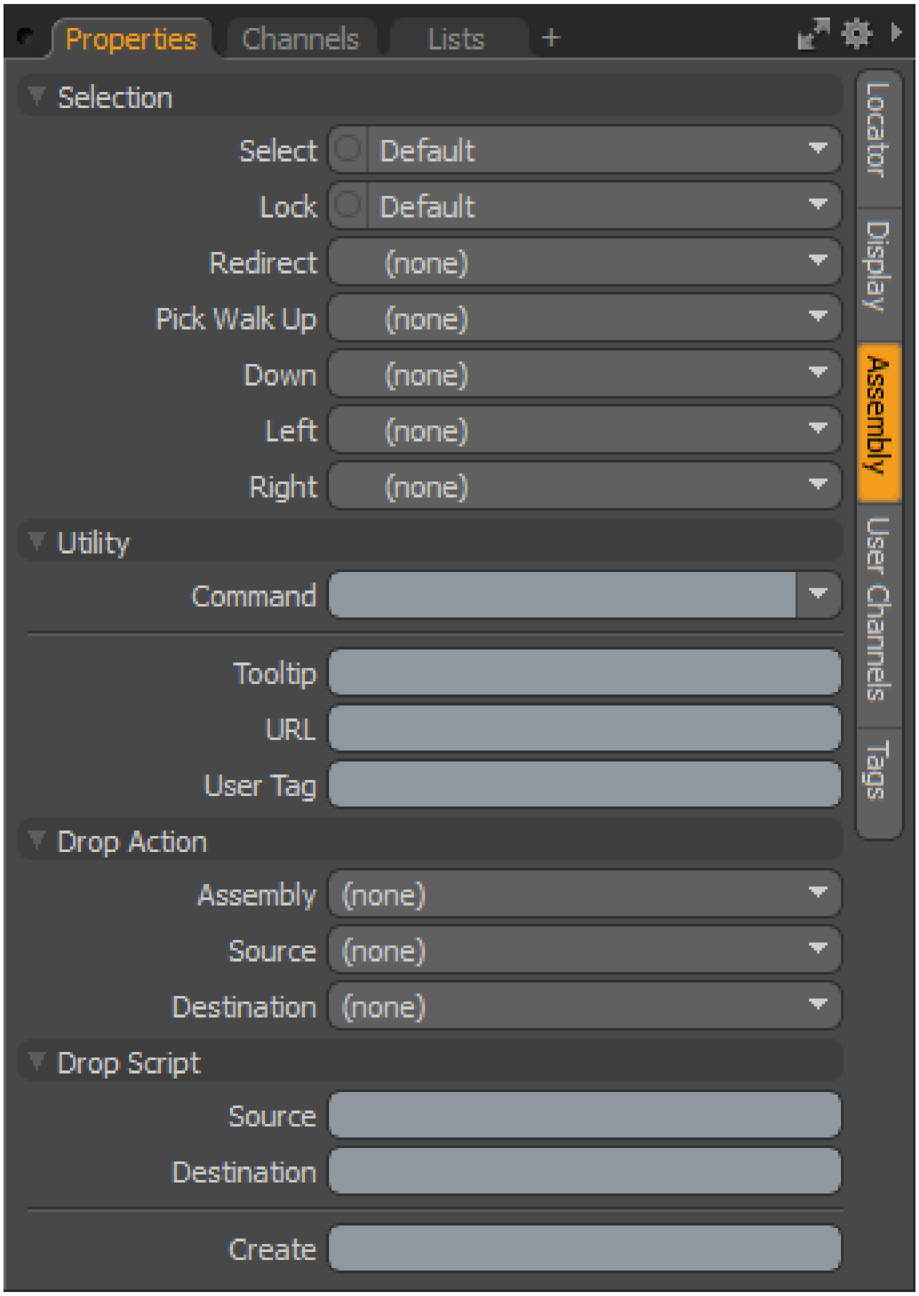
Enable: Provides you a means to toggle the display of the path **On** or **Off**, or to have the path only visible when the item is selected.

Time In/Out: **Time In** determines the number of frames before the current time in the Timeline that are displayed by the curve and **Time Out** determines the number of frames after the current time that are displayed.

Show Frames: When enabled, displays the position of the element at each frame as a small dot along the path. Can be useful to see how the element moves in time, but can also get busy for some types of motion and can therefore be disabled.

Show Keys: When enabled, displays the position of the element at each keyframe as a white dot along the path.

Draw Scale: This option extrudes a ribbon shape outward from the motion path that represents orientation angle of the animated element. The larger the scale value, the wider the ribbon representation in the viewport.


Properties
Channels
Lists
+
Locator
Display
Assembly
User Channels
Tags

Selection

Select: Default

Lock: Default

Redirect: (none)

Pick Walk Up: (none)

Down: (none)

Left: (none)

Right: (none)

Utility

Command:

Tooltip:

URL:

User Tag:

Drop Action

Assembly: (none)

Source: (none)

Destination: (none)

Drop Script

Source:

Destination:

Create:

Assembly options

Selection

Select: Controls whether or not items are selectable within any 3D viewport, **Yes** enables direct item selection, **No** disables it. Items designated **No** can still be selected via the Items (Scenes) List.

Lock: When **Lock** is enabled by selecting **Yes**, items are locked from the application of any type of item transform.

Redirect: When an item is selected in a 3D viewport, specifying an item for **Redirect** triggers Modo to automatically select the alternate item instead. Helpful in easing item selections in complex hierarchies, locators with special Display Viewport properties can be used to visualize selectable items in the viewport that redirect to the actual item you wish to modify.

Pick Walk: The arrow keypad can be used as a way to **walk** selections through a hierarchy. By default, parent items are selected by pressing the up arrow, and child items are selected by pressing the down arrow. The **Pick Walk** function allows you to set which items specifically are selected by any of the 4 keys, Up, Down, Left and Right. Say, for instance, you when you reach the bottom of a hierarchy, the foot, you could specify by hitting the left or right arrows the opposing foot would be selected, and by hitting down the head would be selected, an so forth.

Utility

Command: The **Command** option specifies a command to fire when the item is clicked on in the 3D view. This can be any of Modo's commands. There are two commands that are especially useful when setting up assemblies:

- **item.channelHaul** - This selects any user channels on the item and activates the Channel Haul tool.
- **item.channelPopover** - This displays a popover form containing controls for any user channels on the item. If used in conjunction with selection direction this can also be used to display the channels belonging to a specific group.

Tip: The **Command** function is **disabled** on Mesh Items. To **enable** the option you can convert a mesh to a static mesh, by right-clicking on the item's layer name in the **Items** list and select the **Change Type** > **Static Mesh** option from the contextual menu. Note that converting a mesh to a static mesh is a destructive action, so make sure you have a backup copy should further editing be necessary to the Mesh Item.

Tooltip: Tooltips are the small blocks of text that appear, if assigned, when the pointer momentarily stays over an item. These can be notes or reminders to what an item is intended to do in a scene but tooltips can also be useful for an assembly author to provide some contextual instructions on how to work with an item in an assembly.

URL: If a custom URL is assigned to an item, Using the **F1** help feature and clicking the item in the 3D viewport opens that particular location. Useful for you to add custom documentation to an item.

User Tag: This is a text string that can be used to identify particular items within an assembly or scene. These tags can be read by scripts with the **item.userTag** command. An example of use might be to identify items to be deleted or hidden when an assembly script has finished.

Drop Action

Assemblies offer you the ability to create a rig using Modo's many animation modifiers and then save the setup as a Preset for easy application elsewhere. In the following descriptions the item being dragged is referred to as the source item and the item that has received the drop is referred to as the destination item.

Assembly: You can choose from a series of actions that are performed when the preset is initially dropped into the scene from the Preset Browser. The possible actions are as follows:

- **Parent** - Source item is made a child of the destination item.
- **Parent in Place** - Source item is parented as above, but with **Compensation** applied.
- **Match** - Source item is modified to match position scale and rotation of destination.
- **Match Position** - Source item is modified to match the position only of the destination item.
- **Match Rotation** - Dropped item is modified to match the rotation only of the destination item.
- **Match Scale** - Dropped item is modified to match the scale only of the destination item.
- **Insert** - Inserts the source item into the hierarchy of the destination item. The item is positioned and orientated to match the destination item and becomes the parent of the destination item.
- **Insert at Parent** - Similar to **Insert** but the source item matches its position and orientation to the destination item's parent (if a parent item is present).
- **Insert in Place** - The source item is inserted into the destination item's hierarchy as its parent but retains its current position and orientation.
- **Place** - Positions the selected item at the intersection of the mouse pointer and the surface.
- **Place and Align** - Positions the selected item at the intersection of the mouse pointer and surface rotating the item to match the surfaces normal direction.

Source: The Source drop action is performed when the assembly item is dropped onto a scene item. Same action options as **Assembly**.

Destination: As above, but the action is performed when a scene item is dropped onto the assembly item. Same action options as **Assembly**.

Drop Script

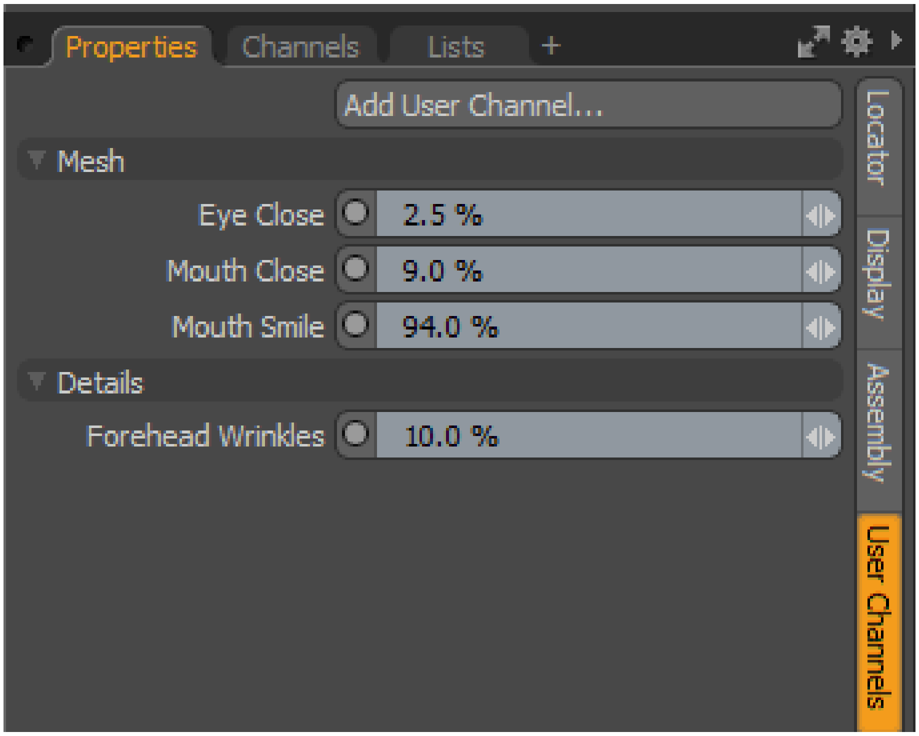
Additional to the above actions, scripts can be assigned to run when certain actions occur.

Source: The assigned script (omitting the @) runs when this item is dropped onto another in the scene. The script gets the source item and destination item passed as arguments. If multiple items are dropped then the script is called for each item in turn.

Destination: Same as above, but the script runs when any item in the scene is dropped onto the item it's assigned to. As with the Source Drop Script the source and destination items are passed to the script as arguments.

User Channels

Add User Channel: User channels are useful in rigging to make simplified controls for complex actions, or to simply give a name to a specific operation. The **Add User Channel** option allows you to add an unlimited number of user defined **Channels** to an item. Once a user channel is defined, it is necessary to connect the channel in the **Schematic** viewport in order to make it operable. Clicking the **Add User Channel** button opens the **Create User Channel** dialog where the actual channel is defined. This panel is fully covered in the Add User Channels page of the documentation.

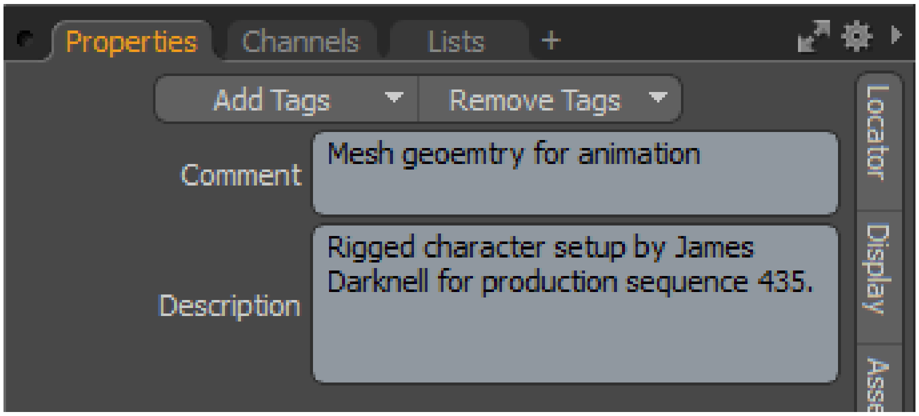


Eye Close 2.5 %
Mouth Close 9.0 %
Mouth Smile 94.0 %
Forehead Wrinkles 10.0 %

As an example in the panel above, a number of user channels have been added, named by morph maps that are linked to the channels, allowing direct animation access. Other user channels are added that create dividers between the groups. Finally, another channel was also added and linked to an image's opacity channel to control the amount of displacement. All of the defined user channels are then added to a channel group so they can remain on the screen for ease of animation.

Tags

The **Tag** panel allows you to define various tags that remain associated to the specific item. These are typically text strings and are useful to add notes to specific elements for other users in a studio, or as reminders as to how something was meant to be used. There are a few pre-defined tags you can choose from or you can create your own specific tag definition.



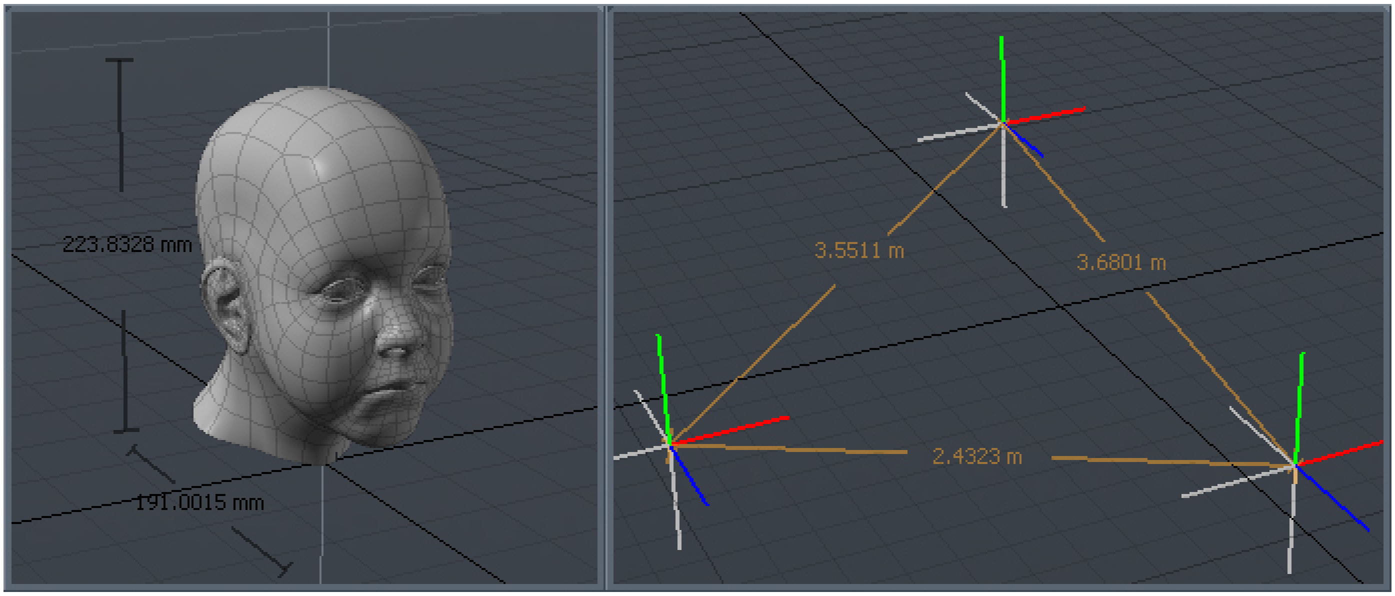
Comment: Mesh geoemtry for animation

Description: Rigged character setup by James Darknell for production sequence 435.

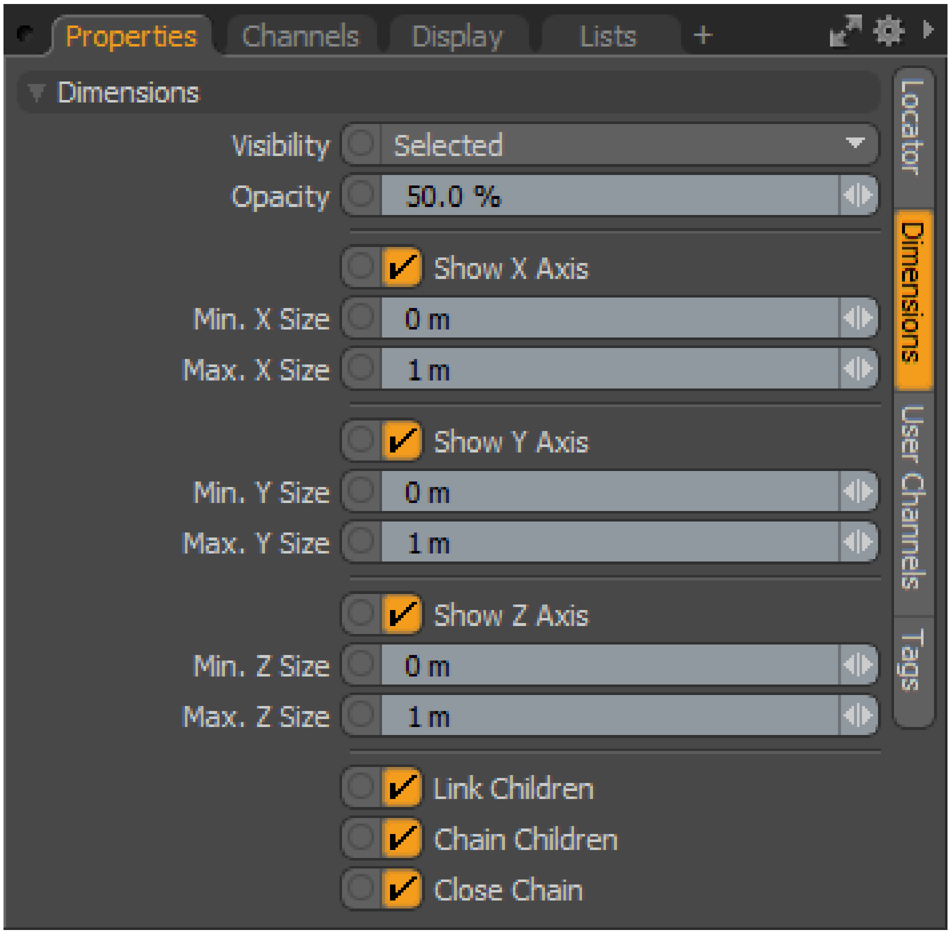
Adding tags is done by pressing the **Add Tags** button, opening the pop-up menu and then selecting a tag type from the **Add Tags** menu. There are also options to **Add Custom Tags**, and **Edit Tag** definition, opening the appropriate panel in the Modo preferences. For more information, see Preferences.

Additional Draw Options

For all item types there are additional GL drawing properties that allow you to customize the UI for rigging in a number of ways, covered here in more detail. Options are toggled on/off with associated buttons in the **Display** sub-tab when the target item is selected. The action creates an additional sub-tab that appears alongside the others in the **Properties** panel.



Show/Hide Dimensions: When this option is **enabled**, an additional sub-tab appears in the **Properties** panel titled **Dimensions**. By default, dimension lines and values for the overall bounding box size appear in the viewport around a Mesh Item, the attributes in the **Properties** panel allow you to further customize the information that is displayed. Linking a series of items together through a hierarchy allows the display of distances between each object's **Center** point, from **parent** to **child** (the dimension values are only displayed when the parent item is selected).



Visibility: Selected

Opacity: 50.0 %

☑ Show X Axis

Min. X Size: 0 m

Max. X Size: 1 m

☑ Show Y Axis

Min. Y Size: 0 m

Max. Y Size: 1 m

☑ Show Z Axis

Min. Z Size: 0 m

Max. Z Size: 1 m

☑ Link Children

☑ Chain Children

☑ Close Chain

Visibility: Determines the visibility of the dimension lines to **Off** not visible, **On** visible and **Selected** only visible when selected.

Opacity: Controls the transparency of the dimension line and numerical value displays in the 3D viewport.

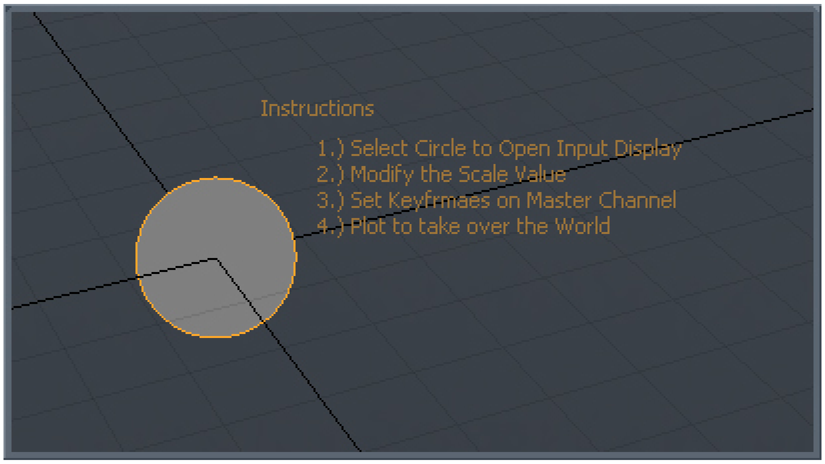
Show X/Y/Z Axis: With these channels you can individually control the display for each axis.

Min X/Y/Z Size/Max X/Y/Z Size: The **Min** and **Max** values can be assigned for each axis and when the defined values for either **Min** or **Max** is reached, the dimension text displays in red in the viewport.

Link Children: When set to **true** (the default option), Modo draws the link lines to the child items, used to disable if the dimension line linking behavior is undesirable.

Chain Child: When set to **true**, Modo draws dimension lines from child to child in hierarchical order instead of from parent to each individual child.

Close Chain: When set to **true**, Modo draws an additional dimension line link back to the parent item from the last child item.

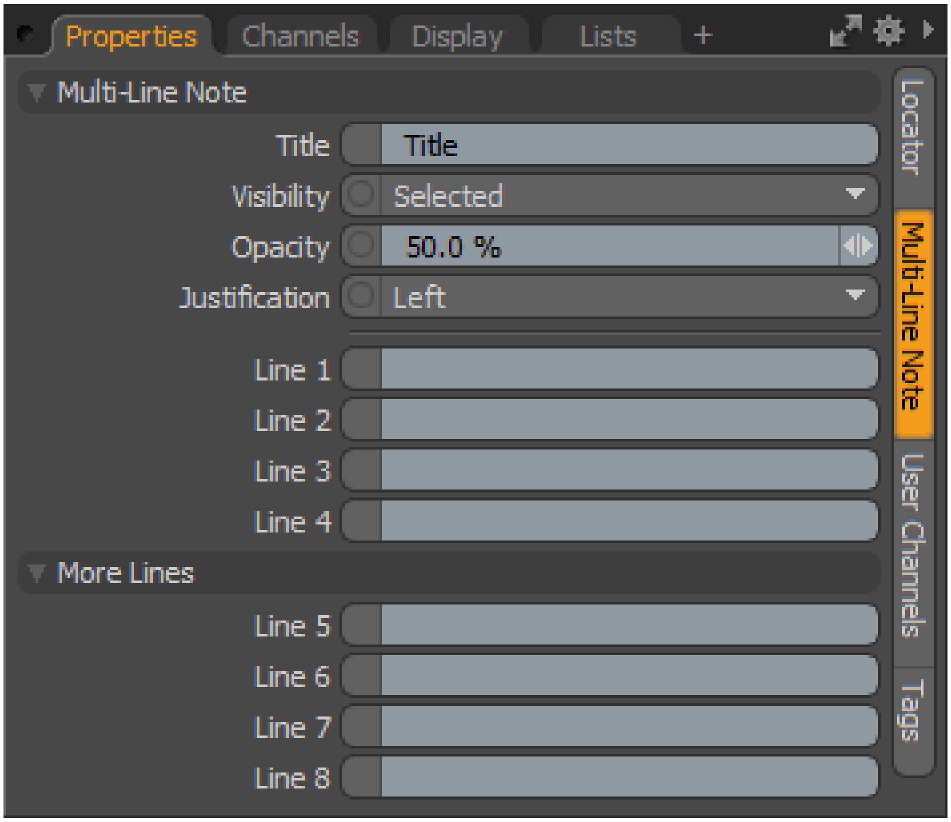


Instructions

1.) Select Circle to Open Input Display
2.) Modify the Scale Value
3.) Set Keyfrmaes on Master Channel
4.) Plot to take over the World

Show/Hide Notes: When this option is **enabled** an additional sub-tab appears in the **Properties** panel titled **Multi-Line Note**. This option allows for on-screen display of custom text strings, for information such as instructions. The text is useful to riggers and TDs, or just to add some helpful reminders for an item when selected.

> **Note:** Display text can also be positioned with the **Label Offset** control.


Properties
Channels
Display
Lists
+
Locator
Multi-Line Note
User Channels
Tags
Multi-Line Note
More Lines
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8

Title: Title
Visibility: Selected
Opacity: 50.0 %
Justification: Left

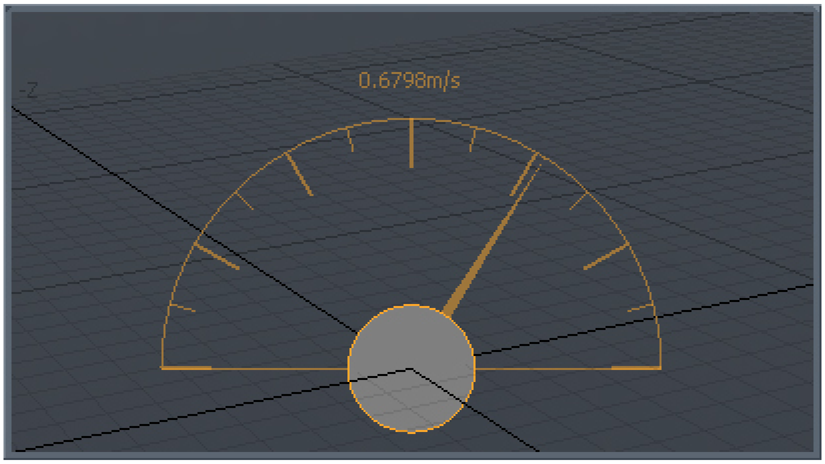
Title: First line of text that displays above the **Line** entries.

Visibility: Determines the visibility of the Dimension lines to **Off** not visible, **On** visible and **Selected** only visible when selected.

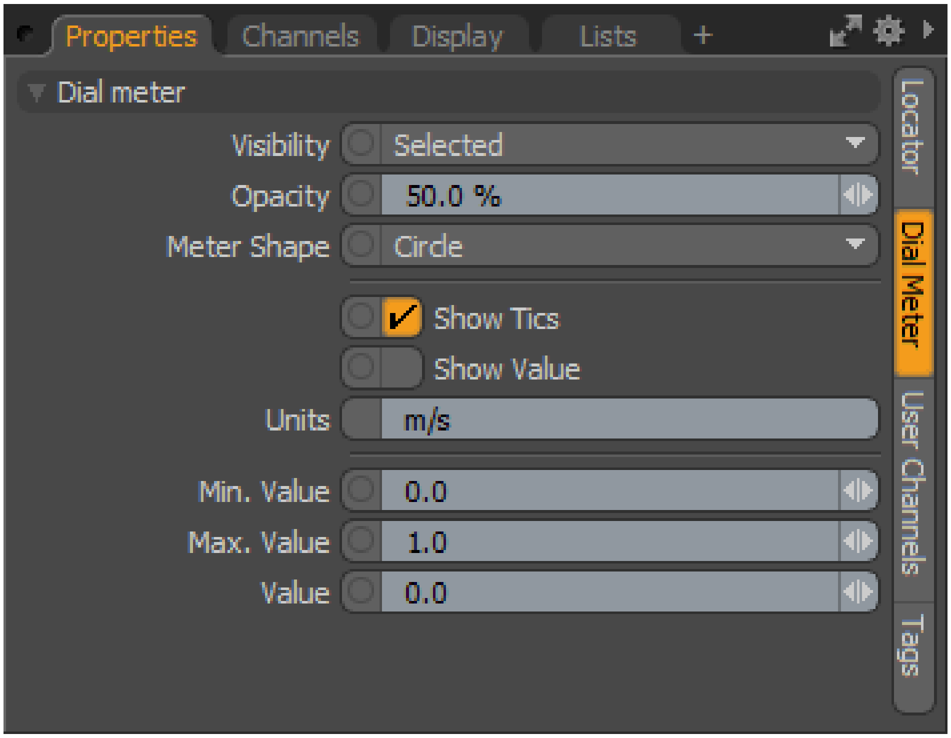
Opacity: Controls the opacity of the dimension line and numerical value displays in the 3D viewport.

Justification: Determines the justification of the text block to **Left**, **Center**, or **Right**.

Line 1-8: Text line entries that display in the 3D viewport.



Show/Hide Meter: When this option is **enabled**, you can display an on-screen analog style gauge graph (looks like a speedometer) that can be rigged to show an individual value in the 3D viewport. An additional sub-tab appears in the **Properties** panel titled **Dial Meter** that allows some customization of the display.



Dial meter

Visibility: Selected

Opacity: 50.0 %

Meter Shape: Circle

☑ Show Tics

☐ Show Value

Units: m/s

Min. Value: 0.0

Max. Value: 1.0

Value: 0.0

Visibility: Determines the visibility of the **Meter** display to **Off** not visible, **On** visible and **Selected** only visible when selected.

Opacity: Controls the opacity of the **Meter** readout and numerical value displays in the 3D viewport.

Meter Shape: You can choose between a full **Circle** meter display and a **Half Circle** display.

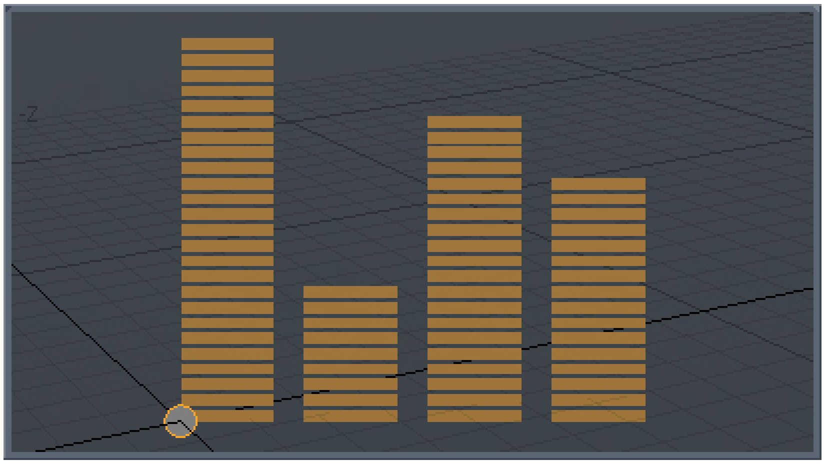
Show Tics: Toggles the visibility of 15° and 30° interval lines in the display.

Show Value: Toggles the visibility of the **Value** information readout above the dial.

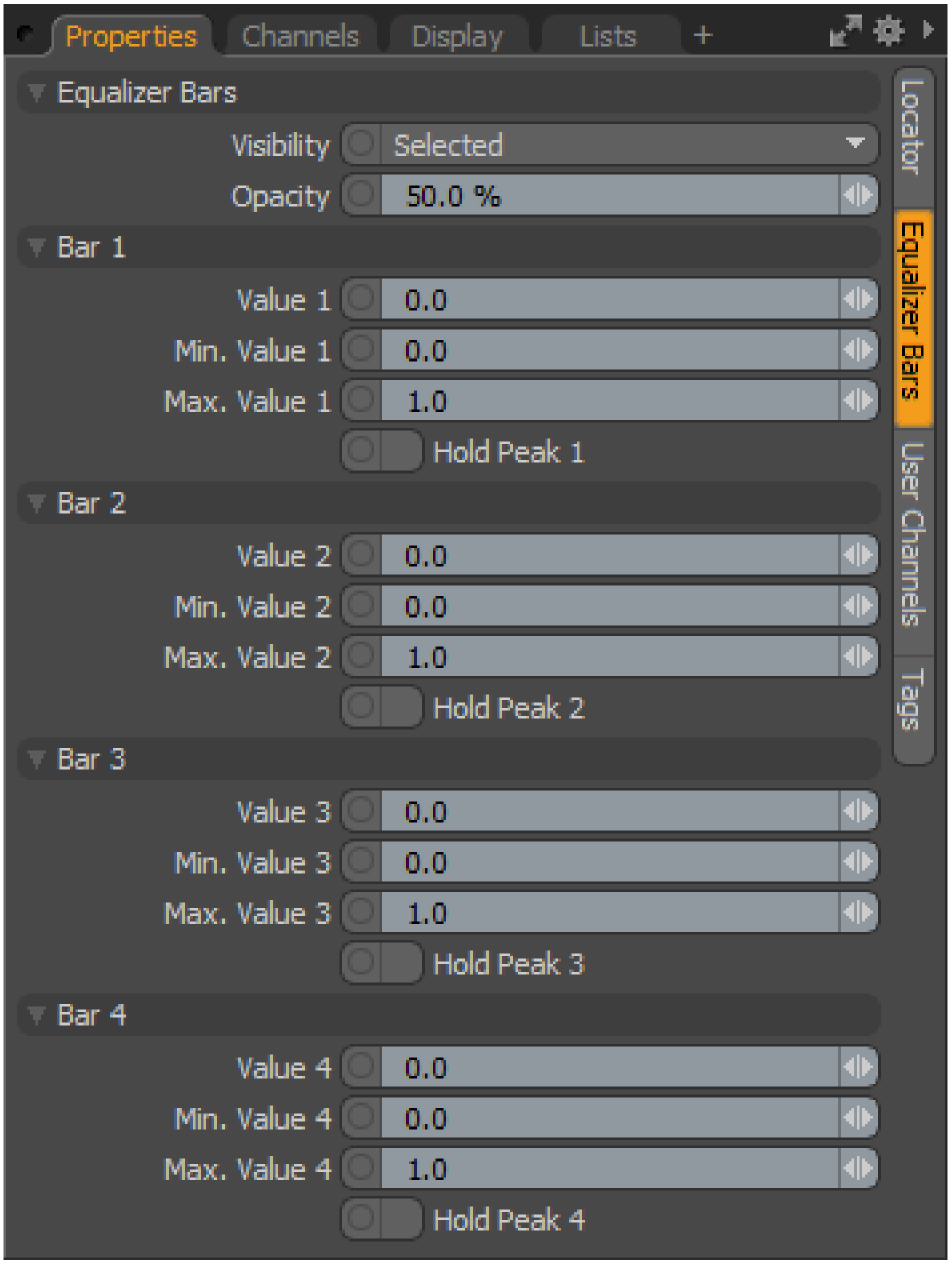
Units: A text display that is positioned next to the **Value** display when the **Show Value** option is set to true.

Min/Max Value: The **Min** and **Max** values determine the lowest value and the highest value displayed in the on-screen meter, controlling the overall position of the **Value** needle.

Value: This is the input channel that determines the actual value displayed in the viewport, as well as the position of the meter needle (relative to the Min and Max values). Additional rigging is required in the Schematic Viewport in order to connect the **Value** channel input from an output.



Show/Hide Level Bars: When this option is **enabled** an additional sub-tab appears in the **Properties** panel titled **Equalizer Bars**. This option is used to display an on-screen equalizer-style bar graph for visualizing multiple on-screen values relative to one another. Additional channels show in the Channels Viewport when **enabled**.



Equalizer Bars

Visibility: Selected
Opacity: 50.0 %

Bar 1

Value 1: 0.0
Min. Value 1: 0.0
Max. Value 1: 1.0
Hold Peak 1

Bar 2

Value 2: 0.0
Min. Value 2: 0.0
Max. Value 2: 1.0
Hold Peak 2

Bar 3

Value 3: 0.0
Min. Value 3: 0.0
Max. Value 3: 1.0
Hold Peak 3

Bar 4

Value 4: 0.0
Min. Value 4: 0.0
Max. Value 4: 1.0
Hold Peak 4

Visibility: Determines the visibility of the **Level Bars** to **Off** not visible, **On** visible and **Selected** only visible when selected.

Opacity: Controls the opacity of the **Level Bars** readout display in the 3D viewport.

Value 1-4: These are the input channels that determine the actual values displayed in the viewport, as well as the scale of the individual level bars (relative to the **Min** and **Max** values). Additional rigging is required in the **Schematic** viewport in order to connect the **Value** channel input from an output.

Min Value 1-4/**Max Value 1-4**: The **Min** and **Max** values determine the lowest value and the highest value per readout, which is displayed in the on-screen **Level Bar**, controlling the overall scale of the bar displayed.

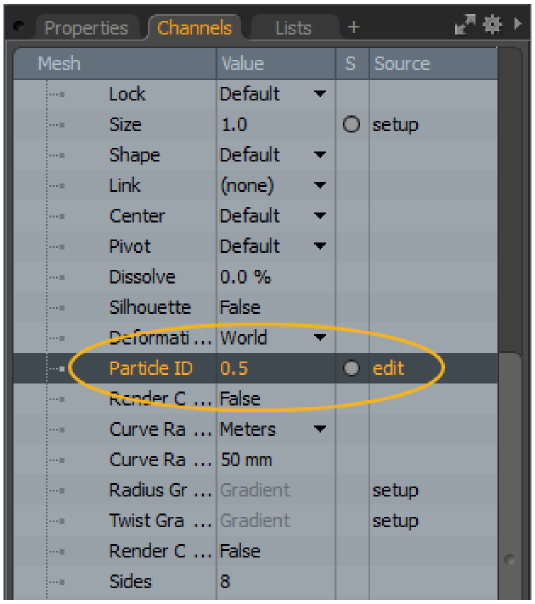
Hold Peak 1-4: Like an audio readout, when the **Max** value is reached for any **Value** readout, an indicator is drawn in the viewport. Toggle the **Hold Peak** value to reset the display.

Note: Modo can display up to 8 bars by referencing the additional channels available in the **Channels** viewport.

Particle ID Channel

Individual items have a Particle ID channel that you can use to control the particle ID assigned to rendered surfaces. You can set this value manually, or drive it by a rig and then feed it to Shaders.

The Particle ID channel allows you to assign a value between 0.0 and 1.0 to a mesh item. You can specify it in the **Channels** list of the mesh item:



Mesh	Value	S	Source
Lock	Default		
Size	1.0	○	setup
Shape	Default		
Link	(none)		
Center	Default		
Pivot	Default		
Dissolve	0.0 %		
Silhouette	False		
Deformati ...	World		
Particle ID	0.5	●	edit
Render C ...	False		
Curve Ra ...	Meters		
Curve Ra ...	50 mm		
Radius Gr ...	Gradient		setup
Twist Gra ...	Gradient		setup
Render C ...	False		
Sides	8		

This channel is useful when using the Gradient or the Texture Switch node. Gradient values range from 0 to 100%, so a Particle ID of 0.5 corresponds to 50%, which results in a mesh with the color in the middle of a Gradient.

For Texture Switch, values are based on the order of connection. So if you have three textures, the first one is 0.0, the second connected texture is 0.5, and the third one is 1.0.

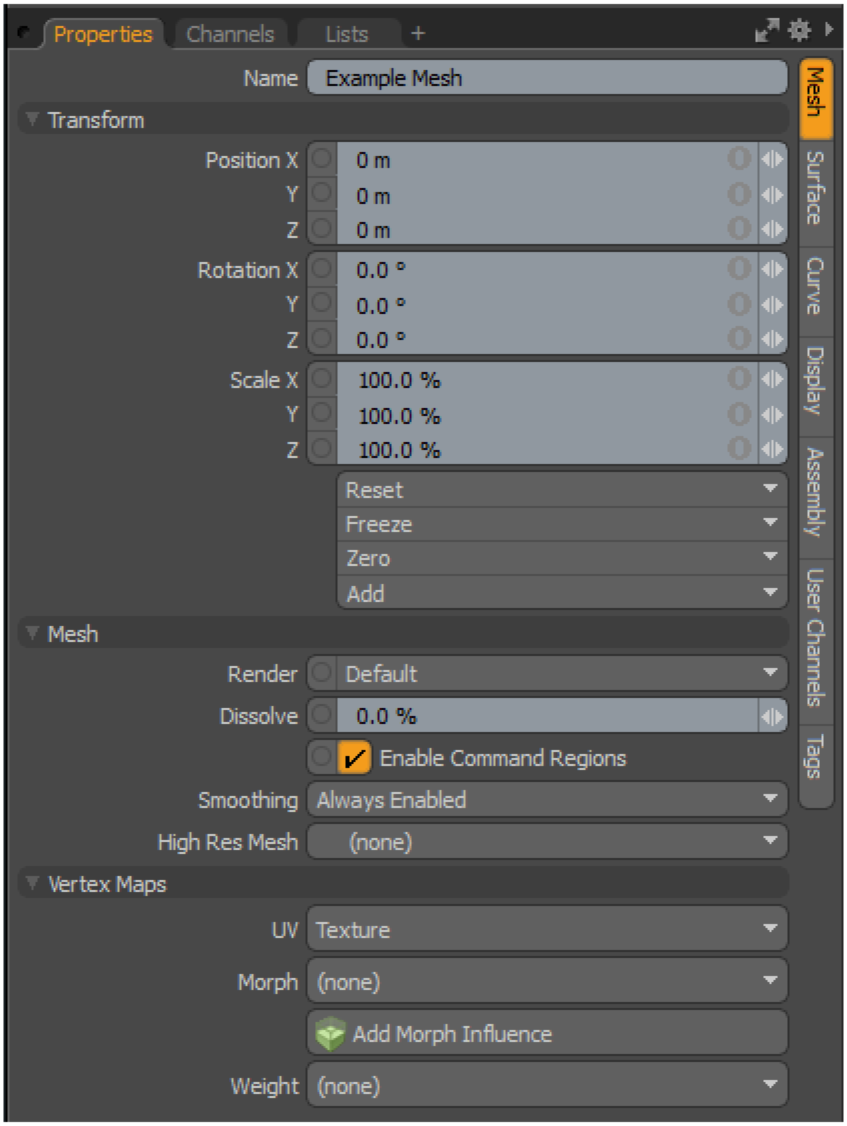
Mesh Item

Mesh Items are often referred to as geometry layers as this is how they appear in the **Item List**, as individual layers. However, these layers can also be thought of more like containers that hold the various geometry components - vertices, edges, and polygons. Mesh Items are usually the heroes of any project; the actual surfaces that are rendered. Each individual mesh layer is its own self-contained object, with its own set of channels, item transforms, as well as UV maps, Weight Maps, and any other associated Vertex Maps.

Mesh Items are animated by transforming their position, rotation and scale channels, or by using Deformers to bend and twist the geometry contained within. Component-level changes modify the contents contained within the layer, and can generally be applied independent of the animation assigned to the item. The difference between component edits and Item-level edits is a very important concept to understand. If you ever find that you have performed Item-level transforms when you intended to do component transforms, you can use the **Freeze** options found on the item's **Properties** panel to return the center location back to the World Origin.

You can add additional **Mesh Items** to a scene by pressing the **Add Item** button at the top of the **Item List** and selecting **Locators** > **Mesh**. Additionally, you can use the menu bar command **Geometry** > **New Mesh Item**. Finally, you can also easily create an empty mesh layer by simply pressing the **N** keyboard shortcut while in the **Model** interface layout. When the **Mesh Item** is selected, its associated attributes become available for selection and you can edit it within the **Properties** viewport panel.

Mesh Tab


Properties
Channels
Lists
+
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags
Transform
Reset
Freeze
Zero
Add
Mesh
Vertex Maps
Add Morph Influence

Name Example Mesh
Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Render Default
Dissolve 0.0 %
☑ Enable Command Regions
Smoothing Always Enabled
High Res Mesh (none)
UV Texture
Morph (none)
Weight (none)

- **Name** - displays the current item name. You can easily change it by clicking within the field and typing the new name.

> **Note:** Please be aware that currently you cannot have multiple Mesh Items that share the same name and that Item names need to be unique to the item. Mesh items that share the same name will automatically be versioned to reflect the amount of items in your scene that share the naming convention; for example if your scenes contains three items all named Cube, the first mesh item created will be named Cube, while the two subsequent Cube items will be named Cube (2) and Cube (3).

The following **Transform** options are available for the **Mesh Item**:

- **Position** - an item transform that allows you to numerically position the item in XYZ space. By default, **Position** transforms originate from the item's **Center** position.
- **Rotation** - an item transform that allows you to numerically set the rotation of the item. By default, **Rotation** transforms originate from the item's **Center** position.
- **Scale** - an item transform that allows you to numerically set the size of the item. By default, **Scale** transforms originate from the item's **Center** position.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Freeze** - returns the item's **Center** position to the world space center (0,0,0) without changing the position of the Mesh Item itself.
- **Zero** - resets the chosen transform property values to **0**, leaving the **Center** position and mesh position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. **Transform** items are the channel groups that store the actual transform values controlling any item's position, rotation, and/or scale. You can add as many **Transform** items as desired for any transform property desired. Adding additional **Transform** items produce an additive effect where each transform group is evaluated before the next, and so on. Additional item transforms are evaluated in their order in the channels list, from the bottom upwards.

  It should be noted that by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead. There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Die to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the Master scene, they must first exist.

The following **Mesh** options are available for the **Mesh Item**:

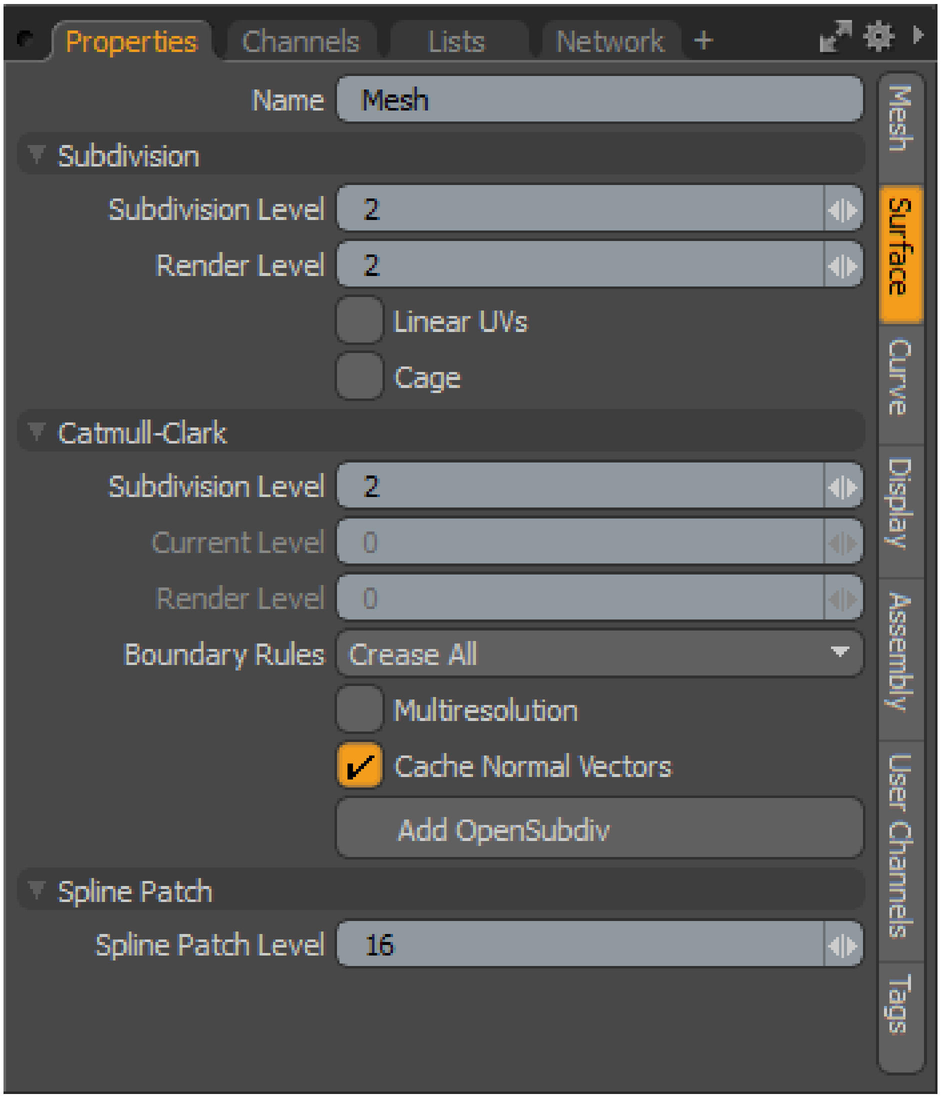
- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable Mesh Items using the visibility column (eye icon) of the **Item List**, when the mesh layer is visible, it contributes to the final rendered scene, and when invisible, it does not. On some instances you may prefer to fix this state, setting the mesh as **On** (enabled) or **Off** (disabled), regardless of layer visibility. These controls are also useful for workflows that auto-toggle visibility, saving you from manually enabling mesh layers for test renders.
- **Dissolve** - controls the visibility of an item layer when rendering. At 0% the item renders normally, utilizing the **Shader Tree** settings. Values above 0% fade the visibility of the item layer, ramping toward 100%, where the item would be completely invisible.

- **Enable Command Regions** - can be used to toggle the automatic **Command Regions** functionality. When enabled, command regions work as defined; when disabled, command region definitions are ignored.
- **Smoothing** - a performance-enhancing option that allows you to control the **Smoothing** applied to a **Mesh Item** on an individual basis. When disabled, the reduced evaluation overhead can be useful to help to increase performance in heavy scenes. **Always Enabled** smooths the entire layer at all times, **Disabled with Deformers** disables any smoothing only when a deformer is attached to the item layer, and the last option, **Always Disabled**, eliminates smoothing for the item at all times.
- **High Res Mesh** - Specify the mesh item to use for baking a normal map. It can either bake the normal textures applied to the mesh item or do object to object baking. For more information, see Working with Bake Items and Baking Tools.

The following **Vertex Maps** options are available for the **Mesh Item**:

- **UV** - provides a convenient means to select the current UV map that is visible in the 3D Viewports. It also allows the ability to easily create a new map by selecting the **New** option. For more information, reference the Vertex Maps topic.
- **Morph** - provides a convenient means to select the current active Morph map. It also allows the ability to easily create a new map by selecting the **New** option.
- **Add Morph Influence** - automatically creates a Morph Influence with the selected Morph map, that is connected to the current mesh. Morph deformers are only visible in viewports that have the **Enable Deformers** option enabled.
- **Weight** - provides a convenient means to select the current Weight Map that is visible in the 3D Viewports. It also allows the ability to easily create a new map by selecting the **New** option.

Surface Tab



Name: Mesh

Subdivision

Subdivision Level: 2

Render Level: 2

☐ Linear UVs

☐ Cage

Catmull-Clark

Subdivision Level: 2

Current Level: 0

Render Level: 0

Boundary Rules: Crease All

☐ Multiresolution

☑ Cache Normal Vectors

Spline Patch

Spline Patch Level: 16

- **Name** - displays the current item name. you can easily change it by clicking within the field and typing the new name.

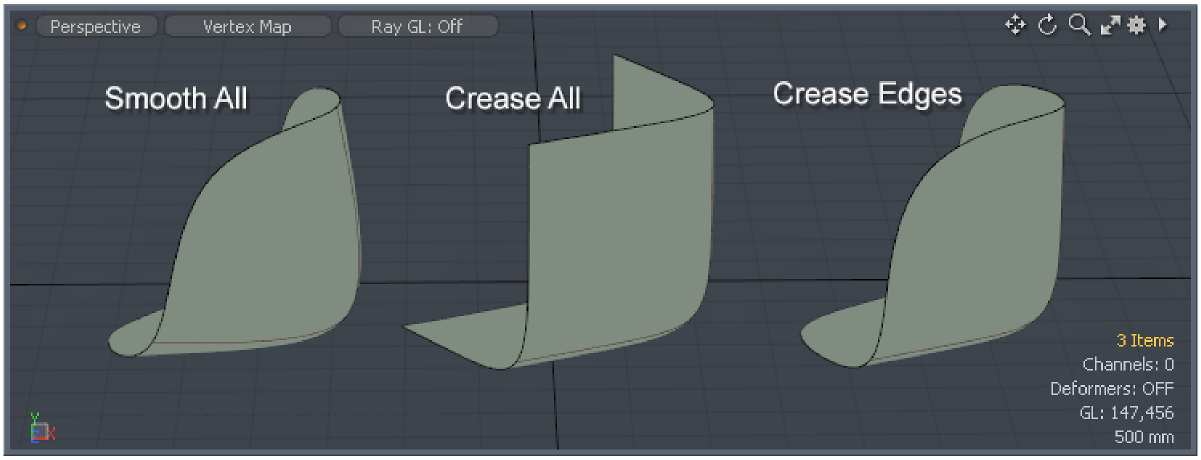
The following are **Subdivision** options for the **Mesh Item**:

- **Subdivision Level** - any selected geometry within a layer can be converted to a subdivision surface by selecting the target geometry and pressing the **Tab** key, converting the geometry. This produces a smoothing effect to the surface, essentially becoming a proxy for the smoothed SDS geometry, which cannot be directly edited. The **Subdivision Level** setting specifies to what degree an SDS model is divided. There are certain functions that override this setting at render time, such as those for **Adaptive Subdivision**. Otherwise, when using the Freezing Geometry command, the **Subdivision Level** controls the number of polygons generated, as it controls the number of polygons displayed in any 3D viewport. This value can be interactively adjusted with the **+/-** (plus/minus) keys on the number pad when a **Mesh Item** is selected.

- **Render Level** - when **Adaptive Subdivision** is disabled, you can set an independent **Render Level** for subdividing the geometry only when any of the Render Commands are invoked. This can be helpful in increasing OpenGL viewport performance by setting a low **Subdivision Level** and a high **Render Level**, keeping a finely subdivided smooth surface only when the scene is rendered.
- **Linear UVs** - Modo automatically switches between linear UVs for straight polygonal geometry and smoothed UVs for subdivision surface (SDS) geometry. There may be some instances where an image map created outside of Modo utilizes linear for an SDS model, in which case enabling **Linear UVs** correctly applies the image map to the Sub-D geometry within Modo.
- **Cage** - toggles any Subdivision Surface geometry in the current mesh layer. This is especially useful in temporarily disabling SDS in complex models without losing the SDS definitions for the geometry.

The following are **Catmull-Clark** options for the **Mesh Item**:

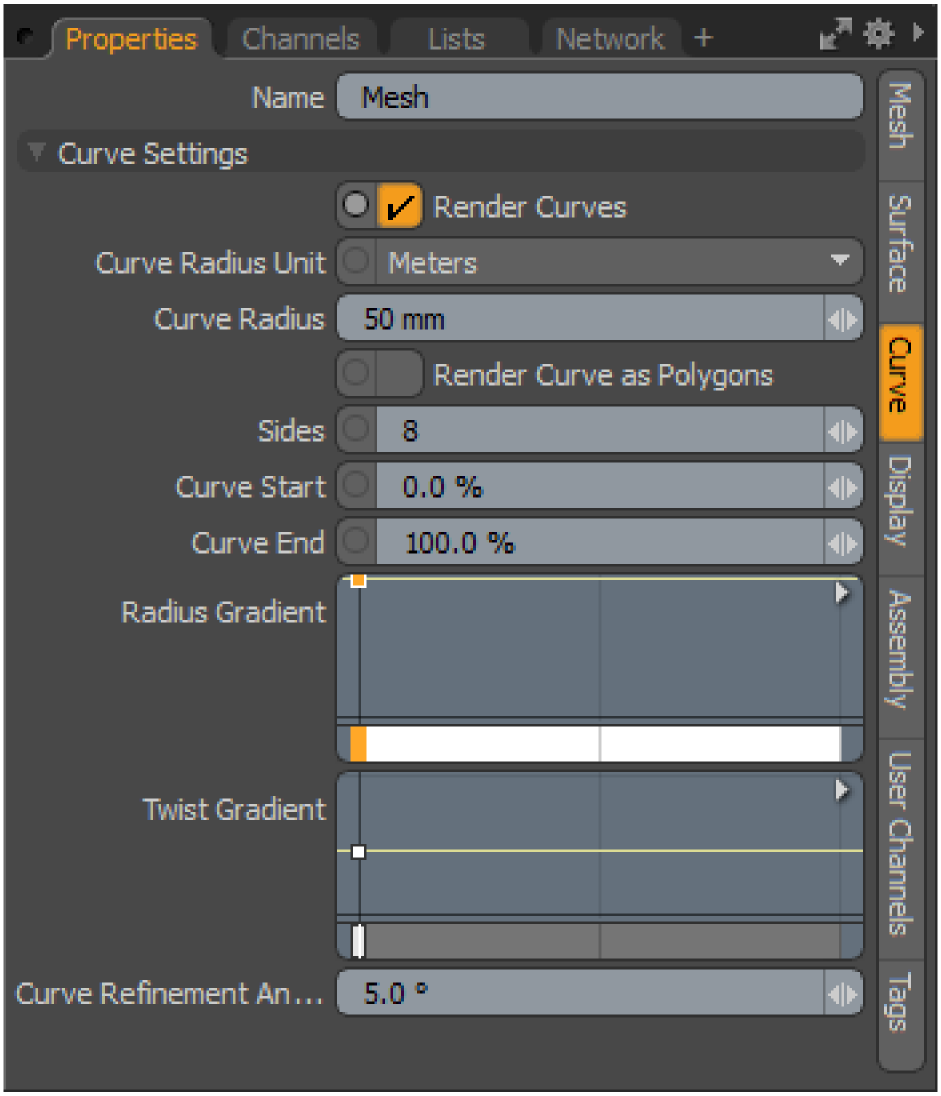
- **Subdivision Level** - you can convert any surface into a Catmull-Clark Subdivision Surface by pressing the **Shift**+**Tab** keyboard shortcut. The Catmull-Clark Subdivision Surfaces are superior to regular Subdivision Surfaces, especially in their handling of edge creasing. However, they are more computationally intensive than regular Subdivision Surfaces and require more memory to render. The **Subdivision Level** controls how finely a surface is subdivided.
- **Current Level** - when the Multiresolution option is enabled, you can use the **Current Level** option to step up and down through the different levels of subdivision (up to the maximum level defined by the **Subdivision Level**). This is useful when sculpting, providing the ability to add different degrees of detail to a model at different levels.
- **Render Level** - when **Adaptive Subdivision** is disabled, you can set an independent **Render Level** for subdividing the geometry only when any of the render commands are invoked. This can be helpful in increasing OpenGL viewport performance by setting a low **Subdivision Level** and a high **Render Level**, keeping a finely subdivided smooth surface only when the scene is rendered.
- **Boundary Rules** - the boundary rules determine how to move the points on the boundaries of a Catmull-Clark Subdivision Surface mesh. Creasing produces a sharper transition between surfaces, while smoothing produces smooth surface transitions. The differences between **Smooth All** and **Crease Edges** are only noticeable on modes that use the Edge Weighting function. The options are:
  - **Smooth All** - all positions on the boundary are moved by the smooth rule of Pixar Subdivision Surfaces.
  - **Crease All** - the points that have two incident edges (valence = 2) are moved by the crease rule; other positions on the boundary are smoothed by two boundary edges.
  - **Crease Edges** - the points that have two incident edges (valence = 2) are moved by the smooth rule; other positions on the boundary are smoothed by two boundary edges.



- **Multiresolution** - only available to a Pixar Subdivision Surfaces model; you can enable this option to step up and down through the subdivision levels when sculpting a mesh, allowing you the ability to add varying degrees of detail without obliterating detain present in other levels. You can adjust the current level using the **Current Level** option, above, or step through the levels using the Multi-res settings in the sculpting toolbox of the **Paint** interface tab.
- **Cache Normal Vectors** - this is useful to enable when sculpting a Multi-resolution mesh. When enabled, normal vectors for the mesh are stored in memory, increasing brush stroke performance. Depending on the system, this increase can be significant. The drawback to the option is that it can increase system memory usage significantly as well, depending on the target mesh's density. On modest systems, the use of the option should be limited to either light meshes or disabled altogether for maximum performance.
- **Add/Remove OpenSubdiv** - when enabled, replace the standard Modo Catmull-Clark subdivision surfaces display algorithm with the high performance Open Subdiv standard. Pressing the button adds an additional option to the panel allowing you to define which **Compute** method is utilized. The **Add Open Subdiv** option is meant as a performance enhancement that can improve playback speeds of deforming Catmull-Clark meshes in the 3D viewport. For general modeling, you should stay with the standard Modo Catmull-Clark setting for best performance. To remove the option, simply press the **Remove OpenSubdiv** button reverting back to the prior settings.
- **Compute** - only appears when you have enabled the OpenSubdiv high performance subdivision surfaces standard. There are four possible options:
  - **Automatic** - when defined, it chooses the best mode, based on the current system. **GPU** mode is the highest priority, if it is available. **Multi Threads** selection is chosen, if no **GPU** mode is available. Otherwise, barring those choices, it defaults to **CPU** mode.
  - **CPU** - when defined, the subdivision is computed by **CPU**, using the main thread.
  - **GPU** - uses OpenCL framework or GLSL Transform Feedback. If the current computer does not support OpenCL, it chooses another mode.
  - **Multi Threads** - selects one of the multi-threading frameworks like GCD, OpenMP, or TBB. For Mac OS X, GCD is selected for multi-thread mode. TBB is chosen for Windows and Linux, if it is available, otherwise it selects OpenMP.

- **Spline Patch Level** - spline patching is a way of parametrically defining a surface with a number of curves. When patching the curves to define the surface, the **Spline Patch Level** controls the number of polygons used. These polygons are not directly editable unless you apply the Freeze command found under the menu bar **Geometry** > **Freeze**.

Curve Tab



Name: Mesh

Curve Settings

☑ Render Curves

Curve Radius Unit: Meters

Curve Radius: 50 mm

☐ Render Curve as Polygons

Sides: 8

Curve Start: 0.0 %

Curve End: 100.0 %

Radius Gradient

Twist Gradient

Curve Refinement An…: 5.0 °

- **Name** - displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Curve Settings** options are available for the **Mesh Item**'s **Curve** tab:

- **Render Curves** - generally curves are non-rendering items in a layer meant to facilitate other functions: visual aids, patching guides, duplication paths, and so forth. However, with the **Curve Radius** set to a value above 0, and the **Render Curves** option enabled, the previously-invisible curves are now renderable. Modo procedurally generates geometry, based on your settings only at render time. Material tags for surfacing can be applied to curves in the same way they are for regular polygons by selecting the curve in **Polygons** mode and assigning a **Material** tag to it, as if it were a normal polygon. Then any material settings assigned in the **Shader Tree** to that material group are applied to the curve at render time.

Rendered curves essentially become tubes that have several benefits over extruded geometry. They reduce the amount of geometry in the scene since render curves are only generated at render time, making for a lighter file size, and their size is editable and animatable as well. Additionally, any morph deformations applied to a curve itself won't impair the final surface, where it keeps a consistent thickness through the deformation.

- **Curve Radius Unit** - you can define the size of the rendered curve in two ways: using the world unit **Meters**, which defines the size based within the scene, and using **Pixels**, which defines the curve based on the rendered image. Using **Meters** produces a cylinder size that is consistent with the scene itself, and has perspective, scaling in to the distance. When defined as pixels, the resulting cylinder scales appropriately to produce a consistent thickness in the rendered image, regardless of perspective and scaling.
- **Curve Radius** - when enabled, the **Curve Radius** determines the size of the resulting tube that is generated. The radius is centered around the curve itself, so a setting of 12mm would generate a tube with a 24mm diameter. When the **Curve Radius Unit** is defined as **Pixels** then the size of the resulting curve is determined by the actual pixel size along its entire length, and is also centered on the curve location. Keep in mind, as the **Frame** resolution of the render image changes, the **Pixel** scale values do not scale automatically, and need to be manually adjusted to maintain a like-sized representation at the higher resolution.
- **Render Curve as Polygon** - due to the procedural nature of the rendered curve, it was previously not possible to add polygon-specific shading effects to the resulting cylinders, such as Displacement. When the option **Render Curves as Polygons** is enabled, the resulting curve is generated using micropoly tessellation, allowing displacement.
- **Sides** - when the option **Render Curves as Polygons** is enabled, you can define a value to represent the number of lateral divisions around the resulting cylinder. Lower values produce a blocky cylinder, while higher values produce a smooth, round cylinder.
- **Curve Start/End** - define positions along the curve where the procedural geometry generates start and end, respectively. This allows you to, for instance, animate a growing vine, by animating the **Curve End** value from 0% to 100% over a length of frames.
- **Radius Gradient** - controls the diameter along the length of the generated curve. Using a mini-gradient input, the radius can be modified with gradient keys that represent a percentage of the **Curve Radius** value. The gradient input's width from 0% to 100% represents the curves overall length, with the base on the left side. (The base of the curve itself is represented by the small circle at its origin.) Keys can be added by middle-clicking along the length of the gradient input, then clicking and dragging the key icon up or down, adjusting the value. The grayscale bar, directly below, gives visual feedback as to how the values fade from one key to the next. Black produce no width, and white produce 100% maximum defined width, with the shades of gray attenuating the radius in between.

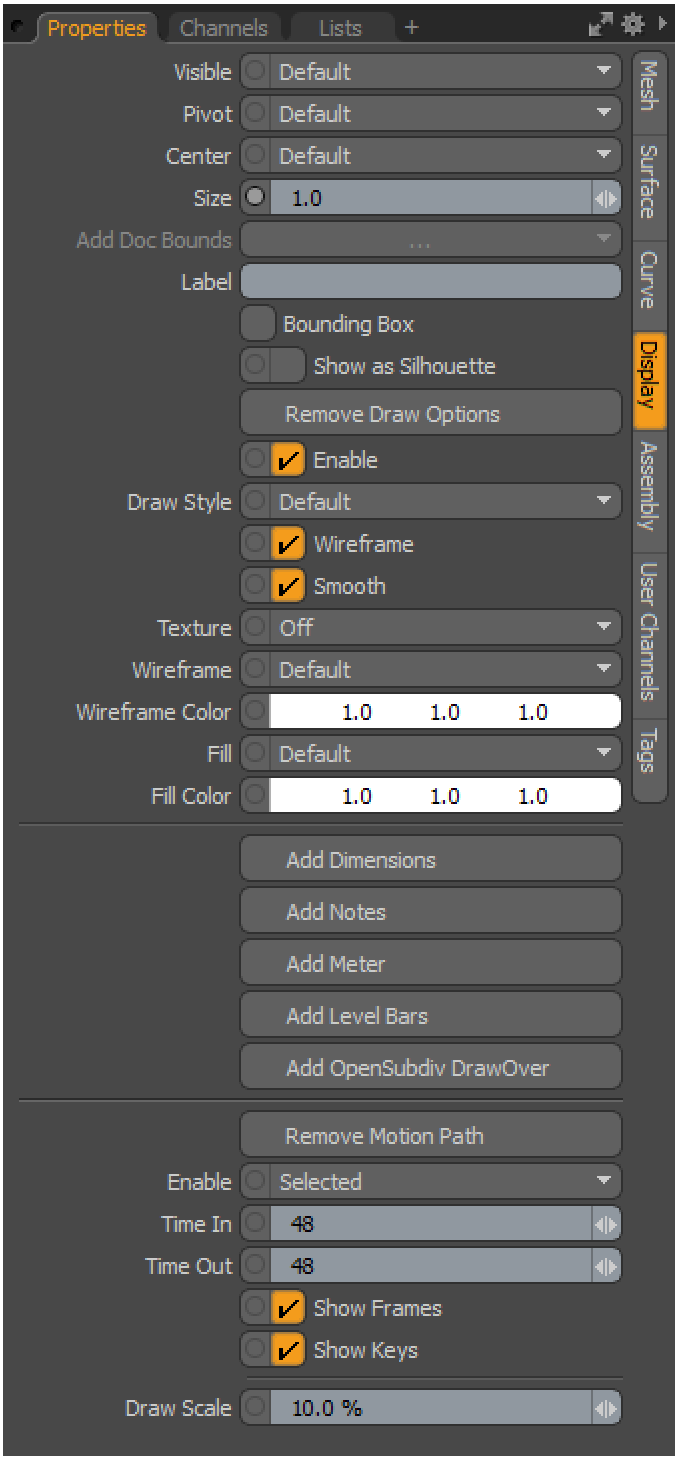
Note: The **Radius Gradient** values affect all curves within the layer in the same manner.

- **Twist Gradient** - controls the amount of twisting along the length of the generated curve, such as what might be seen in the stripes of a candy cane. Using a mini-gradient input, the twist amount can be modified with gradient keys that represent a degree of rotation, defined as a percentage. The gradient input's width from 0% to 100% represents the curve's overall length, with the base on the left side. (The base of the curve itself is represented by the small circle at its origin.) Keys can be added by middle-clicking along the length of the gradient input, then clicking and dragging the key icon up or down, adjusting the value. The grayscale bar directly below gives visual feedback as to how the values fade from one key to the next. Black produces no rotation amount of 0°, and white produce 100% maximum rotation amount of 360°, with the shades of gray attenuating the twisting in between.

Note: The **Twist Gradient** values can be driven higher than 100%, when opening the **Gradient Editor**, and affects all curves within the layer in the same manner.

- **Curve Refinement Angle** - controls how curves are quantized for display and freezing. This is analogous to the patch resolution for subdivision surfaces.

Display Tab


Properties
Channels
Lists
+
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags
Remove Draw Options
Add Dimensions
Add Notes
Add Meter
Add Level Bars
Add OpenSubdiv DrawOver
Remove Motion Path

Visible Default
Pivot Default
Center Default
Size 1.0
Add Doc Bounds ...
Label
☐ Bounding Box
☐ Show as Silhouette
☑ Enable
Draw Style Default
☑ Wireframe
☑ Smooth
Texture Off
Wireframe Default
Wireframe Color 1.0 1.0 1.0
Fill Default
Fill Color 1.0 1.0 1.0
Enable Selected
Time In 48
Time Out 48
☑ Show Frames
☑ Show Keys
Draw Scale 10.0 %

- **Visible** - sets visibility for the item itself in the 3D viewport: **Default** respects viewport settings, **Yes** sets the item as visible always, **No** disables visibility entirely.

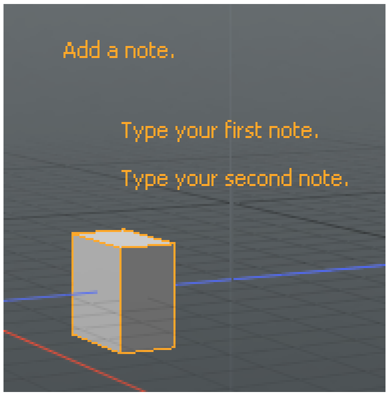
- **Pivot** - sets visibility for the 3D item pivot: **Default** respects viewport settings, **Yes** sets the pivot as always visible, **No** disables pivot visibility entirely.
- **Center** - sets visibility for the 3D item center: **Default** respects viewport settings, **Yes** sets the center as always visible, **No** disables center visibility entirely.
- **Size** - a multiplier that adjusts the display size of the item's representations of camera, light, and locator icons. This does not affect the representations of mesh layers.
- **Label** - typing in text for the **Label** displays the text, positioned next to the 3D item, in the viewport for easy identification. It also adds two additional properties to control the display of the label in the viewport: **Show Label** and **Label Offset**.
- **Show Label** - temporarily enables/disables the display of the label in the viewport without losing settings; this is only available when a label is specified.
- **Label Offset** - specifies the distance away from the locator where the label appears; this is only available when a label is specified.
- **Bounding Box** - when enabled, **Mesh Items** display as a simple wireframe box around the geometry's total volume, instead of displaying the geometry itself.
- **Show as Silhouette** - when enabled, items draw only as a solid shape with no interior details. For the purposes of animation, it can be beneficial to view certain elements as a silhouette to ensure that key poses are expressing certain information. This works in conjunction with the **Enable Silhouette** setting of the 3D (OpenGL) Viewport display options, which must be enabled (the default state) for the silhouette to be visible.

Tip: The **Show as Silhouette** display option is only applicable to the **Advanced OpenGL** viewport display mode.

- **Add/Remove Draw Options** - opens/removes additional properties for adjusting how items display in the 3D Viewports. When selected, additional properties appear.
- **Add/Remove Dimensions** - when this option is enabled (by default), dimension values for the overall bounding box size appear in the viewport around the **Mesh Item**. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Add/Remove Notes** - when this option is enabled, you can define multiple lines of text (up to 9) that show up when the item is selected. An additional **Multi-Line Note** sub-tab shows in the **Properties** viewport, allowing further display customizations.

To add a note:

1. In the **Items** list, select the mesh item you want to add a note to.
2. In the **Properties** list, on the bottom right panel, open the **Display** tab and click **Add Notes**.
3. Open the **Multi-Line Note** tab.
4. Set the options and type the text you want displayed in the viewport.



- **Add/Remove Meter** - when this option is enabled, you can display an on-screen analog-style meter graph (looks like a speedometer) that can be rigged to show individual values in the 3D viewport. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.
- **Add/Remove Level Bars** - when this option is enabled, you can display an on-screen equalizer-type graph for visualizing multiple on-screen values. An additional sub-tab shows in the **Properties** viewport, allowing further display customizations.

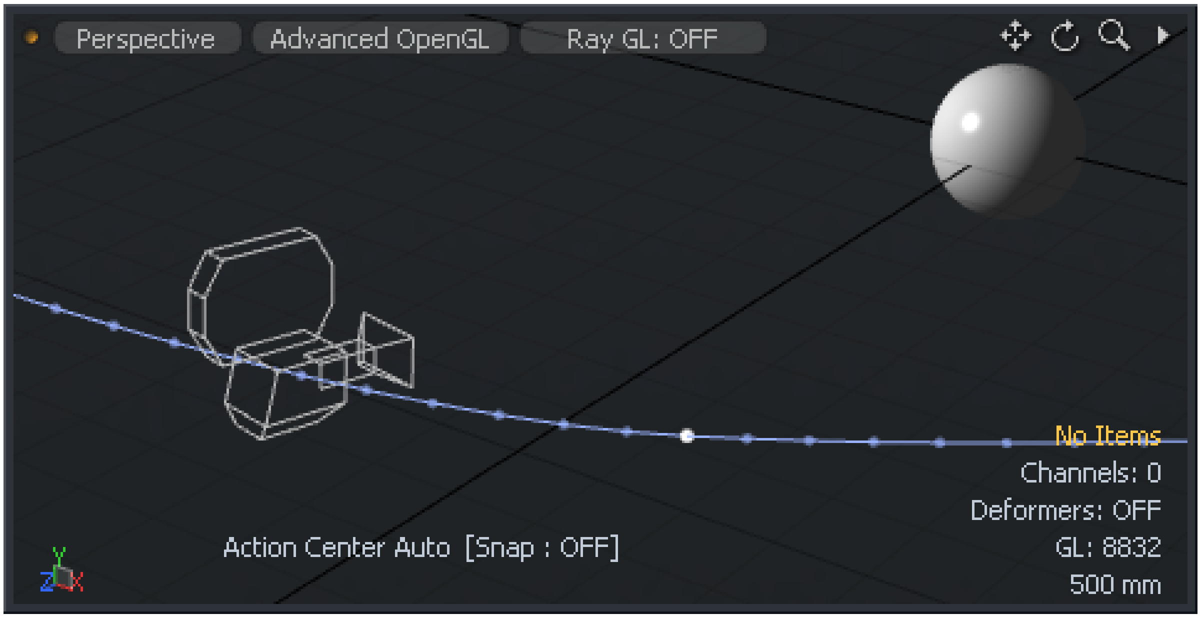
Note: For more details on display customization, see Additional Draw Options.

- **Add OpenSubdiv DrawOver** - Enables the OSD drawing override, which draws Pixar subdivision surface using OpenSubdiv 3.0 instead of native Catmull-Clark and Subdivision polygon mesh drawing. This improves the drawing speed for high subdivision levels. When this package is added to mesh, instanced mesh or group item, The OSD draw package displays the subdivision surfaces instead of the standard drawing. The supported drawing styles are:
  - Shaded (in all solid styles)
  - Wireframe (wireframe style)
  - Wireframe Overlay (wireframe on shaded model)

Note: The OSD drawing override doesn't support texture mapping and the adaptive subdivision of OSD.
In case of meshes consisting of mixed polygon types, other types of polygons, such as faces or curves, are not drawn.

- **Enable** - toggle the display of the **Draw Options** without losing values.
- **Draw Style** - allows you to choose a specific draw style for an individual Mesh Item; **Default** respects viewport settings.
- **Wireframe** - disabling the **Wireframe** checkbox disables the wireframe overlay on 3D Mesh Items.

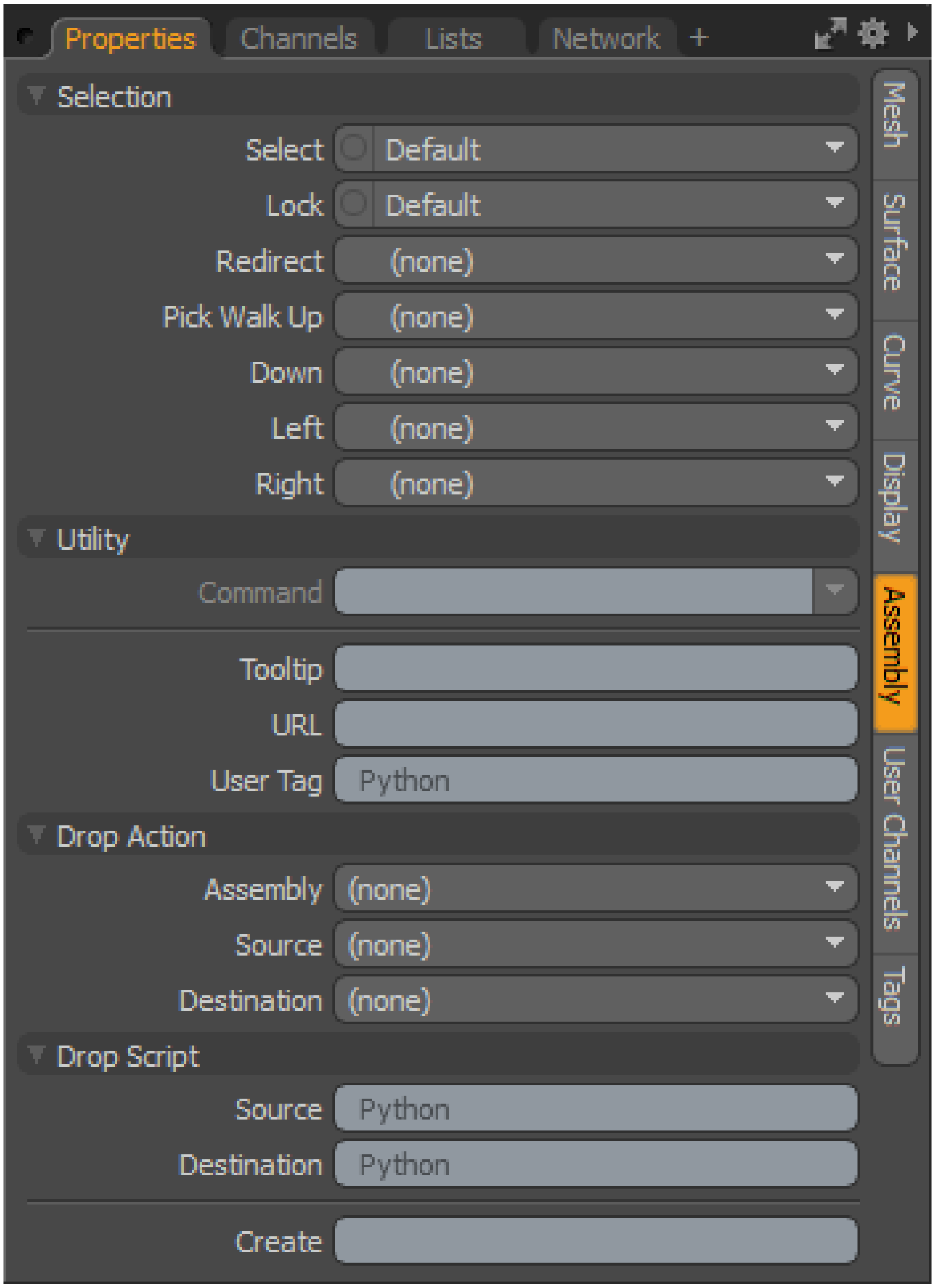
- **Smooth** - disabling the **Smooth** checkbox disables the smoothing applied to polygon surfaces, making polygons appear faceted.
- **Texture** - controls the display of texture layers on a **Mesh Item**. **Off** disables texture display and **Texture** draws the texture without shading (this can make it easier to see a texture for painting), and **Shaded Texture** takes into account the shading of the surface to which the texture is applied.
- **Wireframe** - adjusts the draw color of the wireframe overlay of the 3D item in the 3D viewport.
- **Wireframe Color** - when **Wireframe** is set to **Custom**, the **Wireframe Color** sets the color of the wireframe.
- **Fill** - if **Draw Style** is set to **Solid**, this value controls the color of the 3D item's fill.
- **Fill Color** - when **Fill Color** is set to **Custom**, this value specifies the RGB color of the fill of the **Solid** in the 3D viewport.



- **Add/Remove Motion Path** - to display the motion path for specific items within a scene (shown above for the **Camera** item for illustrative purposes), you can select the **Add Motion Path** option, revealing additional properties and controlling the display of the path in the 3D GL viewport.
- **Enable** - provides you with a means to toggle the display of the path **On** or **Off**, or to have the path only visible when the item is **Selected**.
- **Time In/Out** - the **Time In** option determines the number of frames before the current time in the Timeline that are displayed by the curve; **Time Out** determines the number of frames after the current time that are displayed.
- **Show Frames** - when enabled, displays the position of the element at each frame as a small dot along the path. This can be useful to see how the element moves in time, but can also get busy for some types of motion and can therefore be disabled.

- **Show Keys** - when enabled, displays the position of the element at each keyframe as a white dot along the path.
- **Draw Scale** - determines the size of the resulting motion path in the 3D viewport.

Assembly Tab


Properties
Channels
Lists
Network
+
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

Selection	
Select	Default
Lock	Default
Redirect	(none)
Pick Walk Up	(none)
Down	(none)
Left	(none)
Right	(none)
Utility	
Command	
Tooltip	
URL	
User Tag	Python
Drop Action	
Assembly	(none)
Source	(none)
Destination	(none)
Drop Script	
Source	Python
Destination	Python
Create	

The following **Selection** options are available for the **Mesh Item**:

- **Select** - controls whether or not items are selectable within any 3D viewport: **Yes** enables direct item selection, **No** disables it. Items designated **No** can still be selected via the Items (Scenes) List.

- **Lock** - when enabled, items are locked from the application of any type of item transform.
- **Redirect** - when an item is selected in a 3D viewport, specifying an item for **Redirect** triggers Modo to automatically select the alternate item instead. This is helpful in easing item selections in complex hierarchies. Locators with special Display Viewport properties can be used visualize selectable items in the viewport that redirect to the actual item you wish to modify.
- **Pick Walk** - the arrow keypad can be used as a way to **walk** selections through a hierarchy. By default, parent items are selected by pressing **up**, and child items are selected by pressing **down**. The **Pick Walk** function allows you to specify which items, specifically, are selected by any of the four keys: **Up**, **Down**, **Left**, and **Right**. For instance, when you reach the bottom of a hierarchy, the foot, you could specify (by pressing the left or right arrows) the opposing foot would be selected, and by hitting down the head would be selected, and so forth.

The following **Utility** options are available for the **Mesh Item**:

- **Command** - specifies a command to fire when the item is clicked on in the 3D view. This can be any of Modo's commands. There are two commands that are especially useful when setting up assemblies:
  - **item.channelHaul** - this selects any user channels on the item and activate the **Channel Haul** tool.
  - **item.channelPopover** - this displays a pop-over form, containing controls for any user channels on the item. If used in conjunction with selection directions, this can also be used to display the channels belonging to a specific group.

Tip: The **Command** function is disabled on Mesh Items. To enable the option, you can convert a mesh to a **Static Mesh** by right-clicking on the item's layer name in the **Item List** and selecting the **Change Type** > **Static Mesh** option from the contextual menu. Note that converting a mesh to a static mesh is a destructive action, so make sure you have a backup copy should further editing be necessary to the Mesh Item.

- **Tooltip** - small blocks of text that appear, if assigned, when the pointer momentarily stays over an item. These can be notes or reminders to what an item is intended to do in a scene but tooltips can also be useful for an assembly author to provide some contextual instructions on how to work with an item in an assembly.
- **URL** - if a custom URL is assigned to an item, using the **F1** help feature and clicking the item in the 3D viewport opens that particular location. This is useful for you to add custom documentation to an item.
- **User Tag** - a text string that can be used to identify particular items within an assembly or scene. These tags can be read by scripts with the **item.userTag** command. An example of use might be to identify items to be deleted or hidden when an assembly script has finished.

The following **Drop Action** options are available for the **Mesh Item**:

Assemblies offer you the ability to create a rig using Modo's many animation modifiers and then save the setup as a preset for easy application elsewhere.

Tip: You can drag and drop items over each other directly in the 3D viewport, pausing over an item automatically opens the drop action popup to customize the drop action.

In the following descriptions, the item being dragged is referred to as the **source** item and the item that has received the drop is referred to as the **Destination** item.

- **Assembly** - you can choose from a series of actions that are performed when the preset is initially dropped into the scene from the **Preset Browser**. The possible actions are as follows:
  - **Parent** - the source item is made a child of the destination item.
  - **Parent in Place** - the source item is parented as above, but with **Compensation** applied.
  - **Match** - the source item is modified to match position, scale, and rotation of destination.
  - **Match Position** - the source item is modified to match the position only of the destination item.
  - **Match Rotation** - the dropped item is modified to match the rotation only of the destination item.
  - **Match Scale** - the dropped item is modified to match the scale only of the destination item.
  - **Insert** - inserts the source item into the hierarchy of the destination item. The item is positioned and orientated to match the destination item and becomes the parent of the destination item.
  - **Insert at Parent** - similar to **Insert** but the source item matches its position and orientation to the destination item's parent (if a parent item is present).
  - **Insert in Place** - the source item is inserted into the destination item's hierarchy as its parent but retain its current position and orientation.
  - **Place** - positions the selected item at the intersection of the mouse pointer and the surface.
  - **Place and Align** - positions the selected item at the intersection of the mouse pointer and surface, rotating the item to match the surface's normal direction.
- **Source** - the source **Drop Action** is performed when the assembly item is dropped onto a scene item. The same action options are available as for **Assembly**.
- **Destination** - as above, but the action is performed when a scene item is dropped onto the assembly item. The same action options are available as for **Assembly**.

Additional to the above actions, scripts can be assigned to run when certain actions occur:

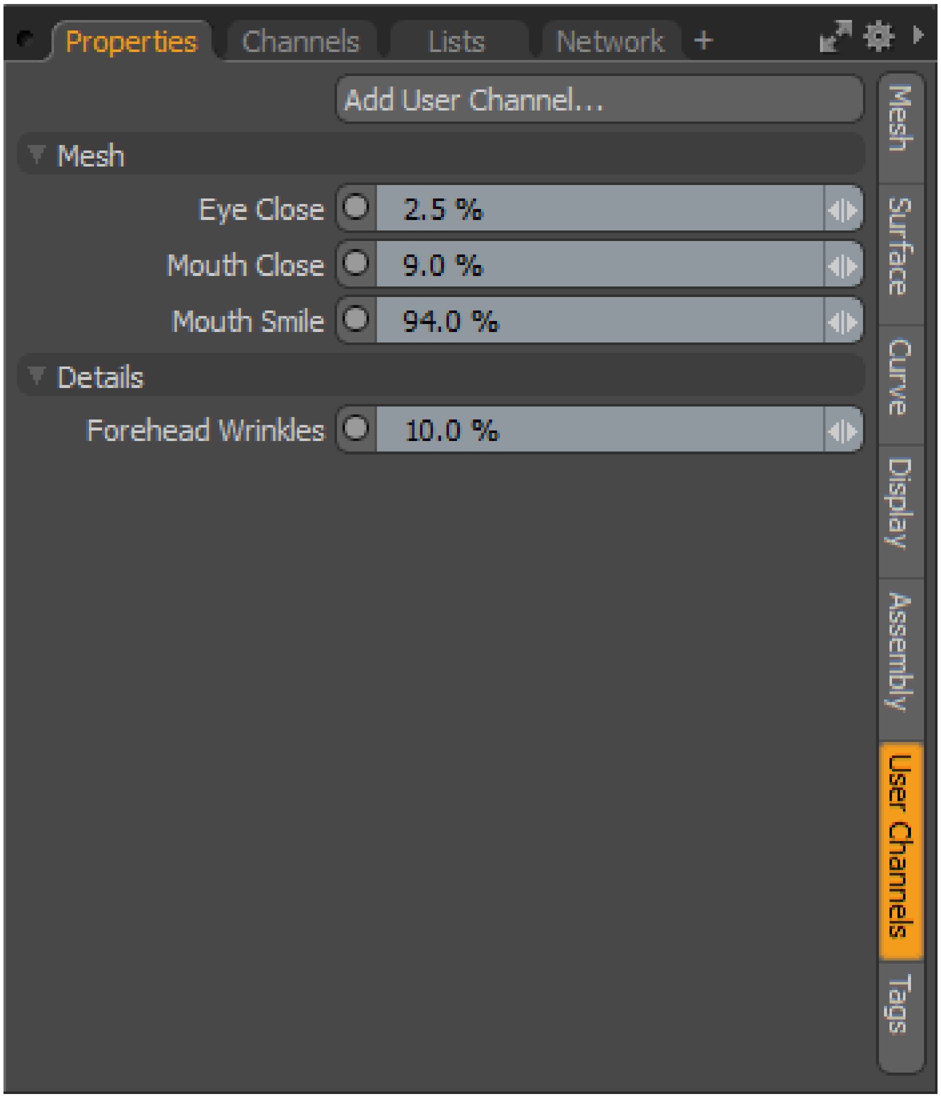
- **Source** - the assigned script (omitting the @) run when this item is dropped onto another in the scene. The script gets the source item and destination item passed as arguments. If multiple items are dropped, then the script is called for each item in turn.
- **Destination** - same as above but the script runs when any item in the scene is dropped onto the item it's assigned to. As with the source **Drop Script** the source and destination items are passed to the script as arguments.

- **Assembly** - you can choose from a series of actions that are performed when the preset is initially dropped into the scene from the **Preset Browser**. The possible actions are as follows:
    - **Parent** - the source item is made a child of the destination item.
    - **Parent in Place** - the source item is parented as above, but with **Compensation** applied.
    - **Match** - the source item is modified to match position, scale, and rotation of destination.
    - **Match Position** - the source item is modified to match the position only of the destination item.
    - **Match Rotation** - the dropped item is modified to match the rotation only of the destination item.
    - **Match Scale** - the dropped item is modified to match the scale only of the destination item.
    - **Insert** - inserts the source item into the hierarchy of the destination item. The item is positioned and orientated to match the destination item and becomes the parent of the destination item.
    - **Insert at Parent** - similar to **Insert** but the source item matches its position and orientation to the destination item's parent (if a parent item is present).
    - **Insert in Place** - the source item is inserted into the destination item's hierarchy as its parent but retain its current position and orientation.
    - **Place** - positions the selected item at the intersection of the mouse pointer and the surface.
    - **Place and Align** - positions the selected item at the intersection of the mouse pointer and surface, rotating the item to match the surface's normal direction.
- **Source** - the source **Drop Action** is performed when the assembly item is dropped onto a scene item. The same action options are available as for **Assembly**.
- **Destination** - as above, but the action is performed when a scene item is dropped onto the assembly item. The same action options are available as for **Assembly**.

Additional to the above actions, scripts can be assigned to run when certain actions occur:

- **Source** - the assigned script (omitting the @) run when this item is dropped onto another in the scene. The script gets the source item and destination item passed as arguments. If multiple items are dropped, then the script is called for each item in turn.
- **Destination** - same as above but the script runs when any item in the scene is dropped onto the item it's assigned to. As with the source **Drop Script** the source and destination items are passed to the script as arguments.

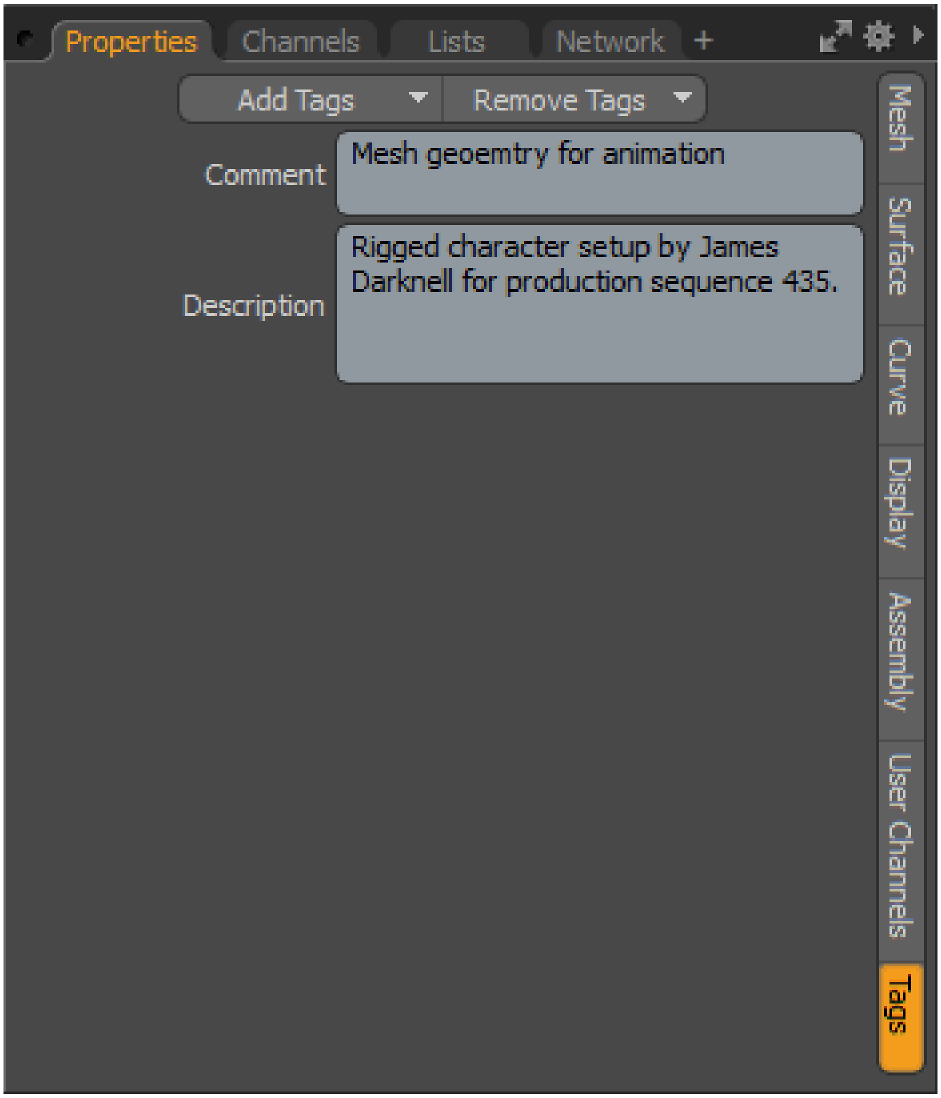
User Channels Tab


Properties
Channels
Lists
Network
+
Add User Channel...
Mesh
Eye Close
2.5 %
Mouth Close
9.0 %
Mouth Smile
94.0 %
Details
Forehead Wrinkles
10.0 %
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

- **Add User Channel** - user channels are useful in rigging to make simplified controls for complex actions, or to simply give a name to a specific operation. The **Add User Channel** option allows you to add an unlimited number of user-defined **Channels** to an item. Once a user chanel is defined, it's necessary to connect the channel using the **Schematic** viewport, in order to make it operable. Pressing the **Add User Channel** button opens the **Create User Channel** dialog, where the actual channel is defined. This panel is fully covered in the Add User Channels topic.

As an example in the panel above, a number of user channels have been added and named by morph maps that are linked to the channels, allowing direct animation access. Other user channels are added that create dividers between the groups. Finally, another channel was also added and linked to an images opacity channel to control the amount of displacement. All of the defined user channels are then added to a channel group so they can remain on the screen for ease of animation.

Tags Tab


Properties
Channels
Lists
Network
+
Add Tags
Remove Tags
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

Comment: Mesh geoemtry for animation

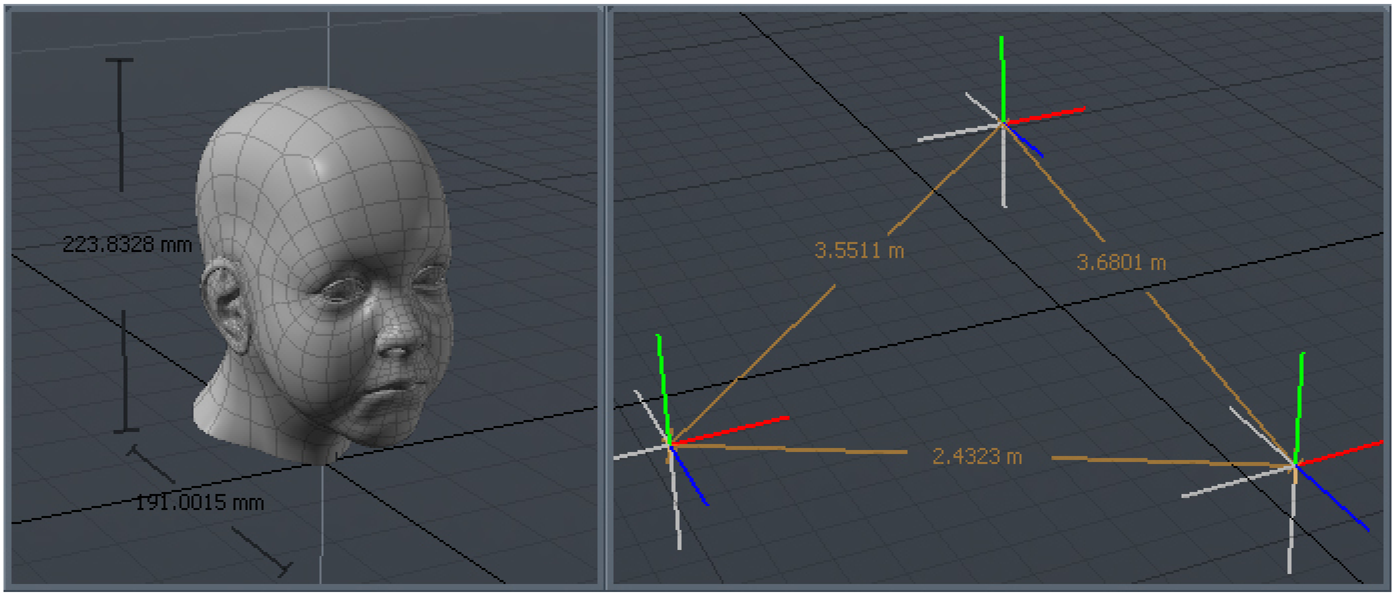
Description: Rigged character setup by James Darknell for production sequence 435.

The **Tag** panel allows you to define various tags that remain associated to the specific item. These are typically text strings and are useful to add notes to specific elements for other users in a studio, or as reminders as to how something was meant to be used. There are a few pre-defined tags you can choose from or you can create your own specific tag definition.

Adding tags is done by pressing the **Add Tags** button, opening the pop-up menu and then selecting a tag type from the **Add Tags** menu. There are also options to **Add Custom Tags**, and **Edit Tag** definition, opening the appropriate panel in the Modo preferences.

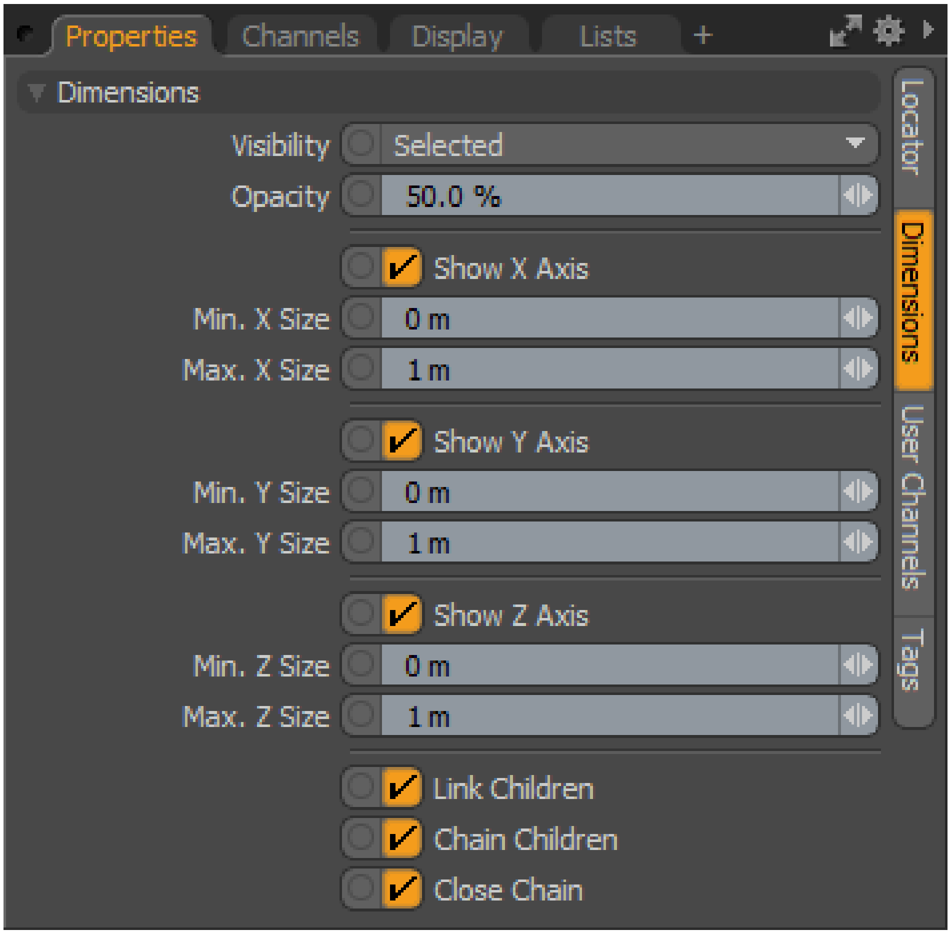
Additional Draw Options

For all item types there are additional GL drawing properties that allow you to customize the UI for rigging in a number of ways, covered below in more detail. Options are toggled on/off with associated buttons in the **Display** sub-tab when the target item is selected. The action creates an additional sub-tab that appeares alongside the others in the **Properties** panel.



- **Show/Hide Dimensions** - when this option is enabled, an additional sub-tab appears in the **Properties** panel titled **Dimensions**. By default, dimension lines and values for the overall bounding box size appear in the viewport around a **Mesh Item**. The attributes in the **Properties** panel allow you to further customize the information that is displayed. Linking a series of items together through a hierarchy allows the display of distances between each object's **Center** point, from parent to child (and only when the parent item is selected the dimension values display).

Dimensions Tab



Visibility: Selected

Opacity: 50.0 %

☑ Show X Axis

Min. X Size: 0 m

Max. X Size: 1 m

☑ Show Y Axis

Min. Y Size: 0 m

Max. Y Size: 1 m

☑ Show Z Axis

Min. Z Size: 0 m

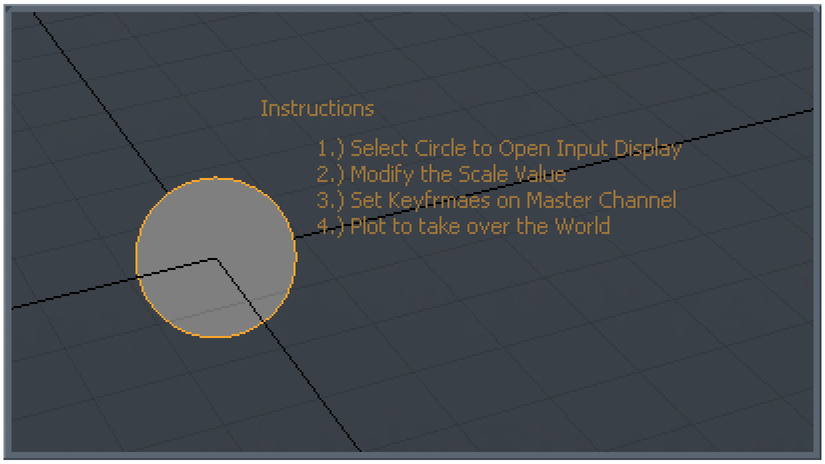
Max. Z Size: 1 m

☑ Link Children

☑ Chain Children

☑ Close Chain

- **Visibility** - determines the visibility of the dimension lines to **Off** (not visible), **On** (visible), and **Selected** (only visible when selected).
- **Opacity** - controls the transparency of the dimension lines and numerical values displayed in the 3D viewport.
- **Show X/Y/Z Axis** - with these channels you can individually control the display for each axis.
- **Min X/Y/Z Size/Max X/Y/Z Size** - the minimum and maximum values can be assigned for each axis and, when the defined values for either **Min** or **Max** are reached, the dimension text displays in red in the viewport.
- **Link Children** - when set to true (the default option), draw the link lines to the child items. This is used to disable if the dimension line linking behavior is undesirable.
- **Chain Child** - when set to true, draw dimension lines from child to child in hierarchical order instead of from parent to each individual child.
- **Close Chain** - when set to true, draw an additional dimension line link back to the parent item from the last child item.



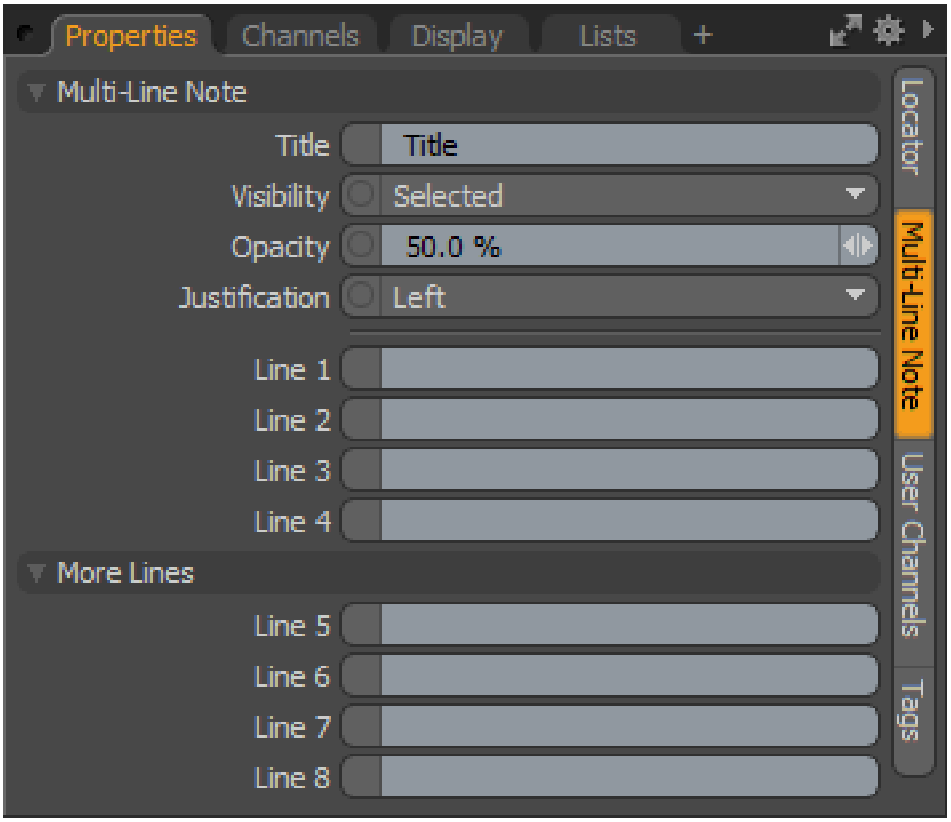
Instructions

1.) Select Circle to Open Input Display
2.) Modify the Scale Value
3.) Set Keyfrmaes on Master Channel
4.) Plot to take over the World

- **Show/Hide Notes** - when this option is enabled, an additional sub-tab appears in the **Properties** panel titled **Multi-Line Note**. This option allows for on-screen display of custom text strings, for information such as instructions. The text is useful to Riggers and TDs, or just to add some helpful reminders for an item when selected.

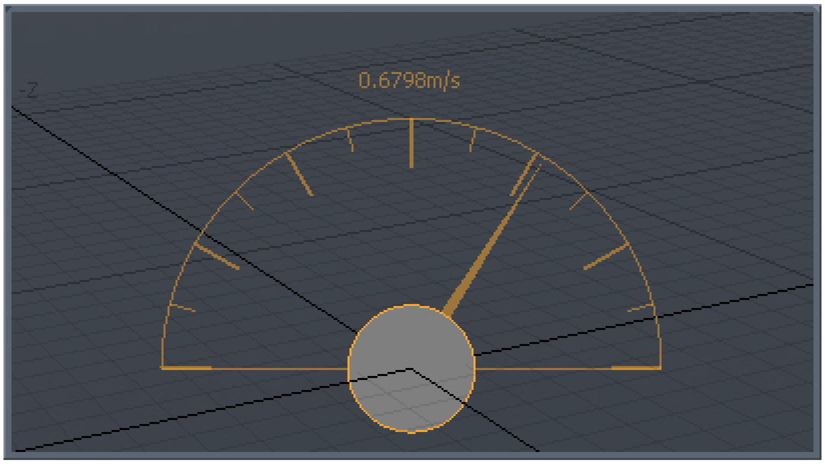
Note: Display text can also be positioned with the **Label Offset** control.

Multi-Line Note Tab


Properties
Channels
Display
Lists
+
Multi-Line Note
Locator
Multi-Line Note
User Channels
Tags
More Lines

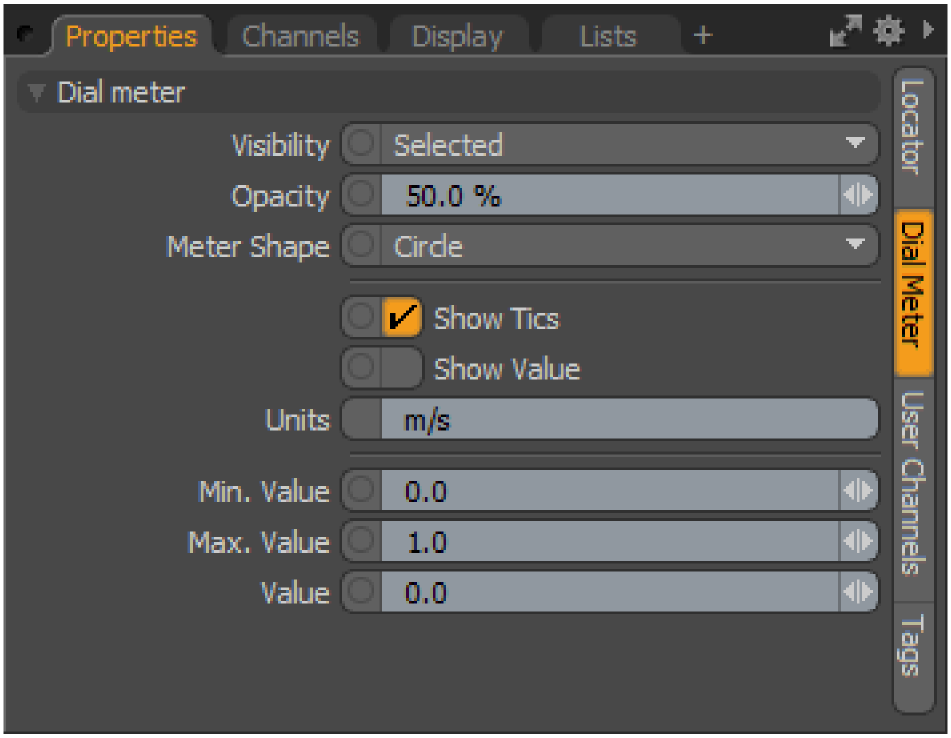
Title: Title
Visibility: Selected
Opacity: 50.0 %
Justification: Left
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8

- **Title** - the first line of text that display above the **Line** entries.
- **Visibility** - determines the visibility of the dimension lines to **Off** (not visible), **On** (visible), and **Selected** (only visible when selected).
- **Opacity** - controls the opacity of the dimension line and numerical value displays in the 3D viewport.
- **Justification** - determines the justification of the text block to **Left**, **Center**, or **Right**.
- **Line 1-8** - the text line entries that display in the 3D viewport.



- **Show/Hide Meter** - when this option is enabled, you can display an on-screen analog-style gauge graph (looks like a speedometer) that can be rigged to show an individual value in the 3D viewport. An additional sub-tab appears in the **Properties** panel, titled **Dial Meter**, which allows some customization of the display.

Dial Meter Tab



Dial meter

Visibility: Selected

Opacity: 50.0 %

Meter Shape: Circle

☑ Show Tics

☐ Show Value

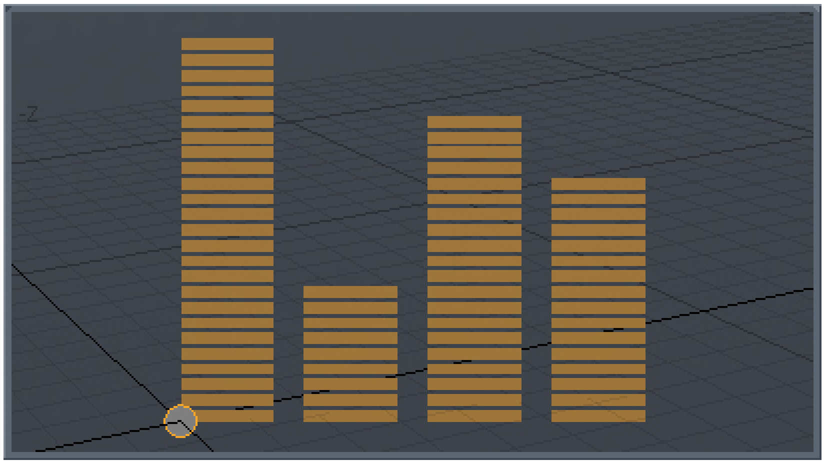
Units: m/s

Min. Value: 0.0

Max. Value: 1.0

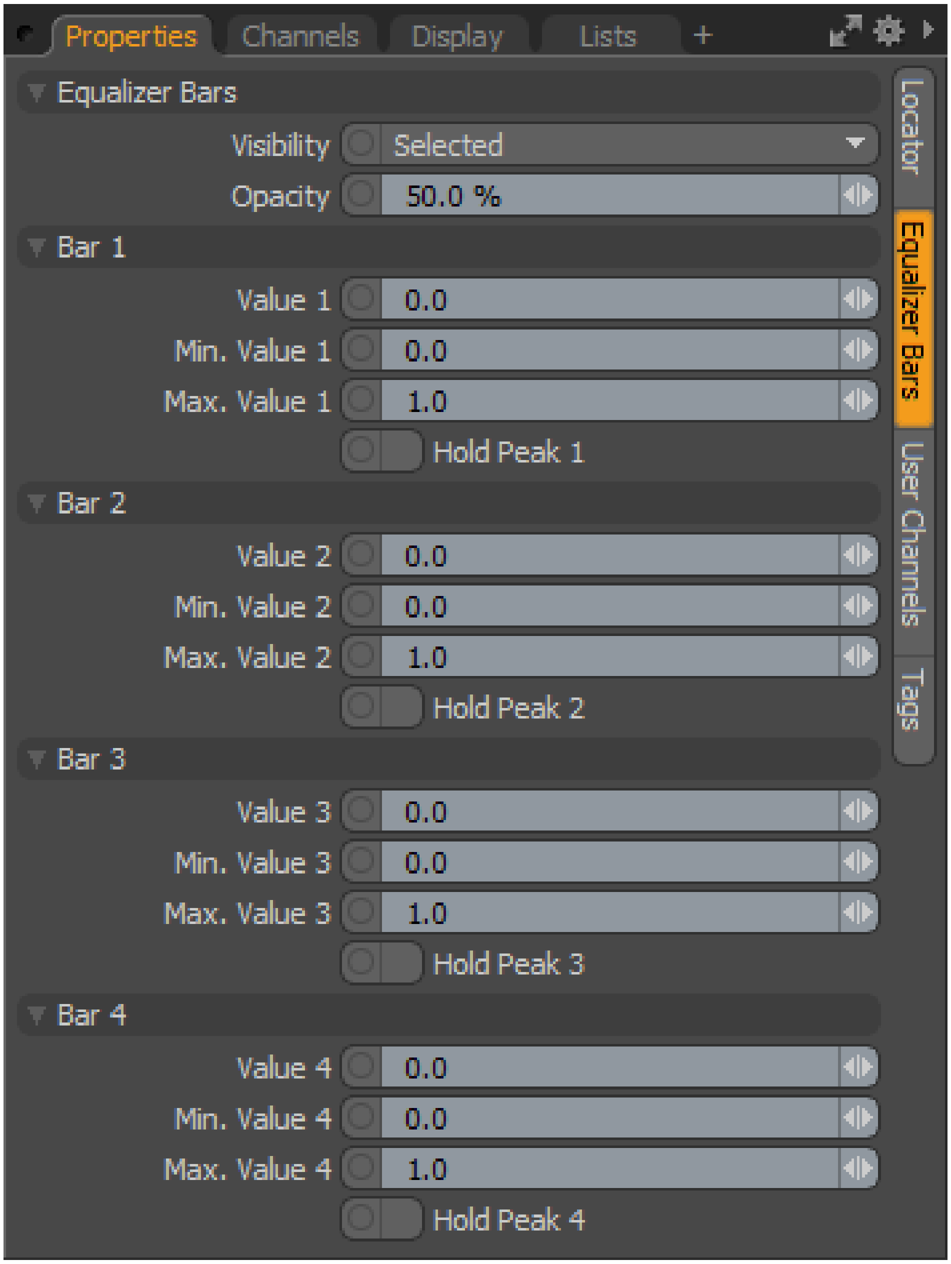
Value: 0.0

- **Visibility** - determines the visibility of the meter display to **Off** (not visible), **On** (visible), and **Selected** (only visible when selected).
- **Opacity** - controls the opacity of the meter readout and numerical value displays in the 3D viewport.
- **Meter Shape** - choose between a full **Circle** meter display and a **Half Circle** display.
- **Show Tics** - toggles the visibility of 15° and 30° interval lines in the display.
- **Show Value** - toggles the visibility of the **Value** information readout above the dial.
- **Units** - a text display that is positioned next to the **Value** display when the **Show Value** option is set to true.
- **Min**/**Max Value** - the minimum and maximum values determine the lowest value and the highest value that is displayed in the on-screen meter, controlling the overall position of the **Value** needle.
- **Value** - this is the input channel that determines the actual value displayed in the viewport, as well as the position of the meter needle (relative to the **Min** and **Max** values). Additional rigging is required in the Schematic viewport in order to connect the **Value** channel input from an output.



- **Show/Hide Level Bars** - when this option is enabled, an additional sub-tab appears in the **Properties** panel titled **Equalizer Bars**. This option is used to display an on-screen equalizer-style bar graph for visualizing multiple on-screen values relative to one another. Additional channels show in the Channels viewport, when enabled.

Equalizer Bars Tab



Equalizer Bars

Visibility: Selected

Opacity: 50.0 %

Bar 1

Value 1: 0.0

Min. Value 1: 0.0

Max. Value 1: 1.0

○ Hold Peak 1

Bar 2

Value 2: 0.0

Min. Value 2: 0.0

Max. Value 2: 1.0

○ Hold Peak 2

Bar 3

Value 3: 0.0

Min. Value 3: 0.0

Max. Value 3: 1.0

○ Hold Peak 3

Bar 4

Value 4: 0.0

Min. Value 4: 0.0

Max. Value 4: 1.0

○ Hold Peak 4

- **Visibility** - determines the visibility of the Level Bars to **Off** (not visible), **On** (visible), and **Selected** (only visible when selected).
- **Opacity** - controls the opacity of the level bars readout display in the 3D viewport.
- **Value 1-4** - these are the input channels that determine the actual values displayed in the viewport, as well as the scale of the individual level bars (relative to the **Min** and **Max** values). Additional rigging is required in the Schematic viewport in order to connect the **Value** channel input from an output.

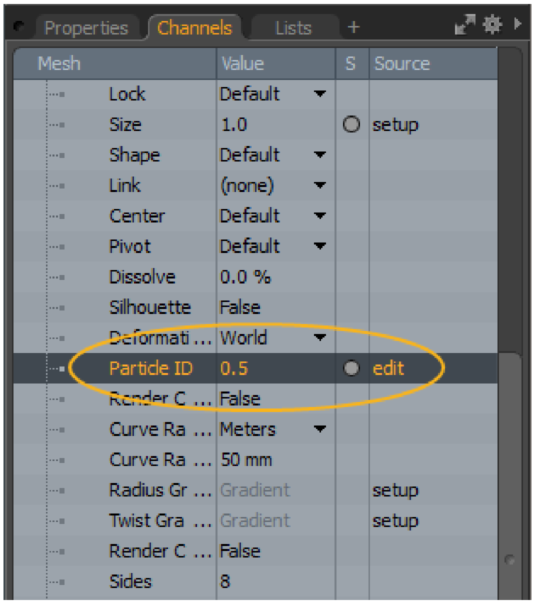
- **Min Value 1-4**/**Max Value 1-4** - the minimum and maximum values determines the lowest value and the highest value, per readout, that is displayed in the on-screen level bar, controlling the overall scale of the bar displayed.
- **Hold Peak 1-4** - like an audio readout, when the **Max** value is reached for any value readout, an indicator is drawn in the viewport. Toggle the **Hold Peak** value to reset the display.

Note: Up to eight bars total can be displayed by referencing the additional channels available in the **Channels** viewport.

Particle ID Channel

Individual items have a Particle ID channel that you can use to control the particle ID assigned to rendered surfaces. You can set this value manually, or drive it by a rig and then feed it to Shaders.

The Particle ID channel allows you to assign a value between 0.0 and 1.0 to a mesh item. You can specify it in the **Channels** list of the mesh item:



Mesh	Value	S	Source
Lock	Default		
Size	1.0	○	setup
Shape	Default		
Link	(none)		
Center	Default		
Pivot	Default		
Dissolve	0.0 %		
Silhouette	False		
Deformati ...	World		
Particle ID	0.5	●	edit
Render C ...	False		
Curve Ra ...	Meters		
Curve Ra ...	50 mm		
Radius Gr ...	Gradient		setup
Twist Gra ...	Gradient		setup
Render C ...	False		
Sides	8		

This channel is useful when using the Gradient or the Texture Switch node. Gradient values range from 0 to 100%, so a Particle ID of 0.5 corresponds to 50%, which results in a mesh with the color in the middle of a Gradient.

For Texture Switch, values are based on the order of connection. So if you have three textures, the first one is 0.0, the second connected texture is 0.5, and the third one is 1.0.

Replicator Item

Replicators are a means of generating render time duplicates of geometry in a scene, allowing you to render billions of polygons with full support for all of Modo's shading and rendering capabilities, such as global illumination, soft shadows, and refractions.

Replicators are similar in concept to Modo's instances. They derive all their shape and shading information from a base mesh, but unlike an instance, replicas can't be manipulated directly. You cannot position, rotate or scale individual replicas with the usual transform tools. Replicators generate positions using a mesh item as a point source, utilizing the vertex positions to place the individual replicas. Their positioning can also be further manipulated by adding a surface particle generator item, allowing for a greater degree of randomization and overall control. For more information, see CSV Point Cache.

Replicators require two mesh items, the first being a point source item, and the other is a prototype. The point source object acts as the surface, determining the position of the replicated objects, and the prototype is the object that is duplicated. Consider the example of a dry creek bed littered with river stones. The creek bed is the point source determining the placement of the stones and the stones themselves are the prototypes. You can add variety to a Replicator by adding multiple mesh items to a group defined in the Groups Viewport and assigning that as the prototype instead of the single mesh item. When specified as such, Modo randomly grabs a member of the group to replicate, providing additional variety to the cloned objects.

Adding a Replicator

1. Select the point source item in the **Items** list, hold **Ctrl**, and click the prototype item, so both items are highlighted.

Note: Selection order is important.

2. In the menu bar, click **Item** > **Replicators** > **Add Replicator**.

   A **Replicator** item is added to the **Items** list and a series of bounding boxes are positioned over the surface of your point source geometry. Each bounding box represents the individual replicas.

Tip: If you'd prefer to set up your replicator manually, or just wish to understand what the **Add Replicator** function is doing, follow these steps.

1. Click **Add Item** at the top of the **Items** list, then choose **Particles** > **Replicator** from the list.

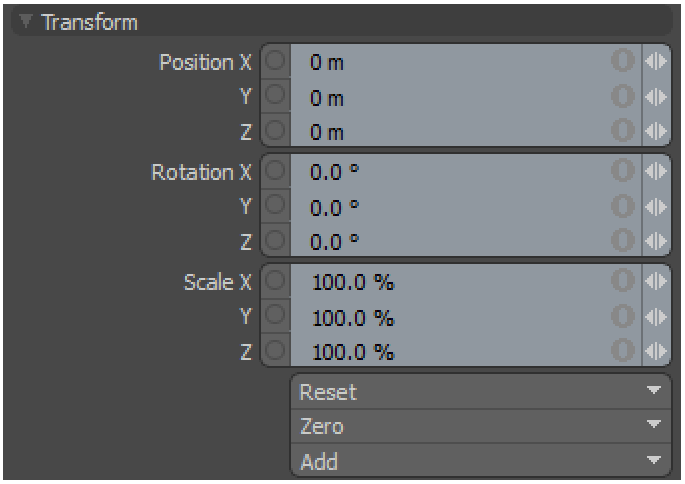
Tip: Selecting the **Replicator** item itself displays its attributes in the **Property** panel.

2. In the **Instancing** section of the Replicator's properties, choose the appropriate geometry in the **Point Source** and **Prototype** selectors.

There are a number of other settings that allow you to add some random variations to the replicas. Further randomization is possible through the use of a surface particle generator. You can right-click on the **Replicator** item and choose the **Scatter Replicas** option from the pop-up menu to assign one, adding the particle layer to the **Items** list. For additional options and details related to Replicator usage, see Replicators.

Replicator Properties

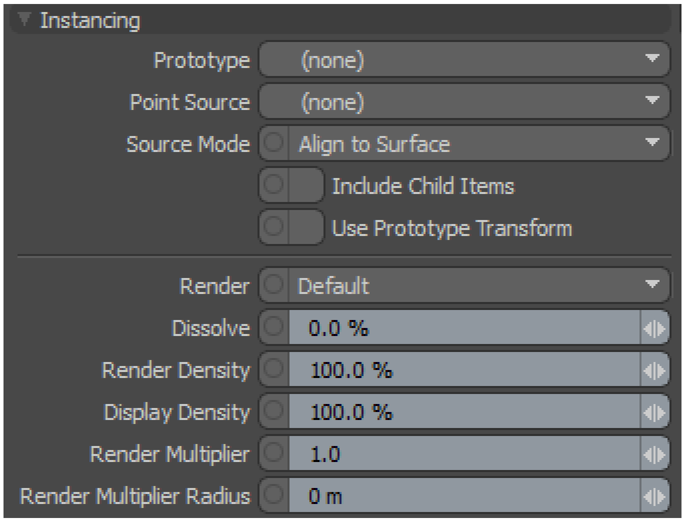
Transform


Transform
Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.
Transform	

Position	An item transform that allows you to numerically position the item in XYZ space. By default, position transforms originate from the center position. Modifications to the position of the Replicator act as an offset from the original point source defined positions, allowing you to transform the entire array.
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, rotation transforms originate from the center position. Modifications to the rotation of the Replicator act as an offset from the original point source defined positions, allowing you to transform the entire array.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale transforms originate from the center position. Modifications to the position of the Replicator act as an offset from the original point source defined positions, allowing you to transform the entire array.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and item position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. You can simply transform the target item with one of the various transform tools, or by editing the value input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the channel list while keeping the default **0,0,0** values. A necessary step for referencing, in order to override the channels, they must first exist.

Instancing



Instancing

Prototype (none)

Point Source (none)

Source Mode Align to Surface

Include Child Items

Use Prototype Transform

Render Default

Dissolve 0.0 %

Render Density 100.0 %

Display Density 100.0 %

Render Multiplier 1.0

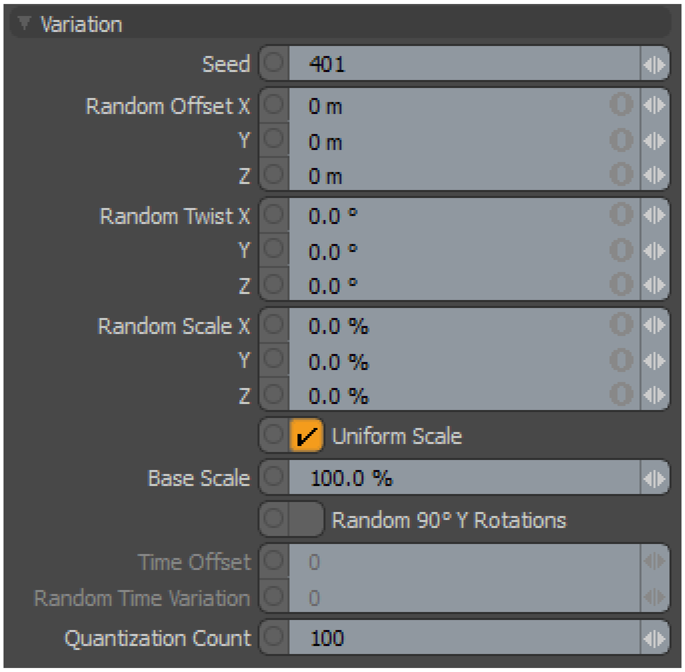
Render Multiplier Radius 0 m

Prototype	The **Prototype** object is the Mesh Item that is duplicated over the point source. This can be something as simple as a single polygon or cube, or something as complex as a tree or high-rise office building. If your object is made up of multiple Mesh Item layers, make sure to parent the items together in a hierarchy and select the **Include Child Items** toggle. This also works for animated items.
Point Source	The **Point Source** geometry is the surface that controls the positioning of prototype items. The default behavior is a replica generated for each vertex of the point source geometry. Each prototype is aligned with each point source vertex, based on the point of origin for the prototype (not the pivot) and the positive-Y of the prototype is aligned with the vertex normal direction.
Source Mode	There are four different source modes Modo can use to generate replicas: • **Point Data** - uses the position of each vertex in a mesh, but the prototype retains its original orientation. • **Align to Surface** - uses the vertex position as well as the normal direction to align each replica, also taking into account subdivision surface smoothing. • **Use Polygons** - generates a replica for each polygon, positioned at the center of each polygon and aligned with the polygon's normal direction. **Note:** If you need to generate replicas over a surface with a displacement map, you need a surface generator.

	• **Use Polygon with Connectivity** - generates a replica for each polygon, positioned at the center of each polygon and aligned with the polygon's normal direction. Additionally, it aligns **X** and **Z** direction of the particle axis to neighbor connecting quad polygons. This is useful when you want to duplicate object along a strip of polygons. For example, telegraph poles along a street or stitching on a shoe.
Include Child Items	When enabled, Modo replicates a parent/child hierarchy. When disabled, only the parent item is replicated. If your prototype geometry consists of only a single Mesh Item layer, you can leave this setting disabled.
Use Prototype Transform	By default, replicated elements are not affected by item transforms of the source Prototype, their positions are strictly defined by the vertex positions of the point source relative to the source's center position. This means that position, rotation and scale changes applied to the base item don't translate to the replicated elements. When enabled, the source item transforms affect the replicated elements, each offset the same value from their initial point source positions.
Render	This dropdown menu allows you to select from 3 options. When set to **Default**, you can enable/disable Mesh Item using the visibility column (👁) of the **Items** list. When the mesh layer is visible, it contributes to the final rendered scene and when invisible, it doesn't. In some cases you may prefer to fix this state, setting the mesh as **On** (enabled) or **Off** (disabled) regardless of visibility. This is useful for workflows that auto-toggle visibility: you don't have to manually enable mesh layers for test renders.
Dissolve	Fades all replicated items from view. At 100%, the item renders normally, while values lower than 100% fade all replicas ramping toward 0%, where the replicated items become completely invisible.
Render Density	Controls the number of generated replicas. Normally, an item is generated at each vertex position, so for a 10x10 grid, you get 121 vertices, therefore 121 replicas. Setting the **Render Density** to 50%, the Replicator randomly generates replicas reducing the overall number by the approximate density value. Values below 100% reduce the number of replicas, however, values above 100% don't generate additional replicas. You may wish to use a 'Surface Generator' to provide better control over the number of generated items.
Display Density	Only the bounding box of replicated items is displayed in the 3D viewport. If the number of replicated items becomes too great, you may wish to reduce the number of bounding boxes displayed by adjusting this value. A setting of 0% disables the bounding box's display without affecting the final rendered image.
Render	The **Render Multiplier** and **Render Multiplier Radius** settings allow you to add extra

Multiplier	replicas for each point and scatter them. This field specifies how many copies of the object you want to scatter. **Note:** The additional copies can only be seen on the render.
Render Multiplier Radius	Specifies the size of the area in which the copies are scattered.

Variation



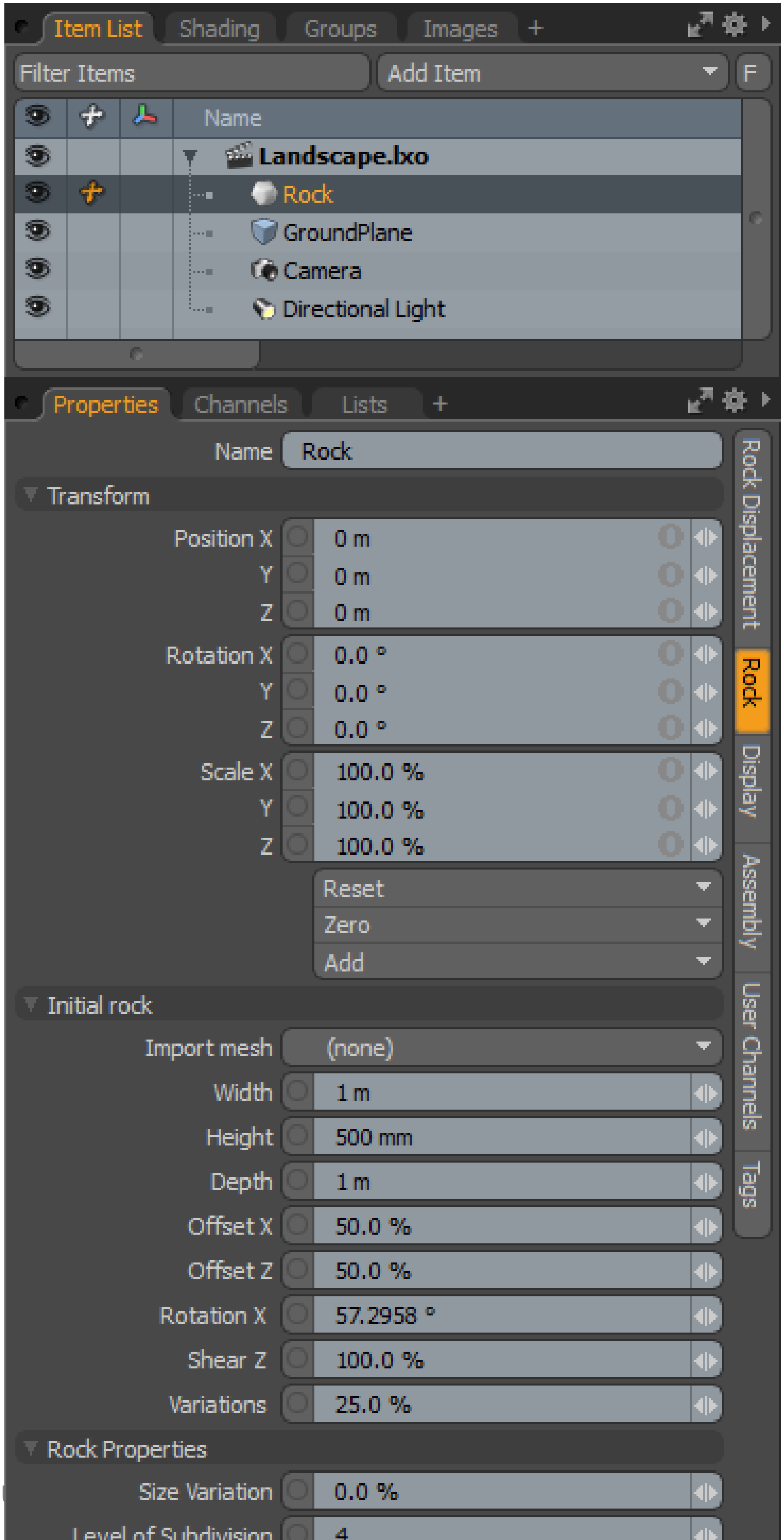
Variation
Seed 401
Random Offset X 0 m
Y 0 m
Z 0 m
Random Twist X 0.0 °
Y 0.0 °
Z 0.0 °
Random Scale X 0.0 %
Y 0.0 %
Z 0.0 %
☑ Uniform Scale
Base Scale 100.0 %
☐ Random 90° Y Rotations
Time Offset 0
Random Time Variation 0
Quantization Count 100

Seed	This value is an initial setting used when random numbers are generated. Different **Seed** values produce different random variations, however you want to use the same **Seed** value when it is necessary that items retain the same variations.
Random Offset	When set to any number above 0 for X, Y or Z, it provides a range from where random values can be generated. For example, with an offset value of 1m, the replicated item is randomly offset up to 1 meter from its initial position. Values of 0 disable this functionality.

Random Twist	When set to any number above 0 degrees for X, Y or Z, it provides a range from where random values can be generated. For example, with an twist value of 90 degrees, the replicated items are randomly rotated 90 degrees in either direction from initial position. Values of 0 disable this functionality.
Random Scale	When set to any amount above 0%, produces a random scaling amount anywhere within the range for the value specified. For example, with a setting of 50%, replicated items vary anywhere between 50% to 150% of the item's original size. Values of 0 disable this functionality.
Uniform Scale	When using **Random Scale** values, all three axes are generated independently. When **Uniform Scale** is enabled, random scaling of generated replicas would be uniform across all axes eliminating squashing or stretching.
Base Scale	Scales the source object before **Random Scale** is applied, so the **Random Scale** value is adjusted based on the scale of the original object. At the default 100%, replicas are scaled either larger or smaller than the original object. Below 100% replicas are scaled smaller, and above 100% replicas are scaled larger than the original object.
Time Offset	Offsets the playback of an animation applied to replicated objects. It can also be used to move the random span of time when combined with **Random Time Variation**. For example, if this is set to 5 seconds and the variation is set to 1 second, then the replicas get their motion from the animation of the prototype between 5 and 6 seconds into the future.
Random Time Variation	Selects a span of time for replicas to vary their animation within from their current time. So, if this is set to 1 (second), then the replicas play their animations on a random start frame between the current time and one second in the future.
Quantization Count	When using **Random Time Variation**, this option limits the number of unique deformed individual meshes. This is useful for improving performance when working with a large number of replicas. The default is 100, which means that the replicator uses 100 unique variations and the rest of the replicas are instances of those.

Rock Item

The **Rock Item** automatically generates geometry that can be made to look like a variety of stone elements, such as pebbles, rocks and boulders. The initial mesh is based on a subdivided cube that can be deformed and displaced into the final rendered shape. Multiples can be generated within a specified area defined by the **Radius** value (relative to the position of the Rock item). The Rock item also allows a unique displacement to be applied to existing Subdivisions surface geometry using the **Import Mesh** option.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
Properties
Channels
Lists
+
Rock Displacement
Rock
Display
Assembly
User Channels
Tags
Reset
Zero
Add

Landscape.lxo
- Rock
- GroundPlane
- Camera
- Directional Light

Name: Rock

Transform

Position X	0 m
Y	0 m
Z	0 m
Rotation X	0.0 °
Y	0.0 °
Z	0.0 °
Scale X	100.0 %
Y	100.0 %
Z	100.0 %

Initial rock

Import mesh	(none)
Width	1 m
Height	500 mm
Depth	1 m
Offset X	50.0 %
Offset Z	50.0 %
Rotation X	57.2958 °
Shear Z	100.0 %
Variations	25.0 %

Rock Properties

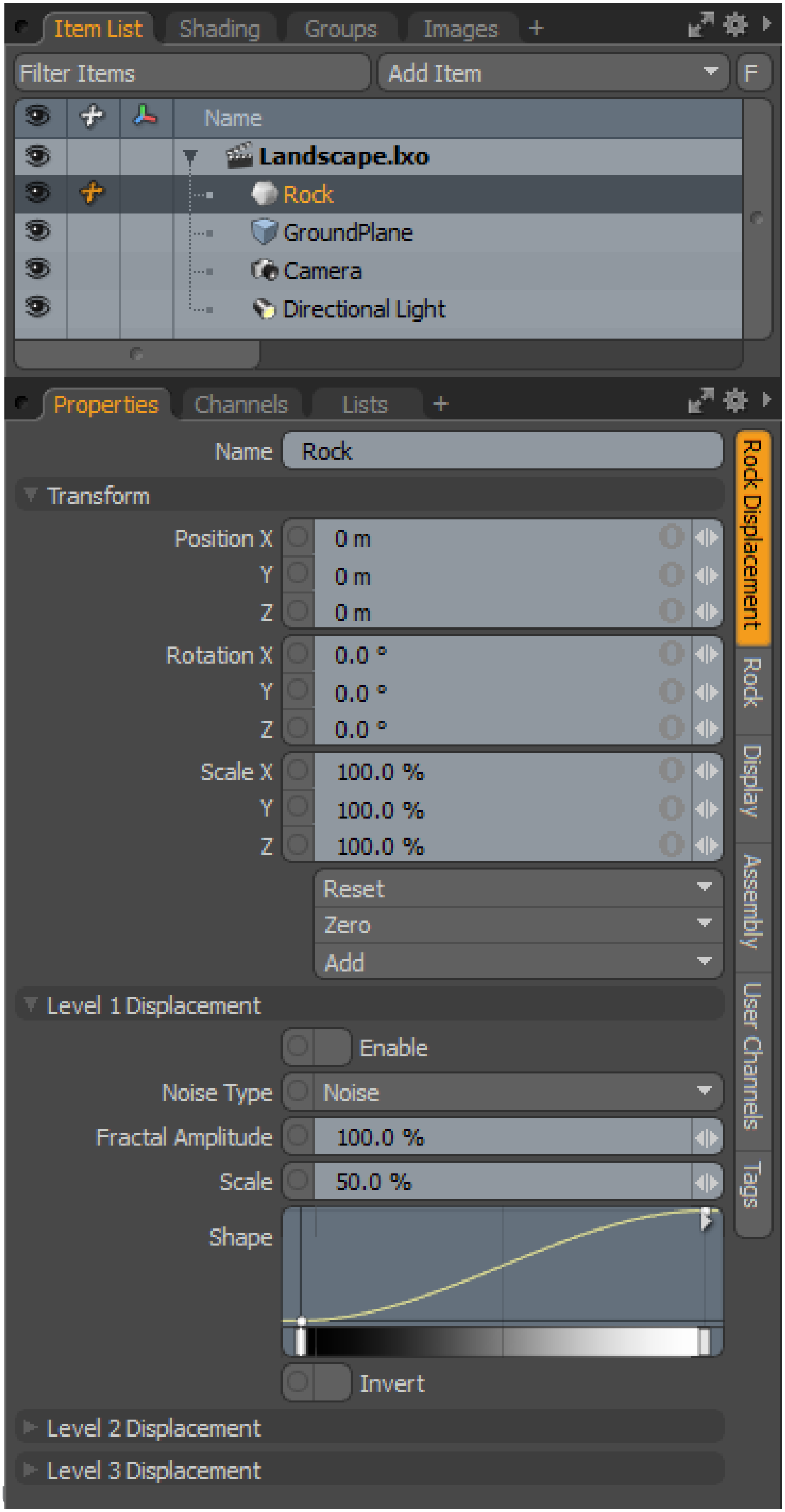
Size Variation	0.0 %
Level of Subdivision	4

Rock Options

Option	Description
Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space.
Rotation	An item transform that allows you to numerically set the rotation of the item.
Scale	An item transform that allows you to numerically set the size of the item.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Freeze	Returns the item's center position to the world space center (0,0,0) without changing the position of the Mesh Item itself.
Zero	Resets the chosen transform property values to **0**, leaving the center position and mesh position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, say a joint, to have a base value of 0,0,0, but still be located away from the World Origin.
Add	Adds additional transform items to the associated item, or if they do not yet exist, it simply adds them. Transform items are the channel groups that store the actual transform values controlling any item's position, rotation and/or scale. You can add as many transform items as desired for any transform property desired. Adding additional transform items produces an additive effect where each transform group is evaluated before the next and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards. By default, new items do not have any transform items associated with them (even though they are visible in the **Properties** panel). This is useful as an optimization as only the necessary transforms are created on an as-needed basis, reducing scene overhead. There are several actions that add

Option	Description
	these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values' input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Because of this fact, you may need to specifically create item transforms when referencing, because in order to override the channels in the master scene, they must first exist.
Initial Rock - The initial mesh that is modified is based on a cube, defined by the size values. The cube is then further modified by the offset and shear values to produce the basic shape.	
Import Mesh	This option can be used to specify an existing Mesh Item layer that stands in as the base geometry for the resulting rock item(s). It can also be used as a way to apply the unique displacement properties of the Rock generator to any geometry in the scene. For best results the specified mesh should be a subdivision surfaces model, consisting of mostly quad geometry. When specified, the resulting mesh is a duplicate of the source (the original mesh is not modified).
Width/Height/Depth	These options determine the physical size of the overall bounding box of the generated rock mesh(es).
OffsetX/Z	These values control the offsetting of the opposing horizontal and/or vertical edges of the bounding box, generating a tapering effect. An **Offset** of 0% retains the rectangular shape of the initial box and a value of 100% creates a pyramid shape. Negative values increase the size of the rock's top plane.
Shear Z	This value laterally shifts the opposing edges of the base plane of the initial rock form.
Variations	Determines the amount of randomness in the various settings that determine the shape. Especially useful when more than one Rock is defined, as it adds randomness to the shape of subsequent rock forms.
Rock Properties	
Size Variations	Defines the amount of overall scale variations when more than one rock is defined.
Level of Subdivision	The level of subdivision determines the roundness and overall mesh density

Option	Description
	of the resulting rock forms.
Number of Rocks	Defines the number of Rock forms generated.
Seed	This value is the initial value used when generating random values. Each **Seed** produces different random results when the **Variations** values are above 0%.
Radius	The **Radius** value defines a circular domain where rock is generated within (when multiples are generated using the **Number of Rocks** value).


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
Landscape.lxo
Rock
GroundPlane
Camera
Directional Light
Properties
Channels
Lists
+
Rock Displacement
Rock
Display
Assembly
User Channels
Tags

Name: Rock

Transform

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Reset
Zero
Add

Level 1 Displacement

Enable

Noise Type: Noise

Fractal Amplitude: 100.0 %

Scale: 50.0 %

Shape

Invert

Level 2 Displacement

Level 3 Displacement

Rock Displacement Options

Option	Description
Level 1 Displacement	The three **Displacement** controls work together to create realistic variations in the resulting rocks. All the **Displacement** options work basically the same way as it would if a texture layer was added to the **Shader Tree** and set to the **Displacement** effect. The one main difference in the approach used by the Rock item is that the displacements are applied sequentially, meaning that each subsequent layer that is enabled displaces the geometry using the normals from the prior layer. There are three separate Displacement layers that all have identical functionality.
Enable	Turns displacement on or off for that layer. Any layer can be enabled or disabled, although their order does have an effect of the final result, as each layer displaces on top of the previous layer. General rule of thumb is that large form-defining displacements can be done first, then the second layer for the major details, and finally the last layer produces the smallest detail. The amplitude should be reduced in each subsequent layer for best results.
Noise Type	Defines a procedural noise layer that displaces the rock's surface. There are two choices, the **Noise** option and the **Cellular** option. Both are identical with their **Shader Tree** counterparts. **Noise** produces a somewhat cloud-like puffy distortion to the surface, and **Cellular** produces a bubbly bumpy look, or when inverted, a chipped sort of look.
Fractal Amplitude	Represents the strength of the displacement. For best results, subsequent layers should have their amplitude reduced for each layer level the proper proportional amount.
Scale	Defines the average size of the fractal features.
Shape	The **Shape Gradient** modifies the contour profile of the generated **Noise**. If it is thought of as a basic linear displacement, the curve values can be used to modify that gradient producing subtle or dramatic changes in the shape profile. The middle mouse button is used to insert keys into the mini gradient and the left mouse button can be used to select key positions and change their value, moving up or down. Upwards displaces outward and down displaces less at that position. The small widget in the upper right corner offers some options to open a full size gradient editor of modify the

Option	Description
	view style of the in-line gradient editor.
Invert	Reverses the values for the nose layer producing a negative effect.

Static Mesh Item

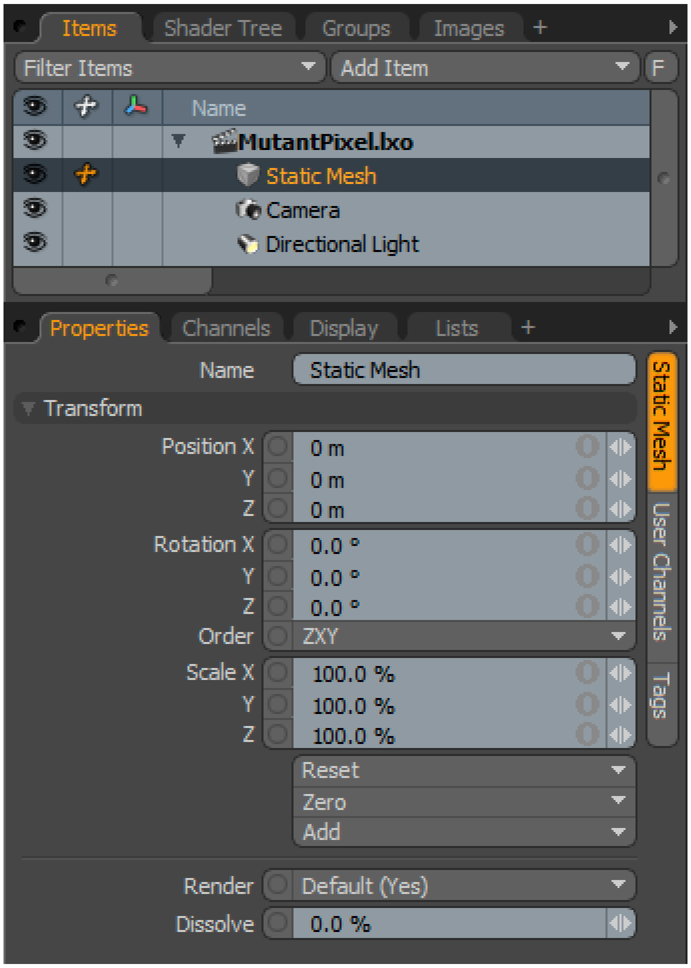
Static Mesh Items can streamline workflows where extremely dense meshes are used. Static meshes, as their name suggests are non editable, essentially frozen objects.

A good amount of memory and processing power is used in Modo so that objects can be edited: each and every vertex ready to accept any type of transform or modification. When a mesh gets really dense, this overhead can bog down a scene file. Adding a few items like this to a scene can substantially reduce viewport interactivity. A static mesh is a means to solve the issue by removing the overhead associated with editing a mesh, increasing overall performance and minimizing file size. Static meshes render like regular meshes and can be positioned by regular item level transformations. They can also be used with Replicators.

Static meshes are made by converting regular Mesh Items. To convert, In the **Items** list, right-click over the Mesh Item layer containing the geometry you wish to convert, and select **Change Type** > **Static Mesh** from the pop-up menu. Once converted, all component level mesh editing is disabled, including morph maps and deformers (which are removed). Subdivisions are frozen as well, converting any subdivided geometry into triangulated polygons. Basically, when converting to a static mesh, Modo is using the freeze command and then triangulating the model. Static Mesh Items were implemented mainly to facilitate the rendering of high-poly CAD data and dense organic geometry, such as trees.

For special cases where a mesh uses displacement maps, deformers and such, you can pre-freeze the item prior to converting to static mesh. In the menu bar click **Geometry** > **Freeze...** and select **Freeze Displacement** and **Freeze Deformation** in the pop-up dialogue.

Note: For further editing, a static mesh can be converted back into a editable mesh layer, however, due to the freezing of the mesh, some information gets lost.


Items
Shader Tree
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Static Mesh
Camera
Directional Light
Properties
Channels
Display
Lists
+
Transform
Reset
Zero
Add
Static Mesh
User Channels
Tags

Name Static Mesh

Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Order ZXY
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Render Default (Yes)
Dissolve 0.0 %

Option	Description
Name	This data field displays the current item name. Change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space. By default, position transforms originate from the center position.
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, rotation transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By

Option	Description
	default, scale transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and item position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. You can simply transform the target item with one of the various transform tools, or by editing the value input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the channel list while keeping the default **0,0,0** values. A necessary step for referencing, in order to override the channels, they must first exist.
Render	This dropdown menu allows you to select from three options. When set to **Default**, you can enable/disable Mesh Items using the visibility column (👁) of the **Items** list. When the mesh layer is visible, it contributes to the final rendered scene and when invisible, it doesn't. In some cases you may prefer to fix this state, setting the mesh as **On** (enabled) or **Off** (disabled) regardless of visibility. This is useful for workflows that auto-toggle visibility: you don't have to manually enable mesh layers for test renders.
Dissolve	Controls the visibility of an item layer. At 0% the item renders normally, utilizing the Shader Tree settings. Values above 0% fade the visibility of the item layer ramping toward 100%, where the item becomes completely invisible.

Note: Static meshes cannot be used as point sources for Replicators or as dynamic objects, as they produce no collision shape for the dynamics engine to calculate.

Transforming Items

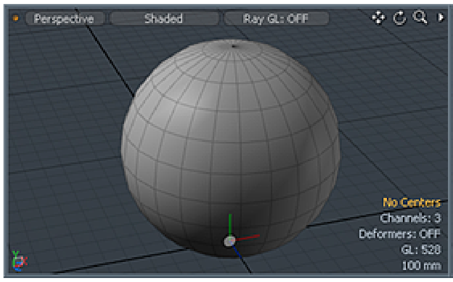
Learn how you can transform items in Modo. Centers and Pivots outlines the differences between these two selection modes, and Transform Item Tools contain detailed information about the transform tools of Modo.

Centers and Pivots

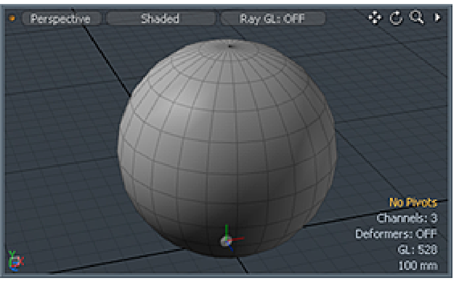
The virtual representation of three-dimensional space within Modo is fixed. You can modify the units that represent scale and change what axis represents the up direction, but otherwise the world is a set known quantity. Everything happens within these fixed coordinates. More importantly, every coordinate value is calculated as an offset from the World Origin; the position where all axis values are 0. When working with any item in Modo, such as a mesh layer, a camera or a light, Modo provides two elements for determining the origin of transformation operations. This occasionally causes some confusion, but the definitions below can help you understand their use.

Center: This is the origin of the Mesh Item layer, regardless of the position of any geometry within the layer. It is the location that sits at the World Origin when all the item transforms are at their default values, that is, when the item has not been moved, rotated or scaled. It is also the default origin from where all item transformations occur, especially rotate and scale.

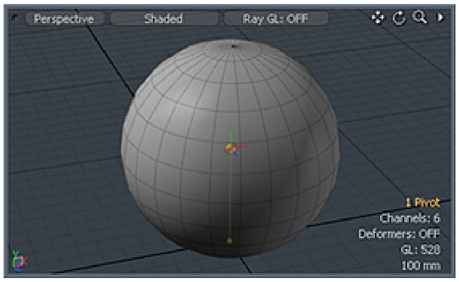
Pivot: This is an offset from the item's center. When a pivot is specified, the rotation and scaling originate from this location instead of the item's center. Pivots are usually used when you need to animate the location around which an item rotates, for example a rolling cube, or when you need the rotation point to be in a different location from the item's center for any reason (such as when using weight deformers).


Perspective
Shaded
Ray GL: OFF
No Centers
Channels: 3
Deformers: OFF
GL: 528
100 mm

Center element (at origin)


Perspective
Shaded
Ray GL: OFF
No Pivots
Channels: 3
Deformers: OFF
GL: 528
100 mm

Pivot element


Perspective
Shaded
Ray GL: OFF
1 Pivot
Channels: 6
Deformers: OFF
GL: 528
100 mm

Pivot Offset (note the line to the center)

Usage

In Center or Pivot selection modes, you first select the respective element to work with, then apply any of the normal transform tools to edit their location. Since they use the regular transforms, they also automatically support all the action centers, axis and snapping functionality.

> **Note:** Editing center positions is only applicable when working with mesh layers. Moving the center for any other element, such as a camera or a light actually moves the element itself, so you need to edit the pivot instead, to change the point of origin.

Constraints default to using an item's center, so if the pivot is offset, make sure to set the constraint to explicitly use the pivot.

> **Note:** Editing a center position after animation has been applied changes the way the item animates based on the new origin.

Transform Item Tools

While some of the standard modeling tools are useful for working with items, there are a few layout and setup-specific tools available that can speed up these workflows. These item-specific tools are differentiated from component-specific ones by their purple icon color scheme. For information about using a specific tool, click its link below.

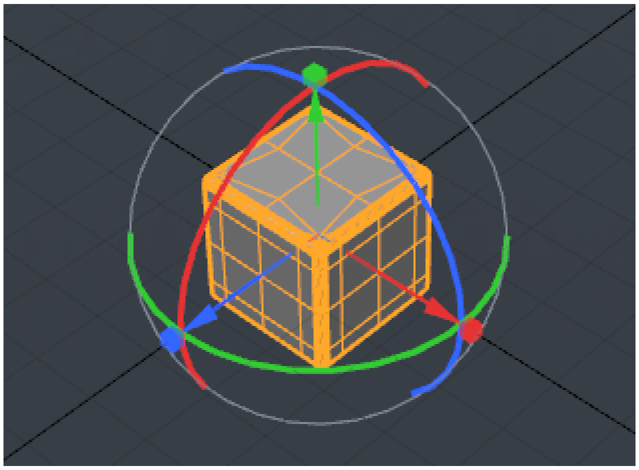
Transform		
	Transform	Combines move, rotate, and scale transformations into a single tool.
	Move	Moves a selection across a specific distance.
	Rotate	Rotates a selection by a specific number of degrees.

Transform		
	Axis Rotate	Rotates a selection, affecting a single axis at a time.
	Scale	Makes any selection larger or smaller by changing the scale percentage factor.
	Uniform Scale	All three axes are constrained, forcing all scaling operation to be uniform.

Clone		
	Clone	Creates duplicates of the selected geometry.
	Scatter Clone	Generates a random point cloud onto which clones of the original geometry are placed
	Curve Clone	Clones geometry along a user-defined curve.

Array		
	Array	Creates copies of the currently selected geometry in an ordered pattern.
	Radial Array	Copies the selected geometry into a circular or helix pattern.

Transform



The Transform tool combines move, rotate and scale transformations into a single tool. By using the various handles you can quickly position, rotate and scale in one convenient tool. Hold **Ctrl** to constrain the position transform to a single axis when hauling, and the rotation transform to 15° increments. After applying a transform amount, middle-click to re-apply the previous offset amount. This way, you can easily apply an edit multiple times. Activate the command in the **Model** viewport's **Basic** sub-tab, or by pressing **Y**.

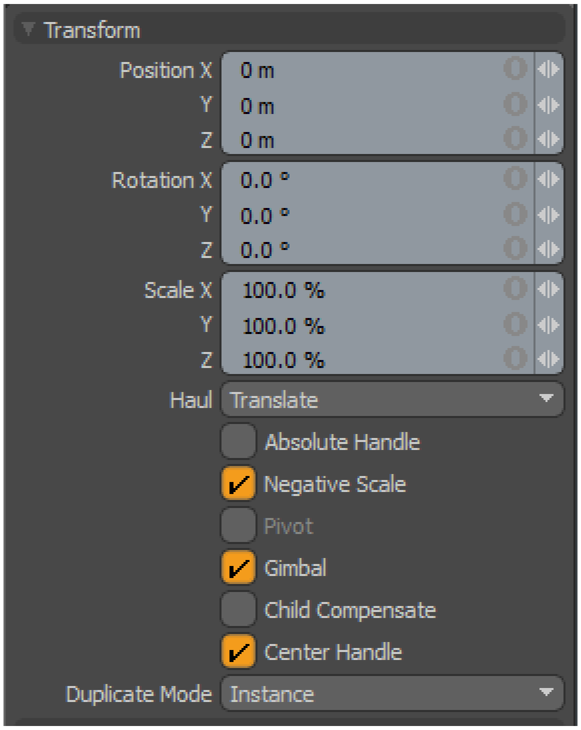
In the component modes (vertices, edges, and polygons), the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected in the **Items** list. In **Items mode**, the tool affects all the currently selected items in the **Items** list. If nothing is selected, then the tools appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the Center and Pivot modes, it modifies the center or pivots respectively.

Note: You need to select either the center or the pivot itself, depending on the mode, prior to activating the tool, as in **Items mode**, with no selection, the tool appears to do nothing.

When you define any of the falloffs, application of the tool can be attenuated across a user-defined area. For more information on using falloffs, see Using Falloffs.

Once you activate the tool, Modo draws the relevant tool handles in the viewport: arrows for moving, boxes for scaling and circles for rotating. At this point, you can click and haul anywhere in the desired viewport to interactively transform the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the function to one or two axes. The various tool handles are color-coded to match the coloring of the different axes: red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then click and drag to modify. To interactively scale the size of the handles themselves, use the **-** (minus) key to decrease size and **=** (equals) to increase their size.

The Transform tool also respects the settings for action center, and originates the appropriate action from the defined position. For more information on understanding and using actions centers, see Specifying Action Centers and Falloffs .



Transform

Position X 0 m

Y 0 m

Z 0 m

Rotation X 0.0 °

Y 0.0 °

Z 0.0 °

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

Haul Translate

☐ Absolute Handle

☑ Negative Scale

☐ Pivot

☑ Gimbal

☐ Child Compensate

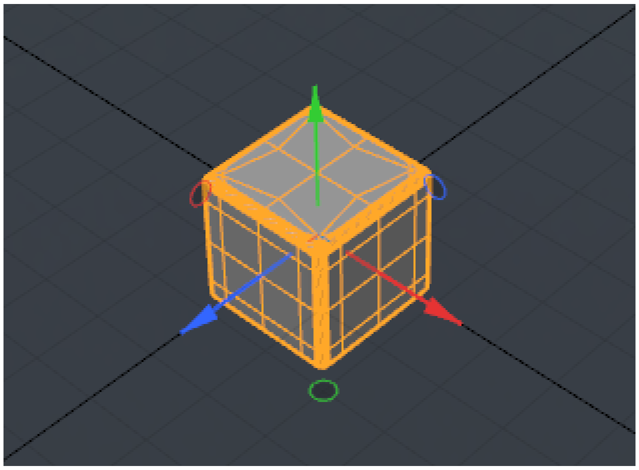
☑ Center Handle

Duplicate Mode Instance

Option	Description
Transform	
Position	Use the X, Y and Z input fields to apply specific offset distance values, calculated from the position the object was in when the tool was activated. When adjusting an object interactively in the viewport, the **Position** input fields display the current offset distance applied, providing useful feedback.
Rotation	Use the X, Y and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, these input fields display the current offset distance applied, providing useful feedback.
Scale	Use the X, Y and Z input fields to apply specific scaling values. When adjusting an object interactively in the viewport, these input fields display the current scale factor applied, providing useful user feedback.

Option	Description
Haul	You have three options available when hauling in the viewport, that is, dragging the mouse when not using the tool handles. **Translate** allows you to move the selection, **Rotate** allows you to rotate, and **Scale** scales the selection.
Absolute Handle	Applies an additional function to the tool handles allowing you to scale the handle itself without affecting the transform amount. In some cases, this provides finer control over very large or very small adjustments.
Negative Scale	Enables/disables Modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, you can always input negative values in any of the input fields directly.
Pivot	Allows you to easily modify an item's pivot position without the need to go into Pivots mode.
Gimbal	When enabled, you can assign rotation values numerically, but may arrive at a situation that produces gimbal lock, the loss of one rotation axis when another's value is too close. You can avoid this by disabling this setting, but then you cannot apply specific values within the input fields.
Child Compensate	
Center Handle	
Duplicate Mode	Instance Duplicate

Move



The Move tool allows you to move, or translate a selection across a specific distance. Use the various handles to easily constrain the movement to one or two axes. Holding the **Ctrl** key constrains move to a single axis when hauling. After applying a transform amount, middle-click to re-apply the previous offset amount. This way, you can easily apply an edit multiple times. Activate the command in the **Model** viewport's **Basic** sub-tab, in the menu bar under **Edit** > **Transform** > **Move** or by pressing **W**.

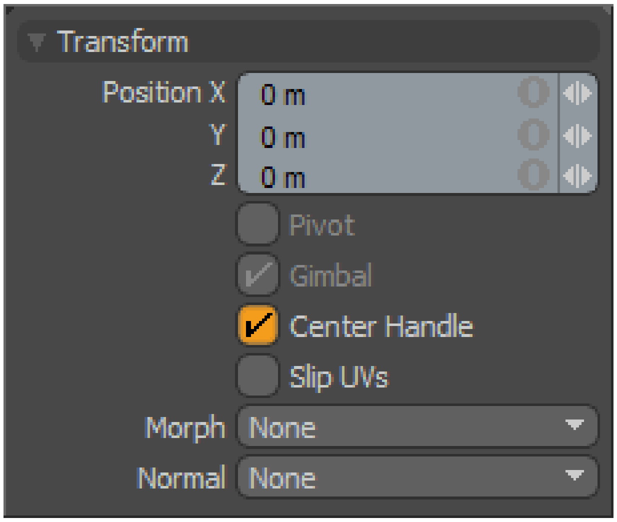
In the component modes, (vertices, edges, and polygons), the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected in the **Items** list. In **Items mode**, the tool affects all the currently selected items in the **Items** list, if nothing is selected, then the tools appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the Center and Pivot modes, it modifies the center or pivots respectively.

Note: You need to select either the center or the pivot itself, depending on the mode, prior to activating the tool, as in **Items mode**, with no selection, the tool appears to do nothing.

When you define any of the falloffs, application of the tool can be attenuated across a user-defined area. For more information on using falloffs, see Using Falloffs.

Once you activate the tool, Modo draws the relevant tool handles in the viewport. At this point, you can click and haul anywhere in the desired viewport to interactively transform the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the tool, the arrows to a single axis, and the circles to two axes. The various tool handles are color-coded to match the coloring of the different axes: red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then click and drag to modify. To interactively scale the size of the handles themselves, use the **-** (minus) key to decrease size and **=** (equals) to increase their size.

The Move tool also respects the settings for action center, and originates the appropriate action from the defined position. For more information on understanding and using actions centers, see Specifying Action Centers and Falloffs .



Transform

Position X 0 m

Y 0 m

Z 0 m

☐ Pivot

☑ Gimbal

☑ Center Handle

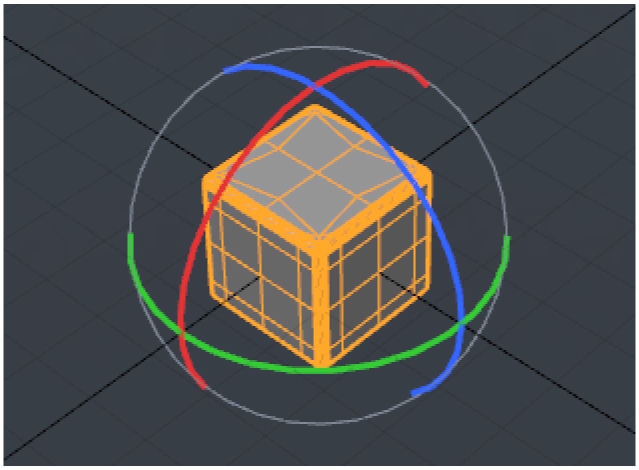
☐ Slip UVs

Morph None

Normal None

Option	Description
Transform	
Position	Use the X, Y and Z input fields to apply specific offset distance values, calculated from the position the object was in when the tool was activated. When adjusting an object interactively in the viewport, the **Position** input fields display the current offset distance applied, providing useful feedback.
Pivot	Allows you to easily modify an item's pivot position without the need to go into Pivots mode.
Gimbal	The **Gimbal** setting is grayed out in Move mode.
Child Compensate	When disabled, Modo passes along the transform value to any child items of the current selection. When enabled, the child items ignore any transform values.

Rotate



The Rotate tool allows you to rotate a selection by a specific number of degrees. By using the various handles you can easily constrain the movement to a single axis. Hold **Ctrl** to constrain the rotation to fixed 15° intervals. After applying a transform amount, middle-click to re-apply the previous offset amount. This way, you can easily apply an edit multiple times. Activate the command in the **Model** viewport's **Basic** sub-tab, in the menu bar under **Edit** > **Transform** > **Rotate** or by pressing **E**.

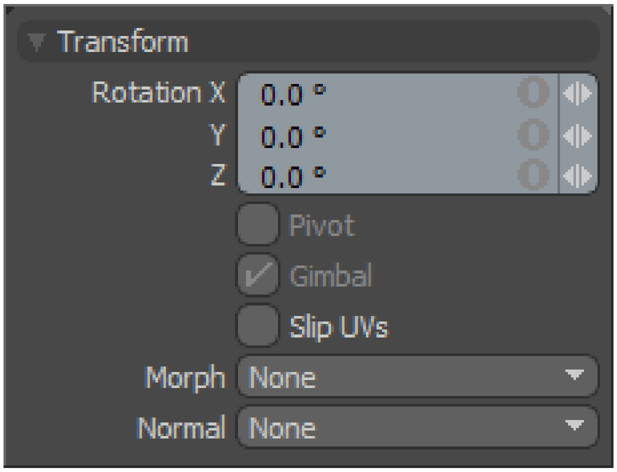
In the component modes (vertices, edges, and polygons), the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected in the **Items** list. In **Items mode**, the tool affects all the currently selected items in the **Items** list. If nothing is selected, then the tools appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the Center and Pivot modes, it modifies the center or pivots respectively.

Note: You need to select either the center or the pivot itself, depending on the mode, before activating the tool, as in **Items mode**, with no selection, the tool appears to do nothing.

When you define any of the falloffs, application of the tool can be attenuated across a user-defined area. For more information on using falloffs, see Using Falloffs.

Once you activate the tool, Modo draws the relevant tool handles in the viewport. At this point, you can click and haul anywhere in the desired viewport to interactively transform the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the tool to a single axis. The tool handles are color-coded to match the coloring of the different axes, red for X, green for Y and blue for Z. In some action center modes, there is an additional gray screen space handle around the outside of the tool handles, which rotates the selection based on the viewport camera's position. Hover the mouse over a specific handle until it changes to a yellow color and then click and drag to modify. To interactively scale the size of the handles themselves, use the **-** (minus) key to decrease size and **=** (equals) to increase their size.

The Rotate tool also respects the settings for action center, and originates the appropriate action from the defined position. For more information on understanding and using actions centers, see Specifying Action Centers and Falloffs .


Transform

Rotation X 0.0 °

Y 0.0 °

Z 0.0 °

☐ Pivot

☑ Gimbal

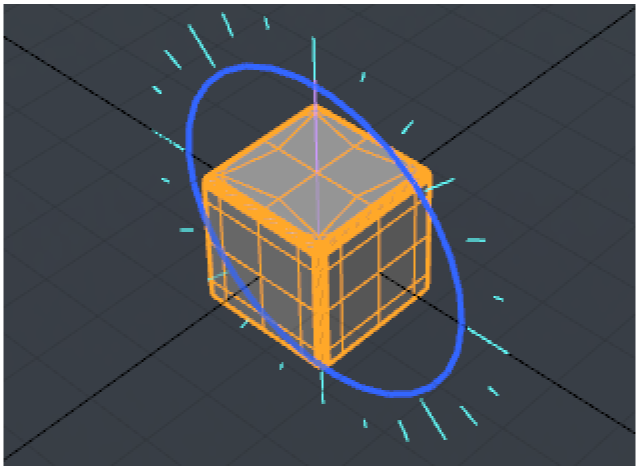
☐ Slip UVs

Morph None

Normal None

Option	Description
Transform	
Rotation	Use the X, Y and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, these input fields display the current offset distance applied, providing useful feedback.
Pivot	Allows you to easily modify an item's pivot position without the need to go into Pivots mode.
Gimbal	When enabled, you can assign rotation values numerically, but may arrive at a situation that produces gimbal lock, the loss of one rotation axis when another's value is too close. You can avoid this by disabling this setting, but then you cannot apply specific values within the input fields.

Axis Rotate



The Axis Rotate tool allows you to rotate a selection by a specific number of degrees. It is different from the standard Rotate tool in that it is limited to editing a single axis at a time, determined by the Work Plane position. It also has a different on-screen control display than the other transform tools, providing a greater degree of control over the amount of rotation depending on how long or short you drag out the handle length. Hold **Ctrl** to constrain the rotation to fixed 15° intervals. After applying a transform amount, middle-click to re-apply the previous offset amount. This way, you can easily apply an edit multiple times. Activate the command in the **Model** viewport's **Basic** sub-tab (you may need to press **Shift** to show alternate tools). Axis Rotate is also available in the menu bar, under **Edit** > **Transform** > **Axis Rotate** or by pressing **Alt**+**Shift**+**E**.

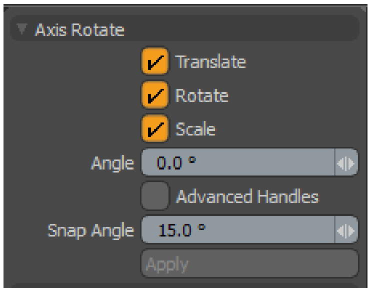
In the component modes (vertices, edges, and polygons), the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected in the **Items** list. In **Items mode**, the tool affects all the currently selected items in the **Items** list. If nothing is selected, then the tools appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the Center and Pivot modes, it modifies the center or pivots respectively.

Note: You need to select either the center or the pivot itself, depending on the mode, prior to activating the tool, as in **Items mode**, with no selection, the tool appears to do nothing.

When you define any of the falloffs, application of the tool can be attenuated across a user-defined area. For more information on using falloffs, see Using Falloffs.

Once you activate the tool, Modo draws the relevant tool handles in the viewport. At this point, you can click and haul anywhere in the desired viewport to interactively transform the geometry. Depending on the action center, this may also set the rotation center based on the mouse click position intersecting the Work Plane. Additionally, you may prefer to utilize the on screen display, either clicking on the wheel to rotate to a specific amount, or by dragging the handles on the value bar across the top of the viewport (when **Advanced Handles** is enabled). When you click on the wheel, a handle is drawn displaying the number of degrees. When using this handle, the rotation sensitivity is adjusted based on the distance the mouse is from the handles as you drag. This can be thought of as extending a lever. The longer the lever, the more mouse movement is required to rotate the element and thus the granularity of control is increased. To interactively scale the size of the handles themselves, use the **-** (minus) key to decrease size and **=** (equals) to increase their size.

The Rotate tool also respects the settings for action center, and originates the appropriate action from the defined position. For more information on understanding and using actions centers, see Specifying Action Centers and Falloffs .


Axis Rotate

☑ Translate

☑ Rotate

☑ Scale

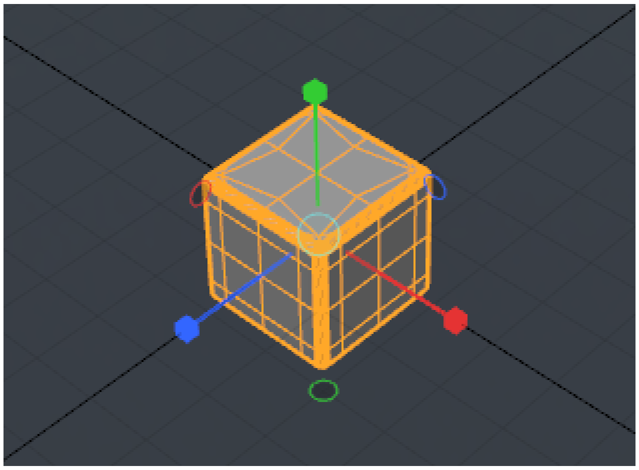
Angle 0.0 °

☐ Advanced Handles

Snap Angle 15.0 °

Option	Description
Axis Rotate	
Angle	Apply a specific rotational value, calculated in degrees. When adjusting an object interactively in the viewport, these input fields display the current rotation amount, providing useful feedback.
Advanced Handles	Determines whether or not Modo displays the additional information in the viewport.

Scale



The Scale tool allows you to make any selection larger or smaller by changing the scale percentage factor. To quickly scale in a single axis dimension, use the various handles. After applying a transform amount, middle-click to re-apply the previous offset amount. This way, you can easily apply an edit multiple times. Activate the command in the **Model** viewport's **Basic** sub-tab, in the menu bar under **Edit** > **Transform** > **Scale** or by pressing **R**.

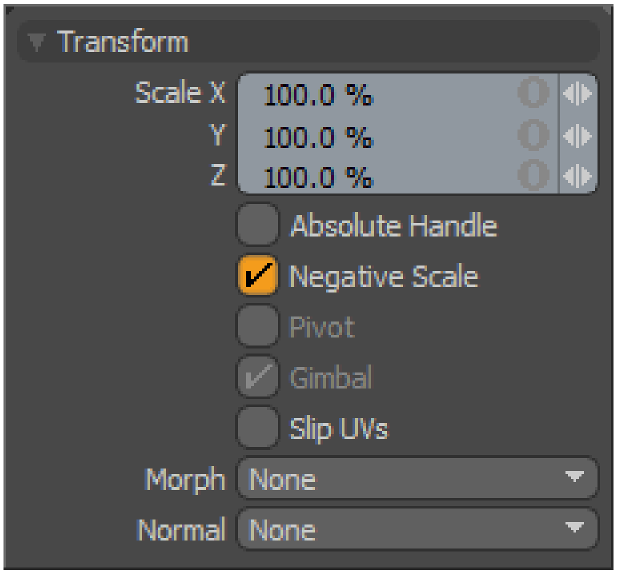
In the component modes (vertices, edges, and polygons), the tool affects the selected geometry directly. When nothing is selected, it affects all active foreground geometry. Active foreground items are those that are visible and selected in the **Items** list. In **Items mode**, the tool affects all the currently selected items in the **Items** list. If nothing is selected, then the tools appear to do nothing. In material mode, it works the same way as in any of the component modes. Lastly, in the Center and Pivot modes, it modifies the center or pivots respectively.

Note: You need to select either the center or the pivot itself, depending on the mode, prior to activating the tool, as in **Items mode**, with no selection, the tool appears to do nothing.

When you define any of the falloffs, application of the tool can be attenuated across a user-defined area. For more information on using falloffs, see Using Falloffs.

Once you activate the tool, Modo draws the relevant tool handles in the viewport. At this point, you can click and haul anywhere in the desired viewport to interactively transform the geometry. Additionally, you may prefer to utilize the various tool handles, constraining the application of the function to one or all three axes. The tool handles are color-coded to match the coloring of the different axes, red for X, green for Y and blue for Z. Hover the mouse over a specific handle until it changes to a yellow color and then click and drag to modify. To interactively scale the size of the handles themselves, use the **-** (minus) key to decrease size and **=** (equals) to increase their size. Additionally, for the Scale tool, there is a small circle at the intersection of all three handles that you can use to uniformly scale the selection interactively.

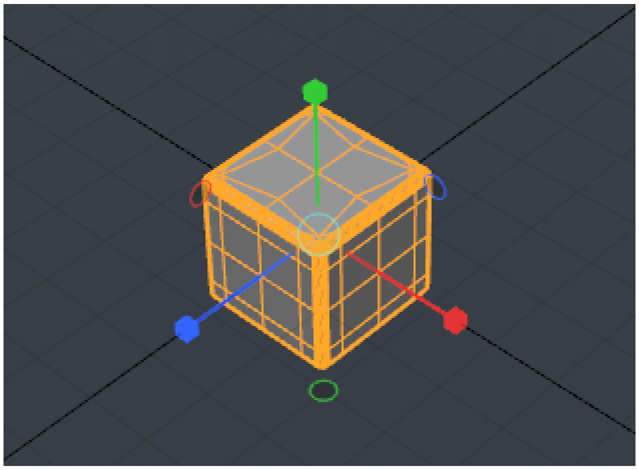
The Transform tool also respects the settings for action center, and originates the appropriate action from the defined position. For more information on understanding and using actions centers, see Specifying Action Centers and Falloffs .



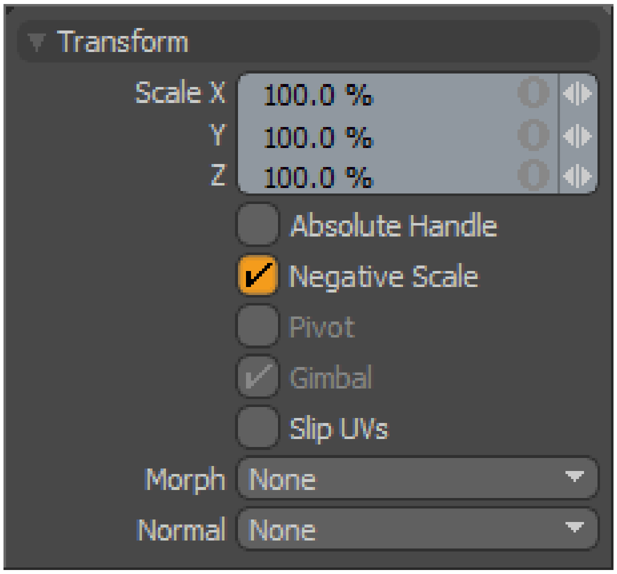
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
☐ Absolute Handle
☑ Negative Scale
☐ Pivot
☑ Gimbal
☐ Slip UVs
Morph None
Normal None

Option	Description
Transform	
Scale	Use the X, Y and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, these input fields display the current factor applied, providing useful feedback.
Absolute Handle	Applies an additional function to the tool handles, allowing you to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.
Negative Scale	Enables/disables Modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, you can always input negative values in any of the input fields directly.
Pivot	Allows you to easily modify an item's pivot position without the need to go into Pivots mode.
Gimbal	This setting is not available in scale mode.

Uniform Scale



The Uniform Scale tool is identical to the normal Scale tool, except that you can haul anywhere in the viewport and all three axes are constrained, forcing all scaling operating to be uniform. This ensures the mesh doesn't become distorted when scaling.



Transform

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

☐ Absolute Handle

☑ Negative Scale

☐ Pivot

☑ Gimbal

☐ Slip UVs

Morph None

Normal None

Option	Description
Transform	
Scale	Use the X, Y and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, these input fields display the current factor applied, providing useful feedback.

Option	Description
Absolute Handle	Applies an additional function to the tool handles, allowing you to scale the handle itself without affecting the transform amount. In some cases this provides finer control over very large or very small adjustments.
Negative Scale	Enables/disables Modo's ability to interactively apply negative scale values when manipulating the tool handles in the viewport. When disabled, you can always input negative values in any of the input fields directly.
Pivot	Allows you to easily modify an item's pivot position without the need to go into Pivots mode.
Gimbal	This setting is not available in scale mode.

Transform Modifiers

Transform modifiers are any function that changes the way transforms are applied to items in Modo. Falloffs are the primary modifier, but Modo's snapping options also affects transforms when applied in **Items mode**. For more information on falloffs and snapping, see Using Falloffs and Applying Snapping.

Duplicating Items

There are various ways of duplicating items in Modo. You can add **Instances** to your scene, which are items that don't have their own geometry, but reference a source object. You can also add a **Replicator** that generates replicas of your source mesh at render time. The properties of the **Replicator** item influence the arrangement of the replicas. Instances gives you detailed information on what **Instances** are and how to use them. The following Replicators topic describes how you can generate copies of your item with the help of **Replicators**.

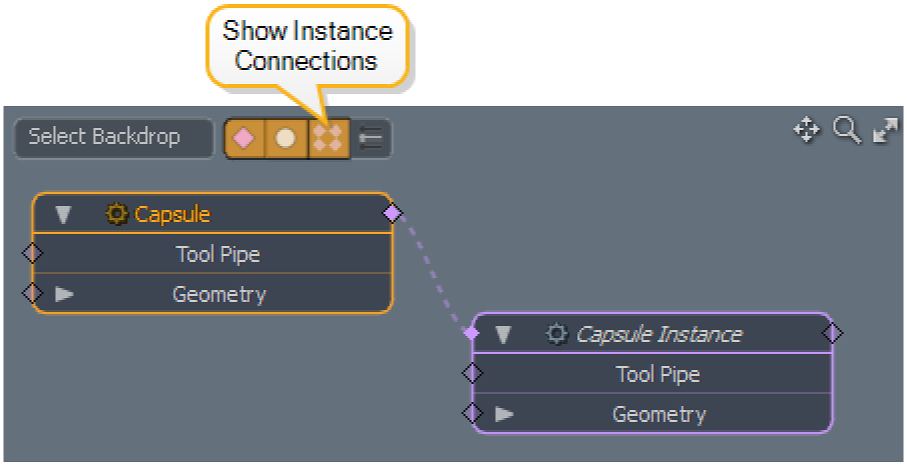
Instances

Instances are a special item type in Modo that add little additional data overhead to a scene, as they contain no actual geometry. Instead, they reference a source object at render time, allowing you to generate a huge number of duplicates for an item with little additional memory usage or performance hit. As an added benefit, you can modify the source geometry at any time and the changes automatically propagate to each instance. For example, consider a model of a computer case, where each bolt is an instance of a single mesh. When the client decides the bolt needs to be a hex head rather than a standard phillips head, all instances of the bolt can be easily updated by simply modifying or replacing the geometry in the source item.

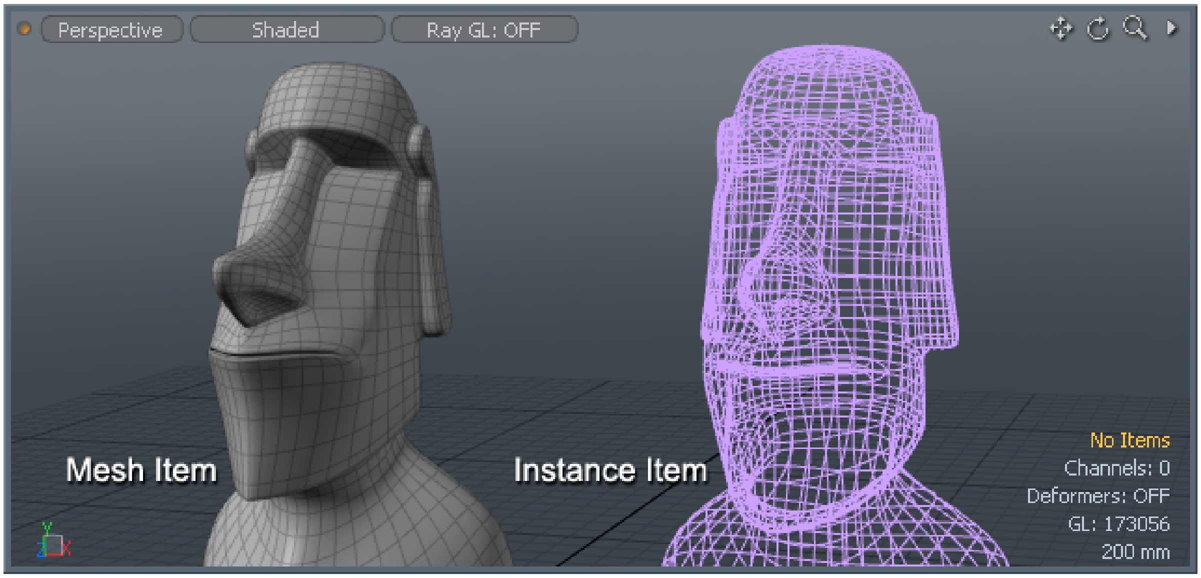
Instances are as easy to create as a normal duplicate, and many cloning and array tools offer additional functionality to generate Instances instead of direct geometric clones. Instances themselves can be treated like normal mesh layers, each having its own set of transforms and render options. You can select them like any other item, and position, scale and rotate them. This gives you a great deal of control over placement of the copies. While instances reference their shading and material settings from the source object, you can assign an item mask through a Material Group Item in the **Shader Tree** and target custom surface settings to a specific instance. For more information on item masks, see Material Group.

Mesh instances are differentiated in the **Items** list by their pink cube **Shader Tree** icon (as opposed to the standard blue-gray cube icon for Mesh Items). They are also distinct in that the layer name is displayed italicized and their unselected wireframe representation in the viewport is always, by default, pink in color.

In the Schematic, instances are highlighted in pink when you select their source item. Clicking the **Show Instance Connections** button at the top of the Schematic viewport draws a dashed line between the instance and its source item, indicating their connection.



Instances are not limited to Mesh Items, they are useful whenever you have a large number of identical items. If a series of streetlights in a scene all shared identical settings, you can instance the first light and then later on, just as with Mesh Items, changes made to the source light propagate to all the instances.



Creating Instances

You can create instances in several ways:

- Select a Mesh Item in **Items mode**, and press **Shift**+**D** on the keyboard.
- Right-click an item layer in the **Items** list and select the **Instance** (or **Instance Hierarchy**) option from the context menu.
- Select an item layer in the **Items** list or directly in the 3D viewport and in the menu bar, click **Item** > **Instance**.
- Select a Mesh Item in **Items mode**, activate the Move or the Transform tool, and hold **Ctrl**/**Cmd**+**Shift** while dragging on a handle.
- Right-click a Replicator in the schematic viewport and select **Replicators** > **Freeze Replicator** to convert the selected Replicator into a series of individual Instances.
- In **Items** selection mode, many of the duplication tools, such as Clone, Curve Clone, and Radial Array, provide an option to generate instances. With the tool active, simply enable the instance checkbox and the clones become instances instead of generating full geometry. Alternate versions of the tools are generally available by the small arrow in the bottom right corner of a tool's button. Click and hold to open the pop-up menu to select the alternate tools.

Selecting Instances

Once an instance has been created, you can work with it like any other item in the scene. Select and transform it directly in the 3D viewport or use the numerical settings found in the **Properties** panel when the item is selected. While it is possible to select and modify the center and pivot position of an instance, this actually moves the instance itself. To modify a center, you need to edit the source item. However, since instance placement is based on the item's center, modifying the center point of the source moves all the instanced items.

If a complex scene makes it difficult to locate the source item of an instance, select the instance and then in the menu bar click **Select** > **Source of Instance**. You can also access this command by right-clicking the instance in the **Items** list and choosing **Select Source of Instance** from the pop-up menu.

You can convert an instance to a normal Mesh Item for further editing in a number of ways:

- Select an instance and in its **Properties** tab, click the **Convert to Mesh** button.
- Right-click on the item layer itself to open the context menu and choose **Change Type** > **Mesh** option.
- In the menu bar, click **Item** > **Change Type** > **Mesh**.

Building a Mask for an Instance

To apply surfacing to an instance that is different from the source mesh, assign an item mask: right-click the target instance and in the pop-up menu, select **Create Item Mask**. The resulting **Shader Tree** material group can then be populated with any number of **Shader Tree** layers.

Tip: The difference between cloning (duplicating) geometry in a component mode versus cloning while in **Items mode** is that in the latter case, each clone becomes its own item layer in the Items list. Cloning in a component mode merely duplicates the geometry within the same existing item layer. If you feel the Items list becomes too cluttered with instances, you can group layers: click on the top instance layer, and press and hold **Shift** and click on the bottom layer to gang-select all layers, then press **Ctrl**+**G** to automatically parent all the selected items to a group locator in the Items list. You can then toggle the visibility of all the child items by clicking the arrow icon (▸) preceding the group layer's name. Likewise, you can toggle visibility of the instances themselves by enabling/disabling the eye (👁) of the parent group (assuming that the preference to do so isn't disabled; see Defaults Preferences).

Replicators

Replicator image by Simon Lundberg

Replicators are a means of generating render time duplicates of geometry in a scene, allowing you to render billions of polygons with full support for all of Modo's shading and rendering capabilities.

The Replicator is more like a controller than an item. Unlike other items, which have visible representations in the scene, Replicators are only visible once their **Prototype** and **Point Source** options have been populated. Once defined, the Replicator controls how duplicates of the item are copied onto the target point source.

Replicators are also unique in that they require these additional elements to function. The prototype is simply any Mesh Item layer within the given scene. The point source defines the positioning of the duplicates; by default the center position of the initial mesh layer is positioned precisely at the location of each vertex in the point source item, so while you don't have direct control over the positions of replicated meshes, like with instances, you can indirectly control their locations with the point source by adjusting the vertex positions.

Creating Point Sources

A point source can be any mesh layer, so you can create one the same way you create geometry. This could even be something as simple as a flat plane created using the Box tool. If you apply a Replicator to a single polygon plane, you get four replicas at each corner (vertex position) of the polygon. In order to get additional replicas, the polygon requires additional subdivisions. To do this, increase the **Segments** option found in many of the primitive geometry tools, or apply the Subdivide command in the **Mesh Edit** sub-tab of the modeling toolbox, after you've created the geometry. If you sculpt this subdivided plane, adding hills and valleys, the replicas are still positioned at each vertex, but instead of facing upward, they align themselves to the normal direction of the background vertex (which is an average of the normal direction of all the surrounding polygons) following the contours of the surface. This indirectly allows you to control position and to a degree rotation by simply using the shape of the point source geometry. Virtually any kind of geometry Modo creates can be used as a point source, including the various curves.

Note: Replicators align the center of the prototype mesh to the vertex position of the point source. Therefore, you need to make sure the prototype geometry is positioned appropriately in relation to its center. By default, this is the World 0,0,0 position for each layer. You can easily align the center in relation to the geometries' bounding box using the menu bar command **Edit** > **Center to Bounding Box**. You can also manually transform the center, while in center selection mode. For more on working with centers, see Selecting Items.

Creating Replicators

You can create Replicators like you would create any other item.

1. In the **Items** list, click **Add Item**, and in the **Preset Browser**, select **Particles** > **Replicator**.

   This adds a blank Replicator Item to the **Item List**.
2. In the item's **Properties** panel, manually populate the **Point Source** and **Prototype** fields with the appropriate items.

Tip: You can right-click a replicator and select **Replicators** > **Freeze Replicator** to convert the selected Replicator item into a series of individual Instances. See Freezing & Replicators for more information.

You can also automatically populate these options using selection order:

1. First, in the **Items** list or 3D viewport, select the target item to be the point source, then select the prototype item.
2. In the menu bar, click **Item** > **Replicators** > **Add Replicator**.

   Your geometry is instantly cloned onto the source item.

Adding Randomness

When used as a point source, most geometry produces replicas that you may find too regular in placement. In some cases this geometric uniformity to the positions may be what you need, but in most cases, a more natural, randomized placement is the goal. To achieve this, use the **Variation** settings of the Replicator itself to randomize the positions as an offset from the vertex position. Adding some scale and twist (rotation) randomness also helps to reduce the uniformity of the Replicas. When you need additional randomness, there are more versatile options available. For more information on the Replicator Item's properties, see Replicator Item.

You can introduce randomness by starting at the point source level. When fine control over replicas isn't necessary, you can utilize a point cloud. Point clouds are collections of vertices, with no edges and no polygons. For example, they are useful point sources for splashes of water droplets, or exploding shrapnel. Before you can create a point cloud, you need to enable their view as they are disabled by default:

1. Hover the mouse over the 3D viewport and press **O** to open the **Viewport Options** panel.
2. Click the **Visibility** sub-tab and enable the **Show Vertices** option.

You can then draw individual vertices using the Vertex tool or create geometry using a primitive tool, such as the Sphere, and then use the Remove command to eliminate all the edges and polygons, leaving only vertices behind. Then you can use the deform tools to push and pull vertices into position. The Jitter function is also a great way to further randomize the point source. Point Clouds can procedurally create a volume of particles that are assigned as point sources in the same way as any other mesh layer in the Replicator Item.

More Randomness

Sometimes randomizing the point source itself isn't desirable or viable. Modo has another option to randomize placement called a Surface Particle Generator, which randomly places points all over the surface of a target item, where the orientation of each particle is dependent on the underlying surface where it is placed. In order to take advantage of a surface particle generator, you can add one manually in the **Items** list.

1. Click **Add Item** > **Particle** > **Point Clouds** > **Surface Particle Generator**.
2. In the surface particle **Properties** panel, assign the item to place the particles upon by specifying it as the target item under the **Source Surface** option.

   If the resulting particle should be further limited to a specific area of the target item, the **Material Tag** option allows the selection of a Material associated with the item, allowing for further confinement of the generated surface particles.
3. Once the particles are defined, assign the surface particle generator as the point source in the Replicator's **Properties** tab to produce the intended result.

To create a surface generator automatically, right-click on the Replicator item in the **Items** list and select **Scatter Replicas** from the contextual menu. Once defined, you can adjust the settings of the surface particle generator item to refine the placement of the objects. Further, the density, normal facing direction and size can be controlled for surface particle items using texture layers of the **Shader Tree**, by setting the appropriate effect.

You can assign randomized surface particles using the **Shader Tree**. This can be in cases where the **Shader Tree** is used for displacement, and the evaluation of the surface particles needs to take this into account (a surface particle generator doesn't take displacement on a surface into account). In these cases, using a **Shader Tree** Surface Generator is the preferred particle generation method. This adds a few benefits as well, since surface generators are Shader Tree-aware, meaning you can mask them with Group Items assign other **Shader Tree** effects to them. The surface generator can be added to the **Shader Tree** using the **Add Layer** function and then specified as the **Point Source** in the Replicator's **Properties**. Adjustments to its many attributes in the **Properties** panel further refine the placement of the replicas. For more information on adding and working with **Shader Tree** layers, see Shader Tree.

Controlling Placement

Sometimes you need specific control over the placement of replicated items. When using texture layers in the **Shader Tree** is not accurate enough for the intended purpose, you can draw individual points using the Vertex tool, and generate the points directly on a surface using the background constraint.

1. Create a new Mesh Item layer and then specify that as the point source in the Replicator's **Properties** panel.
2. Draw the vertices with the tool active. Remember to hold **Ctrl** to draw additional vertex points while the tool is active.

   If there is also a need to control the orientation and scale of the Replicator's, a vertex transform map allows that.
3. Once the map is assigned to the layer with the named Vertex Map and target particle selected, activate the Scale or Rotate tools to apply the transformation to the replicated element.

   You can only see the results of the vertex transform map when a Replicator is assigned to the target point source.

Particles and Simulations

Modo offers a complete particle dynamics system. Particles are just points, and any particle simulated can be assigned as a point source in a Replicator, providing for a virtually unlimited number of uses. Particle simulations work in their basic form by starting with adding a Particle Simulation item and then creating an emitter, a radial emitter being the easiest to work with (for more information, see Particle Emitters). Next, by assigning the particle simulation item as the point source in the Replicator, the source mesh now animates along with the simulated particles. There are much more options related to particle simulations than can be covered here, so for more information regarding working with particles and simulations, see Particles and Simulations.

Working with Groups

Another way you can make your scene more realistic when using Replicators is to replicate more than a single mesh. This can be done easily by collecting multiple elements into a group and the Replicator then randomly grabs members of the group when placing individual clones. Not to be confused with items parented to a Group Locator Item.

To create a group, follow these steps:

1. Open the Groups Viewport on the right side of the screen and the select all the Mesh Item layers you wish to add to the group.
2. Within the **Groups** viewport, click the **New Group** button and define a new group.

   The opening dialog allows you to enter a name for the group with an option to make all selected items members of the named group.
3. Press **OK** to create the group and add the selected items.

Once the group is created, define the named group as the prototype source in the Replicator itself. Modo then randomly grabs one of the items specified in the group for each replica. This method works great for applying rocks and plants to an outdoor scene, or trees to a forest scene.

> **Tip:** You can control the percentage that a group member is used by cloning the elements. For example, if the Group contains nine red spheres and one blue cube, the blue cube only appears approximately 10% of the time across the surface.

Animating with Replicators

You can also replicate animated items. These animations can be either through keyframes or deformers. To specify an animated item as a prototype, ensure that the item layers are properly set up as a hierarchy, and then specify the topmost parent item as the prototype in the Replicator. After this, enable the **Include Child Items** option to replicate all items in the hierarchy. To add additional randomization, you can use the **Time Offset** and **Random Time Variation** options.

In order to animate the Replicator itself, you can animate the settings of the Replicator and/or surface generator, or animate the point source directly. You can apply normal keyframed position and rotation transforms to the point source, but scaling also affects the replicas. In order to scale the point source properly, you need to add a morph to the item and then animate the morph over time. Please see Morph Influence for more on applying and animating a morph.

Billboards

Sometimes it can be handy to use flat polygon planes to clone an image map with a Replicator, maybe used as a forest of distant trees. Using a single polygon, assign the image map to the polygon face and then use that mesh layer as the prototype and assign the point source. To keep the planes from rotating away from the camera, giving away their flat nature, you can assign the Look-At particle modifier, keeping the faces oriented toward the current render camera item. To assign this option, select the Replicator item in the **Items** list and in the menu bar, click **Items** > **Replicators** > **Replicator Face Camera**. This assigns an additional deformer to the scene, keeping the replicas oriented correctly. Although, it may be necessary to make sure the original flat plane is facing toward the correct front direction (Z+); use the **Front** viewport projection type to ensure the correct orientation.

Freezing & Replicators

Replicators work well if you intend to keep your geometry within Modo, however, you may need to export geometry to an external application, or possibly you just need absolute control over the placement of a few individual clones. For such cases, Modo offers the ability to freeze Replicators and their associated particle generators, converting them to Mesh Items with only vertices. The following freeze commands are found in the menu bar under **Items** > **Replicators**

Freeze Replicator: This command converts the selected Replicator Item into a series of individual Instances that all reference the original Replicator prototype. The resulting instances all become children under a **Group Locator** folder in the **Items** List. Each instance has its own transform values for editing individual elements.

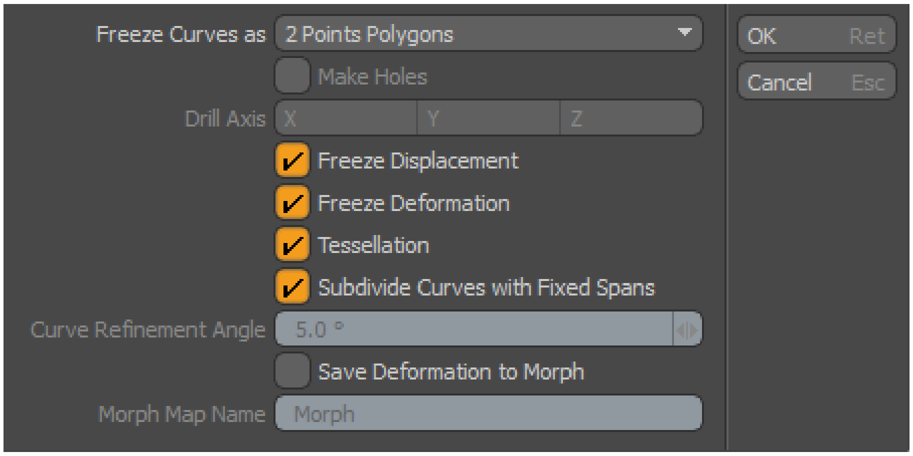
Freeze Surface Generator: This converts the procedurally generated **Shader Tree** Surface Generator points into a Mesh Item layer containing individual vertices for each point. Resulting vertices can be modified with any modeling tools, or edited with the particle paint tools. For more information on these tools, see Using Particle Tools.

Freeze Particle Generator: This converts the procedurally generated Surface Particle Generator points into a Mesh Item layer containing individual vertices for each point. Resulting vertices can be modified with any modeling tools, or edited with the particle paint tools.

Freezing Geometry

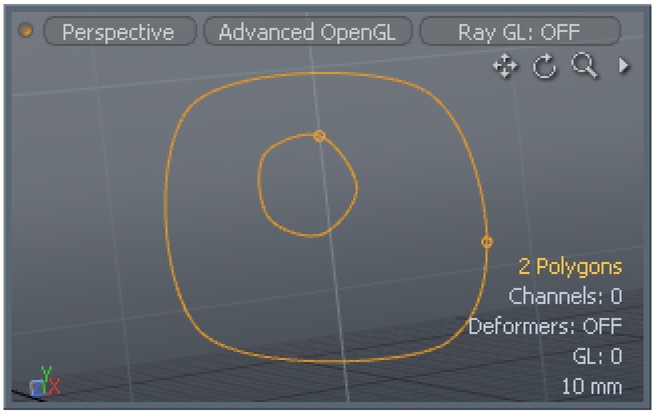
The "freezing" of an object is the act of fixing the geometry into a specific state, such as when a Displacement Map is applied. You can freeze the geometry into a fixed state (as currently-visible in the viewport), allowing for the removal of the Image Maps and/or Deformers. This can be useful when you wish to further edit the model, or export it out of Modo into another application.

Freezing can be applied in a number of situations, to a number of items: Curves, Beziers, Patch Surfaces, subdivided surface geometry, deformed geometry, and Multi-Res sculpted models. The **Freeze** command can be found in the menu bar under **Geometry** > **Freeze**. The command affects any selected geometry when invoked.

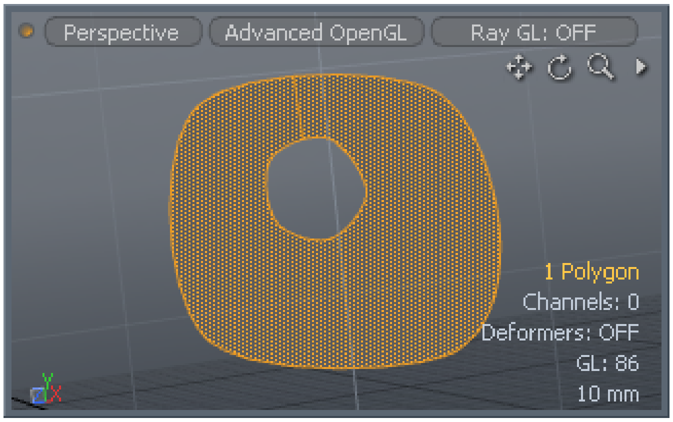

Freeze Curves as: 2 Points Polygons
☐ Make Holes
Drill Axis: X Y Z
☑ Freeze Displacement
☑ Freeze Deformation
☑ Tessellation
☑ Subdivide Curves with Fixed Spans
Curve Refinement Angle: 5.0 °
☐ Save Deformation to Morph
Morph Map Name: Morph

Option	Description
Freeze Curves as	When either of the Modo Curve types are selected for freezing (Adding Curves or Bezier), the **Freeze Curves as** converts the geometry to the selected type.
2 Point Polygons	A two point polygon is simply two vertices connected by an edge. Converting a curve to **2 Point Polygons** produces a continuous string of straight, two point polygon edge segments that approximate the shape of the source curve. Two Point Polygons are not renderable by themselves but they can be extruded into a shell, or used to create a face polygon (n-gon).
Faces	Converts the curve to a multi-sided Polygon (n-gon). If the curve is open, then an edge connect the end location with the start, closing the shape.
Polylines	Polylines are very similar to two-point polygon strings, but also retain some qualities of curves. Polylines have a **Start** and **End** point like a curve, but are made of a string of straight edge segments that approximate the shape of the source curve. Unlike Two-Point Polygons, they render when the **Render Curves** option is enabled, and they are extremely fast to edit and simulate when used as Hair Guides.
Make Holes	When freezing curves as Faces with overlapping shapes, especially those created by the importing of Encapsulated Postscript curves from a vector drawing program, enabling the **Make Holes** option tells Modo to attempt to drill the overlapping shapes automatically.

Two overlapping curves example:


Perspective
Advanced OpenGL
Ray GL: OFF
2 Polygons
Channels: 0
Deformers: OFF
GL: 0
10 mm

Make Holes enabled example:


Perspective
Advanced OpenGL
Ray GL: OFF
1 Polygon
Channels: 0
Deformers: OFF
GL: 86
10 mm

Option	Description
Drill Axis	Depending on the facing direction of the initial curves, the 'Drill Axis' option is used in conjunction with the 'Make Holes' command, determining the axis direction of the automatic Drill operation.
Freeze Displacement	When the **Freeze Displacement** option is enabled, Modo recognizes any Image Map layers assigned to the geometries surface defined as **Displacement** or **Vector Displacement** when freezing and applies them accordingly, retaining the image-based displacements. Once frozen, you can remove the displacement effects. When disabled, Modo ignores displacement layers and only freezes the raw Subdivision cage (also called the Base Mesh). The number of polygons of the final frozen model is determined by the **Subdivision Level** option in the Mesh Item properties, or the **Current Level** when freezing Multi-resolution meshes.

Option	Description
Freeze Deformations	The **Freeze Deformation** option, when enabled, considers any type of Deformer applied to the item for the purposes of posing and animating when freezing the object, applying the effect of the deformer(s) directly to the geometry, and then removes its influence on the mesh (it won't delete the deformer itself though). The target is frozen in its current state (based on the current frame for animated deformers).
Tessellation	The **Tessellation** option freezes the current subdivision surface level for the target mesh for both regular and Pixar subdivision surfaces geometry (based on the viewport display, not the render level). By disabling the **Tessellation** option, and enabling the **Freeze Deformations** option, it is possible to freeze the effects of deformers (such as joints for posing characters) and keep the subdivision surface base geometry intact. (Previously Modo froze the subdivision surface level when freezing deformers. This option separates that function from the application of the deformer, allowing you to pose a model using the various deformers and then freezes the pose and further manipulate the model without creating extra unnecessary geometry).
Subdivide Curves with Fixed Spans	When this option is enabled, the **Freeze** command converts curves based on the Mesh Item setting **Curve Refinement Angle**, generating an identical number of spans between each curve point. Since this value controls how curves are displayed, frozen curves look identical to the source curves viewport display. When this option is disabled, you can then define their own **Curve Refinement Angle** value in the following input field using a more economical adaptive algorithm that generates more divisions around curved areas and less for the straighter sections.
Curve Refinement Angle	When the **Subdivide Curve with Fixed Spans** option is disabled, the **Curve Refinement Angle** value here can be used to define the spans that make up the resulting geometry, based on the angle of change between spans. Smaller angles generate more spans and larger values produce less, using a adaptive algorithm that generates more divisions around curved areas and less for the straighter sections.
Save Deformation to Morph	The **Save Deformation to Morph Map** option allows you to take a deformed mesh and automatically save its current deformed state as a Relative Morph Map. This can then be re-applied as a Morph deformer to produce the same geometric result. To use, set the deformer(s) to the state that is desired for the final result, or if necessary adjust the Timeline to the location of the proper state. With the target mesh selected

Option	Description
	in **Items Mode**, choose the **Geometry** > **Freeze** command from the menu bar, then disable all options except **Freeze Deformations** and the **Save Deformation to Morph Map** option. Define a name under the **Morph Map Name** input field, if desired, and press **OK** to generate the morph. When complete, any attached deformers aredisabled (but still connected to the mesh). To easily view the resulting morph map, change the workspace to the modeling view and then select the named Morph Map from the **Lists** viewport.
Morph Map Name	When the **Save Deformation to Morph Map** option is enabled, this input field allows you to name the resulting morph map. In situations where there is a name collision with an existing map, the previous morph map is overwritten.

Tip: Displacements applied by way of procedural texture layers are not considered when freezing, regardless of the **Freeze Displacements** option. You need to convert the procedurals to Image Maps by way of baking prior to freezing the mesh.

Freeze for Procedural Modeling

The **Freeze** mesh operation is a procedural modeling tool, which rasterizes non-face type polygons into face polygons or specific shapes. This tool works similarly to the traditional Freeze tool.

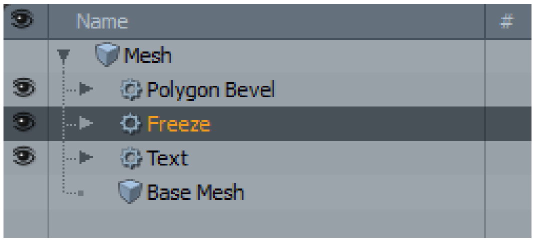
Note: For more information on the Freeze tool, see Freezing Geometry.

This operation is useful, for instance, when working with text. By default, text in your scene is made up of text polygons. Applying mesh operations to text polygons can produce unexpected results. Therefore, it is better to convert them into traditional polygons, then you can apply mesh operations to them without problems.

In addition to converting text to polygons, you can also use this operation to freeze curves or certain parts of a mesh, similarly to how you would use the non-procedural Freeze tool.

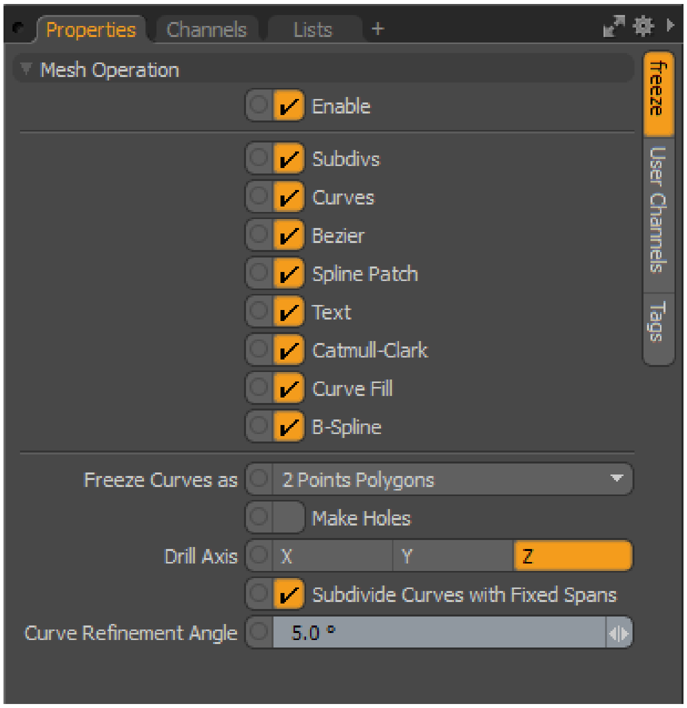
You can add the operation in the **Mesh Operations** list of the **Mesh Ops** tab. Click **Add Operator**, then in the **Add Mesh Operation** panel, click **Edit** > **Freeze**.

The image below shows the **Mesh Operations** list with a Text item, a Freeze operation converting it to two-point polygons, followed by a Polygon Bevel.


Name
#
Mesh
Polygon Bevel
Freeze
Text
Base Mesh

Freeze Tool Properties

You can specify the polygon types to freeze by toggling the settings in the Freeze operation's **Properties** panel.


Properties
Channels
Lists
+
Freeze
User Channels
Tags
Mesh Operation

☑ Enable

☑ Subdivs
☑ Curves
☑ Bezier
☑ Spline Patch
☑ Text
☑ Catmull-Clark
☑ Curve Fill
☑ B-Spline

Freeze Curves as: 2 Points Polygons
☐ Make Holes
Drill Axis: X Y [✓ Z]
☑ Subdivide Curves with Fixed Spans
Curve Refinement Angle: 5.0 °

Freeze Properties	
Freeze Curves as	When any curve type (Curve, Bezier, or B-Spline) is selected for freezing, this option converts the geometry to the selected type. The available options are: • **2 Points Polygons** • **Faces** • **Polylines**
Make Holes	When freezing curves as faces with overlapping shapes, especially those created by importing **.eps** curves from a vector drawing program, enabling this option

Freeze Properties	
	tells Modo to attempt to drill the overlapping shapes automatically.
Drill Axis	Depending on the facing direction of the initial curves, you can use this option with the **Make Holes** command, determining the axis direction of the automatic drill operation.
Subdivide Curves with Fixed Spans	When enabled, the Freeze command converts curves based on the Mesh Item setting **Curve Refinement Angle**, generating an identical number of spans between each curve point. Since this value controls how curves are displayed, frozen curves look identical to the viewport display of source curves. When disabled, you can define your own **Curve Refinement Angle** value, using a more economical adaptive algorithm, which generates more divisions around curved areas and less for the straighter sections.
Curve Refinement Angle	When the **Subdivide Curve with Fixed Spans** option is disabled, the **Curve Refinement Angle** value is used to define the spans that make up the resulting geometry, based on the angle of change between spans. Smaller angles generate more spans, while larger values produce fewer spans, using an adaptive algorithm that generates more divisions around curved areas and fewer for the straighter sections.

Using Dynamics and Particles

Modo provides a useful **Setup** interface, with access to item transformation and duplication tools, a **Preset Browser** to speed up setup tasks, and tools for preparing and refining dynamics and particle simulations. The Dynamics and Particles and Simulations outline the dynamics and particle simulation workflows in Modo, followed by detailed information on each of their aspects.

Dynamics

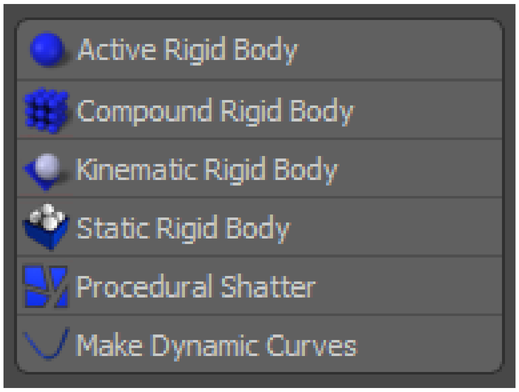
Modo dynamics provide a way to automatically animate and position elements in a scene. Dynamics allow items to be assigned with real-world physical attributes, such as weight, bounce and friction. Then, through a process called simulation, the assigned attributes are evaluated scene-wide, including the further effect of forces over the elements and their interactions with other dynamic elements (collisions). The resulting motion calculations are then cached and saved with the scene for Timeline playback and rendering purposes. Dynamically simulating motion in this way produces highly realistic results that would be very difficult, if not impossible, to generate otherwise.

The Modo dynamics system uses the same global evaluation engine as it does for particles, therefore a lot of the same workflows can be applied to both setups. Due to their deep integration inside Modo, they can also work together, with dynamic items affecting particles and particles affecting dynamic items. This overview page focuses on dynamics specifically, for more information on working with particles, see Particles and Simulations.

Dynamics Workflow Overview

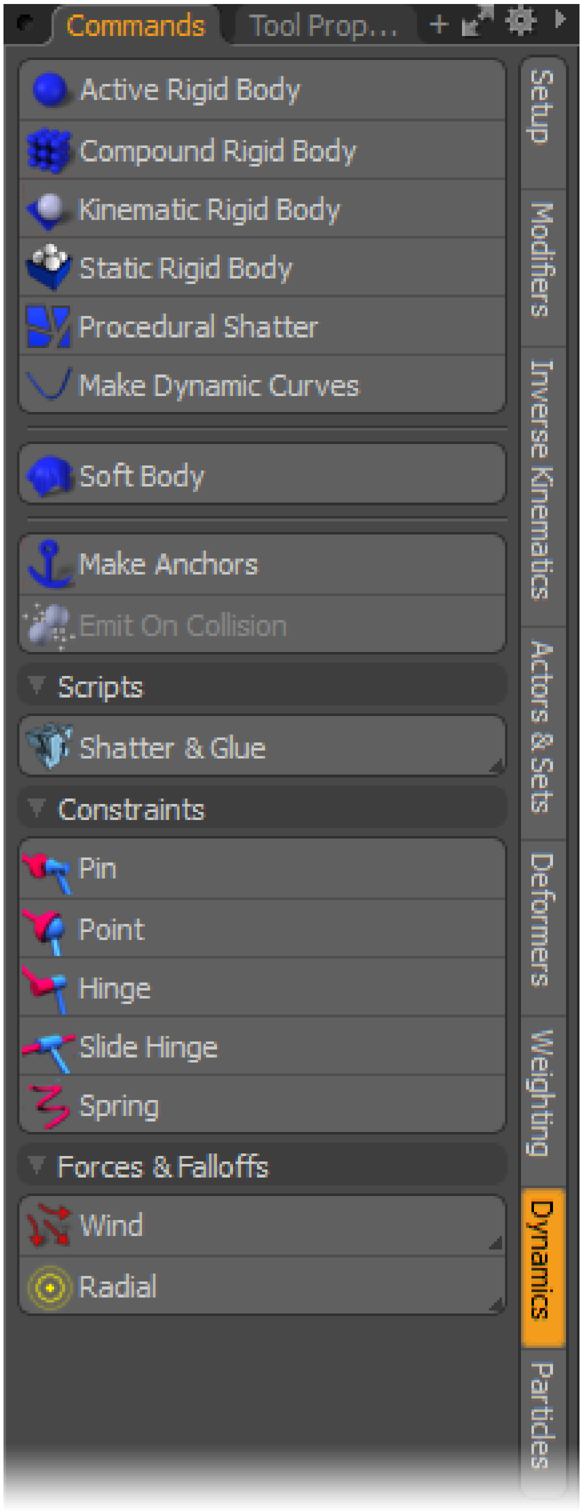
Dynamics first requires a selected element to apply the settings to. While dynamics can be applied to any locator-type item, including cameras and lights, to the simulation engine these non-Mesh Items have no volume to calculate collisions. Therefore, any simulation should start with a Mesh Item. Another important consideration is the location of the center position relative to the geometry. The center location defines the rotation origin as well as the center of gravity for the object.

With the target mesh selected, assign the dynamic properties using the **Dynamics** controls toolbox found in the **Setup** interface. You can also access many of the same commands in the **Dynamics** menu of the menu bar. As with all animated elements, you should be working in Items selection mode when assigning and working with dynamic elements.


Active Rigid Body
Compound Rigid Body
Kinematic Rigid Body
Static Rigid Body
Procedural Shatter
Make Dynamic Curves

The basic workflow to designate an object as dynamic is to select the target element and assign a type of dynamic item to it. Click the appropriate button in the toolbox, such as **Active Rigid Body** (toolbox illustrated above). This assigns the item as dynamic and adds a new **Property** sub-tab called **Dynamics** to the **Properties** panel. Selecting this sub-tab gives you access to all the various dynamics settings. It also adds a new item to the scene, the Dynamics Solver item that defines the global dynamics settings, such as gravity, and stores the cached evaluations of the computed simulation.

Dynamic Item Types


Commands
Tool Prop...
Active Rigid Body
Compound Rigid Body
Kinematic Rigid Body
Static Rigid Body
Procedural Shatter
Make Dynamic Curves
Soft Body
Make Anchors
Emit On Collision
Scripts
Shatter & Glue
Constraints
Pin
Point
Hinge
Slide Hinge
Spring
Forces & Falloffs
Wind
Radial
Setup
Modifiers
Inverse Kinematics
Actors & Sets
Deformers
Weighting
Dynamics
Particles

The **Dynamics** toolbox provides a variety of options for assigning items as dynamic. When any item is tagged as dynamic, controls are added to the item's **Properties** panel under the **Dynamics** sub-tab. The attributes for each different option are actually the same controls, but with different base setting specific to the type of dynamic item chosen. All the various controls are covered in their respective page of the documentation.

Active Rigid Body

An active rigid body is an inflexible, dynamic object that does not deform in any way, reacting to all manner of collisions, constraints, forces, and falloffs, as well as influencing other dynamic items through direct interaction. This setting is great for simulating all kinds of hard objects, like blocks, rocks, bowling pins, broken glass, ski poles; anything that is solid and has weight to it is a good candidate for the active rigid body setting. Active rigid bodies are entirely under the control of the evaluation engine during the course of the simulation and don't respond to keyframed animation. Additional control over active rigid bodies is achieved using Dynamics Forces and Constraints.

For best results, each individual object needs to be its own item layer. Multiple objects within one layer are treated as a single object, but depending on the shape they may produce unsatisfactory results. For Mesh Items, the center location relative to the geometry defines the rotation origin and also acts as the center of gravity for the weight calculations, so it is best to make sure it is located within the volume of the object. Any locator-type item can be tagged as **Dynamic**, but any non-Mesh Items, such as cameras and lights that have no volume to collide with, need to be connected to a constraint in order to keep them from flying away.

Compound Rigid Body

Compound rigid bodies let you attach a collection of separate, but touching items together, treating them as a single active rigid body. To do this, select all the various items and click the **Compound Rigid Body** button. Once designated, you can assign a **Constraint Modifier** that can hold the items together during a collision. If the Constriants **Glue** value is exceeded by an outside force during the simulation, it causes the affected parts of the compound to break off individually.

When assigning the **Compound Rigid Body** option, the dynamics package is added to only a single mesh (the lowest item in the **Items** list). The other associated meshes all derive their dynamics settings from the parent item, with the **Mass** value being equally distributed between all associated layers. Setting a negative 'Compound Glue' value causes the simulation to treat each Item as a separate Dynamic Body without having to make each one dynamic. The same requirements for 'active Rigid Bodies' apply to the individual items of the Compound body.

Kinematic Rigid Body

The **Kinematic Rigid Body** option is applied to animated collision objects; specifically to animated Mesh Items used for collision. Other Locator type items, such as cameras or lights cannot be used for collision shapes. Kinematic rigid bodies share the same properties as Static Rigid Bodies, meaning they are not affected by forces, falloffs or collisions from other objects, but they play any keyframed transform animations assigned to the item (not deformations). This is useful for objects that need to interact with other dynamic items in a controlled way, such as a continuously spinning cage in a lottery ball animation.

Static Rigid Body

A static rigid body is strictly a non-moving collision shape. It doesn't respond to keyframed animation, constraints, forces or falloffs. If the item is animated, the animations are ignored during the simulation, with the object stopping at the first frame where it becomes a participant in the dynamic simulation.

Procedural Shatter

The procedural shatter option is used to dynamically break apart a mesh, without needing to pre-shatter the object manually; you start with one mesh and end up with multiples, but those multiple pieces are represented by just the one procedural shatter item in the **Items** list, keeping things tidy. You can control how the object breaks apart, including how sub-meshes are treated, using the settings of the procedural shatter item. To apply, simply select the target item to shatter and click the **Procedural Shatter** button. The original item at that point becomes a proxy and can be hidden or disabled. Now when a simulation is run, the item is automatically broken apart based on any collisions occurring in the scene.

Dynamic Curves

Dynamic curves allow you to animate curve-specific elements, having them react to gravity, collisions, pins, anchors and falloffs. Dynamics for curves can be assigned to both spline-type curves and the two-point polyline chains used for hair guides. Curves can be open or closed. The dynamics of the curve are based on collisions of the vertices that define the curve itself, acting as if they had zero volume. To create curves that have volume and width, use the combination of the **Render Curves** option of the Mesh Item properties, and the **Use Render Radius** option under the Dynamic Item's properties (enabled by default). Then you can specify a **Curve Radius** value to define the actual size for the simulated curve.

Dynamic curves differ from other dynamic items in that multiple, separate curve elements can all be contained within a single Mesh Item layer, but each reacts individually. For information on how to change this behavior, see Dynamic Item. In addition, the center position for the item's layer relative to the curves within the layer is irrelevant, having no effect on how the curves deform. Also, the **Collision** settings have no effect on curve elements as the collision shape is created automatically. Curves with volume are simulated by creating a virtual collision shape, a capsule, between each vertex of the curve.

Note: Any polygonal mesh elements within the same layer as the curves are ignored during the simulation.

Soft Body

Soft bodies are a good way to animate flexible items, such as curtains, flags, table cloths, or rubber tires. For the best results, soft bodies need some type of collision element or Anchor in order to keep them from flying away. The collision object could be a bar holding up a banner, or a table setting under a tablecloth. These collision items should be designated as any one of the rigid body dynamic types. Alternately anchors attach soft body elements to specific points on the rigid body mesh, fixing them in place. You can also use forces to interact with soft bodies; for example, you can assign wind or turbulence to add some additional realism to the soft body results.

Soft body elements can be open or enclosed Mesh Items, and do not require a specific center position. The shape of the resulting soft body is calculated by adding tiny virtual springs between vertices of the mesh. There are different settings for each type of spring; for **Stiffness**, **Struct** (Structure) and **Sheer**.

Since only the vertices of the soft body are calculated for collision, there may be cases where sharp or pointed areas of the colliding mesh show through. To correct this, increase the **Margin** value under **Collision** for the target object. For more information, see Dynamic Item.

Anchors

The **Make Anchors** options is used to connect soft body items to rigid body items. For best results, the rigid body item should be a passive rigid body, to keep it from flying away. The **Anchor** option could be useful to connect a flag to a flagpole. In order for the anchor to work, follow this specific workflow:

1. In **Items mode**, select the target soft body item and the target passive rigid body item.
2. With the items selected, change into the **Vertices** mode (press **1** on the keyboard) and select at least one vertex on the soft body item, and at least one more on the rigid body (selection order isn't important).
3. With the vertices selected, click **Make Anchors** in the **Dynamics** toolbox.

   The two items are now rigidly connected together with an additional anchor item added to the soft body.

To break the connection, simply delete the resulting anchor item. To make adjustments, remove the anchor and recreate it, following the above instructions.

Scripts

Modo provides ready-made Python scripts as a shortcut to add dynamic behavior to elements in your scene. There are currently two scripts available: **Dynamic Curves as Hair** and **Shatter & Glue**.

Dynamic Curves as Hair

This adds hair type dynamics to selected Dynamic Curves. This can be used to add dynamics to hair by applying it to Hair Guides. For example, you can add hair guides, and with the guides selected, click **Dynamic Curves as Hair** to add hair dynamics. You can then see the behavior by playing the simulation.

Shatter & Glue

Combines Shatter with Glue (as defined by a Constraint Modifier) so that the item shatters subject to a user-defined level of glue between the shatter pieces.

To apply this script, highlight the item and click **Shatter & Glue**. You must first set the **Dynamic Glue Strength** for the object, which corresponds to the Glue Value property for the Constraint Modifier. You then select the **Shatter Type** to determine how object shatters. For a full explanation of different Shatter Types and their properties, see Shatter

Collision Detection

Collision in dynamics is the calculations that restrict intersections between surfaces. For example, it is what allows a ball to bounce and roll down a staircase. Without it, the dynamic bodies simply pass through one another with no regard or knowledge of the opposing item. In the **Dynamic** sub-tab in the item's **Properties** panel are the settings that control the object and how it acts. The most important setting in this tab is in regards to collision, as it defines how the dynamics computations see the object.

By default, Modo assigns a collision type based on a best guess from the actual geometry in the item layer. For a flat polygon this is Plane or for a round object, like a ball, this is Sphere. However, sometimes Modo guesses wrong. The wrong collision shape can cause intersection errors and produce unsatisfactory results, so it's a good idea to check that the best collision shape is being assigned. In most cases, the best results are obtained using the simplest collision shape appropriate to the geometry. Collisions relating to soft bodies are only required when there are issues with the surface intersecting itself, which you can fix by enabling the **Self Collision** option.

Tip: To increase evaluation performance of rigid bodies, low resolution collision geometry can be created separate from the high resolution mesh to simplify calculations during the simulation. Once the dynamics are evaluated and cached, the low resolution collision objects can be set to **Render-No** and the high resolution object can be parented to the low res collision object for final rendering.

Constraints

If dynamics were strictly limited to only calculating collisions between objects, it would very quickly become daunting to produce even modestly complex simulations. Depending on collisions exclusively to generate the mechanics of, say, a rotating drive shaft for a vehicle would no doubt introduce errors and be quite costly to calculate. Constraints are joints that connect two dynamic bodies together, producing much more accurate results than collision detection alone, with very little calculation overhead. Constraints come in a variety of types that cover almost any type of joint you need. You can also stack more than one constraint to produce an even more complex effect, such as combining a point constraint with a slide constraint to produce a swinging and sliding motion, much like riding suspended from a zip-line, a ride where the participant slides down a long, taut cable suspended between two posts.

Forces

In the real world, objects are constantly subject to numerous forces that affect their motion in subtle and not-so-subtle ways; wind, gravity, friction among others. Dynamics offers a variety of additional specific forces to produce even more realistic physical simulations. Gravity is the chief overall governing force, acting on all dynamic bodies globally. Because of this ubiquity, it is part of the Dynamics Solver item, affecting all items globally. Its default value is meant to simulate the same downward pull at full size (1:1 scale) as gravity in the real world. If a simulation isn't reacting as you might expect, the scene may not be modeled to real world size, as very large items react much differently under the same circumstances as very small items. You can scale their objects, or scale the simulation to match more closely the expected result. For more information on scaling the simulation, see Dynamics Solver.

Modo also offers seven additional force types that gives you further control of the simulation: **Linear**, **Drag**, **Radial**, **Turbulence**, **Vortex**, **Wind** and **Curve Force**. These are added as locator-type item layers in the **Items** list and, by default, affect dynamic items globally, just as the gravity value does, although this can be modified using the **Schematic** view. Each force has its own particular function which is covered fully in the Dynamics Forces topic.

Falloffs

Falloffs can be very useful in providing additional control over forces. For instance, if wind was affecting flower petals drifting lightly on the breeze, and the soft body petal items then landed inside of a box. To keep the ones inside the box from being affected by the wind a falloff would be used to eliminate the wind from inside the box, increasing the realism of the simulation.

Dynamics uses the same Falloffs used for other deformers in Modo. Deformation falloffs are meant to modulate the effects of forces on dynamic items within the area of influence of the falloff itself. Not to be confused with Modo modeling falloffs used for geometry creation and layout. Conceptually they are very similar, but instead of being applied as a function, Dynamics falloffs are added as a locator type item to the **Items** list where you can select them and modify their attributes and even animate them within the scene. In the item's properties, you can define an attenuation range that modulates the force it is applied to.

Unlike forces that affect items globally, falloffs must first be attached to the force item it is attenuating in order to produce a result. This can be done several ways. Generally, it is applied by selecting the target force item prior to adding the falloff, when using the **Add Falloff** button in the **Dynamics** toolbox. When applied to a soft body item, you can use the Deformers Viewport, and select the **Soft Body Influence** from the list. By right-clicking on the item to open the context menu, you can select the **Add Override** option and select one of the falloffs, automatically assigning it to the soft body. Additionally, you can use the **Schematic** viewport to connect the two, dragging and dropping both items into the viewport and connecting the output of the falloff item to the input of the force item. For more information on using the **Schematic** viewport, see Schematic Viewport.

Dynamic Replicators

Dynamic Replicators bring the worlds of dynamics, particles and Replicators together, allowing you to assign Mesh Items to a particle simulation to spray blasts of dynamic, colliding objects. There are two basic types of Dynamic Replicators - kinematic, which act only as animated collision objects (acting like replicated kinematic rigid bodies), but have no internal collision calculated between replicas, and dynamic which act as normal active rigid bodies, generated automatically by the particle simulation.

Kinematic

Replicators combined with a particle simulation can be used strictly for collision shapes, which can be useful when you need the replicated items to affect other non-replicator dynamic rigid body elements, but not be affected by inter-object collisions between replicas. The individual replicas would essentially be kinematic rigid bodies in how they react to other elements. They wouldn't be affected by forces, falloffs or other dynamics objects; they would simply have effect on regular active rigid bodies, similarly to force, causing them to react to the collisions.

To create Kinematic Replicators

1. Create a Particle Simulation of some sort. You can use any of the various emitters available for this.
2. Next, add a Replicator in the **Items** list, in **Add Items** > **Particles** > **Replicator**.
3. Once added, designate the particle simulation as the **Point Source** in the Replicator Item's **Properties** panel, and the target Mesh Item (or a group of items) as the **Prototype** option.
4. Select the Replicator Item in the **Items** list and then in the **Dynamics** toolbox, click **Kinematic Rigid Body** to tag it as one.

   This adds a sub-tab to the **Properties** panel of the Replicator Item titled **Dynamic Replicator**, where the various dynamic attributes can be adjusted, such as the **Collision Shape** for the prototypes and their **Bounce** and **Friction** values.

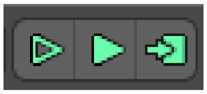
Dynamic

Replicators can also be combined with a particle simulation to generate fully dynamic Mesh Items that collide and react to forces and falloffs. Any type of particle emitter can be designated as the particle source for a Replicators. Adding dynamics allows the replicas to act as individual active rigid body elements, created dynamically by the settings of the particle simulation.

To create a Dynamic Replicator, you need a particle simulation and a Replicator with the target mesh specified as the **Prototype** and the particle simulation specified as the **Point Source**. The workflow is the same as for the Kinematic Replicators above. Once properly assigned, make the Replicator dynamic: select the Replicator in the **Items** list, and in the **Dynamics** toolbox, click **Active Rigid Body** to tag it as one. This action adds a Dynamic Replicator item to the scene. This new item contains the dynamic attribute settings for the dynamic replicas, such as settings for the **Collision Shape**, **Bounce**, **Friction** and **Density**. Select the item in the **Items** list to display the attribute settings in the **Properties** panel.

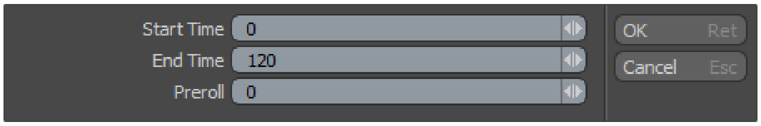
One consideration to keep in mind when making Dynamic Replicators is that the replicated items should be generated from an emitter that is sized large enough to accommodate the volume of replicated Mesh Items without grossly intersecting, while also taking in to consideration the **Emission Rate** and **Emission Speed** values. Additionally the Dynamic Replicator Item has the option **Time Until Collision**, which delays the enforcement of collision detection on emitted Replicators that can also be used to eliminate any collision errors that may occur.

Simulation Controls



There are three buttons used when simulating both dynamics and particles, each with a unique purpose. The two left buttons provide continuous real-time simulation directly in the 3D Viewports, allowing you to adjust settings and move objects and see the results as the settings are changed. The **Preview at Current Time** button (hollow arrow) specifically plays the simulation at the current Timeline position, ignoring keyframed animation. the **Preview Animation** button (solid arrow) cycles through playing the Timeline with all keyframed animations. When you click either button, they change to **Stop** buttons which can then be pressed to stop the simulation playback.

The last button is the **Cache Simulation** button. Click this one to open the **Evaluation** dialog, where you enter the desired frame range and designate a **Preroll** value, a designated number of frames prior to the first frame where the simulation actually begins calculating. This is useful, for instance, when there needs to be a flow of particles at the **Start Time**. The main differences between the real-time playback controls and computing the simulation is that the **Compute Simulation** option fully calculates all Physics Rate (subframe) step values, especially important for collisions to properly evaluate intersections. The real-time controls always use a frame step of **1**, which may not provide enough accuracy for fast or complex dynamic simulations. Once a simulation is cached, it is no longer deleted when a simulation preview is run, and can only be overwritten by computing a new simulation.



Start Time 0

End Time 120

Preroll 0

Additional Dynamics Considerations

Items vs. Components

Animation in Modo applies only to items. Keyframes are always set while in **Items mode** with the modeling of objects done while at the component level, in the vertices, edges and polygons modes. Layers in the items list should be thought of as containers. In **Items mode**, you are editing the entire container, but while in the component modes, you are editing the contents of the container. Dynamics acts only at the item level, where each layer in the **Items** list is its own individual dynamic object. Therefore, if you wish to break apart a wall, for example, each brick needs to be its own individual item layer. You can easily set up stacks of bricks by simply creating the first brick item, and then assigning the **Active Dynamic Body** tag to make the brick dynamic. Next, apply the Array function in **Items mode** to stack subsequent bricks, each inheriting the dynamics of the master object. With the wall created, all it takes is an animated wrecking ball to topple the bricks to the ground, tagged as a **Kinematic Rigid Body**.

Centers and Mass

The weight of rigid body objects is dependent on the position of the item's center. The center point for an object doesn't only represent the center of rotation for the object; when using dynamics, it also represents the center of mass for the object. Dynamics assumes that the weight of an object is equally distributed around the center point, and objects move based on this assumption. If the center point is placed too far away from the item's actual center, or outside the volume of the object, odd behavior is sure to occur. You can see an object's center position by hovering over the 3D viewport in question and pressing the **O** key. In the **Visibility** sub-tab, there is an option to show centers for selected items.

There are several ways to deal with a center that's been moved, the easiest is to select the item in **Items mode** and in the menu bar, click **Edit** > **Center to Bounding Box** > **Center**. This automatically moves the center point to the objects actual bounding box center (a bounding box is based on a model's overall extents). However, a more accurate option is to right-click over the item in the **Items** list and select **Calculate the Center of Mass**, and then use the match tools to move the center to the position of the resulting locator.

You cannot manually (by keyframes) animate active rigid bodies, however, sleeping objects can be animated, and are handed off to dynamics once they are awakened, a mode where dynamic items can be triggered by a specific event. For more information on sleep, see Dynamic Item. If the animated object requires rotations from a position other than the object's true center, modify the **Pivot** position and animate from there to maintain the proper center of mass.

Accuracy

The most frequent issue encountered when calculating a simulation is objects passing through one another (like when you are dropping chain links together to form a chain and they fall right through each other). While it can be caused by a multitude of issues, it is most often caused by the speed at which objects move. At one frame they are apart from each other, the next they pass right through one another. This means the default subframe calculations didn't have ample time to calculate the collisions of the objects. Even though we only see the result at the whole frame values, and a lot can happen between frames.

You can increase this sub-frame accuracy with the **Physics Rate** values found in the Dynamics Solver (properly calculating subframes for all keyframed items, as well). Doing so makes dynamics more accurate, however the additional calculations take longer to compute. Experiment with different values to find one that produces acceptable results with the least number of steps. Keep in mind that sub-frame steps are only calculated for cached simulations. The real-time options always only calculate per-frame, so for fast simulations, it may be necessary to cache the simulation each time to properly preview accurate results.

Schematic Viewport

Dynamic simulations can be controlled nodally using the Schematic Viewport. Actually, in the background a node graph is being created as you add dynamic elements to a scene. It seems hidden until any of the dynamic elements are added to the **Schematic** viewport. As the dynamic elements are added one-by-one, you can see that they are already connected. Dynamics take advantage of the special **Relationship** connectors to define links between items. These connectors look like diamonds instead of circles. They represent bi-directional information flow between elements, unlike regular channel connections that are used to drive a channel's information from one item to another. It is useful to familiarize yourself with the **Schematic** viewport, as it allows advanced options to control and customize the dynamics simulation. For instance, you can connect a force node to an individual Mesh Item and it only affects that single mesh instead of applying itself globally.

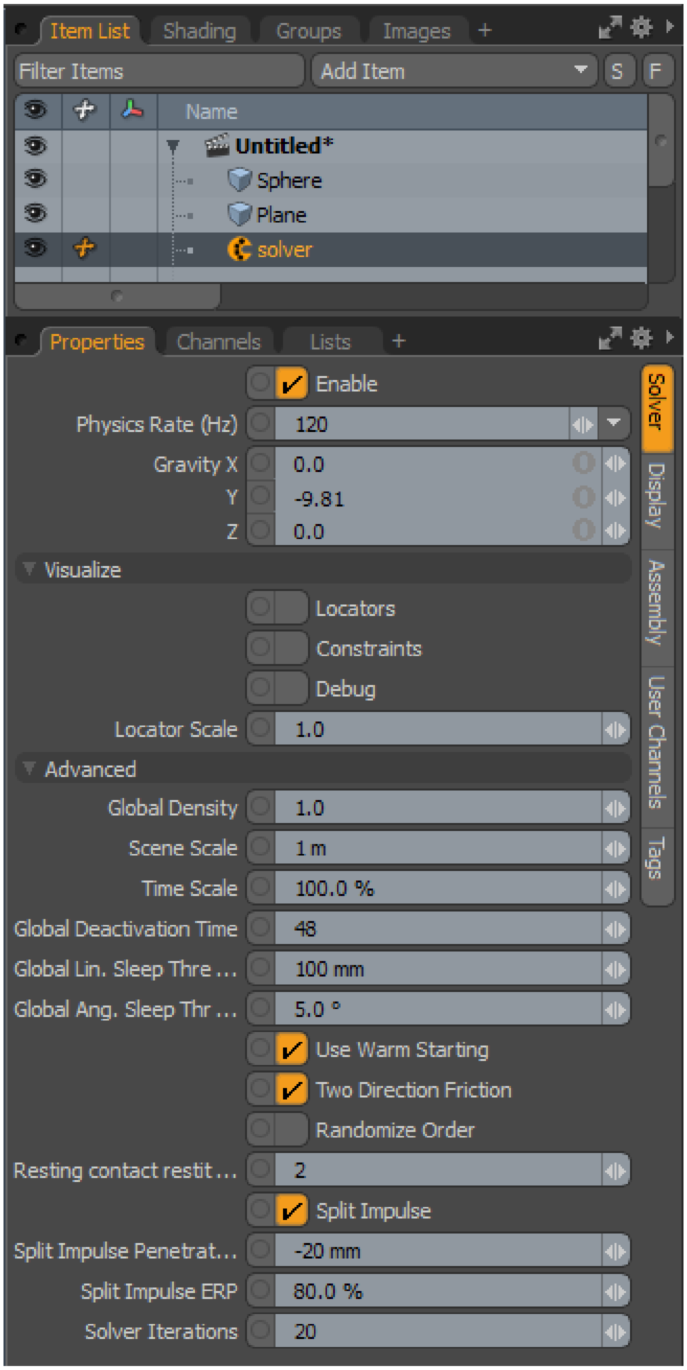
In addition to Mesh Items, Instances can also be tagged as the various types of rigid bodies **Active**, **Kinematic**, and **Static**. While they can be tagged as **Dynamic**, due to their structure Static Mesh Items produce no collisions between other dynamic items.

Bullet Physics

Dynamics is based on Bullet Physics, an open source collision detection and rigid body dynamics library that is free for commercial use, under the ZLib license. Thank you to Erwin Coumans and the Bullet developers for their efforts and generosity.

Dynamics Solver

When any item in the scene is defined as either an **Active Rigid Body**, **Passive Rigid Body** or **Soft Body**, a dynamics solver item is added automatically to the scene in the **Items** list. Only a single instance of the solver item is ever necessary in a scene, as it controls all the global parameters of the simulation, such as time steps, gravity, and so on. If the dynamics solver item is ever accidentally deleted from the scene, you may add another from the menu bar, clicking **Dynamics** > **Add Solver**. With the item selected in the **Items** viewport, a series of attributes can be modified in the **Properties** panel.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
S
F
Name
Untitled*
Sphere
Plane
solver
Properties
Channels
Lists
+
Solver
Display
Assembly
User Channels
Tags

☑ Enable
Physics Rate (Hz) 120
Gravity X 0.0
Y -9.81
Z 0.0

Visualize
☐ Locators
☐ Constraints
☐ Debug
Locator Scale 1.0

Advanced
Global Density 1.0
Scene Scale 1 m
Time Scale 100.0 %
Global Deactivation Time 48
Global Lin. Sleep Thre … 100 mm
Global Ang. Sleep Thr … 5.0 °
☑ Use Warm Starting
☑ Two Direction Friction
☐ Randomize Order
Resting contact restit … 2
☑ Split Impulse
Split Impulse Penetrat… -20 mm
Split Impulse ERP 80.0 %
Solver Iterations 20

Option	Description
Enable	Enabled by default, you can disable this option to temporarily suspend the Dynamics simulation calculations.
Physics Rate (Hz)	Represents the rate at which the dynamic evaluations occur during a

Option	Description
	second of scene time (regardless of the FPS setting). At a FPS rate of 30, the default Physics Rate of 60 provides for 2 calculation steps per frame. Even though you only see the results of a simulation at each whole frame step, there can still be a lot of movement happening between frames, especially with fast moving objects. If objects appear to intersect or are moving through each other, increasing this value helps to eliminate these errors, however, as the rate increases, so does the number of calculations. Higher numbers offer more precise simulations at the expense of greater time to calculate. **Note:** To see the results of the **Physics Rate**, you need to do a full cached simulation, as both real-time preview modes always default to a physics rate of one update per frame.
Gravity X/Y/Z	Control the direction and magnitude of gravity. Gravity is a world force value that affects everything in the scene. The default value of **-9.81** simulates the downward pull of gravity on Earth at a 1:1 scale (meters per second).
Visualize	
Locators	When enabled, draws the center and centers of mass of dynamic objects.
Constraints	When enabled, draws constraints.
Debug Drawing	Draws the internal representation of what the dynamics simulator actually sees. It can be overwhelming to see in complex scenes especially, but can be useful for troubleshooting simulations that don't act as expected.
Locator Scale	Scales the locator drawing.
Advanced	
Body Update Frequency	Determines how often dynamics collects channel and deformations data to update its internal representations for each dynamic item in the scene.

Option	Description
	• **Sim Start**: Internal representation is updated only once at the simulation start. • **Every Frame**: Internal representation is updated at each whole frame value. • **Every Tick**: Internal representation is updated at each **Step**, producing sub-frame accurate movements, but creating a large amount of additional overhead which may greatly slow down the simulation of the scene.
Global Density	Density is the measure of the amount of matter in any given volume. In Modo, geometric objects are technically hollow, and therefore have no density, but if objects were simulated this way, they would just blow away like paper shells. Providing a density value for dynamic objects increases the realism of a simulation. Like metal, as opposed to wood, different materials have different densities, affecting their weight, and therefore the way they move when simulated. The **Global Density** option determines the density of dynamic objects globally in the scene. You can adjust this value, affecting all dynamic objects scene-wide. You can also override the **Global Density** individually per item by setting the **Mass Source** option in the item's dynamic properties to one of the **Local** options and defining a value individually.
Scene Scale	Dynamic evaluations make certain assumptions in their calculations, assuming items to be in roughly real world scale, between the size of a grape up to the size of a van. Simulating items within this range produces the most realistic results, with the ideal size being around 1m in size. Items scaled outside of this range run into issues with accuracy. Instead of scaling the target scene, you can use the **Scene Scale** option to scale the calculations internally, producing better, but slightly less accurate results. The default value of **1 m** is equal to 100%. Changing the **Scene Scale** size to **10 cm** scales the scene up by 10 times (1000%), which useful when simulating very small items. Likewise, changing the **Scene Scale** size to **10 m** scales everything down by a factor of 10 (10%), helpful when simulating very large items.

Option	Description
Time Scale	Generally, a simulation moves items in time, based on real world values of simulated forces. You can use the **Time Scale** option to globally adjust the speed at which a simulation runs simulating slow-motion or super-speed effects when the value is lowered or raised, respectively.
Global Deactivation Time	Defines how long a dynamic object must be below the **Linear** or **Angular** threshold before being put to sleep. The default value is two seconds, so depending on the frame rate of the scene, the default value changes; generally it is 48 frames (at 24fps).
Global Lin. Sleep Threshold	When objects in a Dynamics simulation stop moving, they are put into a sleeping state, so that motion values don't have to be calculated for every object at every frame. This setting defines the lower threshold for putting these objects to sleep. You can override this value individually with the **Linear** option found in the Dynamic Item's properties.
Global Ang. Sleep Threshold	When objects in a dynamics simulation stop rotating, they are put into a sleeping state so that motion values don't have to be calculated for every object at every frame. This option defines the lower global angular threshold for putting these objects to sleep. You can override this value individually with the **Angular** option found in the **Dynamic Item**'s properties.
Use Warm Starting	Dynamics utilizes an iterative algorithm where each iteration is based on the solution of previous iteration. If this option is disabled, the initial solution for the solver algorithm is set to zero for each frame. When enabled, each new iteration uses the last solution of the previous frame. This improves convergence towards a better solution, especially regarding dynamics's stacking stability.
Two Direction Friction	If two objects surfaces are sliding parallel against each other for more than a few frames, errors can be introduced, enabling this option reduces or eliminates these errors.
Randomize Order	Normally, constraints and collisions are calculated in order. Enabling this option calculates the order objects to be randomized. When enabled, this option can aid in the stacking stability of dynamics

Option	Description
	simulations. It has a very small performance cost which is why it is disabled by default.
Resting Contact Restitution Threshold	By default, dynamics implements a mechanism for resting contacts to improve stability, by setting the restitution to zero. It compares the contact lifetime using a threshold to determine resting contact.
Split Impulse	By default, dynamics solves positional constraints and velocity constraints coupled together. This works well in many cases, but the error reduction of position coupled to velocity introduces extra energy (noticeable as bounce). Instead of coupled positional and velocity constraint solving, the two can be solved separately using the **Split Impulse** option. This means that recovering from deep penetrations doesn't add any velocity.
Split Impulse Penetration Threshold	This sets the penetration threshold.
Split Impulse ERP	The Error Reduction Parameter used if a deep penetration happens.
Solver Iterations	Determines the number of iterations between frames that the constraint solver uses to resolve collisions. Increasing this value produces more accurate constraint simulations, but at the expense of longer simulation calculation times.

Dynamic Item

To make an item dynamic, you need to tag it with one of the dynamic item types. Each type determines how the body behaves when moving or colliding with other objects. For example, a **Soft Body** bends and deforms similar to cloth, whereas a **Rigid Body** doesn't deform or break. The different types of dynamic item are described in Dynamic Item Types.

Dynamics are created in the **Setup** layout.

To Make an Item Dynamic

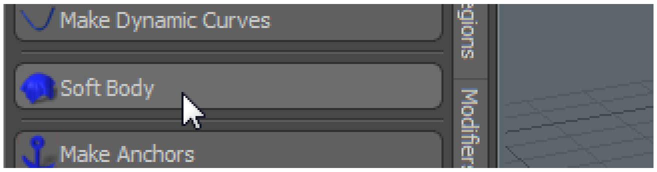
1. Select the **Setup** layout and go into **Items** mode.
2. Select the item.
3. Click the **Dynamics** sub-tab of the **Commands** panel on the left.

This opens the **Dynamics** toolbox.

4. Select the type of dynamic item that you want to apply.

   This tags the selected item with the dynamic item type.

   For example, to apply soft, cloth-like dynamics, select **Soft Body**.


Make Dynamic Curves
Soft Body
Make Anchors
egions
Modifier

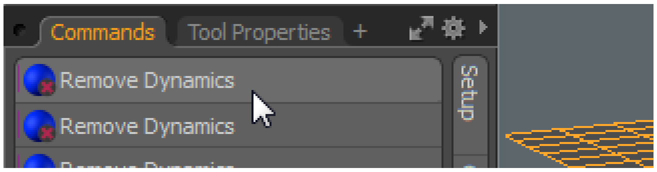
To Remove Dynamics from an Item

To stop an object having dynamic properties:

1. In **Items** mode, select the item.
2. Select **Dynamics** > **Remove Dynamics**.

   OR

   Hold **Shift**. You'll see that most of the dynamic item buttons change to **Remove Dynamics**.


Commands
Tool Properties
+
Remove Dynamics
Remove Dynamics
Setup

   Click any of the **Remove Dynamics** buttons.

To Toggle Dynamics on an Item

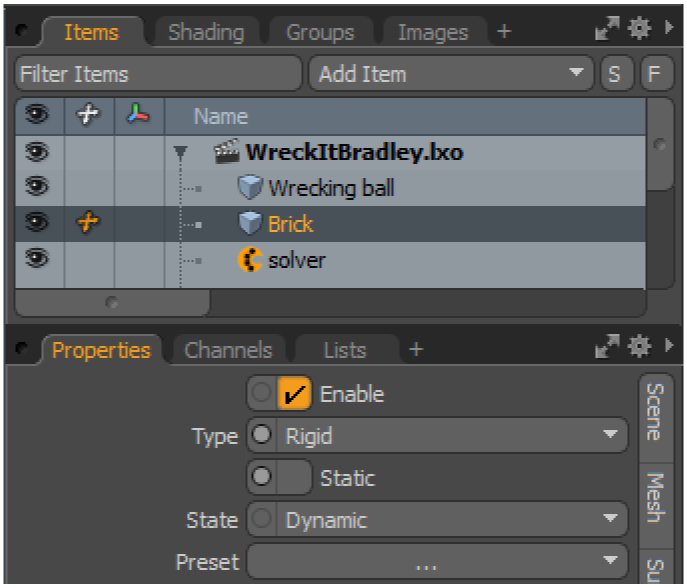
To temporarily disable an item's dynamics, uncheck **Enable** at the top of the **Dynamic** tab in the **Item Properties** panel.

Dynamic Item Properties

A dynamic item has an additional property tab called **Dynamic** in the **Item Properties** panel. The **Dynamic** tab contains the settings for the dynamics simulation.

You can modify the settings to change the behavior of the dynamic item during movement and collision. When a dynamic item is selected, the following attributes appear in the **Properties** panel.

General Properties

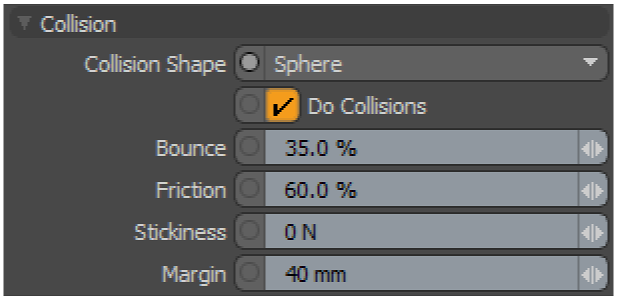

Items
Shading
Groups
Images
+
Filter Items
Add Item
S
F
Name
WreckItBradley.lxo
Wrecking ball
Brick
solver
Properties
Channels
Lists
+
Scene
Mesh
Su

☑ Enable
Type: Rigid
○ Static
State: Dynamic
Preset: ...

General Properties	
Enable	Toggles dynamics on and off for the item. When enabled, Modo applies dynamics to the item when a simulation is run or cached. When disabled, the dynamic properties are ignored.
Type	Dynamic items can be defined as one of three types: • **Rigid**: The item is treated as a solid object with a fixed, unchanging volume. • **Softbody**: A soft body deforms during a collision or when a force is applied. • **Curves**: Curves are used for items that consist of one or more curved strands, such as hair or wires. Selecting each option hides/reveals different attributes in the **Properties** panel.
Static	The associated item does not move during a simulation (even if animated), acting strictly as a collision object. When **Static** is disabled, two options become available in **State**.
State	**State** is enabled when an item is not static. • **Dynamic**: These are collision items controlled exclusively by dynamics. They can also be affected by constraints, forces, and falloffs.

	This is automatically assigned when an object is defined as an **Active Rigid Body**. **Note:** This should be set for animating fur guides. - **Kinematic**: An animated collision object that is controlled by keyframe information. Kinematic objects are unaffected by forces, such as gravity or falloffs. This is automatically assigned when an object is defined as a **Kinematic Rigid Body**.
Preset	Modo offers a series of presets to determine the dynamic properties of an item. If you select a preset, the values are automatically adjusted, and the item takes on the physical properties of the chosen material. You can choose from **Glass**, **Stone**, **Wood**, **Plastic**, or **Steel**.

Collision: Rigid



Collision

Collision Shape: Sphere

☑ Do Collisions

Bounce: 35.0 %

Friction: 60.0 %

Stickiness: 0 N

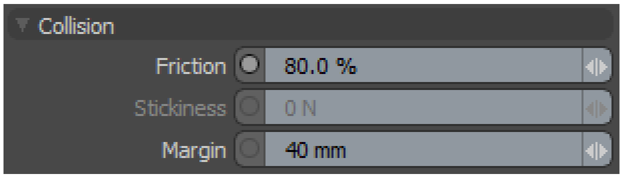
Margin: 40 mm

Collision: Rigid	
Collision Shape	This is the shape to calculate collision contacts between dynamic objects. In general, you should choose the simplest shape that produces satisfactory results. The default collision shape is set to **Hull**, which generally balances accuracy with calculation speed. - **Box**: A rectangular box defined by the bounding box of the geometry's maximum extents.

	• **Sphere**: A sphere that is sized to encompass all the geometry of the object. • **Hull**: This collision shape creates the smallest convex volume that encompasses all the points in this mesh (imagine the mesh enveloped in shrink-wrap). The **Hull** collision type does not support holes or indentations in the object. Although it only approximates the item's shape, it provides very fast collision detection on complex shaped objects. • **Multi Hull**: The default collision option, **Multi Hull** generates a separate convex hull for each polygon island in the mesh. These are then treated as a single, compound dynamic item. • **Mesh**: Uses the mesh of the object for calculating collisions. For subdivision surface models, the actual frozen Sub-D cage is used. **Note:** Avoid this option on **Dynamic Rigid Bodies** (it is fine for **Static** and **Kinematic** ones). Otherwise, performance and simulation stability can suffer negatively. • **Convex Decomposition**: Creates multiple hulls (see above) and connects them together. This works to better approximate the actual shape of the geometry, but can be costly to initially compute with complex shapes. • **Plane**: Generates an infinite ground plane collision shape, originating from the center of the associated geometry's bounding box (not on the item's actual center position).
Do Collisions	Disable this option to remove the object from collision calculations.
Bounce	This is the collision response when the object hits another dynamic object. A value of 0% means the object doesn't bounce or inherit energy from the other dynamic object. A value of 100% means the object inherits all of the energy from the contact object. Values above 100% produce rubber ball-like effects where objects impart more energy from contacts.
Friction	The amount an object slides against another object.

	A value of 0 offers no resistance. A value of 1 stops the object from sliding.
Stickiness	When two dynamic elements collide and they both have a non-zero **Stickiness**, they attach to each other at the collision point. The higher the **Stickiness** value, the stronger the bond between the elements. Lower values allow stuck elements to be broken more easily.
Margin	This value helps the bullet engine to better determine collisions and improves its performance and reliability. It defines a gap around the object used to detect a collision contact. In most cases it should be kept at its default value. When used with the **Mesh** and **Convex Decomposition** settings, this can offset items away from each other, so only a small amount is necessary. **Note:** A setting of 0 may produce errors in collision detection.

Collision: Softbody



Collision

Friction 80.0 %

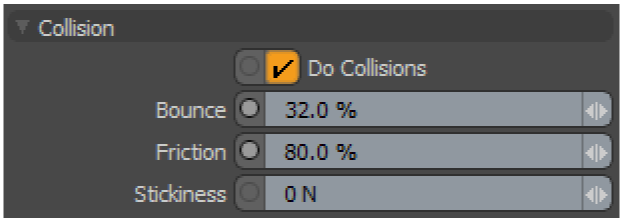
Stickiness 0 N

Margin 40 mm

Collision: Softbody	
Friction	The amount that a soft body slides against another dynamic object. A value of 0 offers no resistance. A value of 1 stops the object from sliding.
Stickiness	When two dynamic elements collide and they both have a non-zero **Stickiness**, they attach to each other at the collision point. The higher the **Stickiness** value, the stronger the bond between the elements.
Margin	This value helps the bullet engine to better determine collisions and improves its performance and reliability. It defines a gap around the object used to detect a collision contact.

	Note: A setting of 0 may produce errors in collision detection.

Collision: Curves



Collision

☑ Do Collisions

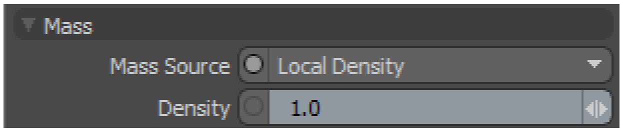
Bounce 32.0 %

Friction 80.0 %

Stickiness 0 N

Collision Curves	
Do Collisions	Disable this option to remove the object from collision calculations (enabled by default).
Bounce	This is the collision response when the curve capsule colliders hit another dynamic object. A value of 0% means the object doesn't bounce or inherit energy from the other dynamic object. A value of 100% means the object inherits all of the energy from the contact object. Values above 100% produce rubber ball-like effects where objects impart more energy from contacts.
Friction	The amount an object slides against another object. A value of 0 offers no resistance. A value of 1 stops the object from sliding.
Stickiness	When two dynamic elements collide and they both have a non-zero **Stickiness**, they attach to each other at the collision point. The higher the **Stickiness** value, the stronger the bond between the elements.

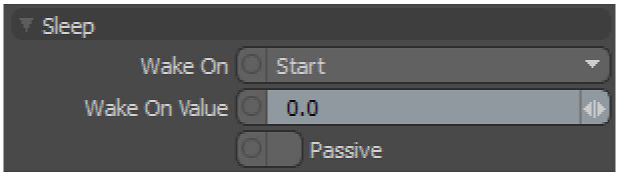
Mass



Mass	
Mass Source	• **World Density**: Calculate the mass by multiplying the calculated volume of the dynamic item by the **Global Density** value defined on the Dynamics Solver item. • **Local Density**: Mass is calculated by multiplying the computed volume by this value. • **Local Mass**: Set the mass manually.
Density	The measure of the amount of matter in any given volume. If the **Mass Source** option is set to **Local Density**, this value is used to calculate the mass.
Mass	The property of an object that gives it bulk and weight. If **Mass Source** is set to **Local Mass**, this value is used for simulations.

Sleep

Note: Sleep is only available for **Rigid Body** items.



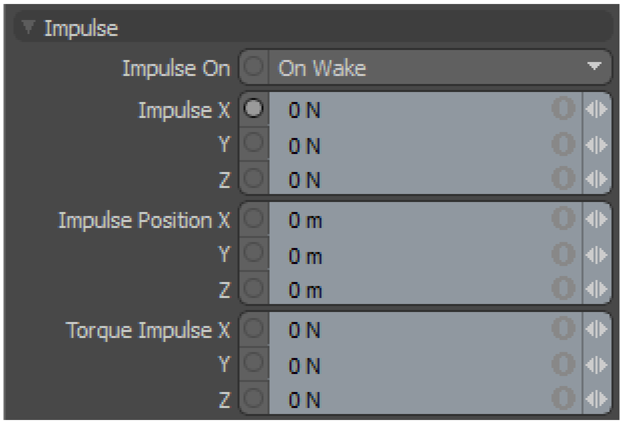
Normally, objects are awake (active) when any dynamic simulation starts and the object begins to react immediately. You can change this behavior, delaying when individual objects wake up and become active, essentially controlling when items begin their participation in a simulation.

Sleep

Wake On	Before the **Wake On** event, items can be keyframe-animated and the object is handed off to dynamics for simulation when the **Wake On Value** threshold is reached. The values defined in the **Impulse** channels are applied when the object wakes up. • **Start**: The object begins the dynamic simulation awake and fully active at the beginning of the scene. • **Collision Velocity**: The object wakes up when it collides with a passive or active object. • **Collision Force**: The object wakes up when it collides with a passive or dynamic object. • **Velocity**: The object wakes up when a certain velocity is reached. Different from **Collision Velocity**, this velocity is determined by keyframe animated motion. • **Trigger**: The object wakes up when the keyframe-able channel, **reWakeOnValue**, is not zero. This has good use with conditional nodes in rigging. • **Frame**: The object wakes up when a specific frame number is reached. • **Time**: The object wakes up when a specific time has been reached, regardless of frame rate. • **Force**: The object wakes up when the force applied by a force item reaches the threshold. **Note:** Use the **Wake On Value** to set a threshold for the **Collision Velocity**, **Collision Force**, **Velocity**, or **Force**.
Wake On Value	This is the threshold that must be reached in order to wake up the object, depending on the **Wake On** option selected.
Passive	When enabled, sleeping objects react to collision forces only. This means they can be pushed around but stop moving the moment the influence stops.

Impulse

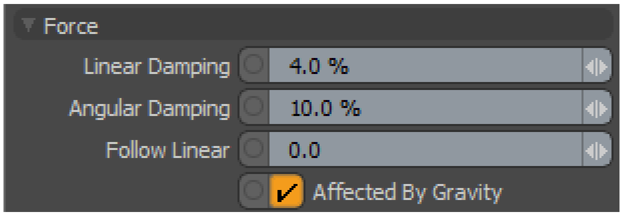
Note: **Impulse** is only available for **Rigid Body** items.



Impulse On: On Wake
Impulse X: 0 N
Y: 0 N
Z: 0 N
Impulse Position X: 0 m
Y: 0 m
Z: 0 m
Torque Impulse X: 0 N
Y: 0 N
Z: 0 N

Impulse	
Impulse On	This option determines the state that causes the **Impulse** value to be applied to the dynamic item. • **Off**: No impulse value is applied to the object. • **On Wake**: The impulse values are applied when the item wakes up, as determined by the **Sleep** settings. • **Continuous**: The impulse value is applied to the object continually at every step in the evaluation.
Impulse X/Y/Z	A directional force that is applied to the object when it wakes up. This is useful when a keyframed object is handed off to dynamics, imparting direction and velocity to the object. Without **Impulse** the velocity would drop, as dynamics has no awareness of the object's velocity prior to waking up.
Impulse Position X/Y/Z	The position the **Impulse** is applied from, affecting the object's trajectory.
Torque Impulse X/Y/Z	A rotational force that is applied to the object when it wakes up. This is useful when a keyframed object is handed off to dynamics, imparting rotation to the object. Without **Torque Impulse** the object would not spin, as dynamics has no awareness of the object's motion prior to waking up.

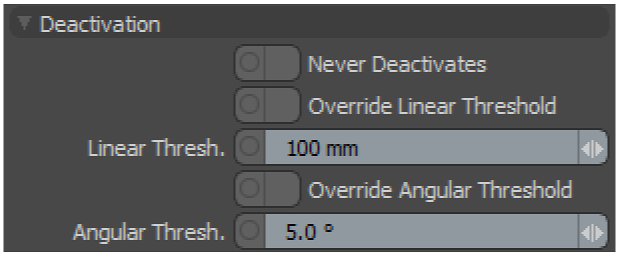
Force


Force
Linear Damping 4.0 %
Angular Damping 10.0 %
Follow Linear 0.0
☑ Affected By Gravity

Force	
Linear Damping	This is a force that acts upon the motion of objects to slow them down over time. A small value here can benefit object stability, since it helps objects come to rest.
Angular Damping	Same as **Linear Damping**, but applies to the rotation of the object. **Note: Angular Damping** is not available for **Soft Body** objects.
Follow Linear	When under the influence of the dynamics system, animated dynamic items ignore their user-assigned kinematic (keyframed) motion. Instead, they act under the forces applied to them, and any collisions encountered. **Follow Linear** applies an amount of force to an item to seek its kinematic position. The setting allows you to influence a dynamic solution by adding keyframes to the item and increasing the **Follow Linear** force value.
Affected by Gravity	Turns gravity on or off for the object.

Deactivation

Note: Deactivation is not available for **Soft Body** items.



Deactivation

○ Never Deactivates

○ Override Linear Threshold

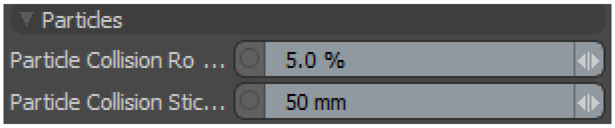
Linear Thresh. 100 mm

○ Override Angular Threshold

Angular Thresh. 5.0 °

Deactivation	
Never Deactivates	When enabled, the item never gets deactivated due to no motion. When disabled, the item is disabled when it meets the defined linear and angular thresholds defined in the solver item (or overridden here).
Override Linear Threshold	Enable this function to put dynamic objects to sleep once they reach a defined upper threshold value, overriding the global value found in the Dynamics Solver. This behaves opposite to the **Wake On** settings; once an object has been below the threshold for the defined length of time, it is put to sleep.
Linear Threshold	The motion threshold value when **Override Linear Threshold** is enabled.
Override Angular Threshold	Enable this function to put dynamic objects to sleep once they reach a defined upper threshold value, overriding the global value found in the Dynamics Solver. This behaves opposite to the **Wake On** settings; once an object has been below the threshold for the defined length of time, it is put to sleep.
Angular Threshold	The rotation threshold value when **Override Angular Threshold** is enabled.

Particles



Particles

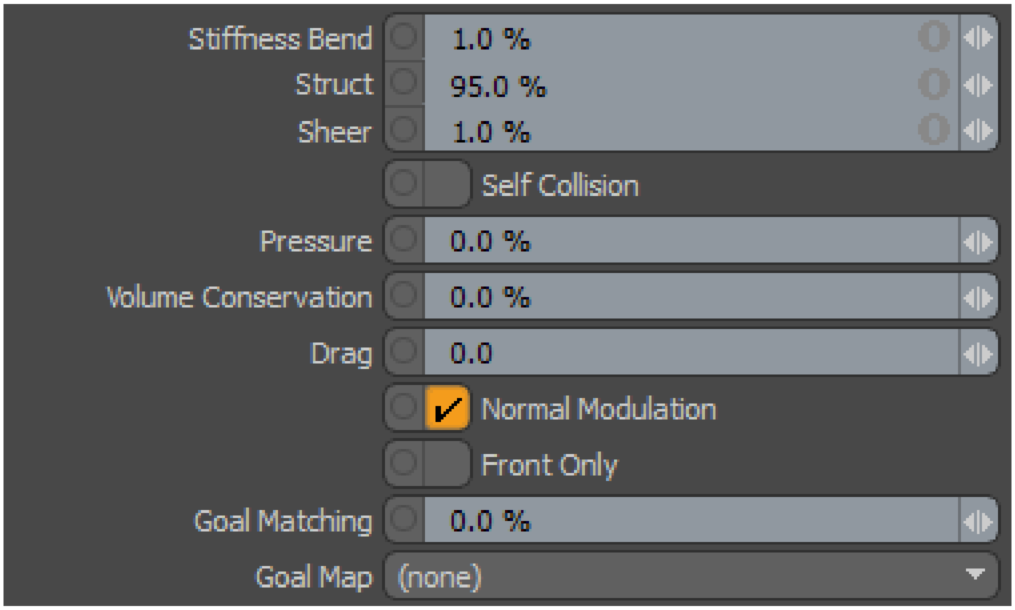
Particle Collision Ro … 5.0 %

Particle Collision Stic… 50 mm

Particles	
Particle Collision Roughness	This adds a random factor to the direction in which particles bounce off the item. At 100% it can bounce up to 90 degrees away from the collision normal.

Particle Collision Stickiness	The distance at which particles try to stick to the surface of an item.

Soft Body



Stiffness Bend 1.0 %
Struct 95.0 %
Sheer 1.0 %
☐ Self Collision
Pressure 0.0 %
Volume Conservation 0.0 %
Drag 0.0
☑ Normal Modulation
☐ Front Only
Goal Matching 0.0 %
Goal Map (none)

Soft Body	
Stiffness Bend	The shape of a Soft Body is determined by virtual springs, evaluated between the vertices in the mesh. These springs help an object maintain its shape by applying a rotational force to maintain the continuity of the surface. Stiffness can be reduced all the way to 0, creating a very soft and saggy fabric type.
Struct	These springs follow along the lines of the mesh (along the polygon edges). This value determines how resistant the springs are to the applied forces when deforming, due to forces such as gravity or turbulence.
Sheer	These springs are generated to add extra connections to the inside of a polygon face (quad polygons forming an x shape). They help to prevent the polygon from collapsing in on itself.
Self Collision	Enable this option to fix issues with the surface intersecting itself.
Pressure	This exerts a force upon the overall volume of an item, similar to how pressure

	reacts on an object in the real world. Increasing values add overall volume, inflating the item. Decreasing (negative) values reduce its volume, deflating the item. **Note:** This option requires that the target mesh be completely watertight (no seams).
Volume Conservation	This option works with the **Pressure** option. You can use it to preserve the volume of the Soft Body. For example, if you crush one part of a balloon, the rest of it expands so that it still contains the same volume of air.
Drag	Determines the loss of energy on a surface that is in motion (say, for jiggling-type of effects).
Normal Modulation	This option multiplies the force applied by the surface normal.
Front Only	When enabled, forces coming from the back-side of a single-sided object are not applied.
Goal Matching	If a goal map is specified, this option acts as a multiplier on the map values.
Goal Map	A Vertex Map that applies to soft body items and controls how much they retain their original shape. It can also be looked at as a stiffness control. In this case, a Weight Map is used. Weight Map values of 1 retain the original position, while values lower than 1 apply an appropriate spring force attempting to retain the initial shape.

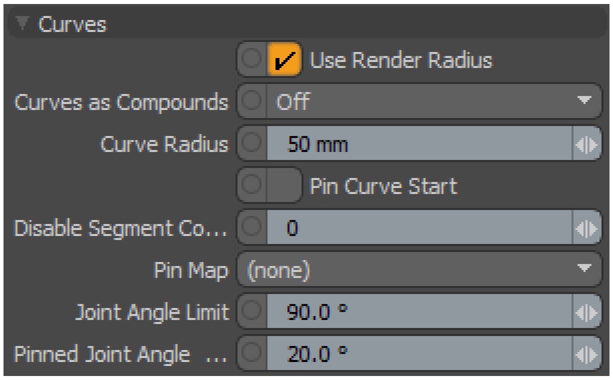
Note: If settings in the Solver and Soft Body properties cause the simulation to become unstable, the Soft Bodies responsible are removed from the simulation. When this happens, the event is reported to the Event log (**System** > **Advanced** > **Event Log**). For more information in the Solver Item, see Dynamics Solver.

Curves

You can apply simulations to various curve types in Modo, using either **Splines** or **Polylines** (fur guides).

The simulations of curves are very much like rigid body calculations, simulating a connected string of rigid capsule shapes drawn between each vertex. Each shape's size is based on the curve radius value in the Mesh

Item properties. The resulting capsule collision shapes have three possible states for dynamics: **Static**, **Dynamic**, or **Kinematic**.


Curves
☑ Use Render Radius
Curves as Compounds: Off
Curve Radius: 50 mm
☐ Pin Curve Start
Disable Segment Co...: 0
Pin Map: (none)
Joint Angle Limit: 90.0 °
Pinned Joint Angle ...: 20.0 °

Curves	
Use Render Radius	When enabled, the resulting capsules generated with dynamics, match the size defined, even when they are modulated with a curve width gradient.
Curves as Compounds	Controls the rigidity of the curves in a simulation, and how multiple curves within a single item layer are treated. • **Off**: All curves are treated individually as soft, deforming elements. • **Per Curve**: All curves are treated individually as rigid, non-deforming elements. • **Item**: All curves are treated as a combined (single) rigid, non-deforming element.
Curve Radius	When the **Use Render Radius** option is disabled, this option determines the collision size for the resulting capsule collision elements.
Pin Curve Start	When enabled, it automatically pins the first vertex of a curve. Pinned vertices remain fixed to their position within the mesh layer, moving only when the layer is animated or its geometry is deformed. This option is mainly used for simulating hair guides, to keep them connected to their character.
Disable Segment Collision Start	The collision capsules that are generated for a simulation are generated between each vertex, creating a chain of collision objects. The first segment begins at the base position of the curve. For pinned curves the first segment is always disabled for collision. The **Disable Segment Collision Start** value allows you to disable collision on additional segments. For example, if there is a curve with 8 segments pinned to a character, and this value is set to 3, then only the last 5 segments of the curve

	would be considered for collision.
Pin Map	Allows you to fix vertices along the curve, meaning they won't move or be affected by the dynamics. This is done by the addition of a Pick Map. Once added, the **Pick Map** appears in the **Pin Map** dropdown list. If you select a **Pick Map**, the targeted vertices aren't affected by curve dynamics.
Joint Angle Limit	Limits the amount of bending allowed between curve segments, this is very useful for fur guide simulations so that the fur doesn't go crazy during fast motions.
Pinned Joint Angle Limit	Same effect as the **Joint Angle Limit**, but works as a separate value for controlling the root rotation of the fur guide.

To Create a Pick Map of Curve Vertices

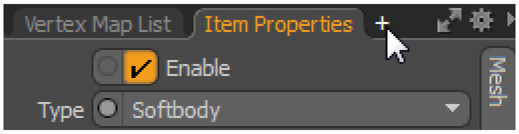
Pick Maps are a type of Vertex Map that designate vertices as selected or not selected. You can create a Pick Map to identify vertices in a curve that should remain fixed during dynamic simulation. Pick Maps are added using the Vertex Map List viewport or the **Vertex Map** menu.

Note: Once created, the Pick Map appears in the **Pin Map** option, in the **Curves** section of the dynamic item properties.

1. Select **Vertex Map** > **Create...**.

   This opens the **Vertex Map** dialog.
2. Set the **Vertex Map Type** to **Pick**.
3. Type a name for the map in **Vertex Map Name**. Click **OK**.
4. To make sure the new map is selected, open the **Vertex Map List** panel.

   Click **+** next to the panel tabs.

   ☑ Enable

   Type ○ Softbody

   Select **Data Lists** > **Vertex Map List**.
5. Ensure your Pick List is selected in the **Vertex Map List** panel, under **Other Maps**.
6. Activate **Vertices** selection mode and select the target vertices in the curve that should remain fixed.
7. Add the vertices to the pick map:

   Select **Vertex Map** > **Set Value...**

This opens the **Set Vertex Map Value** dialog.

Set **Vertex Map Type** to **Pick**.

Set **Value** to 1.0. Click **OK**.

Note: The **Value** determines if the vertex is affected by dynamics. A value of 0 means the vertex is not selected, and is not affected by dynamics. A value of 1 means it is selected, and is affected by dynamics.

Additional vertices can be added or removed from the pick map at a later time. With the Pick Map highlighted, select the vertex and open **Vertex Map** > **Set Value...**. Set the value to 0 to remove the vertex, or 1 to add the vertex.

Constraints

Constraints are basically joints, affecting the movement of dynamic objects in a scene, restricting their motion in very specific ways dependent on the type of constraint applied. Dynamics offers a variety of constraint types, providing a means to produce very controlled dynamic simulations. Additionally, constraints can be combined together (by way of Locators) producing even more control to complex dynamic simulations.

Using Constraints

Constraints in dynamics are simply joints that connect two dynamic bodies together. Each constraint provides a different type of joint, limiting the motions of the dynamic objects in very specific ways. Typically, constraints are used to relate the dynamic motion of an active rigid body item to that of a static or animated passive rigid body item. For instance, if you wanted to animate a pair of dice swaying from the rearview mirror of a car, the car itself (including the mirror) would be defined as a passive rigid body, and the pair of dice within would be defined as an active rigid body. Next, you would assign a point constraint to the mirror and the pair of dice and position it near the mirror. If the car were animated to move as if it was traveling over rough terrain, running a subsequent dynamics simulation would then produce an incredibly realistic secondary swinging motion to the pair of dice without the need to add a single keyframe.

Constraints are also a powerful way to provide control of active dynamic objects, relating the motion of one to that of another. This is handled in the same way as the above example, but both items would be designated as active rigid bodies. Both objects would then move relative to the motions of the other, controlled by the constraint joint. Furthermore, series of constraints can be added to consecutive objects, providing linked motions for the entire length of items. This is applied by selecting the chain of items in order and then applying the constraint.

Applying Constraints

Constraints can be added to scenes like any other item: in the **Items** lists, click **Add Item**. First, select the target items you wish to join together, and select a constraint using the various constraint buttons found in the toolbox of the **Setup** interface.

When created, the constraint item automatically links to the selected items. To link two items together, follow these steps:

1. First select the master or parent item. This is typically the keyframed or static passive dynamic object.
2. Next, select the child or auxiliary item, which is always an active dynamic object.
3. Choose from one of the constraint options available. Click the button to add the item to the scene.

When a constraint is added this way, the constraint item automatically populates the **Body A** and **Body B** options within the constraint. It is also positioned at exactly the middle point between the centers of the two items. Depending on your intentions, this may not be the optimal position for the constraint. The motion of the child item (**Body B**) originates from the position of the constraint item itself, so it needs to be positioned appropriately.

Removing Constraints

Constraints can be removed from the scene like any other item. Simply select the target constraint and press **Del**, or in the menu bar, click **Edit** > **Delete**.

Breaking Joints

You can break a constraint once a certain force threshold is achieved. This essentially disables the constraint, allowing connected objects to fall away from each other. This is useful if, for instance, two objects were tethered together by a chain and when the chain is stressed to a specific point, the links would fall apart, each link then acting as an individual dynamic body, and the tethered item would no longer be under the influence of the constraint. Clever rigging types can also link a morph of the chain to the **isBroken** channel of the constraint, so that once it breaks, the link itself would appear broken. To activate this function, toggle the **Stress Break** option of the Constraint, and set a lower threshold. Once that value or greater is achieved through a combination of the forces applied to the item, including gravity, forces and any force imparted by the item's own movements, the link breaks apart.

Pin Constraint

The pin constraint type is a rigid constraint meant to rigidly attach two or more dynamic items together. This can be useful for attaching together two object with dramatically different mass values, such as the metal head of a hammer attached to its wooden handle. The heavier head affects the whole hammer. It is also useful to rigidly combine shattered objects that are meant to break apart when combined with the **Stress Break** function.

To apply a pin constraint automatically, follow these steps:

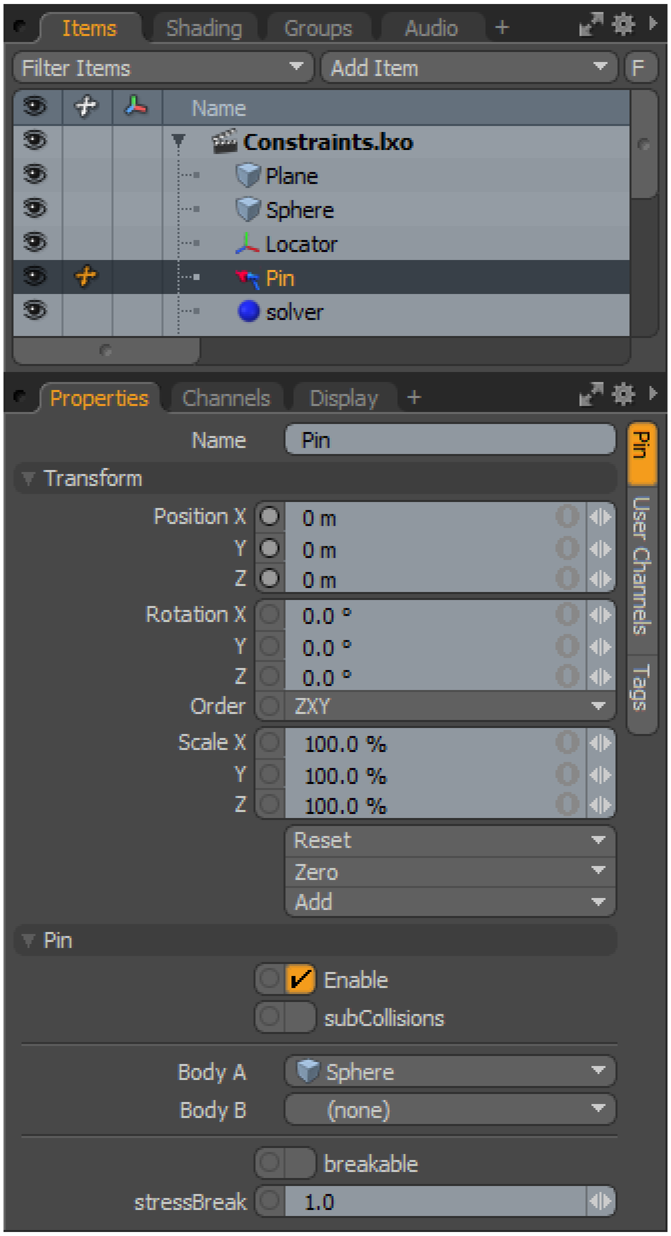
1. Select two items prior to invoking any of the constraint types.

   The first item selected represents the master or parent item and is defined as **Body A** in the constraint, the second item selected is the item that is constrained, also considered the auxiliary or child item, defined as **Body B**.

2. With the items selected, in the **Dynamics** sub-tab, click on the **Pin** constraint button.

   This adds the constraint item to the **Items** list.

The default center location of the pin constraint is exactly halfway between the **Body A** and **Body B** objects with a line drawing to the centers of the constrained items.


Items
Shading
Groups
Audio
Filter Items
Add Item
F
Name
Constraints.lxo
Plane
Sphere
Locator
Pin
solver
Properties
Channels
Display
Pin
User Channels
Tags
Transform
Reset
Zero
Add
Pin

Name Pin

Position X 0 m
Y 0 m
Z 0 m

Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Order ZXY

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

☑ Enable
☐ subCollisions

Body A Sphere
Body B (none)

☐ breakable
stressBreak 1.0

Option	Description
Name	Displays the current constraint item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	These values represent the initial center for the constraint item in 3D world space, and the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item rotates and

Option	Description
	moves based on its initial (resting) position.
Rotation X/Y/Z	These values represent the initial (resting) rotation of the constraining item.
Scale X/Y/Z	These values have no direct effect on constraints.
Pin	
Enable	Toggles the constraint item on or off. When enabled, the constraint item is considered during a dynamics simulation; when disabled, the constraint item is ignored. However, disabled constraints are persistent across Modo sessions, being saved with the scene, and retain their present settings.
SubCollisions	When enabled, allows constrained objects (especially those within a chain, not a literal chain, but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculations are required, however, constrained items may inter-penetrate each other. If this occurs, enable **SubCollisions**.
Body A	Represents the item that another item is constrained to, the parent or master item.
Body B	Represents the item that is constrained, the child or auxiliary item.
Breakable	Allows constraints to be broken apart when a certain stress threshold is reached, defined by the **Break Stress** value. When the threshold is reached, the constraint no longer has any effect on the constrained item, leaving it to be affected by any other forces present in the simulation.
Stress Break	Determines the threshold when the constraint no longer affects the constrained item. The **Breakable** option must be enabled for this value to have any effect on the constraint.

Point Constraint

The point constraint attaches an object to a single point, providing an unrestricted dangling or swaying motion when applied. To visualize this constraint, imagine the constrained item were tied to a piece of string affixed to a specific point in 3D space. As the parent item moves, the constrained item swings relative to the motions of the parent item.

To apply a point constraint automatically, follow these steps:

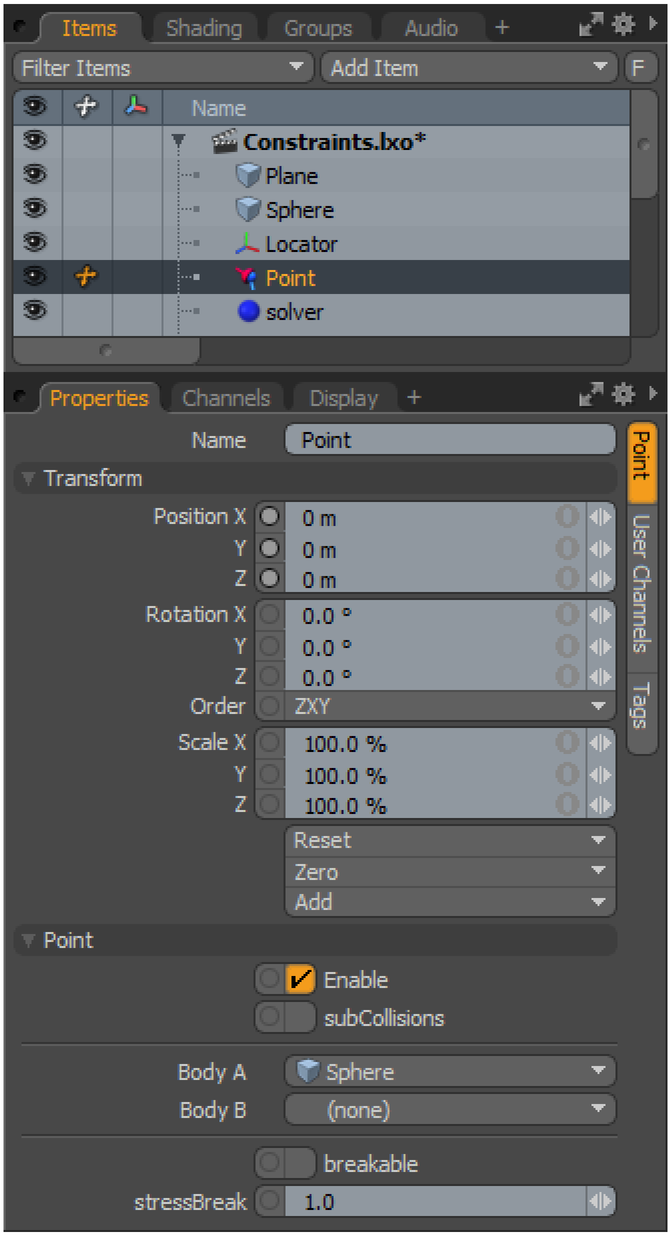
1. Select two items prior to invoking any of the constraint types.

   The first item selected represents the master or parent item and is defined as **Body A** in the constraint, the second item selected is the item that is constrained, also considered the auxiliary or child item, defined as **Body B**.

2. With the items selected, in the **Dynamics** sub-tab, click on the **Point** constraint button.

   This adds the constraint item to the **Items** list.

The default center location of the point constraint is exactly halfway between the **Body A** and **Body B** objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the constraint. The motion of the child item (**Body B**) originates from the position of the constraint item itself, so it may need to be positioned appropriately.


Items
Shading
Groups
Audio
Filter Items
Add Item
F
Name
Constraints.lxo*
Plane
Sphere
Locator
Point
solver
Properties
Channels
Display
Point
User Channels
Tags
Transform
Reset
Zero
Add

Name Point

Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Order ZXY
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

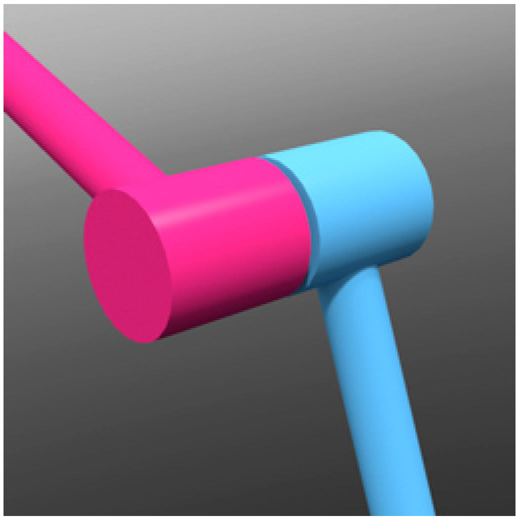
Point

☑ Enable
☐ subCollisions
Body A Sphere
Body B (none)
☐ breakable
stressBreak 1.0

Option	Description
Name	Displays the current constraint item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	These values represent the initial center for the constraint item in 3D world space, and the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item rotates and

Option	Description
	moves based on its initial (resting) position.
Rotation X/Y/Z	These values represent the initial (resting) rotation of the constraining item.
Scale X/Y/Z	These values have no direct effect on constraints.
Point	
Enable	Toggles the constraint item on or off. When enabled, the constraint item is considered during a dynamics simulation; when disabled, the constraint item is ignored. However, disabled constraints are persistent across Modo sessions, being saved with the scene, and retain their present settings.
SubCollisions	When enabled, allows constrained objects (especially those within a chain, not a literal chain, but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculations are required, however, constrained items may inter-penetrate each other. If this occurs, enable **SubCollisions**.
Body A	Represents the item that another item is constrained to, the parent or master item.
Body B	Represents the item that is constrained, the child or auxiliary item.
Breakable	Allows constraints to be broken apart when a certain stress threshold is reached, defined by the **Break Stress** value. When the threshold is reached, the constraint no longer has any effect on the constrained item, leaving it to be affected by any other forces present in the simulation.
Stress Break	Determines the threshold when the constraint no longer affects the constrained item. The **Breakable** option must be enabled for this value to have any effect on the constraint.

Hinge Constraint



The hinge constraint limits the rotation movement of an object to swivel as if fixed by a hinge joint, like that of a door. However, as a door hinge is limited by the collision of the opposing hinge plate, the hinge constraint offers unlimited rotation, much like a wheel axle. You can add a rotation range limit to the joint to produce the more limited motion of the door hinge. The hinge constraint is represented in the viewport as a circle around the point of rotation. Additionally, you can apply a motor force to the joint, making it possible to create wheels that spin and move items forward.

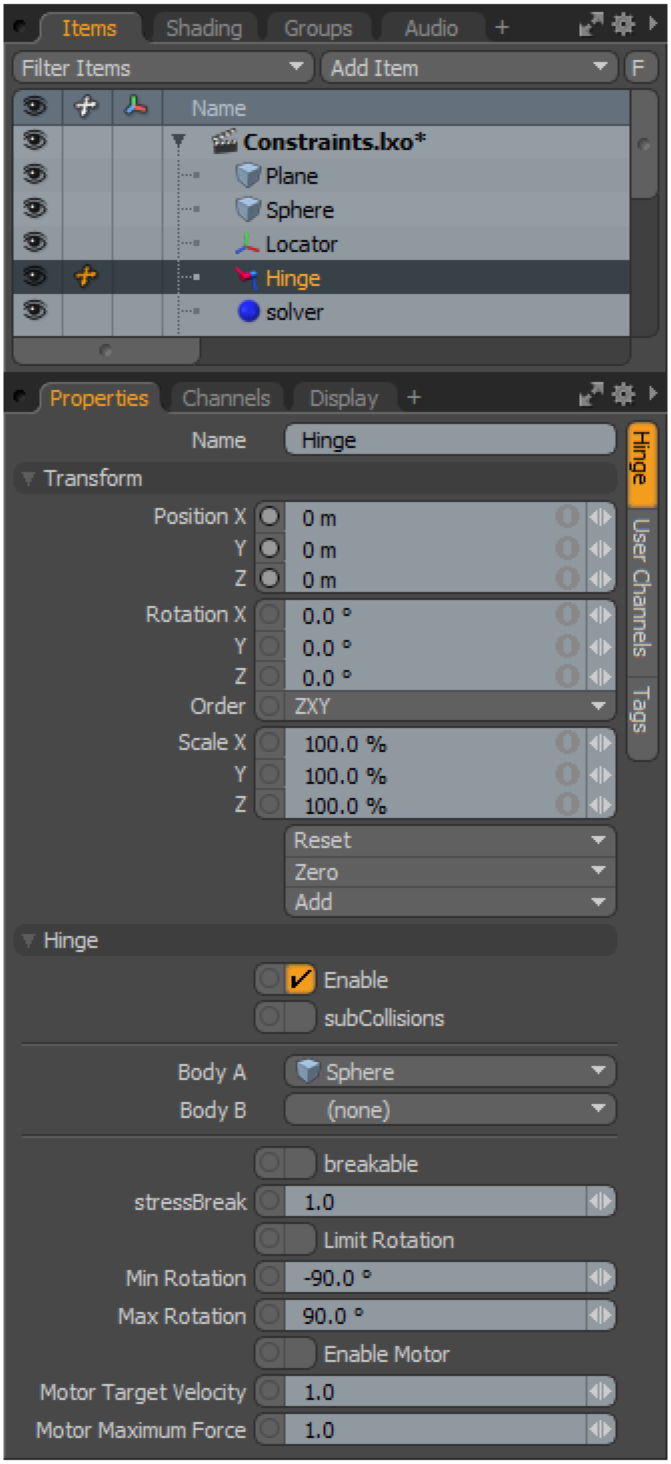
To apply a hinge constraint automatically, follow these steps:

1. Select two items prior to invoking any of the constraint types.

   The first item selected represents the master or parent item and is defined as **Body A** in the constraint, the second item selected is the item that is constrained, also considered the auxiliary or child item, defined as **Body B**.
2. With the items selected, in the **Dynamics** sub-tab, click on the **Hinge** constraint button.

   This adds the constraint item to the **Items** list.

The default center location of the hinge constraint is exactly halfway between the **Body A** and **Body B** objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the constraint. The motion of the child item (**Body B**) originates from the position of the constraint item itself, so it may need to be positioned appropriately.


Items
Shading
Groups
Audio
+
Filter Items
Add Item
F
Name
Constraints.lxo*
Plane
Sphere
Locator
Hinge
solver
Properties
Channels
Display
+
Hinge
User Channels
Tags
Transform
Reset
Zero
Add
Hinge

Name Hinge

Position X 0 m
Y 0 m
Z 0 m

Rotation X 0.0 °
Y 0.0 °
Z 0.0 °

Order ZXY

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

☑ Enable
☐ subCollisions

Body A Sphere
Body B (none)

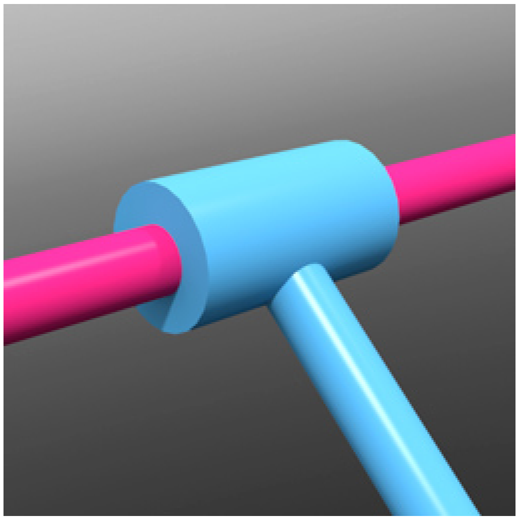
☐ breakable
stressBreak 1.0
☐ Limit Rotation
Min Rotation -90.0 °
Max Rotation 90.0 °
☐ Enable Motor
Motor Target Velocity 1.0
Motor Maximum Force 1.0

Option	Description
Name	Displays the current constraint item name. To change it, click on the field and type in the new name.

Option	Description
Transform	
Position X/Y/Z	These values represent the initial center for the constraint item in 3D world space, and the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item rotates and moves based on its initial (resting) position.
Rotation X/Y/Z	These values represent the initial (resting) rotation of the constraining item.
Scale X/Y/Z	These values have no direct effect on constraints.
Hinge	
Enable	Toggles the constraint item on or off. When enabled, the constraint item is considered during a dynamics simulation; when disabled, the constraint item is ignored. However, disabled constraints are persistent across Modo sessions, being saved with the scene, and retain their present settings.
SubCollisions	When enabled, allows constrained objects (especially those within a chain, not a literal chain, but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculations are required, however, constrained items may inter-penetrate each other. If this occurs, enable **SubCollisions**.
Body A	Represents the item that another item is constrained to, the parent or master item.
Body B	Represents the item that is constrained, the child or auxiliary item.
Breakable	Allows constraints to be broken apart when a certain stress threshold is reached, defined by the **Break Stress** value. When the threshold is reached, the constraint no longer has any effect on the constrained item, leaving it to be affected by any other forces present in the simulation.
Stress Break	Determines the threshold when the constraint no longer affects the constrained item. The **Breakable** option must be enabled for this value to have any effect on the constraint.
Limit Rotation	The hinge is not initially limited in its rotation. By enabling this option, you can constrain the range of rotation that can be applied to the constraint, similar to the motion of a door hinge that is allowed to only swing so far.
Min/Max Rotation	The **Minimum** value sets the lower extreme of the of the rotation clamping, while

Option	Description
Limit	the **Maximum** value limits the upper extreme value.
Enable Motor	Normally, a hinge is only affected by the available forces in a dynamic simulation. This toggle allows you to apply a rotational force to the **Body B** item, making it spin around the hinge constraint's location. Speed of spinning is controlled by the mass of the connected objects, combined with the **Motor Target Velocity** and the **Motor Maximum Force**.
Motor Target Velocity	When this value is other than zero, it applies a rotational motion to the **Body B** item. Like a wheel, positive values rotate the item forward and negative values rotate it in reverse. The direction of the motion is based on the initial position of the constraint. The motor attempts to achieve the defined target velocity, taking into account forces, mass, and any collisions it encounters.
Motor Maximum Force	Determines the strength of how the motor reaches its target velocity and how it reacts to collision items.

Slide Hinge Constraint



The slide hinge constraint combines the dynamic actions of the hinge with a slide constraint, providing the rotation movement of the hinge with the addition of a linear back-and-forth sliding motion.

To apply a slide hinge constraint automatically, follow these steps:

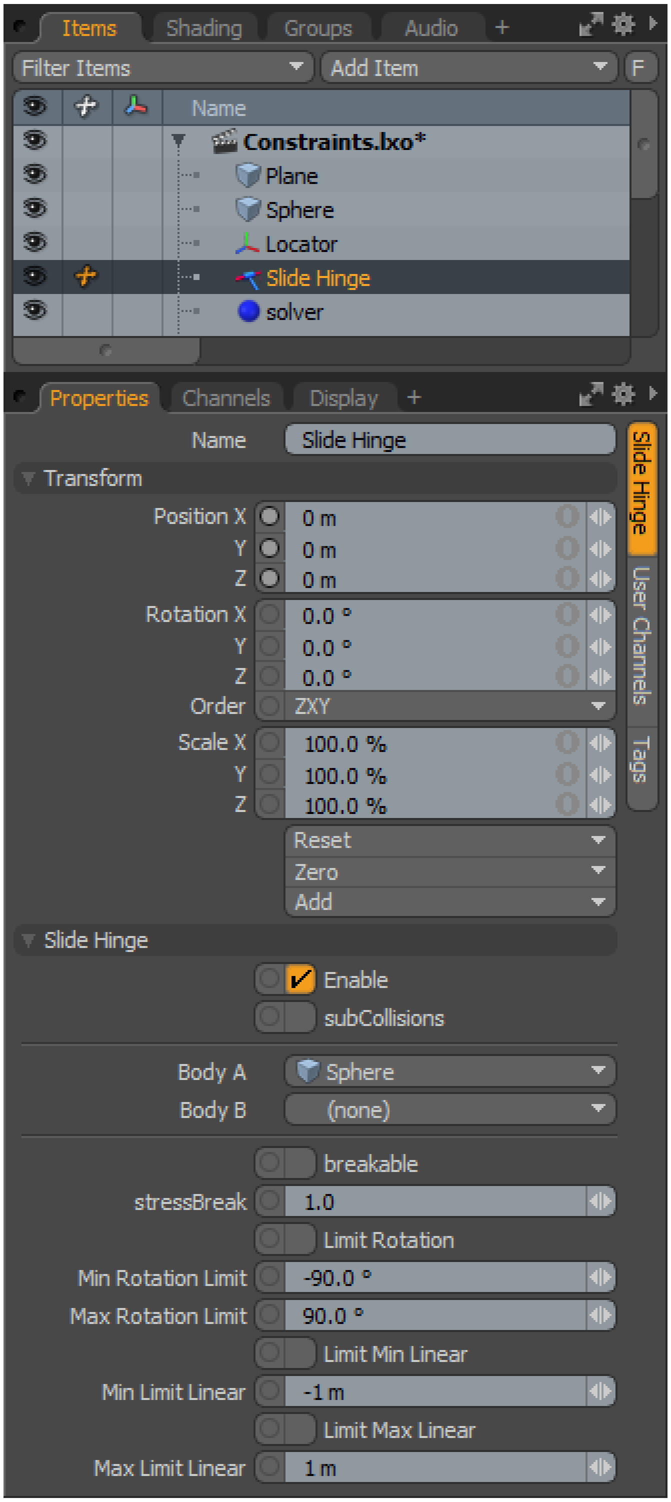
1. Select two items prior to invoking any of the constraint types.

The first item selected represents the master or parent item and is defined as **Body A** in the constraint, the second item selected is the item that is constrained, also considered the auxiliary or child item, defined as **Body B**.

2. With the items selected, in the **Dynamics** sub-tab, click on the **Slide Hinge** constraint button.

   This adds the constraint item to the **Items** list.

The default center location of the slide hinge constraint is exactly halfway between the **Body A** and **Body B** objects with a line drawing to the centers of the constrained items. Depending on your intentions, this might not be the optimal position for the constraint. The motion of the child item (**Body B**) originates from the position of the constraint item itself, so it may need to be positioned appropriately.


Items
Shading
Groups
Audio
+
Filter Items
Add Item
F
Name
Constraints.lxo*
Plane
Sphere
Locator
Slide Hinge
solver
Properties
Channels
Display
+
Slide Hinge
User Channels
Tags
Transform
Reset
Zero
Add
Slide Hinge

Name: Slide Hinge

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable
☐ subCollisions

Body A: Sphere
Body B: (none)

☐ breakable
stressBreak: 1.0
☐ Limit Rotation
Min Rotation Limit: -90.0 °
Max Rotation Limit: 90.0 °
☐ Limit Min Linear
Min Limit Linear: -1 m
☐ Limit Max Linear
Max Limit Linear: 1 m

Option	Description
Name	Displays the current constraint item name. To change it, click on the field and type

Option	Description
	in the new name.
Transform	
Position X/Y/Z	These values represent the initial center for the constraint item in 3D world space, and the center of mass relative to the item it is constraining. Based on its initial position, once a dynamics simulation is invoked, the constraining item rotates and moves based on its initial (resting) position.
Rotation X/Y/Z	These values represent the initial (resting) rotation of the constraining item.
Scale X/Y/Z	These values have no direct effect on constraints.
Slide Hinge	
Enable	Toggles the constraint item on or off. When enabled, the constraint item is considered during a dynamics simulation; when disabled, the constraint item is ignored. However, disabled constraints are persistent across Modo sessions, being saved with the scene, and retain their present settings.
SubCollisions	When enabled, allows constrained objects (especially those within a chain, not a literal chain, but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculations are required, however, constrained items may inter-penetrate each other. If this occurs, enable **SubCollisions**.
Body A	Represents the item that another item is constrained to, the parent or master item.
Body B	Represents the item that is constrained, the child or auxiliary item.
Breakable	Allows constraints to be broken apart when a certain stress threshold is reached, defined by the **Break Stress** value. When the threshold is reached, the constraint no longer has any effect on the constrained item, leaving it to be affected by any other forces present in the simulation.
Stress Break	Determines the threshold when the constraint no longer affects the constrained item. The **Breakable** option must be enabled for this value to have any effect on the constraint.
Limit Rotation	The hinge part of this constraint is not initially limited in its rotation. By enabling this option, you can constrain the range of rotation that can be applied to the constraint, similar to the motion of a door hinge that is allowed to only swing so

Option	Description
	far.
Min/Max Rotation Limit	The **Minimum** value sets the lower extreme of the of the rotation clamping, while the **Maximum** value limits the upper extreme value.
Limit Min/Max Linear	By default, the motion of the constrained item is not limited across the length of the constraining line. Enabling the upper and/or lower limits allows you to restrict the motion to a specific range, determined by the Min/Max position values.
Min/Max Limit Linear	When the **Limit Min/Max Linear** values are enabled, you can define a range as a distance value offset from the constraint's center position.

Spring

The spring is a special constraint that mimics the motion of an actual coiled spring. The spring constraint produces a motion similar to the point constraint, swinging and swaying, with the addition of the stretching and compression action of a spring. It is represented in the viewport by a coil that extends and compresses as objects move during a simulation.

To apply a slide hinge constraint automatically, follow these steps:

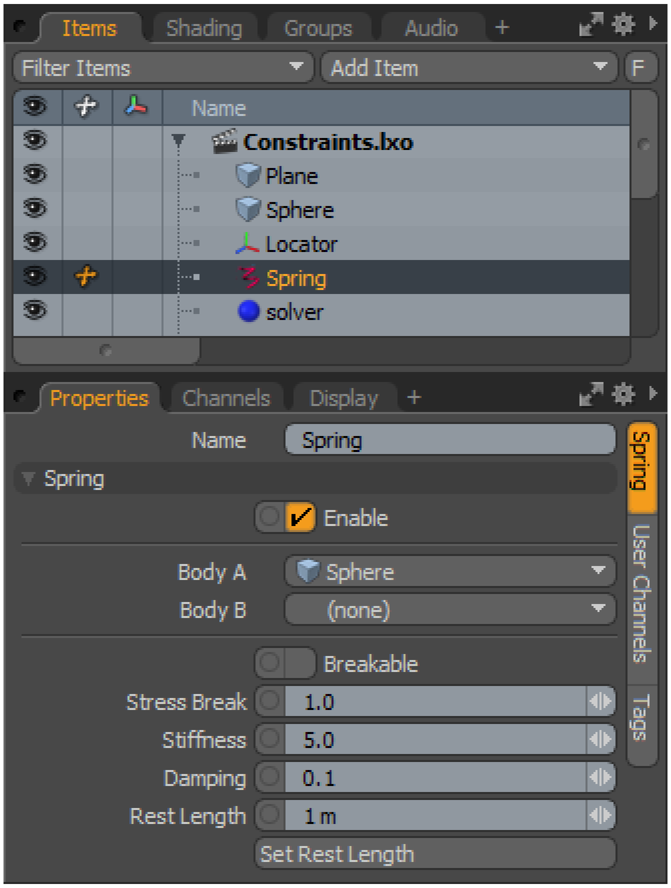
1. Select two items prior to invoking any of the constraint types.

   The first item selected represents the master or parent item and is defined as **Body A** in the constraint, the second item selected is the item that is constrained, also considered the auxiliary or child item, defined as **Body B**.
2. With the items selected, in the **Dynamics** sub-tab, click on the **Spring** constraint button.

This adds the constraint item to the **Items** list.

By default, the spring stretches between the center points of the two selected objects.

Note: While constraints can be added manually to a scene by way of the **Items** list, automatically assigning them to items is the preferred method as it also automatically defines the constraint's position relative to the selected items.


Items
Shading
Groups
Audio
Filter Items
Add Item
F
Name
Constraints.lxo
Plane
Sphere
Locator
Spring
solver
Properties
Channels
Display
Spring
User Channels
Tags

Name: Spring

Spring

☑ Enable

Body A: Sphere

Body B: (none)

☐ Breakable

Stress Break: 1.0

Stiffness: 5.0

Damping: 0.1

Rest Length: 1 m

Set Rest Length

Option	Description
Name	Displays the current constraint item name. To change it, click on the field and type in the new name.
Spring	
Enable	Toggles the spring item on or off. When enabled, the spring is considered during a dynamics simulation; when disabled, the spring is ignored. However, disabled items are persistent across Modo sessions, being saved with the scene, and retain

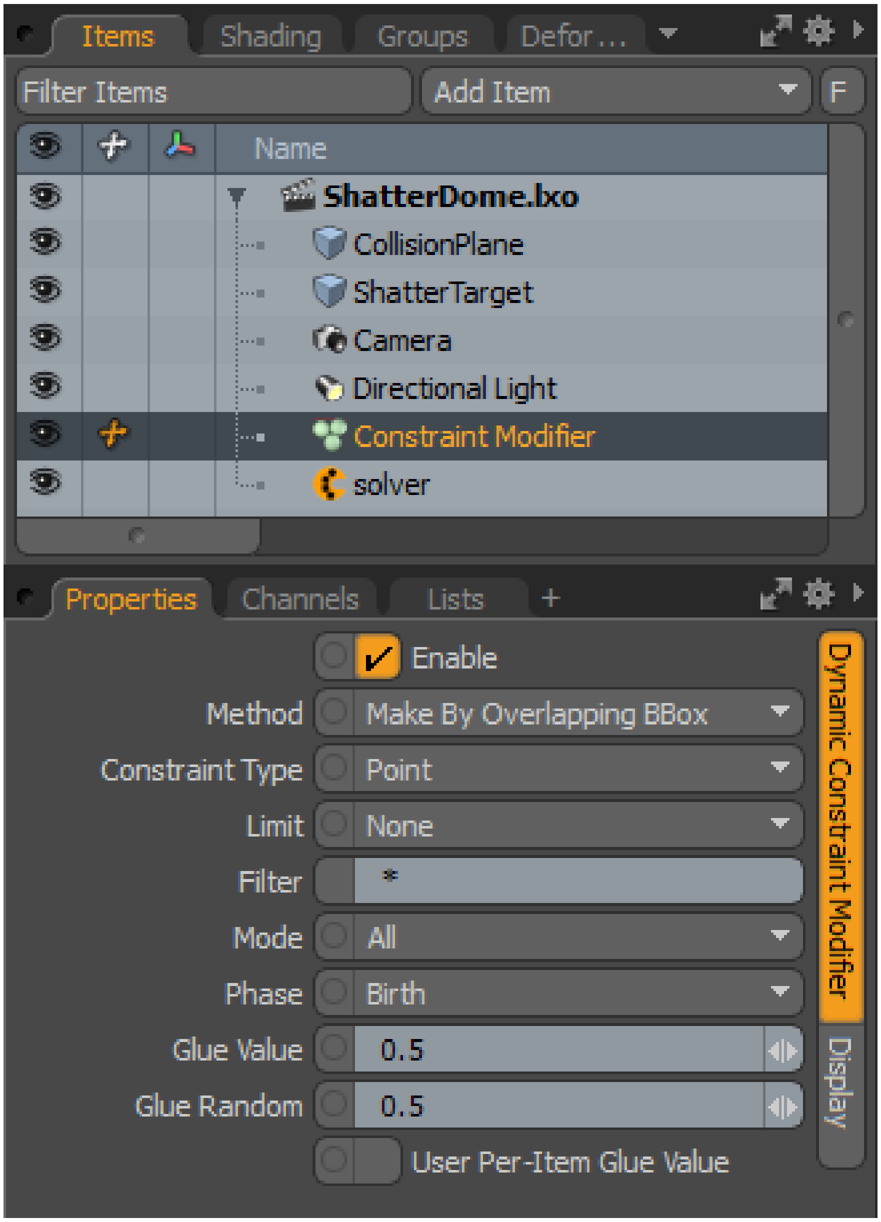
Option	Description
	their present settings.
SubCollisions	When enabled, allows constrained objects (especially those within a chain, not a literal chain, but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculations are required, however, constrained items may inter-penetrate each other. If this occurs, enable **SubCollisions**.
Body A	Represents the item that another item is constrained to, the parent or master item.
Body B	Represents the item that is constrained, the child or auxiliary item.
Breakable	Allows the spring to be broken apart when a certain stress threshold is reached, defined by the **Break Stress** value. When the threshold is reached, the spring no longer has any effect on the constrained item, leaving it to be affected by any other forces present in the simulation.
Stress Break	Determines the threshold when the spring no longer affects the constrained item. The **Breakable** option must be enabled for this value to have any effect on the spring.
Stiffness	The amount of force it takes to compress the spring. A stiffer spring is harder to compress.
Damping	During a simulation, a spring oscillates between extended and compressed. The **Damping** effect reduces the amplitude of the spring's oscillations over time.
Rest Length	Represents the length of the spring when it is not in motion. This can also be considered the distance between the two constraining items when the spring reaches equilibrium.

Constraint Modifier

The constraint modifier works like virtual glue to constrain dynamic items (often shattered items) together during a simulation. It is designed to work with Modo's group system. You can attach groups to this item's dynamics graph in the **Schematic** viewport, connecting the group's relationship connector output to the constraint modifier's dynamics relationship input. For more information about working with the **Schematic** viewport, see Schematic Viewport.

If no groups are attached, all dynamic bodies in the scene are considered for constraint. Another important concept is that constraint modifiers can be layered. So, you could glue all of the pieces together in a shattered object using a weak glue and then layer on another constraint modifier to harden up certain sections. Since there can only ever be a single glue constraint between two bodies, the later created one replaces any earlier ones.

To add a constraint modifier, in the **Items** list, click **Add Item** > **Dynamics** > **Constraints**, and double-click **Contraint Modifier** in the list. When you select the newly added constraint modifier in the **Items** list, the following options appear in the **Properties** panel.


Items
Shading
Groups
Defor...
Filter Items
Add Item
F
Name
ShatterDome.lxo
CollisionPlane
ShatterTarget
Camera
Directional Light
Constraint Modifier
solver
Properties
Channels
Lists
+
Dynamic Constraint Modifier
Display

☑ Enable

Method: Make By Overlapping BBox

Constraint Type: Point

Limit: None

Filter: *

Mode: All

Phase: Birth

Glue Value: 0.5

Glue Random: 0.5

☐ User Per-Item Glue Value

Option	Description
Enable	This option enables/disables the effect of the constraint modifier.
Method	There are six constraining methods to the constraint modifier. The first three create new constraints, and the last three modify existing constraints. **Make By Overlapping Box** checks all of the bodies in the dynamic world and records which ones bounding boxes that overlap. It makes constraints at those points. **Make By Contact** checks all of the bodies in the dynamic world and records which ones have opposing faces that are touching (such as those created by the Shatter command, making constraints at those points. When **Make By Proximity** is selected, there are two other channels that work with this option: the **Proximity Search Range**, and **Max Constraints**. **Proximity Search Range** is the maximum distance, using the body centers, that it looks to find neighboring bodies. **Max Constraints** is the maximum number of constraints that is connected from one body to other bodies. It should be noted that there is only ever allowed one constraint between two bodies, so any connected bodies don't count against the **Max Connections** value. In this mode, if you have the constraint modifier selected, you can see a preview of the constraints that are created. The **Break**, **Decay**, and **Strengthen** options modify existing constraint values when layering subsequent constraints.
Constrain Type	Defines the type of constraint created between the bodies. A **Pin** constraint is a fixed rigid connection and a **Point** constraint rotates around on the connection point.
Limit	Limits the types of items that constraints are added to, providing three options. **None** is for no limits. **Volume Less Than** uses the bodies calculated volume, only considers bodies whose volume is less than the specified **Volume Threshold**, and **Volume Greater Than** only considers bodies whose volume is less than the specified **Volume Threshold**.
Filter	This takes a wildcard string that is compared against the item's name. Considers bodies that are true. If the filter field is set to **Cube***, only items whose name is prefixed **Cube** is considered. **Note:** This option is case sensitive.

Option	Description
Mode	An additional control for constraining items. • **All** - All bodies are considered. If groups are attached to the modifer, only the items in that group are considered, if not, all dynamic items are considered. • **Boundary** - If multiple groups are attached, this only considers creating constraints between items in those groups. • **Area** - This, along with the **Area** channel, divides up the space of the attached bodies and only constrains items that fall into the same section together. The **Area** channel defines how many cubic meters the space should be divided into.
Phase	There are three **Phase** options, **Start** and **Tick** and **Birth**. This option defines when the constraint modifier fires. **Start** is at the beginning of the entire simulation, **Tick** is every simulation sub-step (defined by the **Physics Rate (Hz)** value of the solver item), and **Birth** fires every simulation tick, but only affects newly created rigid bodies.
Glue Value	Defines the strength of the connection between the constrained items for this particular constraint.
Glue Random	Determines a randomizing amount for the **Glue Value** applied to constrained items.
Use the Per-Item Glue Value	If this option is enabled, then the specified **Glue Value** is multiplied by the **Glue Value** defined on the dynamic element itself.
Recovery	When forces hit a constraint they weaken it until it breaks. When the **Make By Contact** constraint method is selected, the **Recovery** value determines how quickly a constraint recovers to full strength.
Volume Threshold	When using the **Limit** option, this value defines the min or max volume that is culled from the constraint.

Dynamics Forces

In addition to gravity applied globally to dynamics, you can also add other forces. A force is any influence that causes a dynamic item to undergo a change in speed or direction. Forces are global in nature and can affect both dynamic items as well as particle items, simultaneously, if desired. There are seven different types of forces available to affect dynamic objects: linear, radial, drag, turbulence, vortex, wind and curve force.

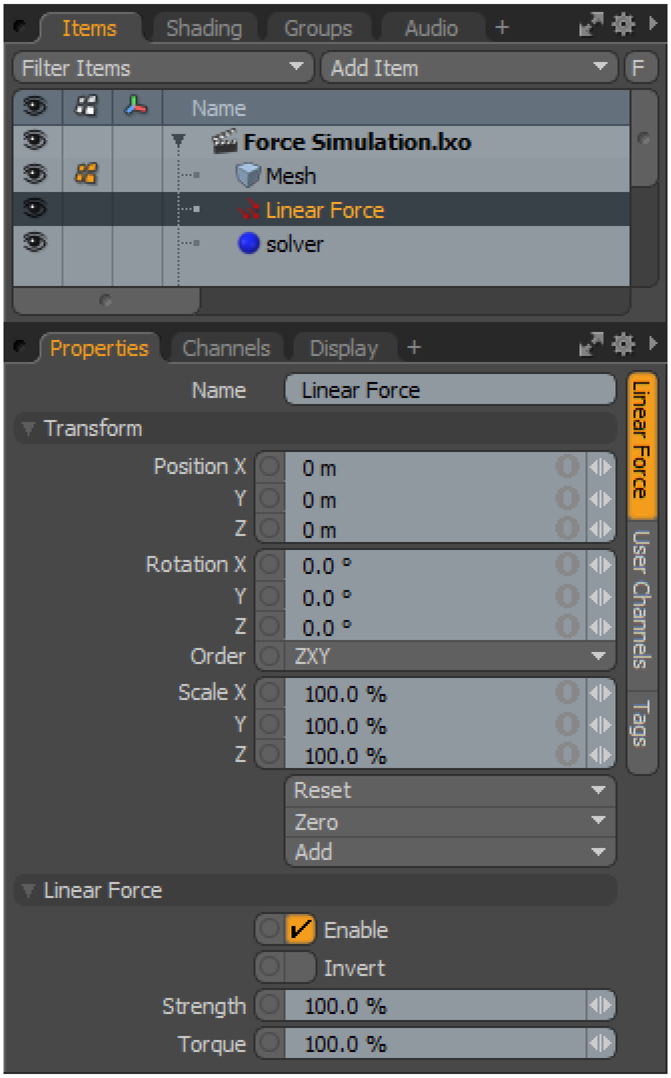
Linear Force

Forces provide a way to further affect and control the motion of dynamic items in a scene. The linear force item applies a constant force globally to all dynamic items in the direction determined by the force's rotational X, Y, or Z values. Linear force is similar to wind, but without the **Speed** dynamic that wind has. The higher the **Strength** value, the stronger the force applied to the dynamic objects.

You can add a linear force item to a simulation in the **Dynamics** sub-tab, by selecting **Forces** > **Linear** in the **Layout** interface's toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Forces**, double-click **Linear Force**. Any force added after a dynamic body auto-connects to the solver item, forces added beforehand need to be connected manually in the **Schematic** view.

Remove force items the same way as any other item: simply select the item and press **Delete** on the keyboard.

When selected, the following attributes appear in the **Properties** viewport.


Items
Shading
Groups
Audio
Filter Items
Add Item
F
Name
Force Simulation.lxo
Mesh
Linear Force
solver
Properties
Channels
Display
Linear Force
User Channels
Tags
Transform
Reset
Zero
Add

Name: Linear Force

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Linear Force

☑ Enable

☐ Invert

Strength: 100.0 %

Torque: 100.0 %

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	The linear force item globally affects dynamic items regardless of its position, therefore these values don't influence the force's effect.
Rotation X/Y/Z	Affect the direction of the linear force options, allowing you to easily change the direction of the force applied to dynamic items. Default forward direction is Y+.
Scale X/Y/Z	The linear force item globally affects dynamic items regardless of its scale, therefore these values are not applicable to it. To adjust the strength of the linear

Option	Description
	force, adjust the **Strength** value.
Linear Force	
Enable	Toggles the force item on or off. When enabled, the force item is considered during a dynamics simulation; when disabled, the force item is ignored. However, disabled forces are persistent across Modo sessions, being saved with the scene, and retain their present settings.
Invert	Reverses the direction of the linear force.
Strength	Acts as a multiplier on an object's motion in the direction the force is oriented toward. The higher the value the greater the strength of the force.
Torque	Acts as a multiplier on rotational movements of dynamic items. The higher the value the greater the strength of the force.

Radial Force

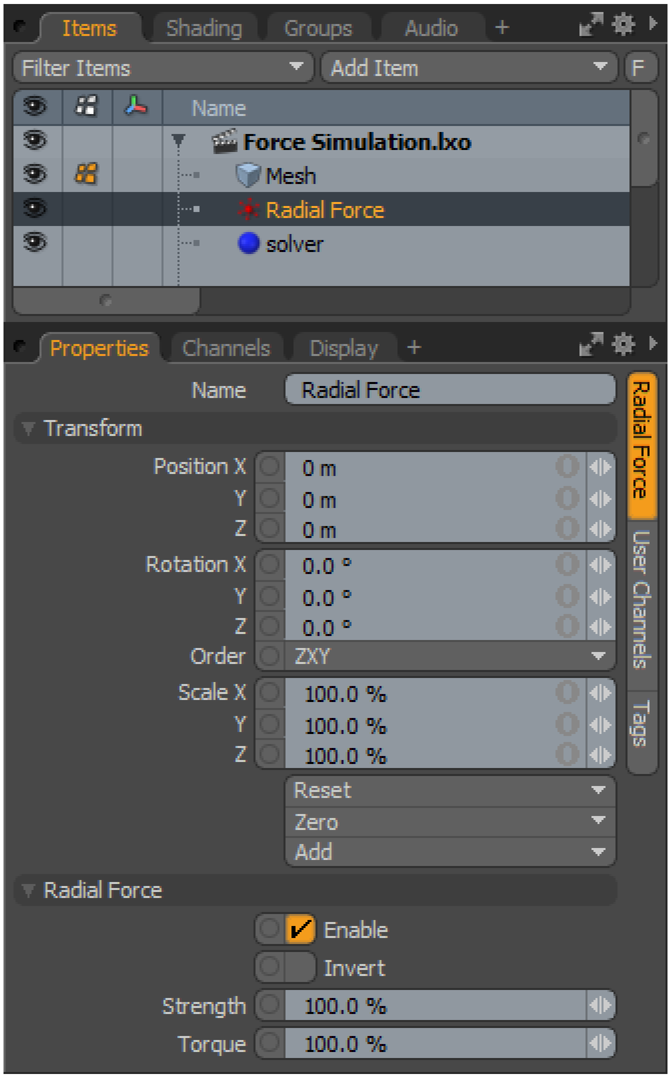
The radial force item applies a force that draws items toward or pushes them away from the force locator center (defined by the item's position).

This force is like a magnet that repels or attracts dynamic objects. The higher the positive value, the stronger the push force applied, the greater the negative value, the stronger the attraction.

You can add a radial force item to a simulation by selecting **Forces** > **Radial** in the **Layout** interface's toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Forces**, double-click **Radial Force**. Any force added after a dynamic body auto-connects to the solver item, forces added beforehand need to be connected manually in the **Schematic** view.

Remove Force items the same way as any other item: simply select the item and press **Delete** on the keyboard.

When selected, the following attributes appear in the **Properties** viewport.


Items
Shading
Groups
Audio
+
Filter Items
Add Item
F
Name
Force Simulation.lxo
Mesh
Radial Force
solver
Properties
Channels
Display
+
Radial Force
User Channels
Tags
Transform
Reset
Zero
Add
Radial Force

Name: Radial Force

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable

☐ Invert

Strength: 100.0 %

Torque: 100.0 %

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	An item transform that allows you to numerically position the force item in XYZ space, representing the point of the repulsion or attraction.
Rotation X/Y/Z	The radial force item affects items radially, therefore these values are not applicable to it.
Scale X/Y/Z	The radial force item globally affects dynamic items regardless of its scale, therefore these values are not applicable to it.

Option	Description
Radial Force	
Enable	Toggles the force item on or off. When enabled, the force item is considered during a dynamics simulation; when disabled, the force item is ignored. However, disabled forces are persistent across Modo sessions, being saved with the scene, and retain their present settings.
Invert	Reverses the pull direction of the radial force.
Strength	Determines how powerful the force is pulling or repelling. The higher the value, the greater the influence of the force. Negative values push dynamic items away, and positive ones pull them toward the radial force.
Torque	Acts as a multiplier on rotational movements of dynamic items. The higher the value the greater the strength of the force.

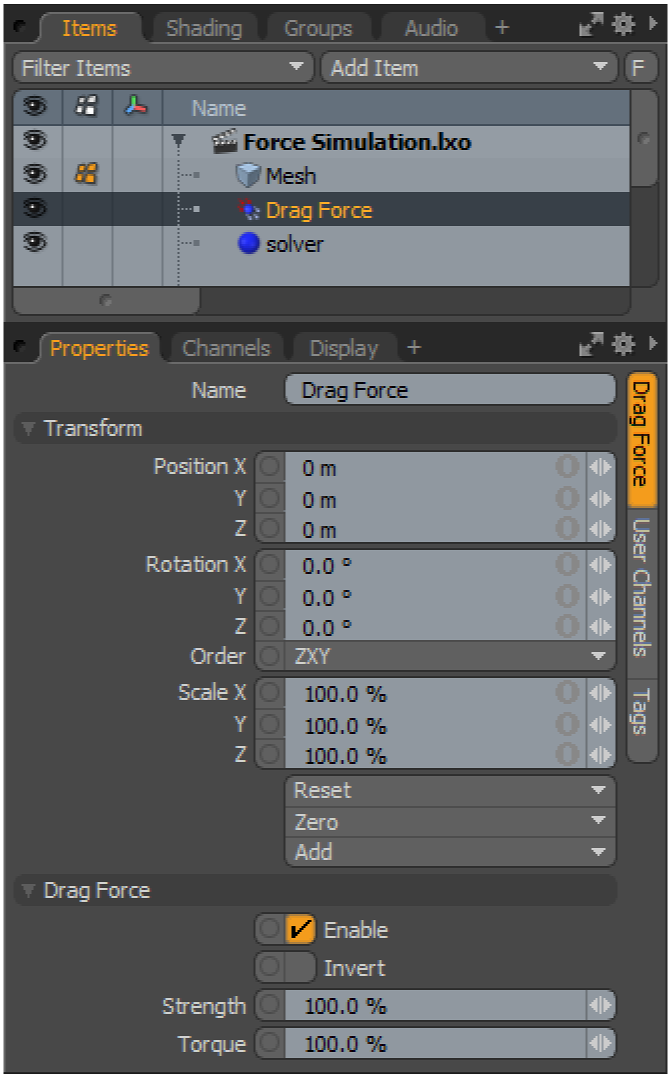
Drag Force

The drag force item applies a counter force to the object's current linear and rotational velocities. This is useful in slowing object motions, like dragging it through molasses. The higher the value, the stronger the drag applied to the dynamic objects.

You can add a drag force item to a simulation by selecting **Forces** > **Drag** in the **Layout** interface's toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Forces**, double-click **Drag Force**. Any force added after a dynamic body auto-connects to the solver item, forces added beforehand need to be connected manually in the **Schematic** view.

Remove force items the same way as any other item: simply select the item and press **Delete** on the keyboard.

When selected, the following attributes appear in the **Properties** viewport.


Items
Shading
Groups
Audio
+
Filter Items
Add Item
F
Name
Force Simulation.lxo
Mesh
Drag Force
solver
Properties
Channels
Display
+
Drag Force
User Channels
Tags

Name: Drag Force

Transform

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Reset
Zero
Add

Drag Force

☑ Enable
☐ Invert
Strength: 100.0 %
Torque: 100.0 %

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	Defines the locator position for the drag force item. By default, the drag force item affects dynamic objects globally, so position of the item in the scene is irrelevant to its effect.
Rotation X/Y/Z	The drag force item globally affects dynamic items regardless of its rotation, therefore these values are not applicable to it.
Scale X/Y/Z	The drag force item globally affects dynamic items regardless of its scale, therefore

Option	Description
	these values are not applicable to it. To adjust the amount of force attenuation, modify the **Strength** value.
Drag Force	
Enable	Toggles the force item on or off. When enabled, the force item is considered during a dynamics simulation; when disabled, the force item is ignored. However, disabled forces are persistent across Modo sessions, being saved with the scene, and retain their present settings.
Invert	Reverses the drag effect on dynamic items.
Strength	Acts as a multiplier on an object's motion. The higher the value, the greater the strength of the drag. Positive values slow dynamic items down, and negative ones impel them to move faster.
Torque	Acts as a multiplier on rotational movements of dynamic items. The higher the value, the greater the strength of the drag.

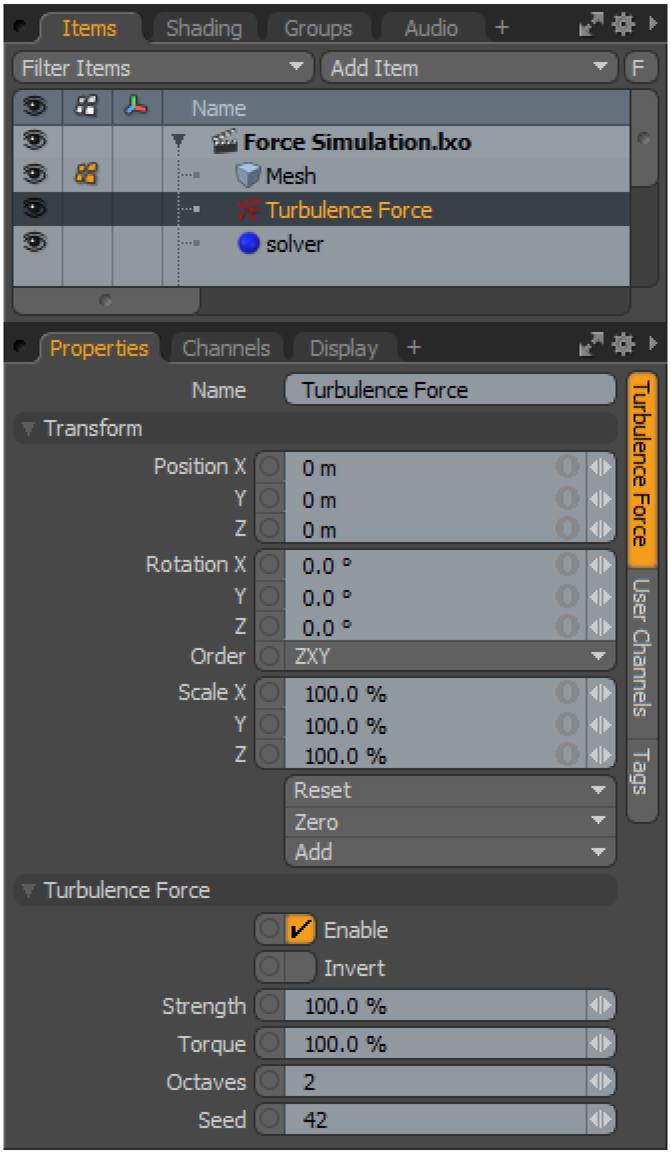
Turbulence Force

The turbulence force item applies a noise-like function to a dynamic item's movements, producing an unsteady or agitated motion. Useful for adding a small or large amount of randomness to simulations, depending on the strength of the effect. The higher the value, the stronger the turbulence applied to the dynamic objects.

You can add a turbulence force item to a simulation by selecting **Forces** > **Turbulence** in the **Layout** interface's toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Forces**, double-click **Turbulence Force**. Any force added after a dynamic body auto-connects to the solver item, forces added beforehand need to be connected manually in the **Schematic** view.

Remove Force items the same way as any other item: simply select the item and press **Delete** on the keyboard.

When selected, the following attributes appear in the **Properties** viewport.


Items
Shading
Groups
Audio
+
Filter Items
Add Item
F
Name
Force Simulation.lxo
Mesh
Turbulence Force
solver
Properties
Channels
Display
+
Turbulence Force
User Channels
Tags
Reset
Zero
Add

Name: Turbulence Force

Transform

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Turbulence Force

☑ Enable
☐ Invert
Strength: 100.0 %
Torque: 100.0 %
Octaves: 2
Seed: 42

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	Changes the position of the turbulence field, which can modify the way in which the turbulence affects dynamic objects.
Rotation X/Y/Z	Determines the rotation of the turbulence field as X, Y & Z values for each axis.
Scale X/Y/Z	Defines the scale of the turbulence field as X, Y & Z values for each axis.

Option	Description
Turbulence Force	
Enable	Toggles the force item on or off. When enabled, the force item is considered during a dynamics simulation; when disabled, the force item is ignored. However, disabled forces are persistent across Modo sessions, being saved with the scene, and retain their present settings.
Invert	Reverses the turbulence field influence on dynamic items.
Strength	Determines how much influence the turbulence field has on the dynamic items. The higher the value, the greater the disturbing strength of the turbulence.
Torque	Acts as a multiplier on rotational movements of dynamic items. The higher the value, the greater the influence of the turbulence.
Octaves	Determines the number of layers for the noise function, where each additional layer adds complexity to the field, subsequently making finer details to the turbulent motion applied.
Seed	Defines the initial value for the noise function, with every different value producing a different random variation to the noise. If a turbulence field isn't producing the results you like, try adjusting this value to sample different randomized fields.

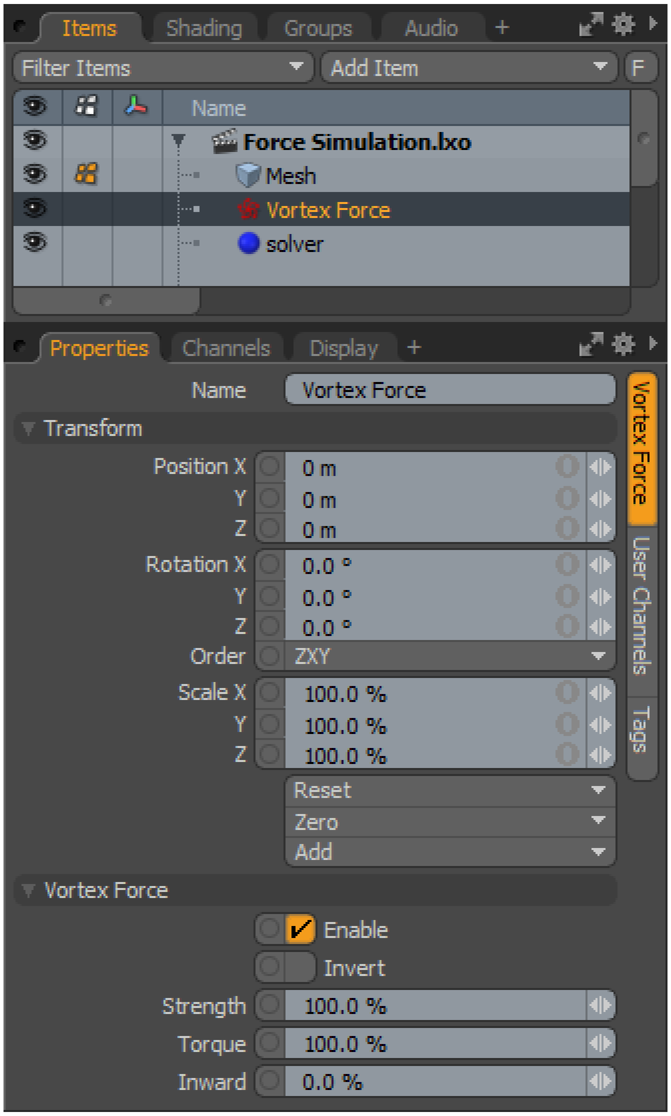
Vortex Force

The vortex force item applies a rotational swirling force, like a whirlpool, spinning dynamic items around the force locator's center (defined by the item's center position).

You can add a vortex force item to a simulation by selecting **Forces** > **Vortex** in the **Layout** interface's toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Forces**, double-click **Vortex Force**. Any force added after a dynamic body auto-connects to the solver item, forces added beforehand need to be connected manually in the **Schematic** view.

Remove Force items the same way as any other item: simply select the item and press **Delete** on the keyboard.

When selected, the following attributes appear in the **Properties** viewport.


Items
Shading
Groups
Audio
Filter Items
Add Item
F
Name
Force Simulation.lxo
Mesh
Vortex Force
solver
Properties
Channels
Display
Vortex Force
User Channels
Tags
Transform
Reset
Zero
Add
Vortex Force

Name: Vortex Force

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable

☐ Invert

Strength: 100.0 %

Torque: 100.0 %

Inward: 0.0 %

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	These values determine the vortex line - the center position for the vortex force, where all items affected by the vortex spin around this central point.
Rotation X/Y/Z	These values determine the angle that the vortex revolves around, defined by the X, Y & Z rotation values.
Scale X/Y/Z	The vortex force item globally affects dynamic items, regardless of its scale,

Option	Description
	therefore these values are not applicable to it. To adjust the influence of the force, modify the **Strength** value.
Vortex Force	
Enable	Toggles the force item on or off. When enabled, the force item is considered during a dynamics simulation; when disabled, the force item is ignored. However, disabled forces are persistent across Modo sessions, being saved with the scene, and retain their present settings.
Invert	Reverses the effect of the vortex on dynamic items, altering its draw and rotation direction.
Strength	Determines how much influence the vortex has on the dynamic items. The higher the value, the greater the rotational influence of the vortex.
Torque	Acts as a multiplier on rotational movements of dynamic items. The higher the value, the greater the influence of the vortex on rotations.
Inward	Controls the inward pull toward the center of the vortex. As items spin around the central point, when the speed increases, the resulting centrifugal force impels dynamic bodies outward. the **Inward** option is a means to oppose that force.

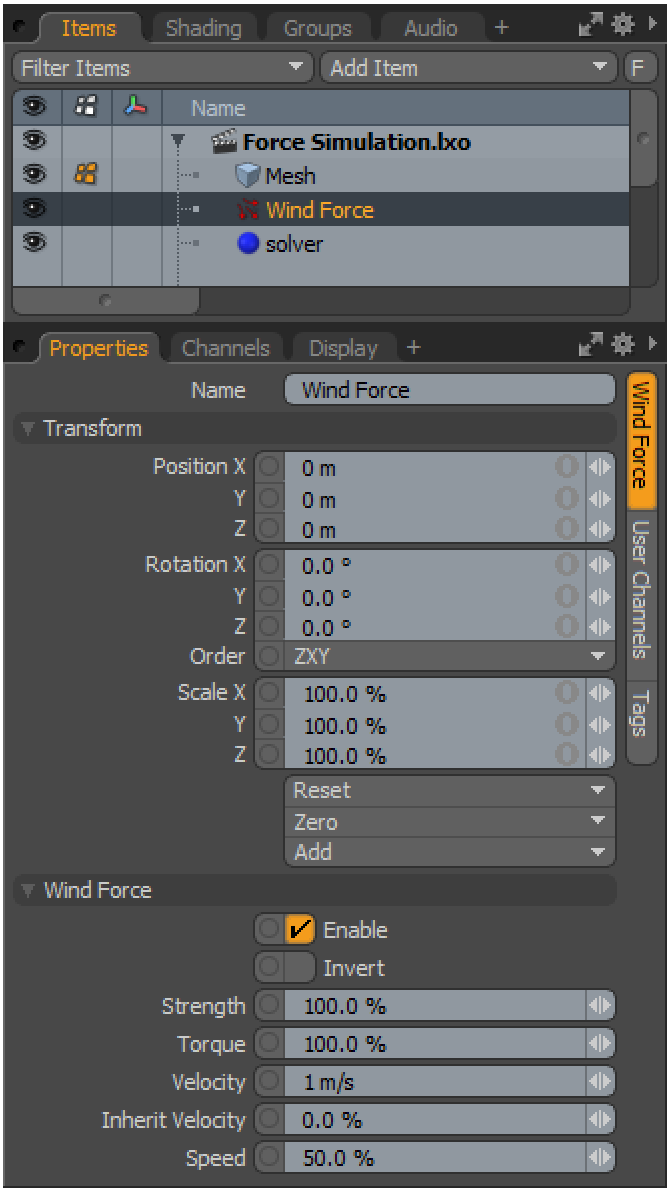
Wind Force

The wind force directionally accelerates or decelerates the speed of dynamic items in a scene until the velocities match that of the wind force.

You can add a wind force item to a simulation by selecting **Forces** > **Wind** in the **Layout** interface's toolbox. Alternatively, in the **Items** list, click **Add Item**, then go to **Forces**, and double-click **Wind Force**. Any force added after a dynamic body auto-connects to the solver item, forces added beforehand need to be connected manually in the **Schematic** view.

Remove Force items the same way as any other item: simply select the item and press **Delete** on the keyboard.

When selected, the following attributes appear in the **Properties** viewport.



Items
Shading
Groups
Audio
+
Filter Items
Add Item
F
Name
Properties
Channels
Display
+
Transform
Reset
Zero
Add
Wind Force
User Channels
Tags

Force Simulation.lxo
Mesh
Wind Force
solver

Name Wind Force

Transform

Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Order ZXY
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Wind Force

☑ Enable
☐ Invert
Strength 100.0 %
Torque 100.0 %
Velocity 1 m/s
Inherit Velocity 0.0 %
Speed 50.0 %

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	The wind force item globally affects dynamic items regardless of its position, therefore these values don't influence the force's effect.
Rotation X/Y/Z	Rotation values affect the direction of the wind force options, allowing you to easily change the direction of the force applied to dynamic items. Default forward

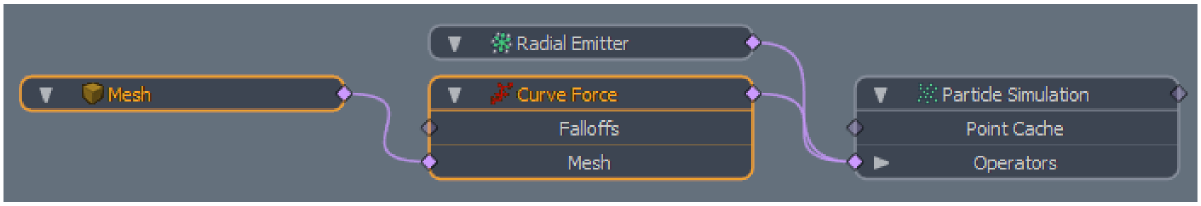
Option	Description
	direction is Y+.
Scale X/Y/Z	The wind force item globally affects dynamic items, regardless of its scale, therefore these values are not applicable to it.
Wind Force	
Enable	Toggles the force item on or off. When enabled, the force item is considered during a dynamics simulation; when disabled, the force item is ignored. However, disabled forces are persistent across Modo sessions, being saved with the scene, and retain their present settings.
Invert	Reverses the directional values of the wind.
Strength	Determines how much influence the wind force has on the dynamic items. The higher the value the greater the push of the wind.
Torque	Acts as a multiplier on rotational movements of dynamic items. The higher the value the greater the influence of the wind.
Velocity	The speed of the wind force. Dynamic items affected by this force don't move faster than the defined value (unless they have their own additional forces or impulses controlling them).
Inherit Velocity	Controls the degree to which the motions of the wind force locator itself affect the wind movement.
Speed	Determines the actual speed at which the force moves or affects its target items. It is defined as a percentage of the **Velocity**.

Curve Force

The curve force allows you to use a drawn curve to determine the directional flow of dynamic items, flowing from the curve's base, the first point, toward its end point. The curve force is useful in cases such as getting particles to flow through a bent hollow tube or along a meandering stream.

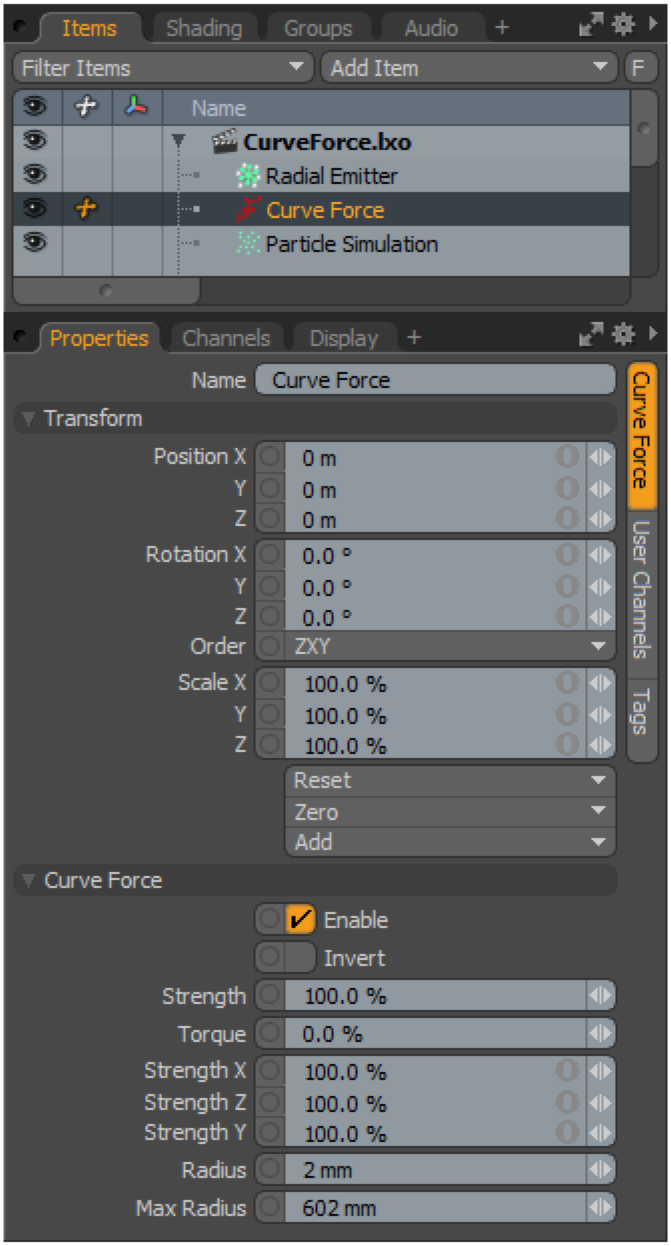
The curve force requires a curve to apply the force with, so you need to draw a curve of some type, either a Hermite Adding Curves or a Bezier curve. It is a good practice to keep curves isolated withing their own mesh layer. With the curve created, you can add a curve force item to a simulation by selecting **Forces** > **Curve** in the **Layout** interface's toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Forces**, double-click

Curve Force. Once created, it is necessary to connect the items manually within the **Schematic** viewport. The Mesh Item containing the curves needs to be connected to the mesh input of the curve force item.


Radial Emitter
Mesh
Curve Force
Falloffs
Mesh
Particle Simulation
Point Cache
Operators

Remove force items the same way as any other item: simply select the item and press **Delete** on the keyboard.

When selected, the following attributes appear in the **Properties** viewport.


Items
Shading
Groups
Audio
Filter Items
Add Item
F
Name
CurveForce.lxo
Radial Emitter
Curve Force
Particle Simulation
Properties
Channels
Display
Transform
Reset
Zero
Add
Curve Force
Curve Force
User Channels
Tags

Name: Curve Force

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable
☐ Invert
Strength: 100.0 %
Torque: 0.0 %
Strength X: 100.0 %
Strength Z: 100.0 %
Strength Y: 100.0 %
Radius: 2 mm
Max Radius: 602 mm

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position X/Y/Z	The curve force item affects dynamic items based on the position of an external curve item, therefore these values don't influence the force's effect.

Option	Description
Rotation X/Y/Z	The curve force item affects dynamic items based on the position of an external curve item, therefore these values are not applicable to it.
Scale X/Y/Z	The curve force item affects dynamic itemsbased on the position of an external curve item, therefore these values are not applicable to it.
Curve Force	
Enable	Toggles the force item on or off. When enabled, the force item is considered during a dynamics simulation; when disabled, the force item is ignored. However, disabled forces are persistent across Modo sessions, being saved with the scene, and retain their present settings.
Invert	Reverses the direction of the curve force where the force flows from the end toward the base.
Strength	Determines how much influence the curve force has on the dynamic items. The higher the value, the greater the influence of the curve force.
Torque	Acts as a multiplier on rotational movements of dynamic items. The higher the value, the greater the influence of the curve force.
Strength X/Y/Z	Acts as a multiplier over individual directional vectors as dynamic items move along the curve.
Radius	Determines the area or volume around the curve where dynamic items try to remain constrained, unless affected by other stronger forces.
Max Radius	Determines the maximum area around the curve where the force tries to attract dynamic items. Anything outside this area is ignored.

Anchor

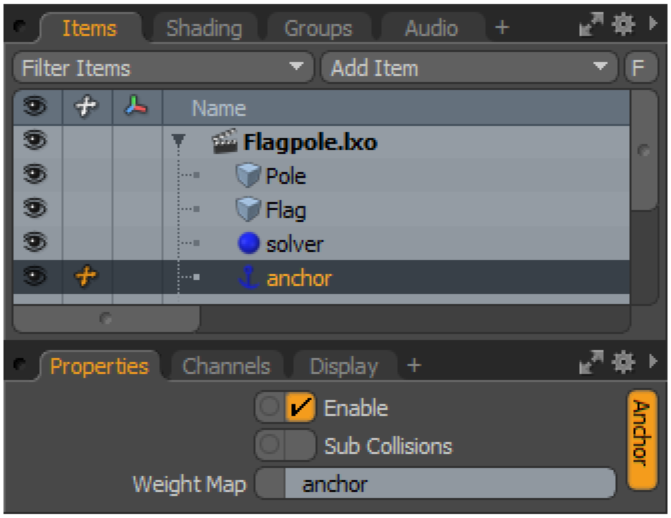
The **Make Anchors** option is used to connect soft body items to rigid body items. For best results, the rigid body item should be a passive rigid body, to keep it from flying away. The anchor option is useful, for example, for connecting a flag to a flagpole. In order for the anchor option to work, follow this specific workflow:

1. First, in **Items selection mode**, select the target soft body item and the target passive rigid body item.

2. With the items selected, change into the **Vertices mode** (press **1** on the keyboard) and select at least one vertex on the soft body item.
3. In the **Dynamics** toolbox, press the **Make Anchors** option.

   The two items are now rigidly connected by an additional anchor item added to the soft body.

To break the connection, simply delete the anchor item; to make adjustments, remove the anchor and recreate it, following the above instructions.



☑ Enable

☐ Sub Collisions

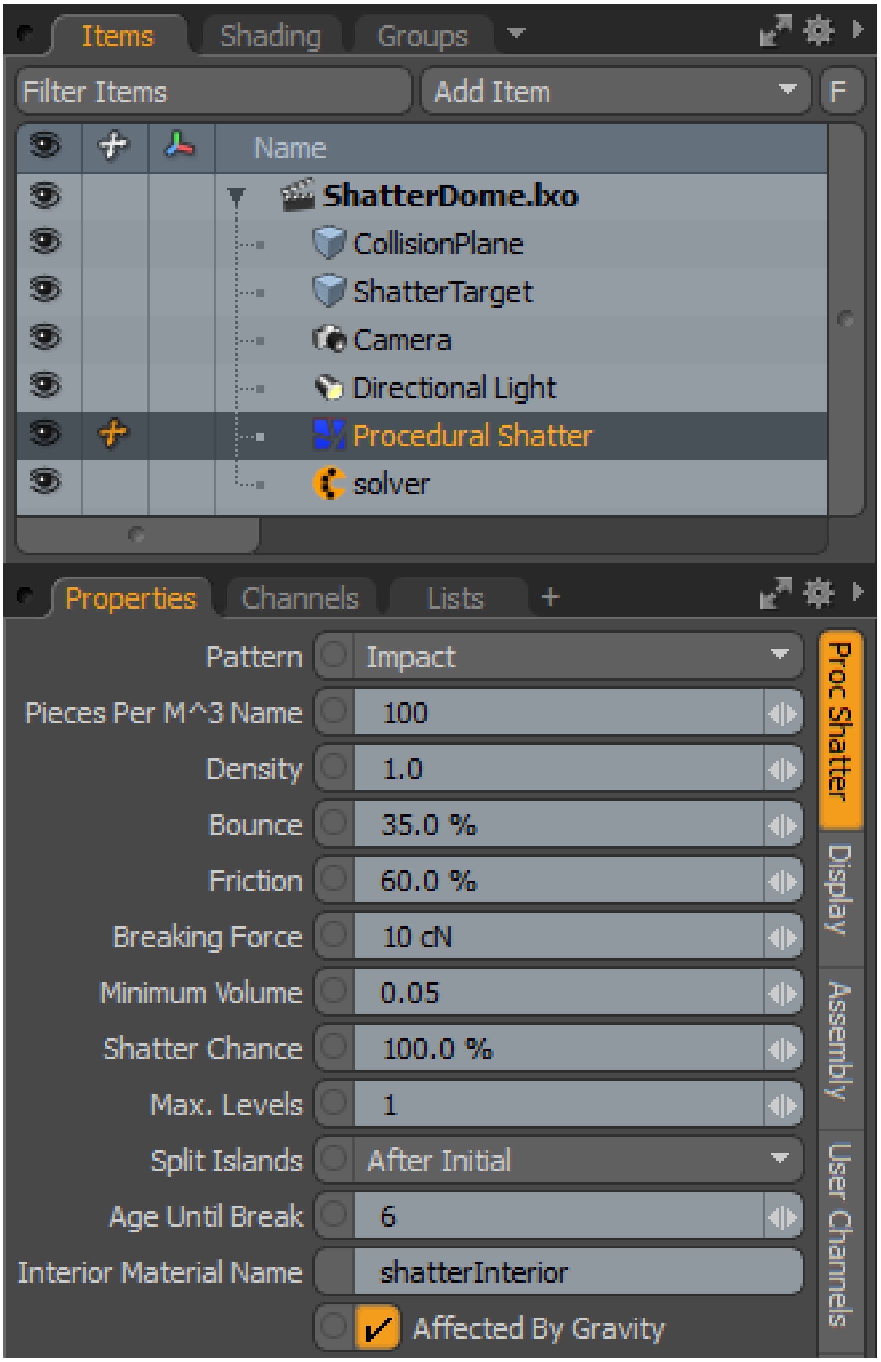
Weight Map: anchor

Option	Description
Enable	Toggles the anchor item on or off. When enabled, the anchor is considered during a dynamics simulation, when disabled, the anchor item is ignored. However, disabled items are persistent across Modo sessions, being saved with the scene, and retain their present settings.
SubCollisions	When enabled, allows constrained objects (especially those within a chain, not a literal chain, but subsequent groupings) to collide with one another. When disabled (the default state), fewer calculations are required, however, constrained items may inter-penetrate each other. If this occurs enable **SubCollisions**.
Weight Map	When the anchor item is assigned using the appropriate steps, a Weight Map is generated automatically on the soft body item, called **Anchor**. This is not a user-assignable feature, but merely feedback that the proper map was generated and assigned.

Procedural Shatter

The procedural shatter option provides a simple way to shatter and break models using dynamics. Any geometry that can be made dynamic can be made to dynamically shatter. Once a procedural shatter item collides with another dynamic item, if enough force is applied, the item breaks apart into various smaller pieces as specified in the procedural shatter item. The controls to apply procedural shattering are found in the **Setup** layout, under the **Dynamics** sub-tab of the toolbox. With the target item(s) selected, you only need to press the **Procedural Shatter** button to apply the effect to that object. This adds a procedural shatter item to the **Items** list, automatically linking it to the selected item(s). That's it. All that's left is to tune the simulation and cache it for final rendering. Tuning can be done by playing the simulation using the simulation **Scene Preview** button (the solid green arrow), running the simulation for the length of the Timeline, and allowing for near real-time feedback as values are adjusted (there is a hiccup in playback as the item collides, this is the shatter evaluation happening and its length is highly dependent on the number of items created).

Note: The original item works like a proxy when combined with procedural shatter, where the shattered geometry is actually stored in the procedural item itself. To keep the original item from rendering, you need to disable it or set it as non-renderable.


Items
Shading
Groups
Filter Items
Add Item
F
Name
ShatterDome.lxo
CollisionPlane
ShatterTarget
Camera
Directional Light
Procedural Shatter
solver
Properties
Channels
Lists
+
Proc Shatter
Display
Assembly
User Channels

Pattern: Impact
Pieces Per M^3 Name: 100
Density: 1.0
Bounce: 35.0 %
Friction: 60.0 %
Breaking Force: 10 cN
Minimum Volume: 0.05
Shatter Chance: 100.0 %
Max. Levels: 1
Split Islands: After Initial
Age Until Break: 6
Interior Material Name: shatterInterior
✓ Affected By Gravity

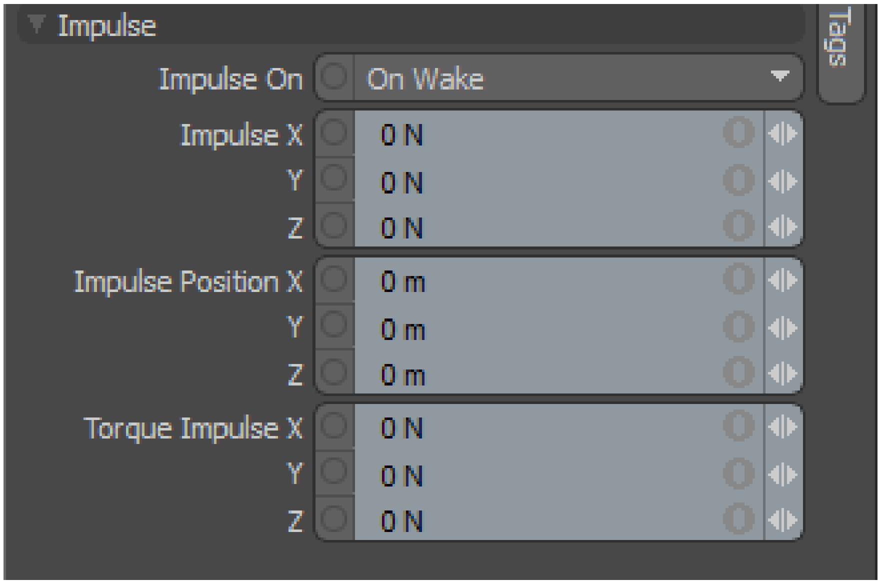
Procedural Shatter Properties

Option	Description
Pattern	Defines how the target object is broken apart for shattering. **Uniform** tries to

Option	Description
	break the item into roughly even sized pieces, **Impact** shatters the object into smaller pieces at its point of impact, with a falloff where the pieces get larger further away from the impact point. **Radial** produces a web-like radial pattern of broken pieces.
Pieces per M^3 Name	This value defines the number of pieces generated by a collision event, basically it is the average number of pieces generated per cubic meter.
Density	This value represents the density, or weight of the shattered object. The greater the density, the greater the force they apply to other dynamic objects.
Bounce	This is the collision response of the shattered pieces, when they hit another dynamic object. A value of 0% means the pieces don't bounce or inherit energy from other dynamic objects it contacts. A value of 100% inherits all of the energy from the contact object. Values above 100% produce rubber-ball like effects where objects impart more energy from contacts.
Friction	The amount the shattered pieces slide against another dynamic object. A value of 0.0 offer no resistance, while a value of 1.0 would keep the object from sliding entirely.
Breaking Force	Defines the amount of force necessary to cause the object to shatter into smaller pieces.
Minimum Volume	This value determines the smallest size a piece can get when shattered.
Shatter Chance	Determines the likelihood that an object breaks apart upon impact. Lower values keep pieces from breaking while higher values allow them to break more readily.
Max. Levels	Defines the number of times an item can be shattered by impacts. A level of one would means the item could be shattered only the first time and its pieces don't break any further, a level of two means shattered pieces can be broken again, and so on.
Split Islands	This option determines how the individual pieces (islands) within an item (overlapping or not) are treated when shattering the mesh. • **Never** - Islands are never treated individually, but as a single solid object. • **Initial** - Only islands in initial source mesh are treated as separate objects. • **After Initial** - Only islands in fractures shards are split into separate objects.

Option	Description
	• **Always** - Islands are always treated as separate objects.
Age Until Break	This value defines the amount of time that must pass before a subsequent shard can be broken again. **Note:** The **Max Levels** value must be above 1 to allow further shattering of subsequent shards.
Interior Material Name	This option defines a material tag that you can use in the **Shader Tree** to add distinct surfacing to the interior areas of the shattered pieces. The process isn't automatic, but you can add a Material Group Item to the **Shader Tree** and set the named Material from here as the **Polygon Tag** defining its mask. Then any other material items added into that group shade only on those surfaces.
Affected By Gravity	Toggles the influence of the gravity force over the procedurally shattering object, turning it on or off. The gravity setting is defined in the dynamics solver item.

Impulse



Impulse On: On Wake
Impulse X: 0 N
Y: 0 N
Z: 0 N
Impulse Position X: 0 m
Y: 0 m
Z: 0 m
Torque Impulse X: 0 N
Y: 0 N
Z: 0 N

Impulse - These values determine the initial state of the dynamic item(s), imbuing them with certain forces

at the beginning of their dynamics simulation, which is defined by the Impulse On value.	
Impulse On	This option determine the state that causes the 'Impulse' value to be applied to the dynamic item. **Off** - With this setting, no **Impulse** value is applied to the object. **On Wake** - With this setting, the **Impulse** values are applied when the item wakes up, for procedural shatter items this is the first frame of the simulation. **Continuous** - With this setting, the **Impulse** value is applied to the object continually, at every time step in the evaluation.
Impulse X/Y/Z	A directional force that is applied to the object when it wakes up. Useful when a keyframed object is handed off to dynamics, imparting direction and velocity to the object, which would otherwise just drop without it as Dynamics has no awareness of the objects velocity prior to waking up.
Impulse Position X/Y/Z	Position of the impulse is applied from affecting the object's trajectory.
Torque Impulse X/Y/Z	A rotational force that is applied to the object when it wakes up. Useful when a keyframed object is handed off to dynamics, imparting rotation to the object, which would otherwise not spin without it, as dynamics has no awareness of the object's motion before waking up.

Baking Dynamics

Once any dynamic item has been simulated and the motions cached, including both rigid bodies and particles, the cache remains as part of the saved scene until it is invalidated or deleted. Modo can render either of these dynamic items directly, using the cached values, but in cases where Modo may not be the final destination (or you want to apply modifiers to the item), you can convert the dynamics cache into keyframes for rigid bodies and geometry for particles. This conversion process is called baking.

Baking Rigid Bodies

Rigid bodies that have been simulated and cached can be converted to animated keyframes, retaining all the motion and nuance of the actual simulation, using the standard Bake Animation command. This is found in the menu bar under **Animate** > **Bake...**. To apply it, simply select the target item(s) and run the command. Keyframes are generated for each frame in the Timeline, based on the **Start** and **End** frame values entered. Baked dynamic items retain their dynamic features. To disable this, in the menu bar, click **Dynamics** > **Make**

Un-dynamic. This removes any tags from the item that would include it in future simulations. For more information on the **Bake Animation** command, see Tools and Commands.

Sometimes dynamics are used to simply place items in a scene, and the actual animated movement of the elements is unnecessary (such as filling a bowl with fruit). In this case, you can simply bake the current item position into the Position and Rotation transform of the item with the command **Dynamics** > **Bake Dynamics Transforms**. The current Timeline position is used for the defined values, so make sure it is located at the desired position. This also retains the dynamic features.

Baking Particles

Particles that have been simulated and cached can be converted to several different formats that provide some powerful options. With the target particle simulation item selected in the **Items** list, apply the menu bar command **Dynamics** > **Convert Particle Cache** > **Particles to Mesh**. This converts the particle positions at each frame into a Mesh Item that is parented to the particle simulator. This layer contains a series of continuous edges (also considered two-point polygons), representing the paths for each individual particle that you can directly modify using any of Modo's modeling tools, including the particle sculpting tools. Once modified to your liking, these paths can then be converted back into the cache as particle paths, replacing the previous values. This is done with the command **Dynamics** > **Convert Particle Cache** > **Particles from Mesh**. This allows for ultimate control over every single particle in a simulation. The **Auto Update** option keeps the cached values continuously updated as you modify the resulting mesh layer.

Sometimes it can be beneficial to simply use the simulation tools to generate particles for still images. In such cases, you can select the desired frame of a cached particle simulation along the Timeline and apply the menu bar command **Dynamics** > **Convert Particle Cache** > **Particles to Cloud**, converting the current positions into a Mesh Item, containing static vertices representing each individual particle from the simulation. The resulting layer can be assigned as a point source to the many particle-related effects that Modo offers, such as Replicators, Blobs or Volumes.

Another interesting effect is the command **Dynamics** > **Convert Particle Cache** > **Particles to Curves** that generates a similar output to that of the **Particles to Mesh** command, however the resulting paths are curves that can be rendered using the **Render Curves** option of the Mesh Item. When combined with gradients controlling thickness and color, and the addition of displacements, an entire world of creative possibilities opens up.

Tip: The menu option **Dynamics > Convert Particle Cache** can be used to convert a simulation into a collection of curves that can be edited individually and converted back to a simulation.

Particles and Simulations

A particle, individually, is just a single vertex, a position in 3D space. But unlike a vertex, you don't create or move particles directly. For simulation's sake, a particle is indirectly given motion by the application of virtual forces acting upon it, and this is where the simulation part comes in. Forces are evaluated, per frame, producing a motion that is a simulation of how an element might react in the real world when placed under similar influence.

The most common force acting on a given particle is gravity, which compels it downward. Another is wind, compelling it forward. Combinations of multiple forces can be applied, and simulated to produce a more dynamic result.

Animating a single element in this way doesn't offer much of a compelling benefit; the true strength of a particle system lies in collections of many, many particles reacting in concert, all moving in relation to one another and reacting to other elements within a given scene. This collection of particles can be used to easily simulate simple phenomena like water dripping from a faucet, to something much more complex, like fireworks or dust kicking up from a spinning wheel.

The Modo particle simulation system uses the same global evaluation engine as it does for dynamics, therefore, a lot of the same workflows can be applied to both setups. Due to their deep integration inside Modo, they can also work together, with particle elements affecting dynamic items and dynamic items affecting particle elements. This overview page focuses on particles specifically. For more information on working with dynamics, see Dynamics.

Tip: In the **Preset** shelf (press **F6**) under **Assets** > **Assemblies** > **Particles** you can find a number of useful assemblies for particle rigging.

Simulation Basics

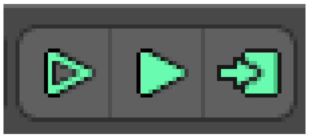
While you don't directly control individual particles, you can directly specify where and how the particles are generated using what is called an emitter. Emitters are the first element in the simulation chain, they generate the particles. Once a particle is born, it is instantly acted upon by forces, the next element in the chain. Forces largely control the motion of the particles. The third element in the chain is optional, but can be a very powerful addition: modifiers provide additional specific functionality, such as evaluating inter-particle interactions. Adding collisions allows particles to interact and collide with dynamic elements. The combination of all these various elements is collectively called a particle system, also often referred to simply as a particle simulation.

Modo provides a robust and flexible node-based particle system, where each element (called a node) is connected in an evaluation chain that is used to calculate the final result. These node setups (called graphs) are created and modified in the Schematic Viewport. For most simple setups though, the connections of particle nodes and modifiers are all done automatically in the background when creating items with the commands found in the **Particles** toolbox, a sub-tab of the **Setup** interface. At any time, you can drag and drop the simulation elements from the **Items** list into the **Schematic** viewport to see how they are connected, and continue building the graph network from there, to create additional complexity.

You can view the particles directly in the 3D viewport using the simulation transport buttons (covered further below). Pressing either of these puts the particles in motion using the current settings. You can modify those settings and get real-time feedback of the changes, allowing you to easily see how different settings affect the outcome. Once a desired look is achieved, the particles can be cached for final rendering, as particles cannot be rendered until this step. Caching them essentially sets the particles into their current state that remains a part of the saved scene (similarly to baking animation frames). The actual rendering of the simulated particles is done by assigning an effect that renders in place of the particle positions. This can be a blob, volume or sprite, each having its own unique look and controls. Additionally, you can assign any geometry element to particles using Replicators.

Workflow

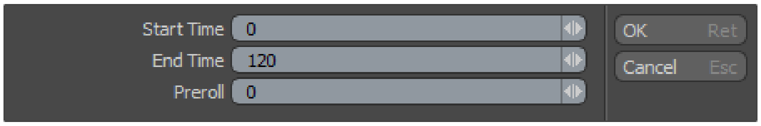
Starting a new simulation only requires the creation of Particle Emitters. This can be done using the particle creation tools found in the **Setup** interface layout, in the **Particles** sub-tab. The most common emitter type is the radial emitter, a very flexible emitter that can be used in a variety of situations. Clicking the toolbox button of the same name creates an emitter in the scene and also additionally adds a simulation item to the items list. These two elements work together to produce a basic particle simulation. The emitter item generates the actual particles based on its attributes, such as the number of particles and their initial speed at birth. The simulation item adds the universal gravity force that governs the particles motion after birth. It also has settings that define the resolution of the simulation and hold the final cached particle data for rendering.



Realtime preview in the 3D viewport is handled by the simulation transport controls found below the main 3D viewport. The two left buttons (seen above) provide continuous, real-time simulation directly in the 3D viewports. This allows you to adjust settings and move objects and see the changes happen live, directly in the 3D viewport. The **Preview at Current Time** button (hollow arrow) specifically plays the simulation at the current Timeline position, ignoring keyframed animation. The **Preview Animation** button (solid arrow) cycles through playing the Timeline with all keyframed animations. When either button is pressed, the buttons themselves change to **Stop** buttons which can then be pressed to stop the simulation playback. Unless the particles are meant to interact with dynamic items or their is animated channels on the particle items, the first (left) button is most often used.

Click the **Simulation Run** button to see the particles moving in the 3D viewport. By selecting the Emitter item in the **Items** list, you can adjust its attributes in the **Properties** viewport panel and see how the new settings affect the generation of the particles in real-time. Adjusting the number of particles or their initial velocity shows up instantly in the viewport. You can also adjust the emitter shape by controlling the **Cone Angle** and **Angle Spread** values. You can make further changes to the emitter by transforming it, changing its angle and position using the standard transform tools while in **Items mode**.

Once you have the settings you need, cache the simulation to the simulation item. To do this click **Compute Simulation** (the arrow pointing to the box). Clicking it opens a dialog.



Start Time 0
End Time 120
Preroll 0

By default, it reflects the scene length values for the **Start** and **End Times**. Adjust these if you want to have a simulation run longer or shorter than the scene itself. The **Start Time** indicates where the simulation begins to be evaluated. Particles don't start to emit until this frame. The **End Time** designates the end of the simulation evaluation. The **Preroll** value allows the particle to be emitted and active prior to the **Start Time** but the values are not cached until that defined frame. This is useful in cases where particles should already exist at the beginning of a sequence. Take, for instance fog in a scene: the **Preroll** option allows the fog to be visible and moving in the scene at the start frame, otherwise the fog would have to roll-in and build up in the scene first, which might not be the desired result.

While particles are visible as points in the 3D viewports, by themselves, they don't directly render. In order to actually see particles in a render, you need to assign them as point sources to an appropriate effect, such as Replicators, Blobs, Volumes, or Sprites. The effect chosen depends greatly on the desired result. Replicators can be used to assign geometry to the particles for effects like bullets firing and shell casings flying. Blobs can be applied to particles with dynamic fluids for liquid type effects. Fire and smoke are perfect candidates for volumes, especially when combined with Shader Tree textures controlling their color and density. Sprites are useful for image-based particles effects, such as falling ashes after an explosion. Each different effect has a **Particle Source** option where the **Particle Simulation** item can be designated. Remember that the simulation must be cached using the appropriate simulation control for the render to actually see the particle.

Particle Simulation

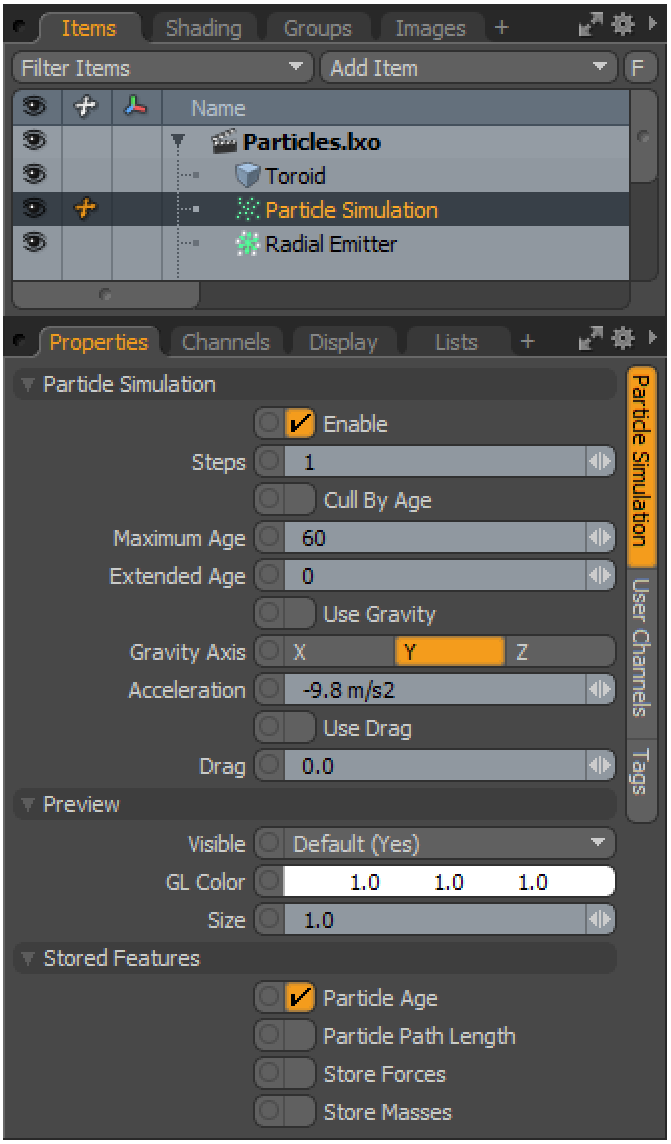
Every particle simulation in Modo requires a particle simulation item, which acts as the manager in the simulation and can control any number of associated emitters. The simulation item is also where Modo stores the cached values when the simulation is run. The cached values are then used for future playback and rendering. There can be multiple simulation items in a single scene, and each is evaluated simultaneously when a simulation is invoked, using any of the simulation controls.

Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface toolbox. You can add a particle simulation item by clicking the **Particle Simulation** button. Alternatively, in the Items list, click **Add Items** > **Particles** > **Simulation** > **Particle Simulation**.

At minimum, a single emitter and a particle simulation item are required for a basic particle simulation. Adding any emitter item to a scene when a particle simulation item is not present adds one automatically and associate them together. While different particle simulation elements and controls can be connected together manually in the Schematic Viewport, selecting the simulation item when adding items automatically connects the elements in a basic simulations, providing the associations necessary for proper evaluation.

You can modify and tweak the values of the simulation prior to actually caching the evaluation for rendering and playback. To do this, in the **Setup** interface, click the **Live Simulation** playback button. For more information on using the particle simulation tools, see Particles and Simulations. When the particle simulation item is selected, the following attributes can be edited in the **Properties** viewport.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Radial Emitter
Properties
Channels
Display
Lists
Particle Simulation
User Channels
Tags

Particle Simulation

☑ Enable

Steps 1

☐ Cull By Age

Maximum Age 60

Extended Age 0

☐ Use Gravity

Gravity Axis X [✓ Y] Z

Acceleration -9.8 m/s2

☐ Use Drag

Drag 0.0

Preview

Visible Default (Yes)

GL Color 1.0 1.0 1.0

Size 1.0

Stored Features

☑ Particle Age

☐ Particle Path Length

☐ Store Forces

☐ Store Masses

Option	Description
Particle Simulation	
Enable	Turns the simulation on or off. When the simulation is disabled, its settings don't produce any results.
Steps	Determines the number of evaluations calculated per frame. The value can be increased up to a maximum of 50 steps to increase the quality of a simulation. The default value of **1** is fine for most basic simulations, but increasing this value allows for more accurate results when dealing with collisions and fast-moving particles. The additional steps are like mini-frames, as the steps are computed sequentially. For

Option	Description
	example, if the **Steps** value is increased to 5, five full evaluations are computed as if they were full frames, but only the results of the fifth frame are actually used. **Note:** Increasing this value slows down the simulation processing, sometimes significantly. To keep the 3D viewport simulation preview speeds fast, the **Steps** value is always fixed at a value of **1**, so when raising this value, you need to create a simulation cache to actually see the results.
Cull By Age	When enabled, it kills any particle that reaches the maximum age value. When disabled, particles live indefinitely, unless affected by a terminator.
Maximum Age	Determines the length of time a particle is active when the **Cull by Age** option is enabled. The format of the time input is determined by the **Preferences** > **Animation** > **Time Format** setting. Common units are **Frames** and **Seconds**.
Extended Age	Randomly extends the **Maximum Age** value up to the defined **Extended Age** value.
Use Gravity	Toggles the effect of gravity on the simulated particles on or off. When enabled, the gravity force affects all related particles associated with the simulation item with a directional pull determined by the **Gravity Axis** and **Acceleration** values.
Gravity Axis	The X, Y or Z selection options determine the axis direction for gravity when the **Use Gravity** option is enabled.
Acceleration	The default value of **-9.8** simulates the downward pull of real-world gravity on Earth at a 1:1 scale (defined as meters per second). Negative values pull in the negative direction of the defined axis and positive values pull in the positive direction.
Use Drag	Toggles the effect of drag on the simulated particles on or off. When enabled, the drag force affects all related particles associated with the simulation item by decreasing the amount of velocity per particle, slowing them down over time. The amount of velocity drag is determined by the **Drag** value.
Drag	Determines the strength of the drag force applied to particles when the **Use Drag** option is enabled.
Preview	
Visible	Determines the visibility of the simulated particles in the 3D GL viewport. The **Default**

Option	Description
	option means that the particles' visibility is determined by the viewport settings, **Yes** means they are always visible and **No** they are never visible. **No (Hide Children)** hides visibility of all related child items.
GL Color	Allows you to define the display color of the simulated particles in the 3D viewport. This option is very helpful in differentiating particles that are emitted from other particles.
Size	Determines the size, in pixels, of the particle representations in the 3D viewport.
Stored Features	Enables the caching of certain extra particle-specific information. This information can be later applied as special shading effects, such as using a gradient to assign color to a volumetric, based on the particles' age. The toggles must be enabled when the simulation is created for the information to be stored and made accessible. By default the particle position and velocity values are always stored.
Particle Age	Stores the overall age of the particle since birth on a per frame basis.
Particle Path Length	Stores the overall length of the particle's path since birth on a per frame basis.
Store Forces	Stores the current force value for the particle on a per frame basis.
Store Masses	Stores the current mass value for the particle on a per frame basis

Particle Emitters

Emitters are the elements that generate the particles in a simulation based on the settings of each individual emitter item. For the simplest simulation, a radial emitter and a particle simulation item are needed. While this creates particles, in order for the particles to actually render, you need to then create and additional renderable item that uses the particles as a source, such as Blobs, Volumes or Sprites. Geometry can also be applied to particles when utilizing Replicators, providing for a lot of interesting effects.

Radial Emitter

The radial emitter is the go-to particle generator for many different types of simulations. It is a single point, defined by the emitter's position, that emits the particles outward based on the emission option settings. It can act like a point emitter, radiating particles in all directions randomly, or act like a cone emitter with ordered particles streaming in a single direction. Using the **Offset** option allows for spherical area and disc shaped emitters. You can apply additional forces and modifiers to further affect and control the particles.

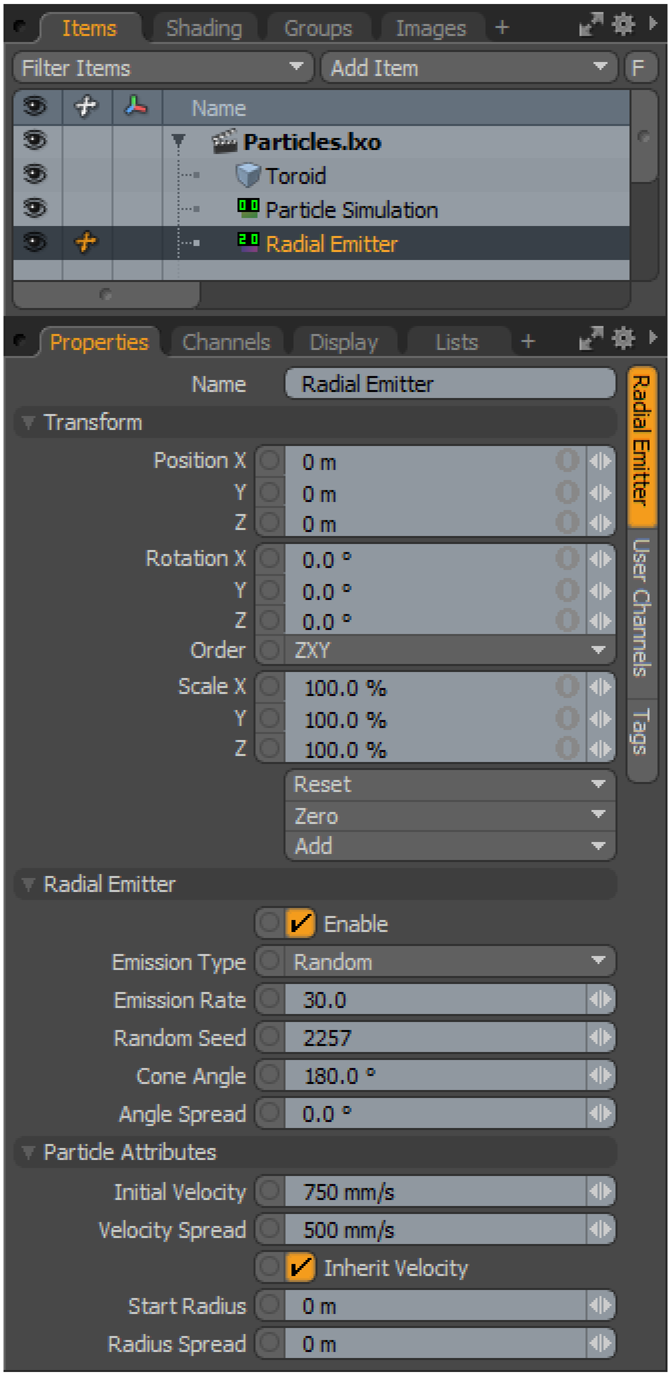
Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface toolbox. Add a radial emitter item by clicking the **Radial Emitter** button. Alternatively, in the **Items** list, click **Add Items** > **Particles** > **Simulation** > **Radial Emitter**.

At minimum, a single emitter and a particle simulation item are required for a basic particle simulation. Adding any emitter item to a scene when a particle simulation item is not present adds one automatically and associates them together. While different particle simulation elements and controls can be connected together manually in the Schematic Viewport, selecting the simulation item when adding items automatically connects the elements in a basic simulations, providing the associations necessary for proper evaluation.

You can modify the values of the simulation before actually caching the evaluation for rendering and playback. To do this, in the **Setup** interface, click the **Live Simulation** playback button. For more information on using the particle simulation tools, see Particles and Simulations.

When the radial emitter item is selected, the following attributes can be edited in the **Properties** viewport.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Radial Emitter
Properties
Channels
Display
Lists
Radial Emitter
User Channels
Tags
Transform
Reset
Zero
Add
Radial Emitter
Particle Attributes

Name: Radial Emitter

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable
Emission Type: Random
Emission Rate: 30.0
Random Seed: 2257
Cone Angle: 180.0 °
Angle Spread: 0.0 °

Initial Velocity: 750 mm/s
Velocity Spread: 500 mm/s
☑ Inherit Velocity
Start Radius: 0 m
Radius Spread: 0 m

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Transform	

Option	Description
Position	An item transform that allows you to numerically position the emitter item in XYZ space.
Rotation	An item transform that allows you to numerically set the rotation of the emitter item. Rotation transforms originate from the **Position** value. **Note:** Changing the angle of the rotation doesn't produce noticeable results unless the **Cone** angle of the emitter is below 180°.
Order	Sets the order that rotations are applied to the emitter item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the emitter item. Scale transforms originate from the **Position** value. Since particles themselves have no inherent size, the scale transform acts to scale the emission values, acting as a multiplier on the particle attributes.
Reset	Resets the selected transform values to **(0,0,0)** returning the items back to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the **Center** position intact. This is done by adding a negative transform item to the Mesh Item's channels.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the value's input field). This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function can also be used to add the selected set of transforms to the **Channels** list while keeping the default **0,0,0** values (a necessary step for referencing, in order to override the channels, they must first exist).
Radial Emitter	
Enable	Turns the generation of particles from the emitter on or off. When the emitter is disabled, its settings don't produce any results.

Option	Description
Emission Type	Determines how the particles themselves are emitted over time: **Random** means that statistically the emission rate is accurate, but the time between individual particle emissions is irregular. **Uniform** produces regular, evenly-spaced individual particle emissions.
Emission Rate	Controls the number of particles emitted per second.
Random Seed	The initial number used when generating the random emission values. Each seed produces different random results. You need to change this when there are multiple emitters in a scene with identical settings.
Cone Angle	Determines the overall emission range outward from the central point, generating a particle and emitting it in a random direction within the specified range. When set to 180°, the particles radiate outward equally in all directions, like a point emitter. When set to less that 180°, the particles radiate only within the specified range. Setting the value to less than 90° produces a cone-shaped emission, similar to a water hose.
Angle Spread	Allows you to define a soft falloff range on the outside edge of the defined **Cone Angle**, where the probability of emission is reduced, attenuating over the specified range.
Particle Attributes	
Initial Velocity	Determines the speed the particle is traveling at birth. Particles remain the same speed until affected by a force or a collision. Gravity and drag are two forces that can be generated by the particle simulation item automatically.
Velocity Spread	Randomly varies the speed of individual particles within the specified value, making the particles appear more varied and natural.
Inherit Velocity	When enabled, the individual particles inherit the speed of the emitter item, if it is in motion, adding its speed to the **Initial Velocity'**. When disabled, the particles ignore the movement of the emitter item.
Start Radius	Offsets the location where the particles generate from, producing a larger area of emission. When set to 0, all particles originate from the same location. When the **Start Radius** is greater than 0, particles originate randomly across the disc, created from the offset.
Radius Spread	Varies the **Start Radius** value randomly within this range, plus or minus, producing a volume within which the particles are generated.

Surface Emitter

The surface emitter generates particles outward from any polygonal surface. Individual particles are emitted randomly from anywhere over the defined surface, perpendicular to the normal facing direction at the location of emission. You can apply additional forces and modifiers to further affect and control the particles.

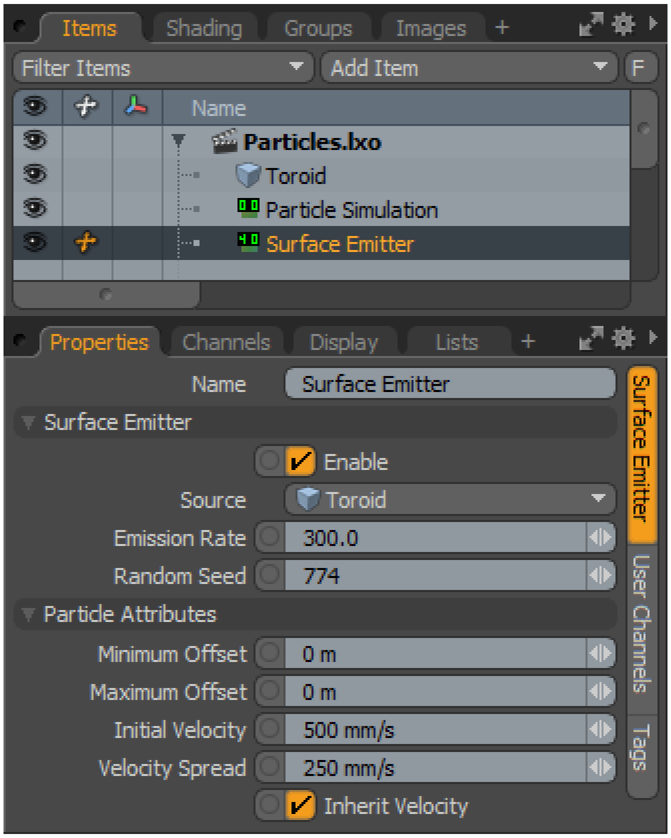
Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. Before adding a surface emitter, you need to create a polygonal Mesh Item to act as the emission source. Add a surface emitter item by clicking the **Surface Emitter** button. Alternatively, in the **Items** list, click **Add Items** > **Particles** > **Simulation** > **Surface Emitter**.

At minimum, a single emitter and a particle simulation item are required for a basic particle simulation. Adding any emitter item to a scene when a particle simulation item is not present adds one automatically and associates them together. While different particle simulation elements and controls can be connected together manually in the Schematic Viewport, selecting the simulation item when adding items automatically connects the elements in a basic simulations, providing the associations necessary for proper evaluation.

Once the surface emitter item is defined, you need to specify the source, which can be any Mesh Item layer that contains at least one polygon. Once defined, you can modify the values of the simulation before actually caching the evaluation for rendering and playback. To do this, in the **Setup** interface, click the **Live Simulation** playback button. For more information on using the particle simulation tools, see Particles and Simulations.

When the surface emitter item is selected, the following attributes can be edited in the **Properties** viewport.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Surface Emitter
Properties
Channels
Display
Lists
Surface Emitter
User Channels
Tags

Name: Surface Emitter

Surface Emitter

☑ Enable

Source: Toroid

Emission Rate: 300.0

Random Seed: 774

Particle Attributes

Minimum Offset: 0 m

Maximum Offset: 0 m

Initial Velocity: 500 mm/s

Velocity Spread: 250 mm/s

☑ Inherit Velocity

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Surface Emitter	
Enable	Turns the generation of particles from the emitter on or off. When the emitter is disabled, its settings don't produce any results.
Source	Allows you to define the specific layer that contains the polygons for particle emission. The source item can be standard polygons, or either type of subdivision surface geometry.
Emission Rate	Controls the number of particles emitted per second.
Random Seed	The initial number used when generating the random emission values. Each seed produces different random results. You need to change this when there are multiple emitters in a scene with identical settings and sources.
Particle Attributes	

Option	Description
Minimum Offset	Allows you to define an offset distance away form the surface, essentially a volume area where particles generate from.
Maximum Offset	Allows you to define a larger volume where particles are less likely to generate, attenuating from the **Minimum Offset** to the defined **Maximum Offset**. **Note:** This value should always be higher than the **Minimum Offset**.
Initial Velocity	Determines the speed the particle is traveling at birth. Particles remain the same speed until affected by a force or a collision. Gravity and drag are two forces that can be generated by the particle simulation item automatically.
Velocity Spread	Randomly varies the speed of individual particles within the specified value, making the particles appear more varied and natural.
Inherit Velocity	When enabled, the individual particles inherit the speed of the emitter item, if it is in motion, adding its speed to the **Initial Velocity**. When disabled, the particles ignore the movement of the emitter item.

Curve Emitter

The curve emitter generates particles outward from any Spline or Bezier curve element. Individual particles are emitted randomly along the length outward from the curve itself, perpendicular to the location of emission. By default, the particle emission strength is statistically even along the curve, based on the **Emission Rate**, unless the distribution value is modified, varying the emission amounts. You can apply additional forces and modifiers to further affect and control the particles.

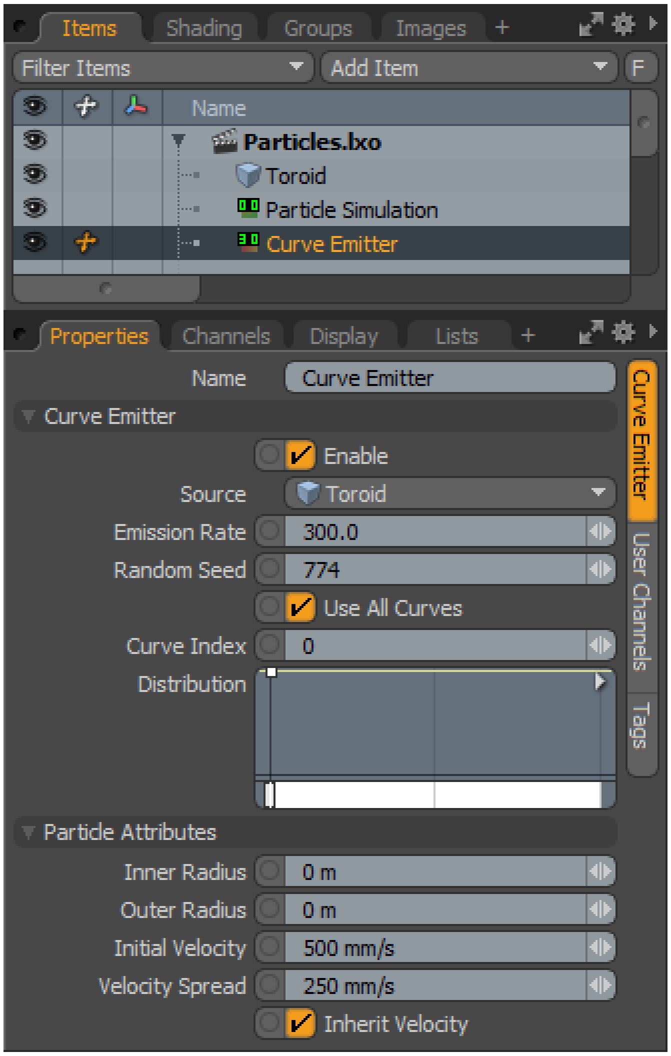
Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. Before adding a curve emitter, you need to create a Spline or Bezier type curve to act as the emission source. Add a curve emitter item by clicking the **Curve Emitter** button. Alternatively, in the **Items** list, click **Add Items** > **Particles** > **Simulation** > **Curve Emitter**.

At minimum, a single emitter and a particle simulation item are required for a basic particle simulation. Adding any emitter item to a scene when a particle simulation item is not present adds one automatically and associates them together. While different particle simulation elements and controls can be connected together manually in the Schematic Viewport, selecting the simulation item when adding items automatically connects the elements in a basic simulations, providing the associations necessary for proper evaluation.

Once the curve emitter item is defined, you need to specify the source, which can be any Mesh Item layer that contains one or more curves. Once defined, you can modify the values of the simulation before actually caching the evaluation for rendering and playback. To do this, in the **Setup** interface, click the **Live Simulation** playback button. For more information on using the particle simulation tools, see Particles and Simulations.

When the curve emitter item is selected, the following attributes can be edited in the **Properties** viewport.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Curve Emitter
Properties
Channels
Display
Lists
Curve Emitter
User Channels
Tags
Curve Emitter
Particle Attributes

Name: Curve Emitter

☑ Enable

Source: Toroid

Emission Rate: 300.0

Random Seed: 774

☑ Use All Curves

Curve Index: 0

Distribution

Inner Radius: 0 m

Outer Radius: 0 m

Initial Velocity: 500 mm/s

Velocity Spread: 250 mm/s

☑ Inherit Velocity

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.

Option	Description
Curve Emitter	
Enable	Turns the generation of particles from the emitter on or off. When the emitter is disabled, its settings don't produce any results.
Source	Allows you to define the specific layer that contains the curves for particle emission. The target layer can contain only a single curve, multiple curves, or both curve and polygonal data; only the curves themselves act as emitters in that case.
Emission Rate	Controls the number of particles emitted per second.
Random Seed	The initial number used when generating the random emission values. Each seed produces different random results. You need to change this when there are multiple emitters in a scene with identical settings and sources.
Use All Curves	When enabled, all the curves within the specified source item act as emission sources, either Spline or Bezier. When disabled, you can specify a curve within the layer to act as the emission source with the **Curve Index** value.
Curve Index	Allows you to specify a particular curve within a layer with more than one curve element (Spline or Bezier) based on its index value, a numeric assignment generated in the order the curves were originally drawn in the layer.
Distribution	Allows you to modify the emission strength of the particles along the curves length. Using a mini-gradient input, the emission is modified with gradient keys that represent the statistical likelihood that a particle is generated at any location along the curve. Much like how a keyframe defines a value in time, these keys represent a value along the length allowing for the variation of emission strength. The gradient input's width represents the individual curve emitter's overall length, with the base on the left side. You can add keys by clicking along the length of the gradient input and then clicking and dragging the key icon up or down to adjust the value. The grayscale bar directly below gives visual feedback as to how the values fade from one key to the next, black produces no particle emission and white produces 100% emission strength, with the shades of gray attenuating the strength in between.
Particle Attributes	
Inner Radius	Allows you to define a cylindrical volume around the curve from where particles generate from.
Outer Radius	Allows you to define a larger volume where particles are less likely to generate,

Option	Description
	attenuating from the **Inner Radius** to the defined **Outer Radius**. **Note:** This value should always be higher than the **Inner Radius**.
Initial Velocity	Determines the speed the particle is traveling at birth. Particles remain the same speed until affected by a force or a collision. Gravity and drag are two forces that can be generated by the particle simulation item automatically.
Velocity Spread	Randomly varies the speed of individual particles within the specified value, making the particles appear more varied and natural.
Inherit Velocity	When enabled, the individual particles inherit the speed of the emitter item, if it is in motion, adding its speed to the **Initial Velocity**. When disabled, the particles ignore the movement of the emitter item.

Source Emitter

The source emitter generates particles outward from other particles. These can be active particle simulations, static particle clouds or even the vertices of a standard Mesh Item. This makes it easy to create fireworks, for instance, where one particle is spawning more particles, leaving a trail behind. You can apply additional forces and modifiers to further affect and control the particles.

Usage

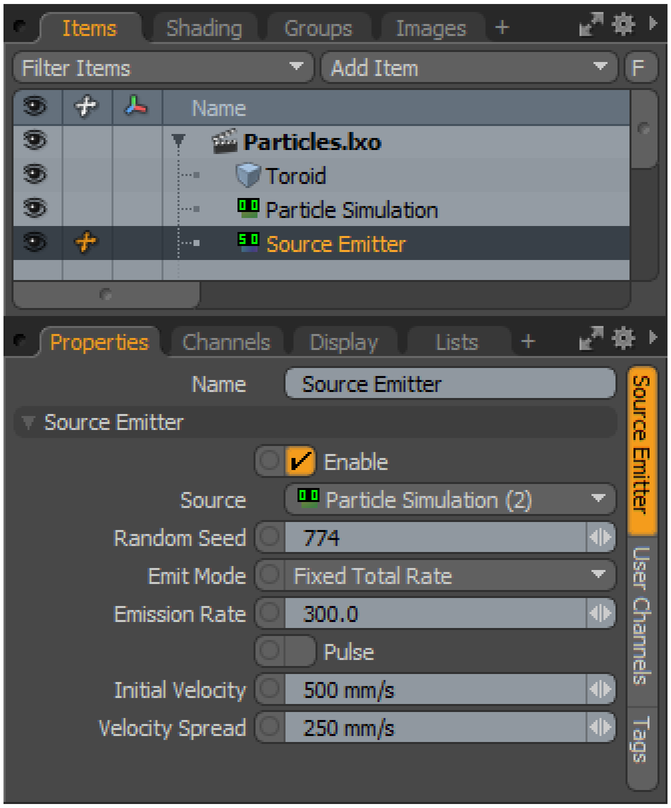
The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox.

1. Before adding a source emitter, you need to create some type of a particle item, typically, another particle simulation and emitter combination.
2. Next, add an additional particle simulation item, separating the settings of the two simulations.
3. With the new particle simulation item selected, add a source emitter item by clicking the **Source Emitter** button. Alternatively, in the **Items** list, click **Add Items** > **Particles** > **Simulation** > **Source Emitter**.

   This ensures the source emitter is connected to the proper particle simulation item.

Once the source emitter item is defined, you need to specify the source, which can be any any particle item, static or dynamic, or if polygonal data is specified, the vertices of the target geometry act as the emission points. Once defined, you can modify the values of the simulation before actually caching the evaluation for rendering and playback. To do this, in the **Setup** interface, click the **Live Simulation** playback button. For more information on using the particle simulation tools, see Particles and Simulations.

When the source emitter item is selected, the following attributes can be edited in the **Properties** viewport.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Source Emitter
Properties
Channels
Display
Lists
Source Emitter
User Channels
Tags

Name: Source Emitter

Source Emitter

☑ Enable

Source: Particle Simulation (2)

Random Seed: 774

Emit Mode: Fixed Total Rate

Emission Rate: 300.0

☐ Pulse

Initial Velocity: 500 mm/s

Velocity Spread: 250 mm/s

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Source Emitter	
Enable	Turns the generation of particles from the emitter on or off. When the emitter is disabled, its settings don't produce any results.
Source	Allows you to define the specific source target item. The source is usually another particle simulation item, where the positions of its particles spawn the particles in the new simulation. It can also be a static particle object, such as a particle cloud, or it can simply be a Mesh Item, where the vertices of the geometry act as emission sources.
Random Seed	The initial number used when generating the random emission values. Each seed produces different random results. You need to change this when there are multiple emitters in a scene with identical settings and sources.
Emit Mode	Controls how the particles are emitted. The following options are available:

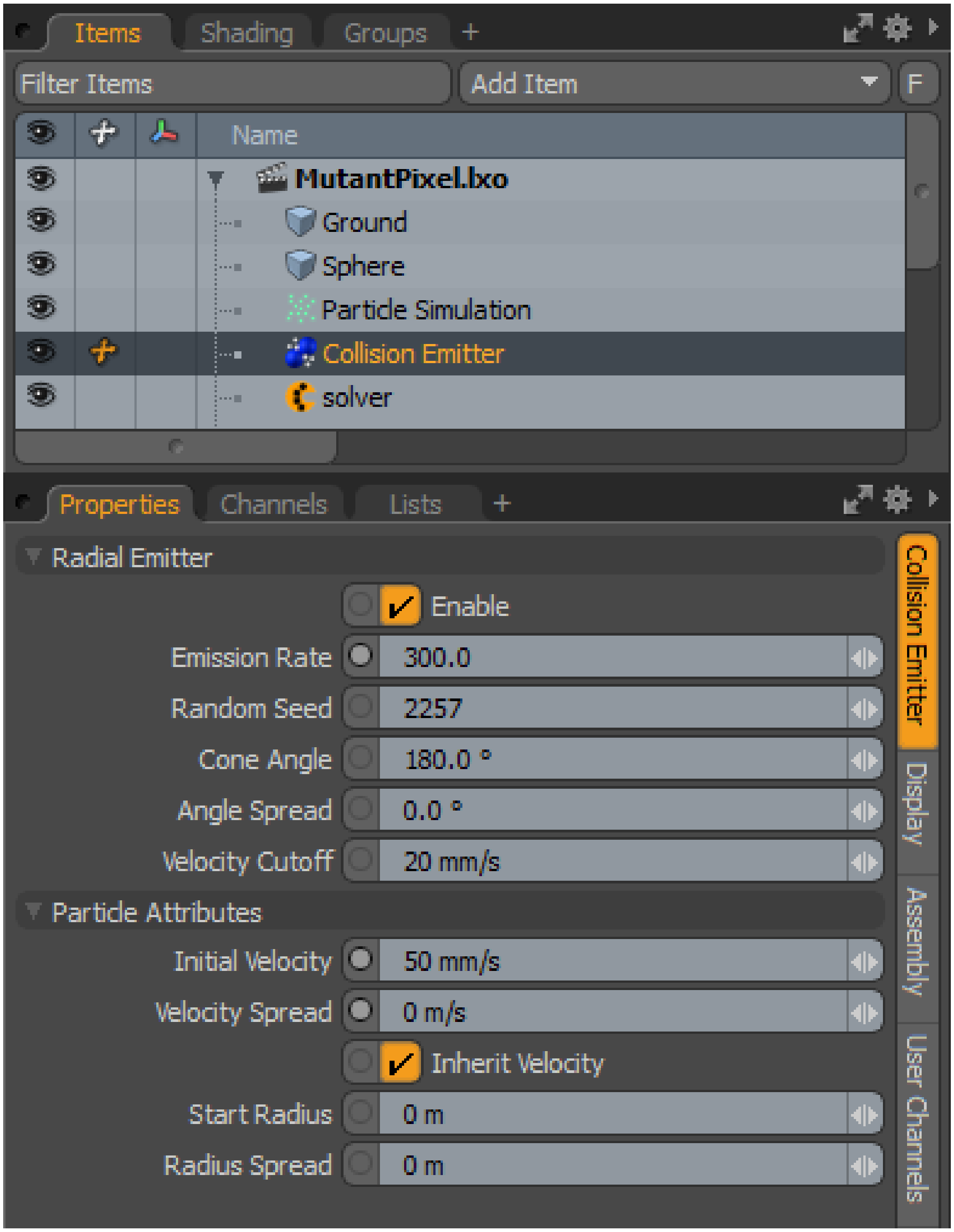
Option	Description
	• **Fixed Total Rate**: Uses a single combined emission rate for all sources. • **Fixed Average Rate**: Uses a single emission rate for source particles individually. • **Per Particle Rate**: Reads the particle creation from the **Rate** particle feature. • **Pulse**: Emits one particle for every source particle when **Pulse** is enabled by a keyframe or trigger.
Emission Rate	Controls the number of particles emitted per second.
Use All Curves	When enabled, all the curves within the specified source item act as emission sources, either Spline or Bezier. When disabled, you can specify a curve within the layer to act as the emission source with the **Curve Index** value.
Curve Index	Allows you to specify a particular curve within a layer with more than one curve element (Spline or Bezier) based on its index value, a numeric assignment generated in the order the curves were originally drawn in the layer.
Distribution	Allows you to modify the emission strength of the particles along the curves length. Using a mini-gradient input, the emission is modified with gradient keys that represent the statistical likelihood that a particle is generated at any location along the curve. Much like how a keyframe defines a value in time, these keys represent a value along the length allowing for the variation of emission strength. The gradient input's width represents the individual curve emitter's overall length, with the base on the left side. You can add keys by clicking along the length of the gradient input and then clicking and dragging the key icon up or down to adjust the value. The grayscale bar directly below gives visual feedback as to how the values fade from one key to the next, black produces no particle emission and white produces 100% emission strength, with the shades of gray attenuating the strength in between.
Pulse	When the **Emit Mode** is set to the **Pulse** option, enabling this toggle by a trigger or a keyframe causes the emitter to emit the particles.
Initial Velocity	Determines the speed the particle is traveling at birth. Particles remain the same speed until affected by a force or a collision. Gravity and drag are two forces that can be generated by the particle simulation item automatically.
Velocity Spread	Randomly varies the speed of individual particles within the specified value, making the particles appear more varied and natural.

Collision Emitter

The collision emitter is used to emit particles into a particle simulation whenever a collision event happens between two or more dynamics bodies. The emitter works the same basic way as the radial emitter, sharing many of the same settings, but the emission position and direction are defined by the collision of a vertex with another dynamics surface.

Usage

With two or more dynamic elements defined as such and selected in the **Items** list, press the **Emit on Collision** button. This adds in the **Collision Emitter** item (automatically connecting them correctly in the **Schematic** view). Now when a simulation is evaluated or cached, particles are emitted at any collision event position between the dynamic items that were selected when the **Emit on Collision** item was added. To add additional items after the fact to the collision emitter, any dynamic item can be added to the Group Item associated with the collision emitter. A **Dynamic Collider** item can be added as well to have the resulting particles collide and otherwise interacts with the dynamic elements. The emitted particles are controlled by the settings of the collision emitter. When selected, it displays the following attributes in the **Properties** panel.


Items
Shading
Groups
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Ground
Sphere
Particle Simulation
Collision Emitter
solver
Properties
Channels
Lists
+
Collision Emitter
Display
Assembly
User Channels

Radial Emitter

☑ Enable

Emission Rate 300.0

Random Seed 2257

Cone Angle 180.0 °

Angle Spread 0.0 °

Velocity Cutoff 20 mm/s

Particle Attributes

Initial Velocity 50 mm/s

Velocity Spread 0 m/s

☑ Inherit Velocity

Start Radius 0 m

Radius Spread 0 m

Option	Description
Enable	Turns the generation of particles from the emitter on or off. When the emitter is disabled, its settings don't produce any results.
Emission Rate	Controls the number of particles emitted per second per collision.

Option	Description
Random Seed	The initial number used when generating the random emission values. Each seed produces different random results. You need to change this when there are multiple emitters in a scene with identical settings.
Cone Angle	Determines the overall emission range outward from each central emission point, generating a particle and emitting it in a random direction within the specified range. When set to 180°, the particles radiate outward equally in all directions, like a point emitter. When set to less than 180°, the particles radiate only within the specified range. Setting the value to less than 90° produces a cone-shaped emission, similar to a water hose.
Angle Spread	Allows you to define a soft falloff range on the outside edge of the defined **Cone Angle**, where the probability of emission is reduced, attenuating over the specified range.
Velocity Cutoff	Defines a lower relative velocity limit where colliding items don't cause an emission.
Particle Attributes	
Initial Velocity	Determines the speed the particle is traveling at birth. Particles remain the same speed until affected by a force or a collision. Gravity and drag are two forces that can be generated by the particle simulation item automatically.
Velocity Spread	Randomly varies the speed of individual particles within the specified value, making the particles appear more varied and natural.
Inherit Velocity	When enabled, the individual particles inherit the speed of the emitter item, if it is in motion, adding its speed to the **Initial Velocity**. When disabled, the particles ignore the movement of the emitter item.
Start Radius	Offsets the location where the particles generate from, producing a larger area of emission. When set to 0, all particles originate from the same location. When the **Start Radius** is greater than 0, particles originate randomly across the disc, created from the offset.
Radius Spread	Varies the **Start Radius** value randomly within this range, plus or minus, producing a volume within which the particles are generated.

Particle Operators

Once any particle is generated by an emitter, its motion is governed by forces. By default this is gravity, provided by the Particle Simulation item. While Dynamics Forces and falloffs can produce interesting effects, it is the particle operators that give you broad control over Particles in motion, providing effects for age, speed, collision and a multitude of other controllers. Flocking controls how particles move relative to its neighbors, dynamic fluid allows particles to cling closely together, simulating viscous fluids, while terminators cause particles to die within the bounds of a falloff. Really, operators can be seen as part of the building blocks that can be added to an overall simulation to provide a huge number of possibilities.

Particle Operator

The particle operator is a general filter/modifier-type item that can be used to process or modify all particles in a simulation based on your selected features. Its use of the **Schematic** view to nodally connect and alter any features makes it very powerful and extremely flexible. In a way, its use can be thought of as adding user channels, because adding features creates additional channels on the operator that can be linked to produce a specific result.

Usage

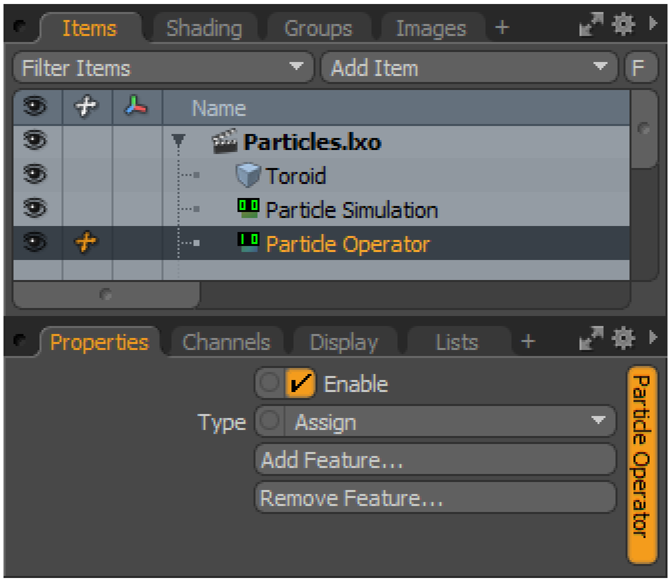
The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. Particle operators require a particle simulation on which to operate. Once a simulation is added, add the particle operator:

1. With a particle simulation selected in the **Items** list, click the **Particle Operator** button in the toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Particles** > **Simulation**, double-click **Particle Operator**.
2. With the created particle item selected in the **Items** list, click the **Add Feature** button found in the **Properties** panel.

   This opens a pop-up dialog that allows you to select the feature and additionally add the feature channel to the operator in the **Schematic** view, if it is already present there.
3. Add as many features are necessary for the intended result.

Since the particle operator is such an open-ended controller, it is largely up to you how you use it based on the feature assignment.

Video: This video demonstrates one use of the Operator.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Particle Operator
Properties
Channels
Display
Lists
Add Feature...
Remove Feature...
Particle Operator

☑ Enable

Type Assign

Option	Description
Enable	Toggles the particle operator item on or off. When enabled, the settings of the operator are evaluated for the associated particle simulation, when disabled, the settings are ignored.
Type	Determines how the feature results integrate with the original particle data values. There are several possible options: • **Assign**: Outputs the values of the operator directly to particle features for every step. • **Integrate**: Adds the outputs to particle features, effectively making them the rate of change. • **New**: Outputs are assigned only to new particles, allowing them to be initialized. • **Trigger**: Outputs are assigned only when the trigger channel is true.
Add/Remove Feature	Allows you to create only the pertinent controls based on the desired results. Each feature represents a specific aspect of an individual particle in a simulation. Each feature type also has two options, the regular feature and a read-only version, which differs in that it contains no input connection. These types of features reduce evaluation overhead, which can be important in complex simulations. It may be necessary to enable the storing of certain features in the simulation item, as not all are calculated by default.

Collector/Emitter

The collector/emitter item works as a co-processor to a particle operator item. When the signal channel on the collector is true, features of the current particle are collected. This serves as a set of new particles that are emitted into a second simulation. Basically, it takes particles from one simulation and moves them to another simulation. The features collected depend on the channels and their links to and from the particle operator and the collector/emitter. This is useful if, for instance, one wanted to change the items from one Replicator into the items of another separate Replicator based on a specified event.

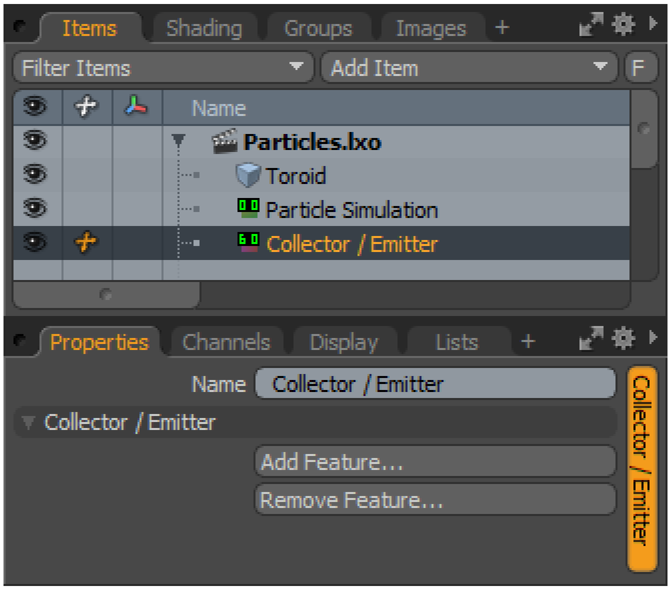
Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. Collector/emitter items require a particle simulation on which to operate. Once a simulation is added, add the collector/emitter:

1. With a particle simulation selected in the **Items** list, click the **Collector/Emitter** button in the toolbox. Alternatively, in the **Items** list, click **Add Item**, navigate to **Particles** > **Simulation**, and double-click **Collector/Emitter**.
2. With the created particle item selected in the **Items** list, click the **Add Feature** button found in the **Properties** panel.

   This opens a pop-up dialog that allows you to select the feature and additionally add the feature channel to the operator in the **Schematic** view, if it is already present there.
3. Add as many features as necessary for the intended result.

Since the collector/emitter is such an open-ended controller, it is largely up to you how you use it based on the feature assignment and schematic connections.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Collector / Emitter
Properties
Channels
Display
Lists
Add Feature...
Remove Feature...
Collector / Emitter

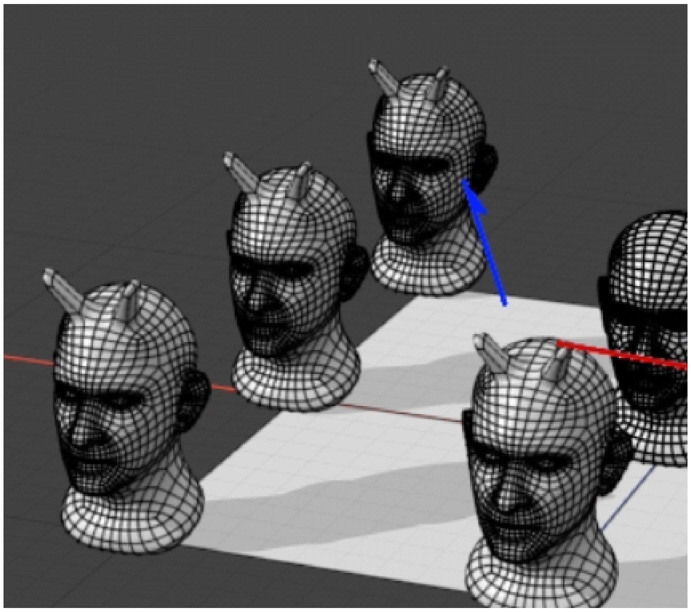
Name: Collector / Emitter

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Collector / Emitter	
Add Feature	Allows you to define custom channels (features) for the item that can then be used to drive other channels. Click the **Add Feature** button opens the **Feature** dialog, where you can select from the available features specific to particles and add them to the collector/emitter when pressing **OK**. Additionally, there is an option to automatically add the the selected feature as a channel to the collector/emitter to the **Schematic** viewport if it is already present.
Remove Feature	Deletes unnecessary feature channels that have been added to the collector/emitter.

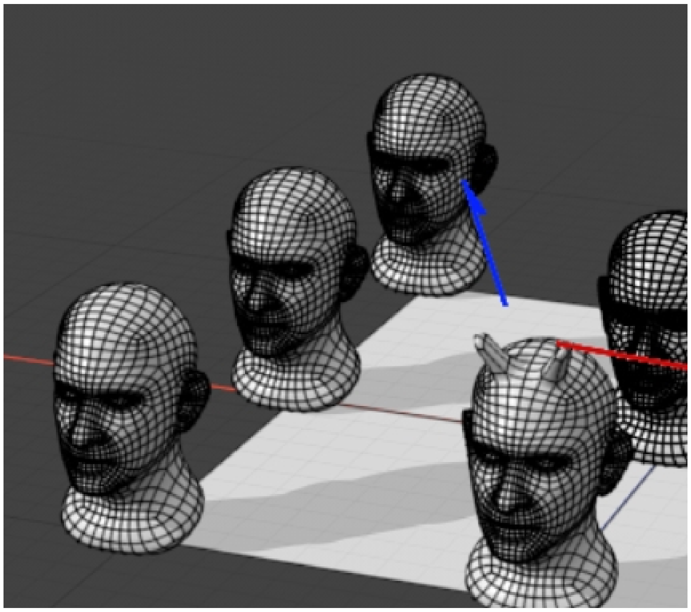
:

Duplicate Geometry Using Particle Clone

The Particle Clone operator creates clones of existing source geometry, which you can then affect with other operations such as falloff or even Mesh Ops on each individual clone, separately. Particle Clone is similar to using replicators, but with clones you have more control over the copies. Replicators also apply any operators that affect the geometry to the copies, whereas clones can be affected individually. In the example images, the bevel on the left affects all copies equally, whereas the cloned mesh is affected by the bevel individually.



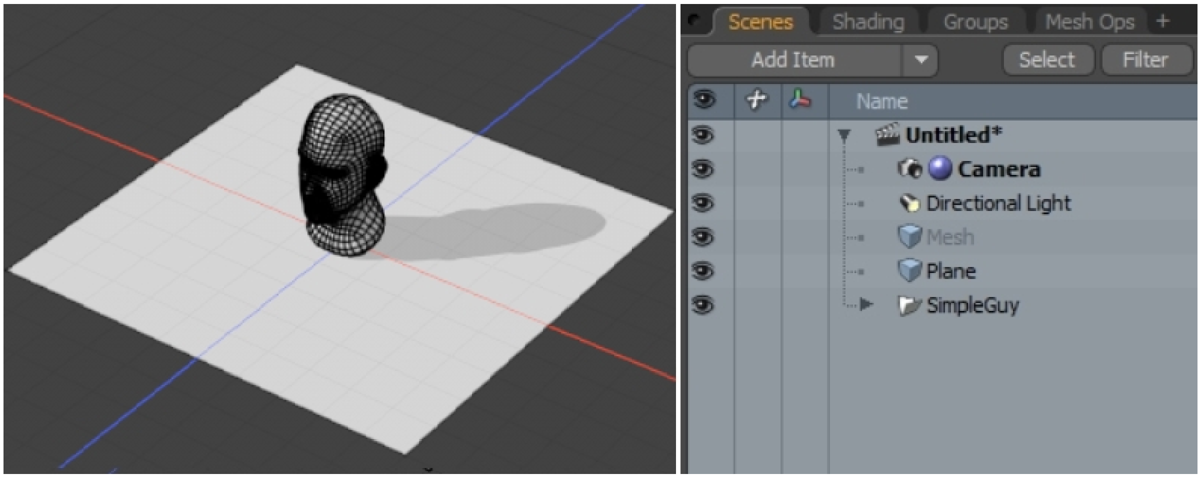
A bevel on replicated geometry copies the bevel to all replicas



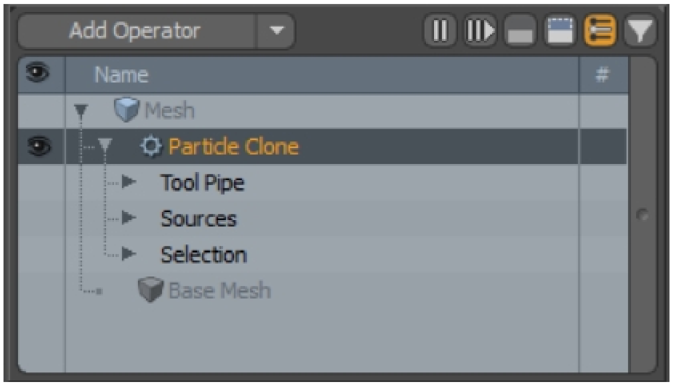
A bevel on cloned geometry only affects the selected clone

Cloning particle geometry can be used to create simulations where particles are affected by operators differently, such as simulations of crowds, but it is slower to render than replicators because of the complexity. We don't recommend using clones for simulations with millions of particles, such as clouds or snow, replicators are more efficient for that volume of particles.

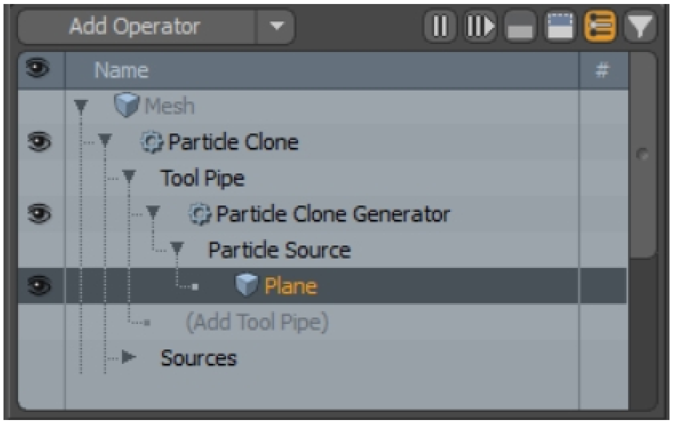
To use Particle Clone, you need some particle geometry and a particle source, in this case a simple figure and a plane:


Scenes
Shading
Groups
Mesh Ops
+
Add Item
Select
Filter
Name
Untitled*
Camera
Directional Light
Mesh
Plane
SimpleGuy

1. Select the default mesh in the scene or press **N** to create a new mesh.
2. Go to the **Mesh Ops** tab and click **Add Operator**.
3. Go to **Duplicate** > **Particle Clone** or use the search bar at the top of the **Add Mesh Operation** viewport to locate **Particle Clone**.
4. Double-click **Particle Clone** to add it to the **Mesh Ops** list.
5. Click the arrow to the left of **Particle Clone** to open its options.

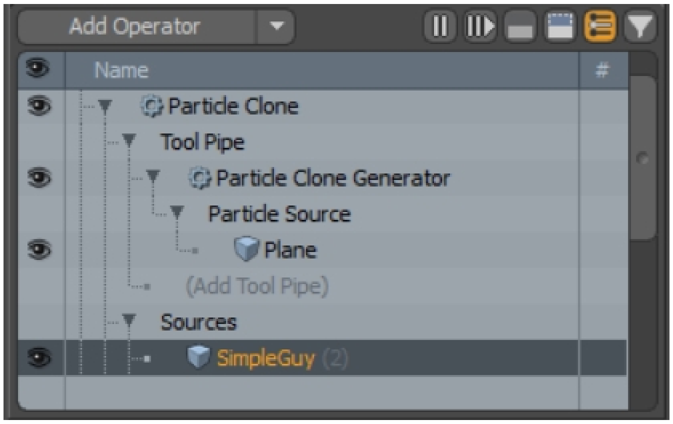

Add Operator
Name
#
Mesh
Particle Clone
Tool Pipe
Sources
Selection
Base Mesh

6. To select where the particles are generated from, drill down into **Tool Pipe** > **Particle Clone Generator** > **Particle Source** and click **Add Particle Source**.
7. Select the **Plane** from the list and double-click it to add the geometry to the **Tool Pipe**.

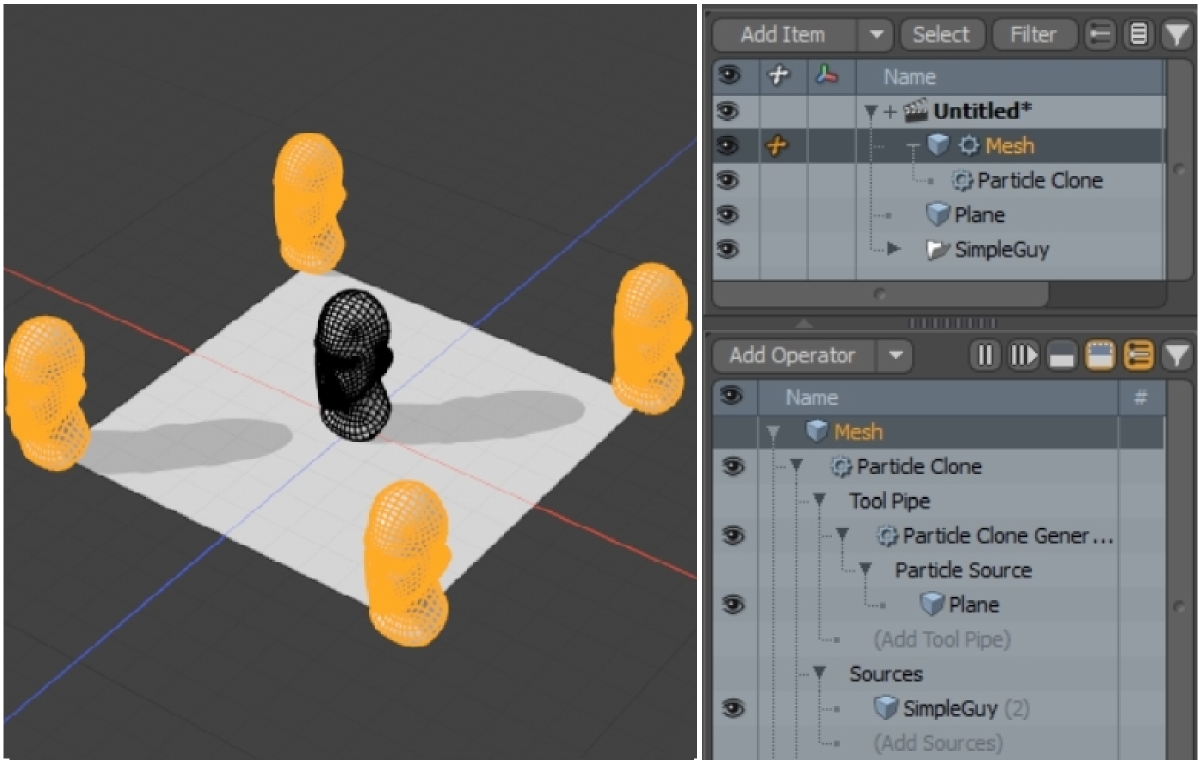

Add Operator
Name
#
Mesh
Particle Clone
Tool Pipe
Particle Clone Generator
Particle Source
Plane
(Add Tool Pipe)
Sources

8. To select the particles to clone, drill down into **Sources** and click **Add Sources**.

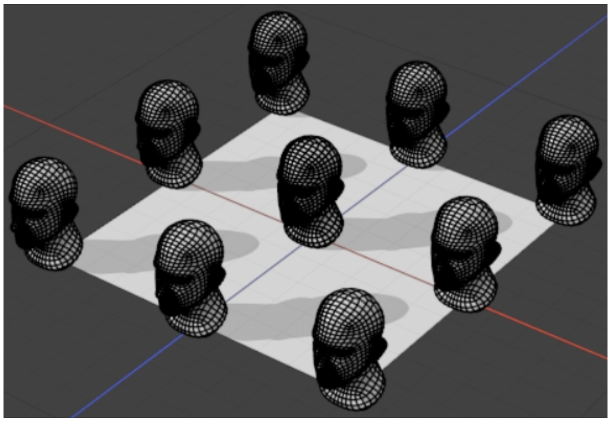
9. Select the **SimpleGuy** from the list and double-click it to add the geometry to the **Sources**.


Add Operator
Name
#
Particle Clone
Tool Pipe
Particle Clone Generator
Particle Source
Plane
(Add Tool Pipe)
Sources
SimpleGuy (2)

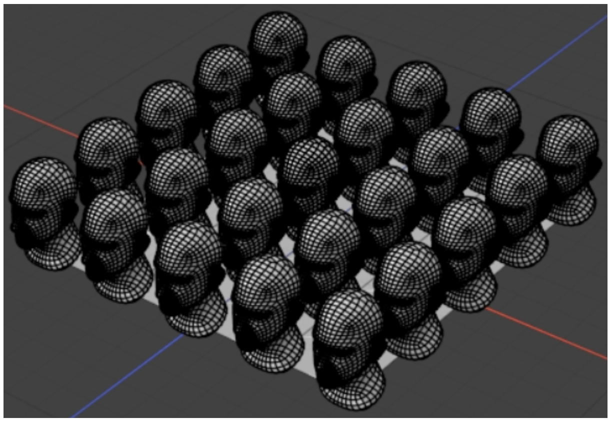
The **Sources** geometry is cloned by the number of vertices in the **Particle Source**, in this simple case just the four vertices from the plane.


Add Item
Select
Filter
Name
Untitled*
Mesh
Particle Clone
Plane
SimpleGuy
Add Operator
Name
#
Mesh
Particle Clone
Tool Pipe
Particle Clone Gener…
Particle Source
Plane
(Add Tool Pipe)
Sources
SimpleGuy (2)
(Add Sources)

10. To increase the number of clones, select the **Plane** in **Polygons** selection mode and press **Shift**+**D**.
11. Click **OK** to subdivide the geometry. Subdividing the plane, which is the **Particle Source**, creates more vertices and so the number of clones increases.

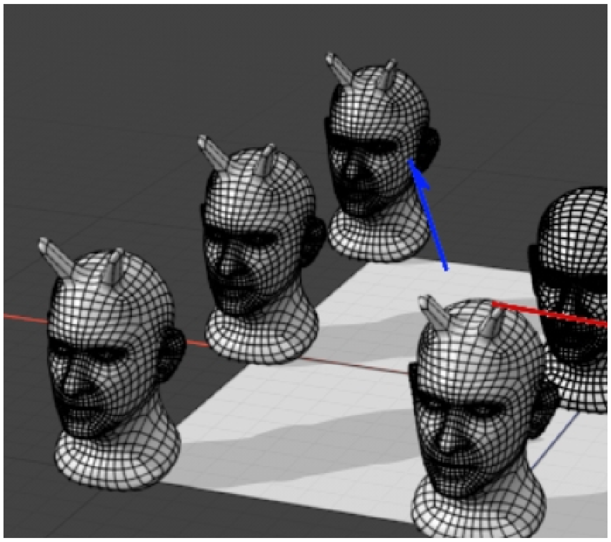


The source plane subdivided once

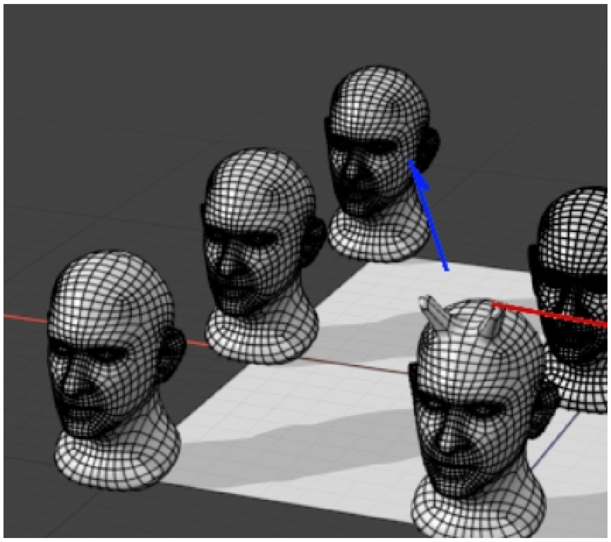


The source plane subdivided twice

One advantage of using Particle Clones over replicators is that you can affect clones individually. For example, adding a Bevel to the source mesh does not add the bevel effect to the clones. Similarly, adding a Bevel to one of the clones does not affect the other clones in the mesh.



A bevel on replicated geometry copies the bevel to all replicas



A bevel on cloned geometry only affects the selected clone

Particle Clone Controls

There are a number of settings available in the **Properties** tab for the **Particle Clone** Mesh Op.

Control	What it Does
Enable	Toggles cloned geometry on and off.
Use World Transform	When enabled, transforms are calculated based on the World Position of the operation. In most cases, this provides expected results.
Selection Type	Sets the component used when a selection is passed to the operation, **Vertex** or **Polygon**.
Replace Source	When enabled, removes the original geometry and creates a replacement clone.
Invert Polygons	When enabled, inverts the normal direction of the cloned geometry.
Merge Vertices	When enabled, merges coinciding vertices into a single vertex. The **Distance** control determines the merge threshold.
Vertex Only	When enabled, only clone vertices not polygons.

Control	What it Does
Distance	When **Merge Vertices** is enabled, the **Distance** value determines the maximum distance between vertices where merging occurs. Vertices outside this value remain unchanged.
Scale Pivot	Sets the center from which the clones start in simulations: **Center**, **Top**, **Bottom**, **Left**, **Right**, **Front**, or **Back**.
Source Index	If the clones have multiple sources, sets the source index order.

Flocking

Flocking is a particle operator that creates forces between particles that produce distinct particles groups or clumps. You have control over how strong the bond between these groups are as well as how quickly they appear in a simulation.

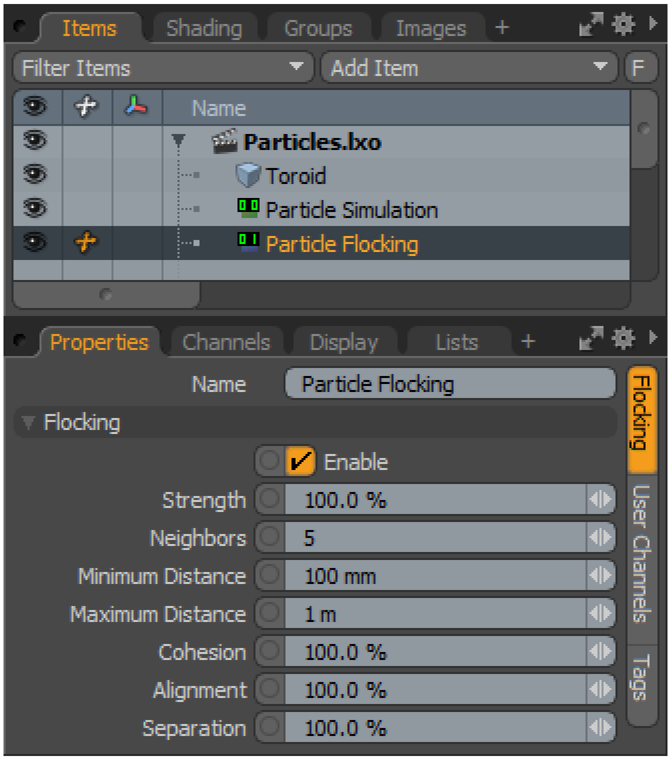
Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. Flocking operators require a particle simulation on which to operate. Once a simulation is added, add the flocking operator:

1. With a particle simulation selected in the **Items** list, click the **Flocking** button in the toolbox. Alternatively, in the **Items** list, click **Add Item** then go to **Particles** > **Simulation**, and double-click **Flocking**.

   When added this way, the flocking operator connects to the operator input of the particle simulation item and affect all emitted particles.

When the flocking item is selected, the following attributes appear in the **Properties** panel.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Particle Flocking
Properties
Channels
Display
Lists
Flocking
User Channels
Tags

Name: Particle Flocking

Flocking

☑ Enable

Strength 100.0 %

Neighbors 5

Minimum Distance 100 mm

Maximum Distance 1 m

Cohesion 100.0 %

Alignment 100.0 %

Separation 100.0 %

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Flocking	
Enable	Toggles the flocking operator item on or off. When enabled, the settings of the operator are evaluated for the associated particle simulation, when disabled, the settings are ignored.
Strength	Controls how much influence the flocking force has upon the target particles. With higher values particles flock sooner in a simulation while with lower values, particles flock later.
Neighbors	Controls the number of particles around a single particle that it considers when deciding which flocking group to join. **Note:** Since this value is per particle, increasing it too high can quickly cause a simulation to slow to a crawl.

Option	Description
Minimum Distance	Determines the distance where grouped particles repel away from each other.
Maximum Distance	Determines the radius distance that a particle uses to look for neighbors to join with.
Cohesion	Controls the strength of the influence that causes particles to move toward the averaged position of its neighbors. It is essentially the strength of the bond that keeps the group together.
Alignment	Controls the strength of the influence that causes particles to orient toward the averaged direction of its neighbors. It is essentially keeping the particles moving in the same direction.
Separation	Controls the strength of avoidance from crowding between neighbors in a group; the repulsion of nearby particles.

Dynamic Collider

The dynamic collider has no adjustable attributes in the **Properties** panel, it is merely meant as a bridge to share data between a particle simulation and a soft or hard body dynamics simulation when calculating collisions between them. This allows you to control which simulations are necessary to share data, which can save valuable calculation time for complex simulations of all types.

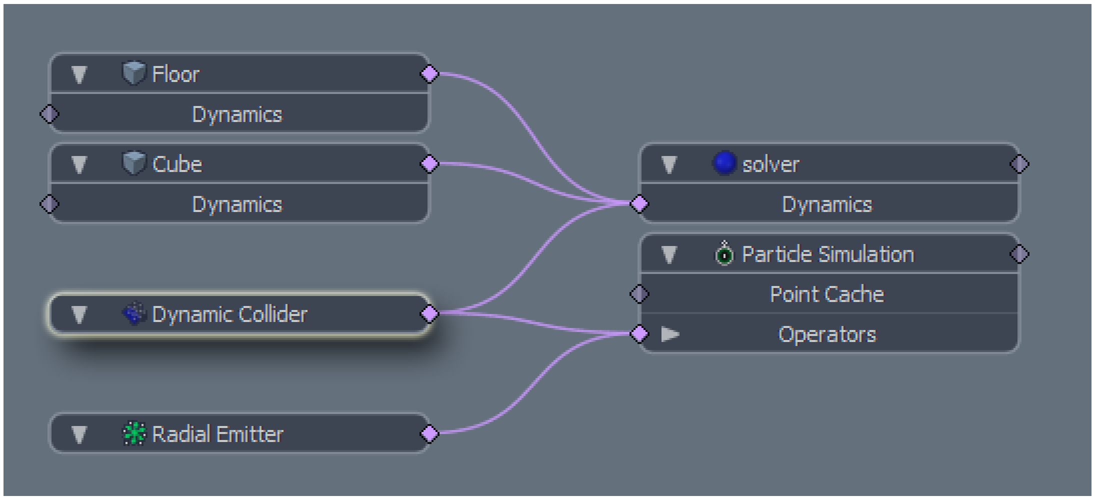
Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. Before adding the dynamic collider, you need to have the dynamic and particle simulation items already present.

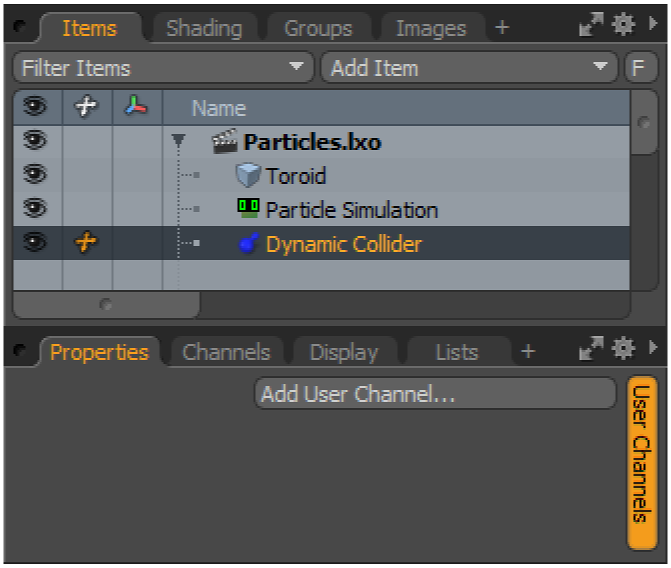
To add the dynamic collider item, follow these steps:

1. Select the dynamics solver item and the particle simulation item, creating the 'Dynamics Collider' item, which connects them as necessary for proper evaluation.
2. With the items selected, click the **Dynamic Collider** button in the toolbox. Alternatively, in the **Items** list, click **Add Item Particles** > **Simulation** > **Dynamic Collider**.

You can also manually connect the dynamic collider elements together via the **Schematic** viewport (illustrated below). For information of working with elements in the **Schematic** viewport, see Schematic Viewport. For more information on using the particle simulation tools, see Particles and Simulations.


Floor
Dynamics
Cube
Dynamics
Dynamic Collider
Radial Emitter
solver
Dynamics
Particle Simulation
Point Cache
Operators

The dynamic collider item connects to the solver and the particle simulation items.


Items
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Dynamic Collider
Properties
Channels
Display
Lists
+
Add User Channel...
User Channels

There are no user-definable attributes or channels related to the dynamic collider.

Dynamic Fluid

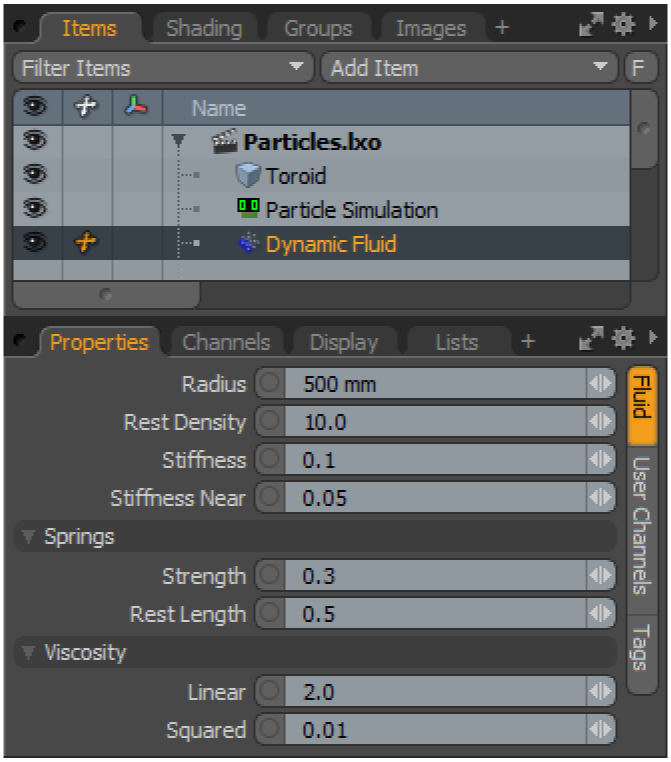
The dynamic fluid item allows you to simulate a variety of viscous liquids. The item itself is attached to a particle simulation item and works to control the particles so that they act in a fluid manner. Fluids are fully integrated into the dynamics of Modo and can be used to affect solid body elements as well, when combined with a dynamic collider item connecting the two simulations. Dynamics Forces can also be added to the simulation, providing additional control over the resulting particle behaviors.

Usage

The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. The dynamic fluid item is added to an existing particle simulation, which at minimum requires an emitter and a particle simulator.

To add a dynamic fluid item, select the target particle simulation and click the **Dynamic Fluid** button in the toolbox. Alternatively, in the **Items** list, click **Add Item**, navigate to **Particles** > **Simulation**, then double-click **Dynamic Fluid**. This automatically wires up the connection between the items.

For proper results, you must use an emitter with some volume to the emission area. By default, Modo emitters are single point, which is infinitely small. The fluid simulation produces particles that, like a liquid, are incompressible and when multiples are spawned from the single point, forces are generated that make the particles appear to shoot off in random directions. When using the radial and curve emitters, you need to define a **Start Radius** that is larger than the **Radius** value of the dynamic fluid item for proper results. Once the item is added to the simulation, particles act in a fluid manner. When the dynamic fluid item is selected, the following attributes are available in the **Properties** panel.


Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Dynamic Fluid
Properties
Channels
Display
Lists
Fluid
User Channels
Tags

Radius 500 mm
Rest Density 10.0
Stiffness 0.1
Stiffness Near 0.05
Springs
Strength 0.3
Rest Length 0.5
Viscosity
Linear 2.0
Squared 0.01

Option	Description
Radius	Density in the resulting fluid volume is determined by neighboring particles

Option	Description
	connecting via virtual springs. To save simulation time, springs are only generated for nearby particles in the simulation. The **Radius** option defines the area around a particle where it searches for neighbors to connect with.
Rest Density	Controls the maximum number of neighboring particles that can connect to a single particle when at rest (meaning they have reached a minimum velocity threshold).
Stiffness	Controls the strength of the inter-particle springs in retaining the **Radius** value. Higher values create springier, bouncier particles that are more apt to act like splashes, where lower values result in softer springs that react more like a viscous fluid.
Stiffness Near	When the distance for resting particles becomes less than the **Radius** value, additional smaller springs are added between particles that control the resting movement of the fluid. This is the **Stiffness Near** setting. It controls the repelling strength between resting particles.
Springs	
Strength	Determines the flexing strength of the virtual springs generated between particles which results in the fluid's plasticity, which influences how quickly or slowly the substance forgets its former shape.
Rest Length	When particles are resting, the spring length is modified to pull neighboring particles together to form the overall density volume. This value determines that minimum spring length value.
Viscosity	
Linear	The **Viscosity** setting controls the velocity transition values between active and resting particles. Higher **Linear** values increase the transition timer creating a thicker fluid.
Squared	When particles in a group reach equilibrium, non-linear forces are applied that repel the neighboring particles to maintain density in the volume. This value defines the strength of that force.

Terminator

The terminator item is a filter that works in conjunction with a falloff to generate specific areas where a particle gets destroyed.

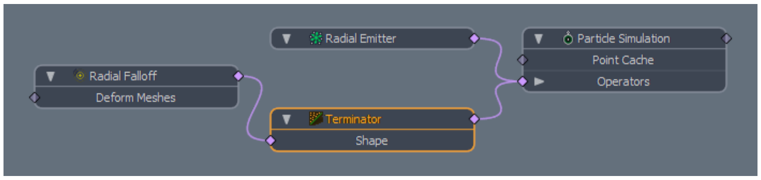
Usage

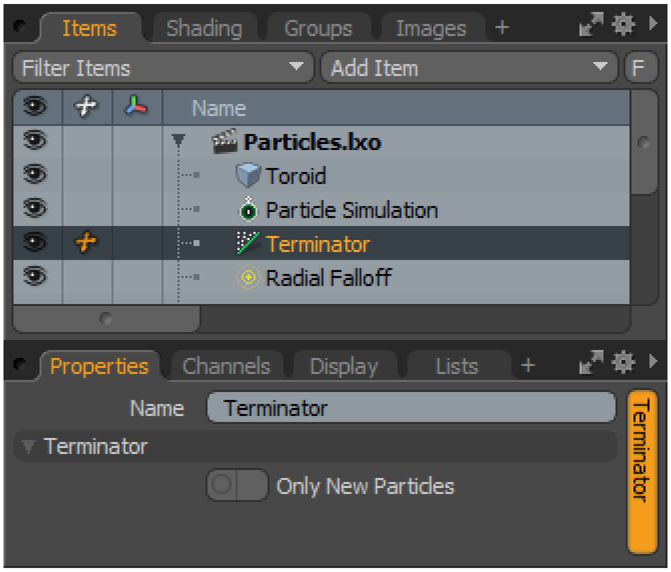
The main particle controls are found in the **Particles** sub-tab of the **Setup** interface's toolbox. Before adding a terminator, add a particle simulation on which the item can operate.

To add a terminator item, follow the steps below:

1. Select the particle simulation in the **Items** list and click the **Terminator** button in the toolbox. Alternatively, in the **Items** list, click **Add Item**, and under **Particles** > **Simulation**, double-click **Terminator**.
2. Add a falloff by clicking **Add Falloff** at the bottom of the toolbox, or in the **Items** list in **Add Item** > **Falloffs**. The type of falloff depends on the effect you need.

Once all the items are created, drag and drop them to the **Schematic** viewport and connect them appropriately. This is demonstrated below with a basic particle simulation. For more information of working with the **Schematic** viewport, see Schematic Viewport.




Items
Shading
Groups
Images
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Terminator
Radial Falloff
Properties
Channels
Display
Lists
Terminator

Name: Terminator

Terminator

Only New Particles

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Terminator	
Only New Particles	When enabled, only newly generated particles are affected by the terminator.

Particle Modifiers

Particle modifiers, or Pmods for short, offer you a way to modify particles in different and interesting ways. The basic modifier applies simple uniform transform controls for particles, whereas the random modifier applies the same thing, but randomly. The step modifier applies stepped transforms based on the particle's ID value, useful when applied to Point Clouds, but not limited to just them. The sieve modifier acts as a conditional controller, modifying particles based on user defined criteria. The expression modifier provides an extremely powerful means to modify particles procedurally and the look at modifier makes it easy to have the particles point toward specific items.

Particle Basic Modifier

Particle modifiers apply any or all of the three basic transforms (Move, Scale and Rotate) to a procedurally generated particle source, allowing you to animate the final elements in interesting ways. The basic particle modifier is applied as the go-between for any item's Point Source, offering interesting ways of animating the resulting items.

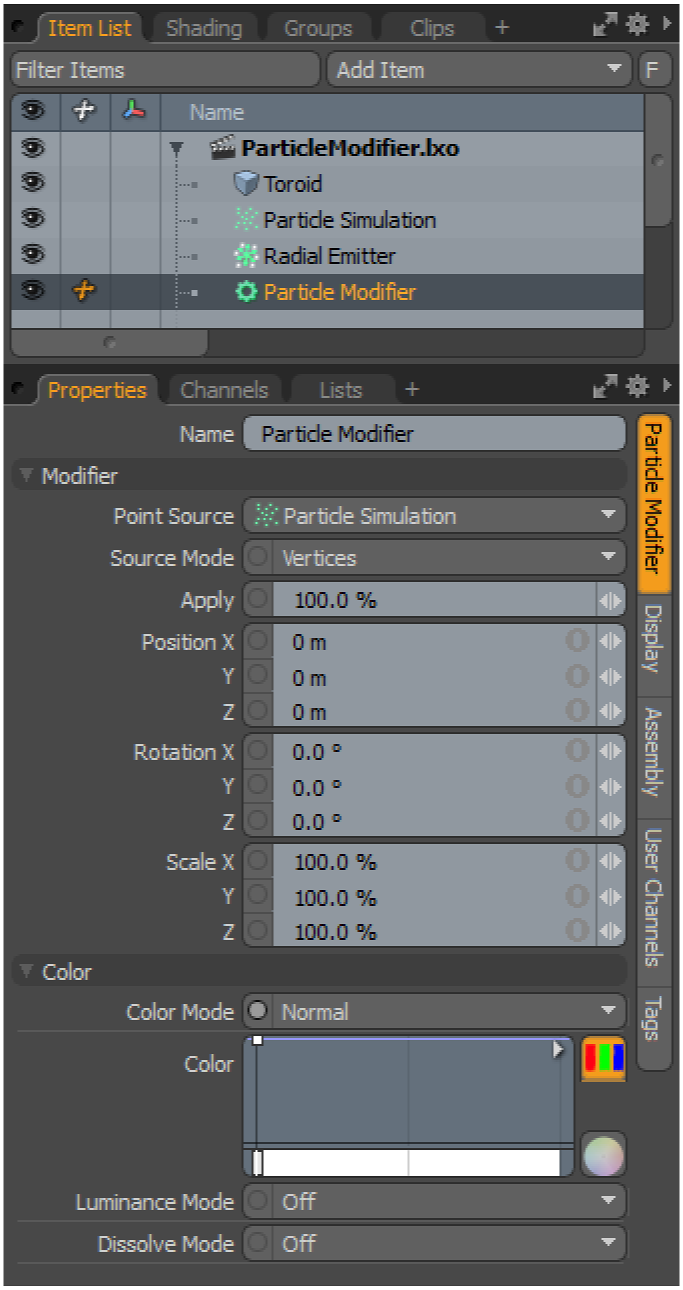
The basic particle modifier is a controller to apply simple transforms to particles, such as translate, rotate and scale. By itself, it doesn't offer any compelling benefits that can't be applied by the particle generators. However, when combined with falloffs and Particle Operators, some really interesting effects can be created, such as particle rotation within the confines of a falloff, or particles scaling when a collision is encountered.

Adding a Particle Modifier

You can add a particle modifier item by using the **Add Item** function of the **Items** List. Click the button to open the menu, click **Items** and select **Particles** > **Modifiers**, and double-click **Particle Modifier**.

Once added, you need to specify the point source that you wish to affect. Once the originating point source is defined, you also need to specify the particle modifier item as the source in the final item. If thought of like a chain, the originating particle generation is defined as the source in the modifier, and then the modified values are fed into the final effect, such as a Replicator, Blob, Volume or Sprite. The source must be defined in this order so as to produce the intended result.

When applying the basic modifier to an active particle simulation, you need to connect the modifier in the proper order within the **Schematic** viewport. For existing networks, simply select the connector between the simulation item and the target and add the modifier. The basic modifier is found in the **Schematic** view in the **Add...** menu under **Particles** > **Modifiers** > **Particle Modifier**. When selected, the following attributes are available in the **Properties** panel.



Item List
Shading
Groups
Clips
Filter Items
Add Item
F
Name
ParticleModifier.lxo
Toroid
Particle Simulation
Radial Emitter
Particle Modifier
Properties
Channels
Lists
Particle Modifier
Display
Assembly
User Channels
Tags

Name: Particle Modifier

Modifier

Point Source: Particle Simulation

Source Mode: Vertices

Apply: 100.0 %

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Color

Color Mode: Normal

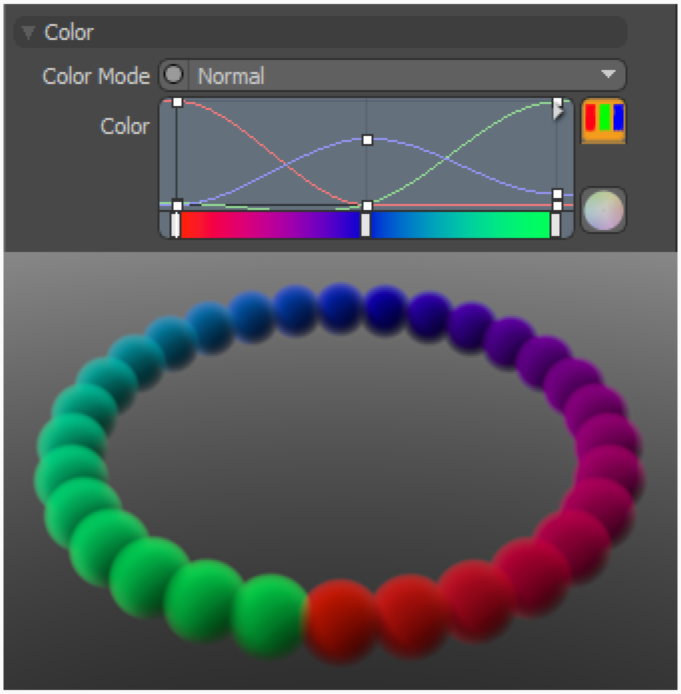
Color

Luminance Mode: Off

Dissolve Mode: Off

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Modifier	
Point Source	Defines the source of the particles that are affected by the modifier.

Option	Description
Source Mode	Specifies the way the particle modifier affects the source. **Vertices** uses the point positions, **Polygons** uses the polygon centers, and **Detached Vertices** uses the point positions of vertices that are not part of polygons.
Apply	Acts as a multiplier, controlling the overall strength of the transforms applied to the source particles.
PositionX/Y/Z	Determines a position value for each individual axis as an offset from the particle's originating position.
RotationX/Y/Z	Determines a rotation value for each individual axis as an offset from the particle's originating orientation.
Scale X/Y/Z	Determines a scaling value for each individual axis as an increase/decrease from the particle's initial size.
Color	Allows you to use mini gradient editors to assign and control different attributes for the resulting particles. These attributes are then passed along to the final effect, working with both volumes and sprites. The gradient values are generated sequentially, dividing the gradient itself into the number of generated particles it is assigned to, each receiving the value for that position along the gradient. For example, a red to blue gradient assigned to a linear array of particles produces one red particle at the start. The particle colors then fade toward the blue color in even increments along the length of the array. There are three different settings that can be applied to their associated surface attributes. Any of these functions are enabled by selecting any option other than **Off** (the default value that disabled the function). The selected **Mode** determines the blending of the gradient with the base setting of the target element.
Color Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle coloring. The **Color** gradient editor that opens allows you to define the actual gradient values that are applied.
Luminance Mode	Determines how the resulting gradient applies, defining the blending mode of the luminosity value as it applies to the base particle luminosity. The **Luminosity** gradient editor that opens allows you to define the actual gradient values that are applied.
Dissolve Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle transparency. The **Dissolve** gradient editor that opens allows you to define the actual gradient values that are determine the transparent

Option	Description
	amount.  A **Normal** color mode gradient applied to a particle generator's radial array

Particle Random Modifier

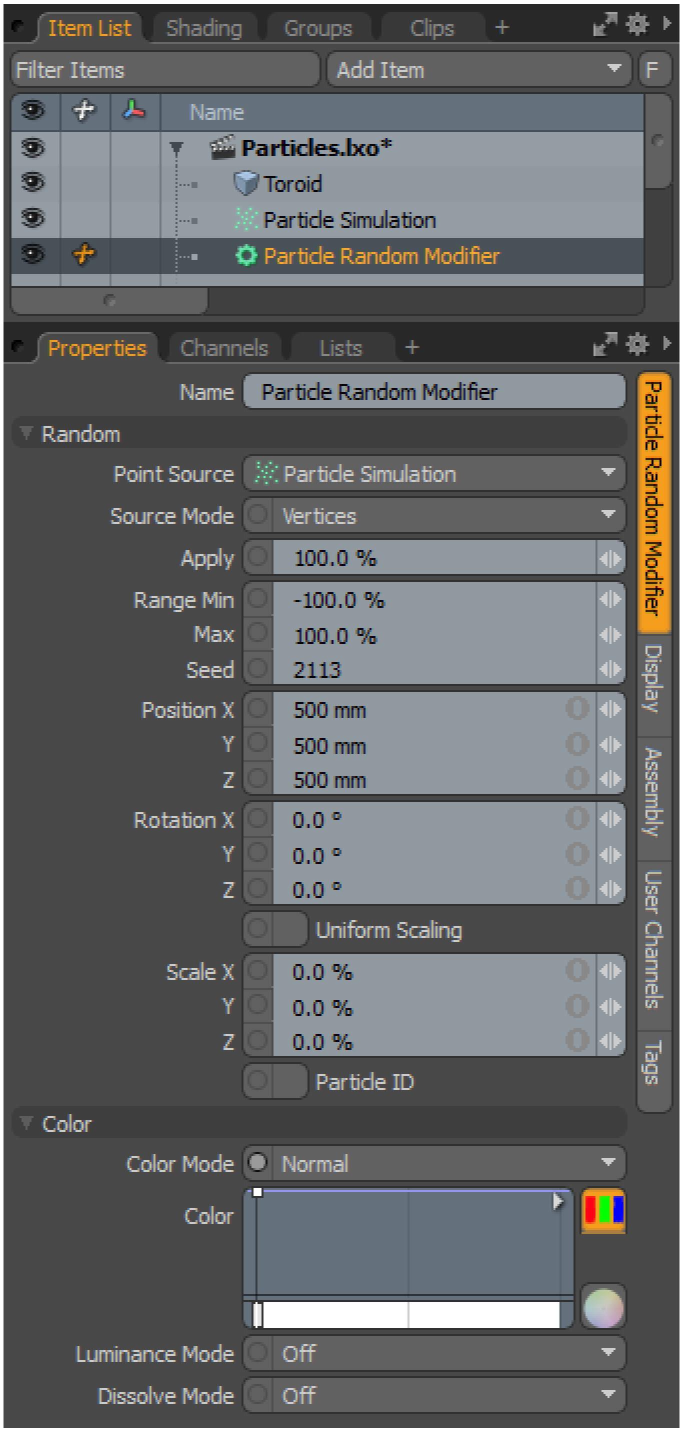
The random particle modifier randomly applies any or all of the three basic transforms (Move, Scale and Rotate) to a procedurally generated particle source. The random particle modifier is applied as the go-between for any item's point source, providing and easy way to add natural variations to the resulting particles. Interesting effects can also be created when combined with falloffs and particle operators.

Adding a Particle Modifier

You can add a random particle modifier item by using the **Add Item** function of the **Items** List. Click the button to open the menu, then under **Particles** > **Modifiers**, double-click **Particle Random Modifier**.

Once added, you need to specify the point source that you wish to affect. Once the originating point source is defined, you also need to specify the particle random modifier item as the source in the final item. If thought of like a chain, the originating particle generation is defined as the source in the modifier, and then the modified values are fed into the final effect, such as a Replicator, Blob, Volume or Sprite. The source must be defined in this order so as to produce the intended result.

When applying the random modifier to an active particle simulation, you need to connect the modifier in the proper order within the **Schematic** viewport. For existing networks, simply select the connector between the simulation item and the target and add the modifier. The random modifier is found in the **Schematic** view in the **Add...** menu under **Particles** > **Modifiers** > **Particle Random Modifier**. When selected, the following attributes are available in the **Properties** panel.


Item List
Shading
Groups
Clips
Filter Items
Add Item
F
Name
Particles.lxo*
Toroid
Particle Simulation
Particle Random Modifier
Properties
Channels
Lists
Particle Random Modifier
Display
Assembly
User Channels
Tags

Name: Particle Random Modifier

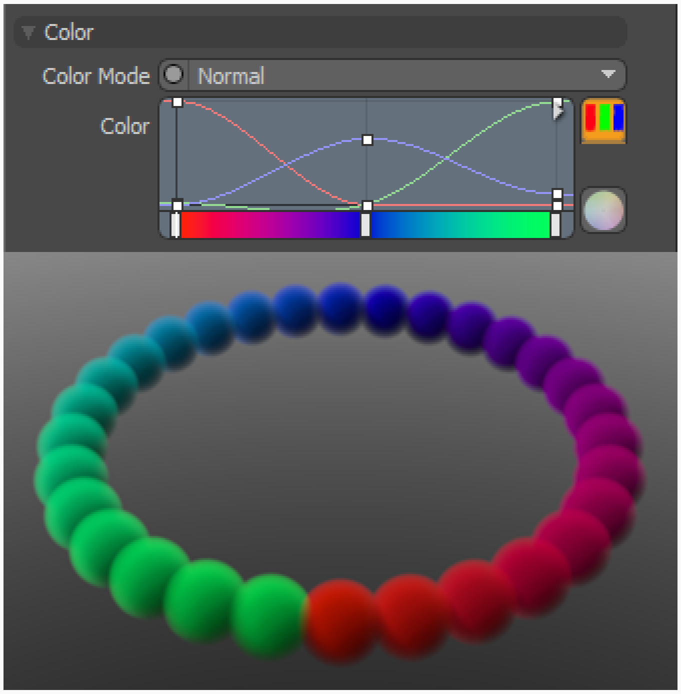
Random

Point Source: Particle Simulation
Source Mode: Vertices
Apply: 100.0 %
Range Min: -100.0 %
Max: 100.0 %
Seed: 2113
Position X: 500 mm
Y: 500 mm
Z: 500 mm
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
○ Uniform Scaling
Scale X: 0.0 %
Y: 0.0 %
Z: 0.0 %
○ Particle ID

Color

Color Mode: Normal
Color
Luminance Mode: Off
Dissolve Mode: Off

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Random	
Point Source	Defines the source of the particles that are affected by the modifier.
Source Mode	Specifies the way the particle modifier affects the source. **Vertices** uses the point positions, **Polygons** uses the polygon centers, and **Detached Vertices** uses the point positions of vertices that are not part of polygons.
Apply	Determines the overall degree that the expression affects the target. At 0% the target is not affected at all, at 100% the target is fully affected. This allows you to transition to the modified state.
Range Min/Max	Determines the amount of transform applied to the target over the entire range of the particles acting as a multiplier on the value. For example, if the **Scale** is set to 200% on all three axes, and the **Range Min/Max** is -100%/100% respectively, then the first particle would be scaled 200%, attenuating the value toward the last particle scaled at 200%. The range is determined in the order of the **Particle ID**.
Seed	The initial number used when generating the random procedural values. Different **Seed** values produce different random variations and can be useful in changing the result.
PositionX/Y/Z	Defines the position values added to the target for each axes.
RotationX/Y/Z	Defines the rotation values added to the target for each axes.
Uniform Scaling	When enabled, the random scaling factor is applied evenly on all axes.
Scale X/Y/Z	Defines the scale values added to the target for each axes.
Particle ID	When enabled, the unique ID identifier values are randomized, instead of sequential of the initial particle.
Color	Allows you to use mini gradient editors to assign and control different attributes for the resulting particles. These attributes are then passed along to the final effect, working with both volumes and sprites. The gradient values are generated sequentially, dividing the gradient itself into the number of generated particles it is assigned to, each receiving the value for that position along the gradient. For example, a red to blue gradient

Option	Description
	assigned to a linear array of particles produces one red particle at the start. The particle colors then fade toward the blue color in even increments along the length of the array. There are three different settings that can be applied to their associated surface attributes. Any of these functions are enabled by selecting any option other than **Off** (the default value that disabled the function). The selected **Mode** determines the blending of the gradient with the base setting of the target element.
Color Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle coloring. The **Color** gradient editor that opens allows you to define the actual gradient values that are applied.
Luminance Mode	Determines how the resulting gradient applies, defining the blending mode of the luminosity value as it applies to the base particle luminosity. The **Luminosity** gradient editor that opens allows you to define the actual gradient values that are applied.
Dissolve Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle transparency. The **Dissolve** gradient editor that opens allows you to define the actual gradient values that are determine the transparent amount.  Color Mode Normal A **Normal** color mode gradient applied to a particle generator's radial array

Particle Step Modifier

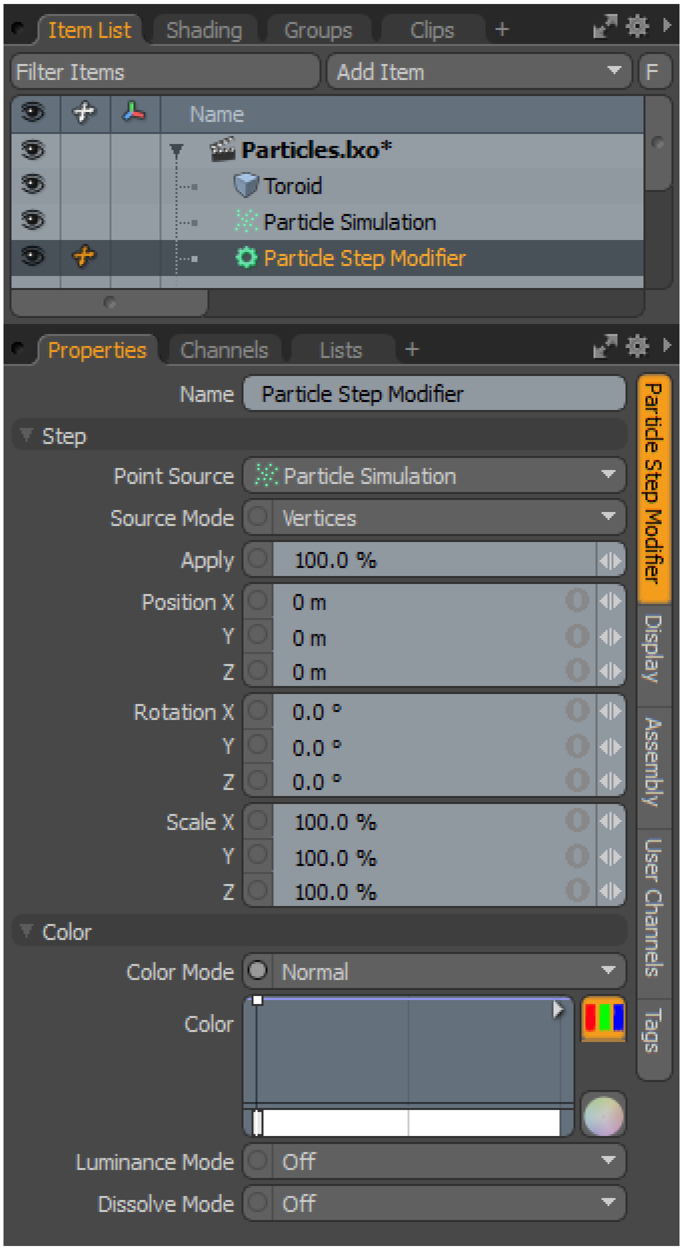
The particle step modifier applies any or all of the three basic transforms (Move, Scale and Rotate) to a procedurally generated particle source, in additive steps. Meaning each subsequent element inherits the transform of its predecessor and then applies the transform again, generating progressively positioned, scaled or rotated elements. The resulting effect on the particles is based on point order, so the transforms are applied in that order. Generally meant for use with the Particle Generator, they can also be applied to the Particle Cloud and the Surface Particle Generator items.

Adding a Particle Step Modifier

You can add a particle step modifier item by using the **Add Item** function of the **Items** List. Click the button to open the menu, then under **Particles** > **Modifiers**, double-click **Particle Step Modifier** option.

Once added, you need to specify the point source that you wish to affect. Once the originating point source is defined, you also need to specify the particle step modifier item as the source in the final item. If thought of like a chain, the originating particle generation is defined as the source in the step modifier, and then the modified values are fed into the final effect, such as a Replicator, Blob, Volume or Sprite. The source must be defined in this order so as to produce the intended result.

When applying the step modifier to an active particle simulation, you need to connect the modifier in the proper order within the **Schematic** viewport. For existing networks, simply select the connector between the simulation item and the target and add the modifier. The step modifier is found in the **Schematic** view in the **Add...** menu under **Particles** > **Modifiers** > **Particle Step Modifier**. When selected, the following attributes are available in the **Properties** panel.


Item List
Shading
Groups
Clips
Filter Items
Add Item
F
Name
Particles.lxo*
Toroid
Particle Simulation
Particle Step Modifier
Properties
Channels
Lists
Particle Step Modifier
Display
Assembly
User Channels
Tags

Name: Particle Step Modifier

Step

Point Source: Particle Simulation

Source Mode: Vertices

Apply: 100.0 %

Position X: 0 m

Y: 0 m

Z: 0 m

Rotation X: 0.0 °

Y: 0.0 °

Z: 0.0 °

Scale X: 100.0 %

Y: 100.0 %

Z: 100.0 %

Color

Color Mode: Normal

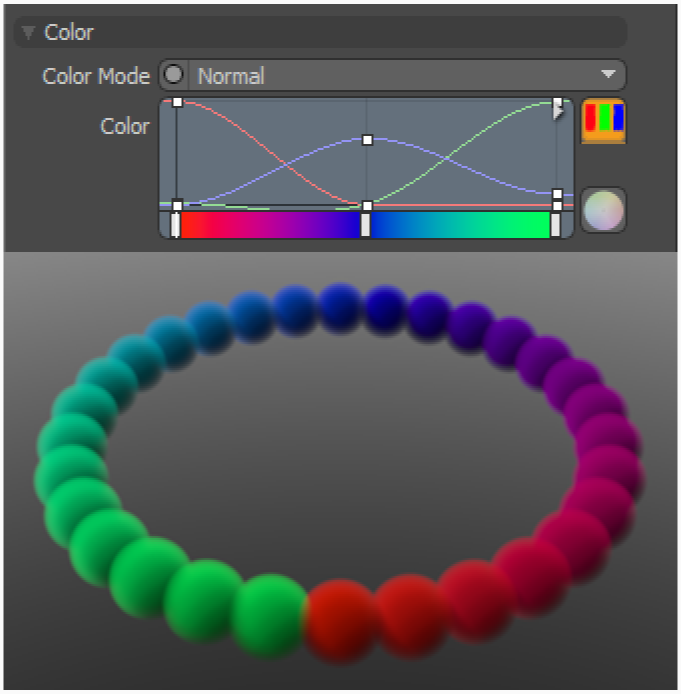
Color

Luminance Mode: Off

Dissolve Mode: Off

Option	Description
Point Source	Defines the source of the particles that are affected by the modifier.
Source Mode	Specifies the way the particle modifier affects the source. **Vertices** uses the point positions, **Polygons** uses the polygon centers, and **Detached Vertices** uses the point positions of vertices that are not part of polygons.
Apply	Acts as a multiplier, controlling the overall strength of the transforms applied to the source particles.

Option	Description
PositionX/Y/Z	Determines a position offset value for the initial particle for each individual axis. Subsequent particles inherit this values and then apply it again, producing a progressive type effect.
RotationX/Y/Z	Determines a rotation offset value for the initial particle for each individual axis. Subsequent particles inherit this values and then apply it again, producing a progressive type effect.
Scale X/Y/Z	Determines a size increase/decrease for the initial particle for each individual axis. Subsequent particles inherit this values and then apply it again, producing a stepped or progressive type effect.
Color	Allows you to use mini gradient editors to assign and control different attributes for the resulting particles. These attributes are then passed along to the final effect, working with both volumes and sprites. The gradient values are generated sequentially, dividing the gradient itself into the number of generated particles it is assigned to, each receiving the value for that position along the gradient. For example, a red to blue gradient assigned to a linear array of particles produces one red particle at the start. The particle colors then fade toward the blue color in even increments along the length of the array. There are three different settings that can be applied to their associated surface attributes. Any of these functions are enabled by selecting any option other than **Off** (the default value that disabled the function). The selected **Mode** determines the blending of the gradient with the base setting of the target element.
Color Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle coloring. The **Color** gradient editor that opens allows you to define the actual gradient values that are applied.
Luminance Mode	Determines how the resulting gradient applies, defining the blending mode of the luminosity value as it applies to the base particle luminosity. The **Luminosity** gradient editor that opens allows you to define the actual gradient values that are applied.
Dissolve Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle transparency. The **Dissolve** gradient editor that opens allows you to define the actual gradient values that are determine the transparent amount.

Option	Description
	 Color Mode Normal A **Normal** color mode gradient applied to a particle generator's radial array

Particle Sieve Modifier

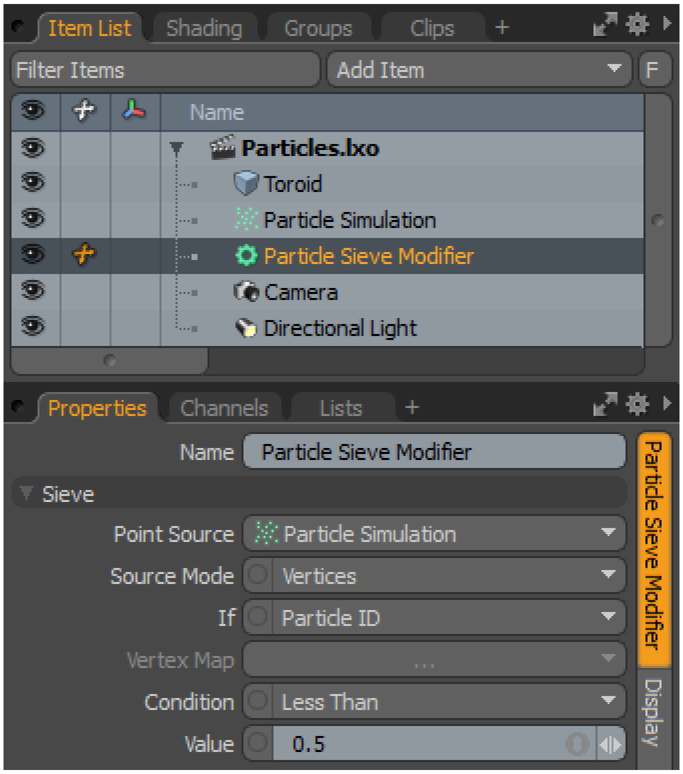
The particle sieve modifier is a conditional filter that allows you to adjust values based on certain specified criteria. Only values that meet the defined criteria are passed along to the output (it doesn't modify the initial point source itself). It is useful in filtering particles based on age, particle ID or a multitude of other options. It can also be equally useful for point clouds. When combined with a surface particle generator, particles can be filtered by slope.

Adding a Particle Sieve Modifier

You can add a particle sieve modifier item by using the **Add Item** function of the **Items** List. Click the button to open the menu, then under **Particles** > **Modifiers**, double-click **Particle Step Modifier**.

Once added, you need to specify the point source that you wish to affect. Once the originating point source is defined, you also need to specify the particle sieve modifier item as the source in the final item. If thought of like a chain, the originating particle generation is defined as the source in the sieve modifier, and then the modified values are fed into the final effect, such as a Replicator, Blob, Volume or Sprite. The source must be defined in this order so as to produce the intended result.

When applying the sieve modifier to an active particle simulation, you need to connect the modifier in the proper order within the **Schematic** viewport. For existing networks, simply select the connector between the simulation item and the target and add the modifier. The step modifier is found in the **Schematic** view in the **Add...** menu under **Particles** > **Modifiers** > **Particle Sieve Modifier**. When selected, the following attributes are available in the **Properties** panel.


Item List
Shading
Groups
Clips
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Particle Sieve Modifier
Camera
Directional Light
Properties
Channels
Lists
Particle Sieve Modifier
Display

Name: Particle Sieve Modifier

Sieve

Point Source: Particle Simulation

Source Mode: Vertices

If: Particle ID

Vertex Map: ...

Condition: Less Than

Value: 0.5

Name	Displays the current item name. To change it, click on the field and type in the new name.
Sieve	
Point Source	Defines the source of the particles that are affected by the modifier.
Source Mode	Specifies the way the particle modifier affects the source. **Vertices** uses the point positions, **Polygons** uses the polygon centers, and **Detached Vertices** uses the point positions of vertices that are not part of polygons.
If	Controls which aspect of the source particles are evaluated for the defined condition. The pop-up menu allows you to choose from one of the many options available.
Vertex Map	When the point sources is a Mesh Item, you can define an associated weight Vertex Map, adding finer control to the sieve. Those vertices with weight values greater than 0.0

	are assumed to have a value of 1, vertices with weight value of less than 0.0, the value is assumed to be 0 (basically, it is a binary on or off behavior).
Condition	Determines under what condition that particles are filtered. This works in conjunction with the **Value** option.
Value	Defines the specific filtering value based on the **Condition** option selected. Value varies dependent on the **If** option. For example, if **Slope** is defined, then the **Value** relates to angle of degrees.

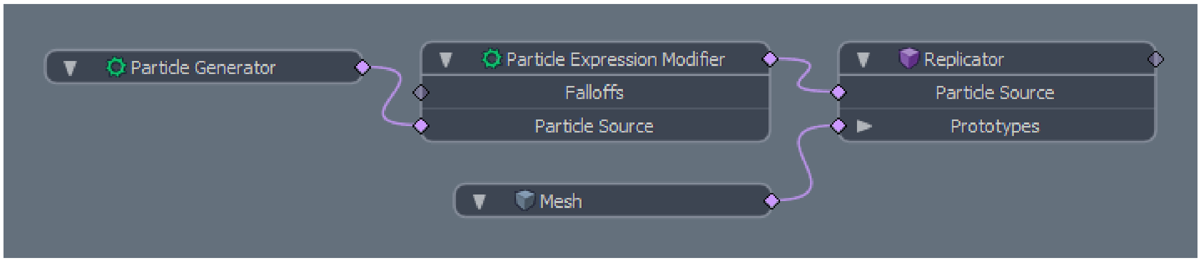
Particle Expression Modifier

The particle expression modifier allows you to apply expressions to particles. An expression is a set of instructions that are represented as a mathematical formula. In the case of this particle expression modifier, certain mathematical controls can be applied to particles to control various aspects of the particles motions or positions, or can assign a control when a certain condition is met. The expression itself controls the way in which the particles are affected, the **Apply** and **Range** options control the degree of its affect, and the transform options control the actual modification that is applied to the target.

Adding a Particle Expression Modifier

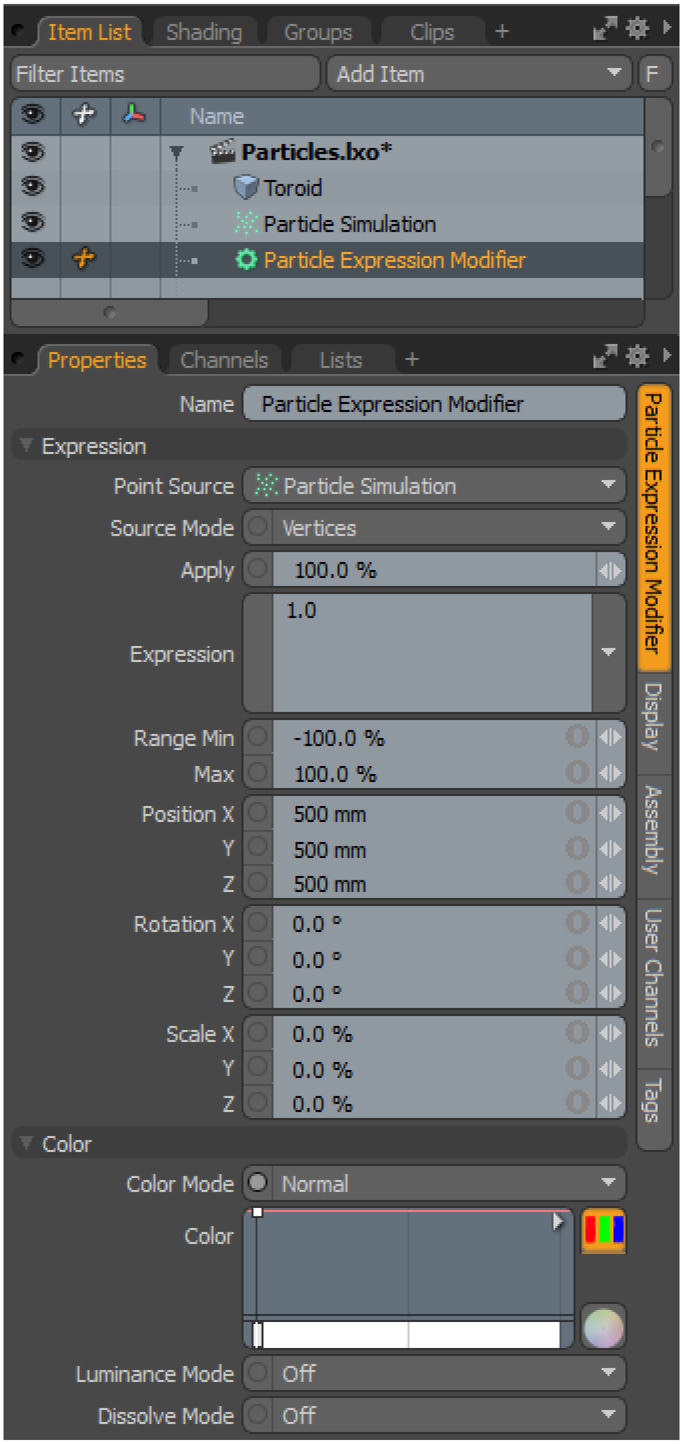
You can add a particle expression modifier item in the **Setup** interface under the **Particles** sub-tab, or by using the **Add Item** button of the **Items** List. Click the button to open the menu, and under **Particles** > **Modifiers**, click **Particle Expression Modifier**.

Once added to the scene, you need to drag the item itself from the **Items** list and drop it in the **Schematic** viewport. The expression modifier is connected between the point source element and the point target items, such as a Replicator. The connection should look like this image.



The expression modifies the particle values from the source, which can be any number of particle items, either static or dynamic. In this case, it is a simple particle Generator. Those values are replaced with the expression evaluations and then passed on to the target. The values returned by the expression are multiplied by the values set in the other channels of the modifier. For example, for a sine wave on the Y axis that moves through the range of the particles with time, the expression is **sin(T+PID)**. Where using **PID** (particle ID) is an offset to the current time applied to the Y position. Then you just need to set the **Position Y** transform option to a positive value, such as 1 m, and all the other Position, Rotation and Scale channels to 0.0. When the Timeline is played, the sine wave transforms the positions of the particles 1 m modulating up and down with the default **Range Min/Max** values of 100%/100%, respectively.

When the particle expression modifier is selected, the following attributes appear in the **Properties** panel.


Item List
Shading
Groups
Clips
+
Filter Items
Add Item
F
Name
Properties
Channels
Lists
+
Particle Expression Modifier
Display
Assembly
User Channels
Tags

Particles.lxo*
Toroid
Particle Simulation
Particle Expression Modifier

Name: Particle Expression Modifier

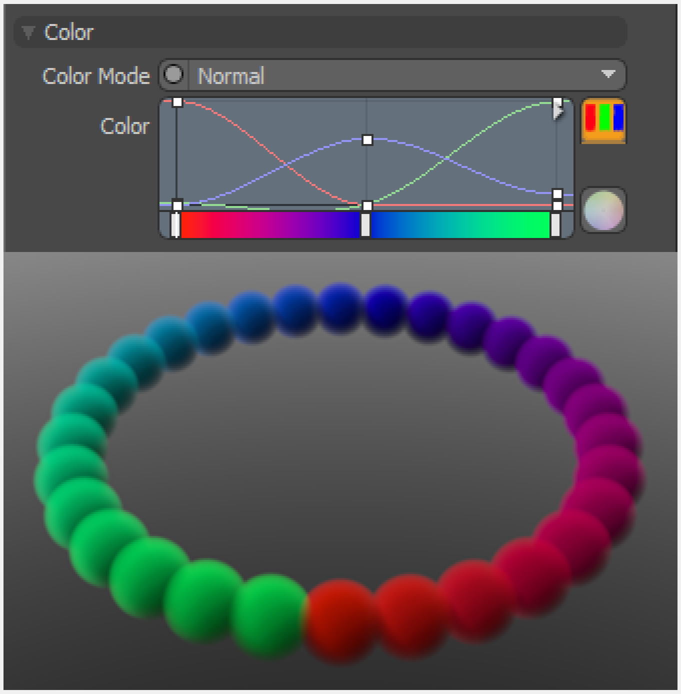
Expression

Point Source: Particle Simulation
Source Mode: Vertices
Apply: 100.0 %
Expression: 1.0
Range Min: -100.0 %
Max: 100.0 %
Position X: 500 mm
Y: 500 mm
Z: 500 mm
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 0.0 %
Y: 0.0 %
Z: 0.0 %

Color

Color Mode: Normal
Color
Luminance Mode: Off
Dissolve Mode: Off

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Expression	
Point Source	Defines the source of the particles that are affected by the expression modifier.
Source Mode	Specifies the way the particle modifier affects the source. **Vertices** uses the point positions, **Polygons** uses the polygon centers, and **Detached Vertices** uses the point positions of vertices that are not part of polygons.
Apply	Determines the overall degree that the expression affects the target. At 0% the target is not affected at all, at 100% the target is fully affected. This allows you to transition to the modified state.
Formula	This is where the actual expression formula is entered.
Range Min/Max	Determines the amount of transform applied to the target over the entire range of the particles acting as a multiplier on the value. For example, if the **Scale** is set to 200% on all three axes, and the **Range Min/Max** is 100%/100% respectively, then the first particle would be scaled 200%, attenuating the value toward the last particle scaled at 200%. The range is determined in the order of the **Particle ID**.
Seed	The initial number used when generating the random procedural values. Different **Seed** values produce different random variations and can be useful in changing the result.
PositionX/Y/Z	Defines the position values added to the target for each axes.
RotationX/Y/Z	Defines the rotation values added to the target for each axes.
Scale X/Y/Z	Defines the scale values added to the target for each axes.
Color	Allows you to use mini gradient editors to assign and control different attributes for the resulting particles. These attributes are then passed along to the final effect, working with both volumes and sprites. The gradient values are generated sequentially, dividing the gradient itself into the number of generated particles it is assigned to, each receiving the value for that position along the gradient. For example, a red to blue gradient assigned to a linear array of particles produces one red particle at the start. The particle colors then fade toward the blue color in even increments along the length of the array. There are three different settings that can be applied to their associated surface

Option	Description
	attributes. Any of these functions are enabled by selecting any option other than **Off** (the default value that disabled the function). The selected **Mode** determines the blending of the gradient with the base setting of the target element.
Color Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle coloring. The **Color** gradient editor that opens allows you to define the actual gradient values that are applied.
Luminance Mode	Determines how the resulting gradient applies, defining the blending mode of the luminosity value as it applies to the base particle luminosity. The **Luminosity** gradient editor that opens allows you to define the actual gradient values that are applied.
Dissolve Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle transparency. The **Dissolve** gradient editor that opens allows you to define the actual gradient values that are determine the transparent amount.  Color Color Mode: Normal A **Normal** color mode gradient applied to a particle generator's radial array

Particle Variables

The following table lists per-particle variables supported by the parser.

Note: For **AGE**, **PATH**, and **MASS** you need to enable these features in the Particle Simulation item, otherwise they return zero.

User channels can be referenced by name in the expression.

Variable	**Meaning**
PX, PY, PZ	Position for X, Y & Z
PPX, PPY, PPZ	Previous Position for X, Y & Z
RX, RY, RZ	Rotation for X, Y & Z
SX, SY, SZ	Scale for X, Y & Z
VX, VY, VZ	Velocity for X, Y & Z
FX, FY, FZ	Force for X, Y & Z
CR, CG, CB	Color for R, G & B
LUM	Luminance
PID	Particle ID
PCNT	Particle Count
AGE	Particle Age
MASS	Particle Mass
SIZE	Particle Size
PATH	The distance the particle has traveled
DISS	Dissolve
RATE	Rate

Variable	Meaning
ITEM	Item
T	Time

Built-in Function

The following table gives an overview of the functions supported by the default implementation. It lists the function names, the number of arguments and a brief description.

Name	Meaning
sin	sine function
cos	cosine function
tan	tangens function
asin	arcus sine function
acos	arcus cosine function
atan	arcus tangens function
sinh	hyperbolic sine function
cosh	hyperbolic cosine
tanh	hyperbolic tangens function
asinh	hyperbolic arcus sine function
acosh	hyperbolic arcus tangens function
atanh	hyperbolic arcur tangens function

Name	Meaning
log2	logarithm to the base 2
log10	logarithm to the base 10
log	logarithm to the base 10
ln	logarithm to base e (2.71828...)
exp	e raised to the power of x
sqrt	square root of a value
sign	sign function -1 if x<0; 1 if x>0
rint	round to nearest integer
abs	absolute value
min	min of all arguments
max	max of all arguments
sum	sum of all arguments
avg	mean value of all arguments
mod(a,b)	returns the remainder of a divided by b
clamp(a, min, max)	given a value a, if a is less then min, min is returned. If greater then max, max is returned. If in between min and max, a is returned.

Built-in Binary Operators

The following table lists the default binary operators supported by the parser.

Operator	Meaning	Priority
=	assignment *	-1
&&	logical and	1
||	logical or	2
<=	less or equal	4
>=	greater or equal	4
!=	not equal	4
==	equal	4
>	greater than	4
<	less than	4
+	addition	5
-	subtraction	5
*	multiplication	6
/	division	6
^	raise x to the power of y	7

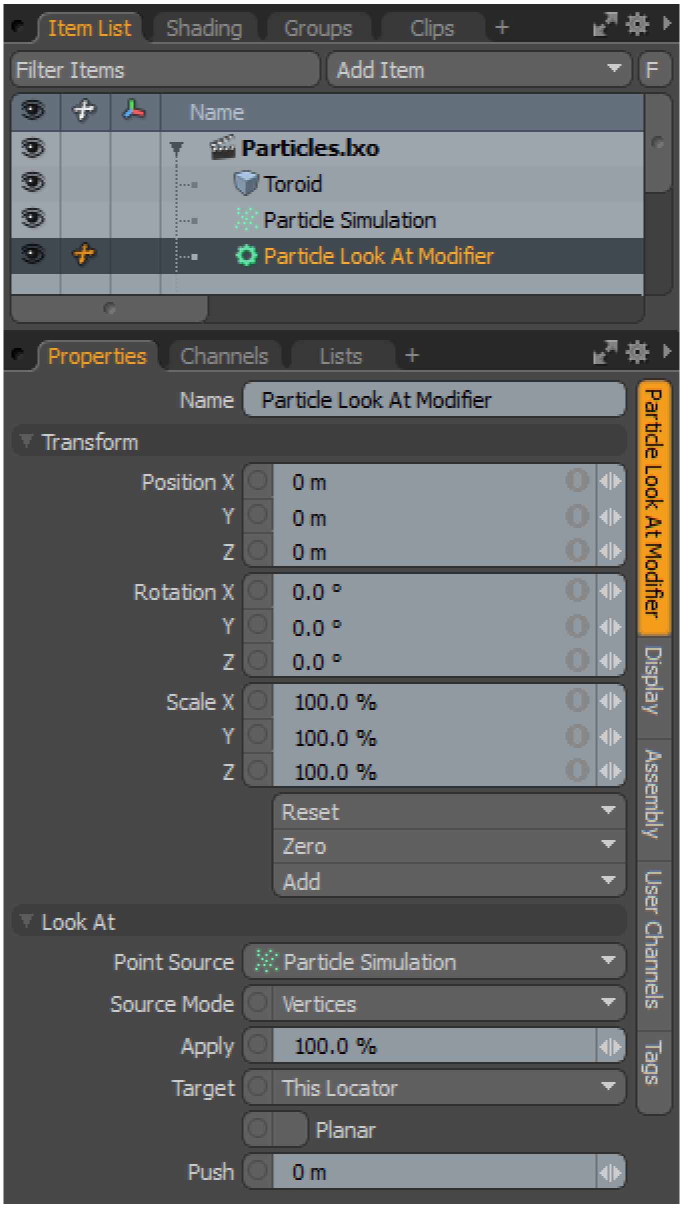
Other Operators

muParser has built in support for the if/then/else operator. It uses lazy evaluation in order to make sure only the necessary branch of the expression is evaluated.

Operator	Meaning	Remarks
?:	if then else operator	C/C++ style syntax

Particle Look At Modifier

The particle look at modifier orients the Z+ direction of the source particles toward a user-defined target item. The target item can be static or animated. When animated, the source particles' rotation values are adjusted so that it always remains oriented toward the target.


Item List
Shading
Groups
Clips
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Particle Look At Modifier
Properties
Channels
Lists
Particle Look At Modifier
Display
Assembly
User Channels
Tags
Reset
Zero
Add

Name Particle Look At Modifier

Transform
Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Look At
Point Source Particle Simulation
Source Mode Vertices
Apply 100.0 %
Target This Locator
Planar
Push 0 m

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new name.
Transform	When using the **Target** option **This Locator**, the **Position XYZ** values control the location of the look at modifier's locator item in the scene, which defines the position that the targeted elements orient towards.
Look At	
Point Source	Defines the source of the particles that are affected by the modifier.
Source Mode	Specifies the way the particle modifier affects the source. **Vertices** uses the point positions, **Polygons** uses the polygon centers, and **Detached Vertices** uses the point positions of vertices that are not part of polygons.
Apply	Acts as a multiplier, controlling the overall strength of the transforms applied to the original particles.
Target	Defines which item in the scene the source is oriented toward. The following options are available: • **This Locator**: Orients toward the locator item of the look at modifier itself. • **Next**/**Previous Particle**: Each particle in a given simulation or cloud is given a unique ID value in the order the particles are generated. The **Next** or **Previous** options can create chain or snaking type effects where particles orient toward earlier or later particles. • **Render Camera**: Orients towards the camera selected as the **Render Camera** in the Render item. • **Direction**: Allows you to specify a particular vector to use for the orientation.
Direction X, Y, Z	If **Target** is set to **Direction**, set the orientation vector using these values. This gives you full control over the particle look-at orientation.
Planar	When enabled, items only rotate around the up axis, especially useful for billboard-style image planes (flat polygons with a transparent image mapped onto it).
Push	An offset determined by the angle between the target and the actual particle location. Positive values push the target items further from the defined target, while negative values bring them closer.

Particle Audio Modifier

The particle audio modifier uses the frequency response of an imported audio file to modify any (or all) of the three basic item transforms (Move, Scale and Rotate) when assigned to a particle source. This provides a means to animate the final elements in unique and interesting ways, that synchronize to the music. The way it works is by dividing the audio spectrum into the same number of samples as the number of particles. At each designated sample location the amount of audio variance is converted to a percentage and applied to the desired transform channel.

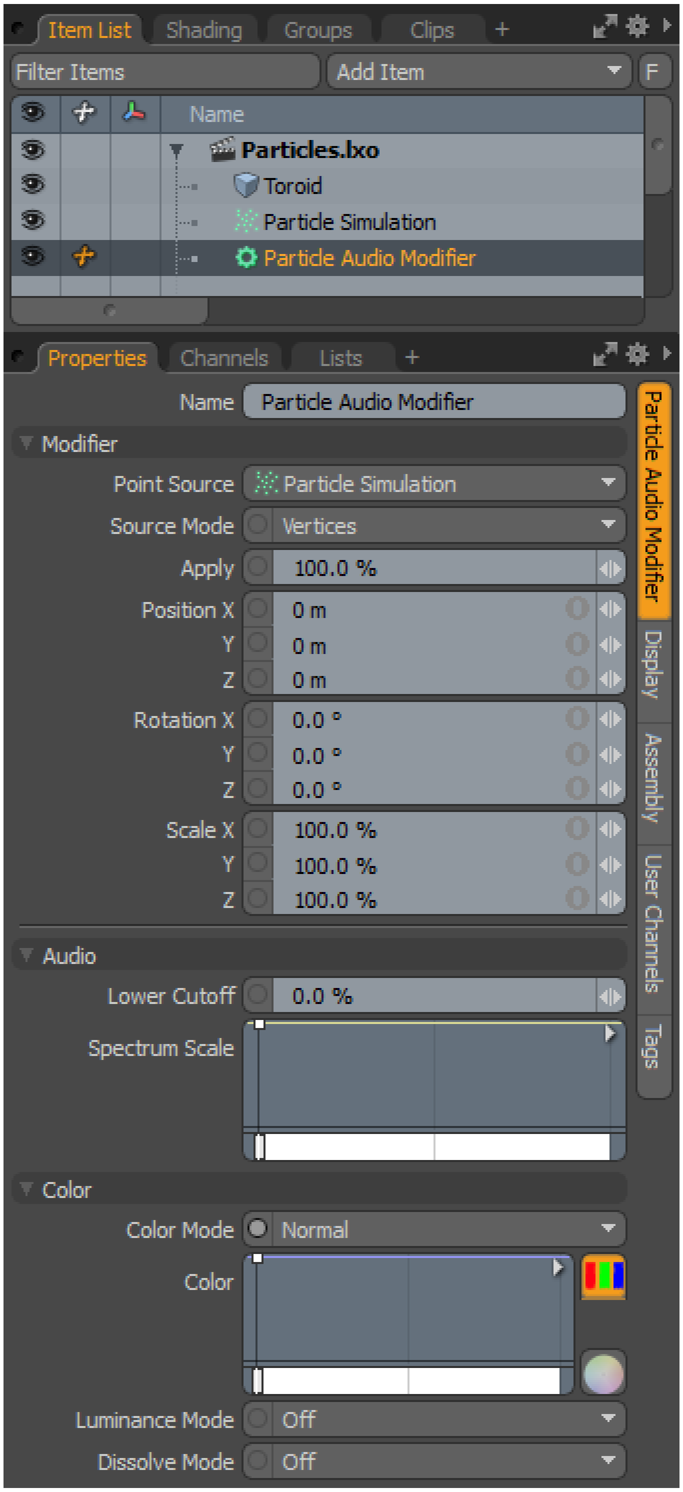
Adding a Particle Audio Modifier

You can add a particle audio modifier item by using the **Add Item** function of the **Items** List. Click the button to open the menu, then select **Particles** > **Modifiers**, and double-click **Particle Audio Modifier**.

Next you need to connect the modifier to a point source and specify the audio file. You can do this by selecting the audio file in the **Audio** tab, then click **Add Selected** in the **Schematic** viewport.

Once added, you need to specify the point source that you wish to affect. Once the originating point source is defined, you also need to specify the particle modifier item as the source in the final item. If thought of like a chain, the originating particle generation is defined as the source in the modifier, and then the modified values are fed into the final effect, such as a Replicator, Blob, Volume or Sprite. The source must be defined in this order so as to produce the intended result.

When applying the audio modifier to an active particle simulation, you need to connect the modifier in the proper order within the **Schematic** viewport. For existing networks, simply select the connector between the simulation item and the target and add the modifier. The audio modifier is found in the **Schematic** view in the **Add...** menu under **Particles** > **Modifiers** > **Particle Audio Modifier**. When selected, the following attributes are available in the **Properties** panel.


Item List
Shading
Groups
Clips
Filter Items
Add Item
F
Name
Particles.lxo
Toroid
Particle Simulation
Particle Audio Modifier
Properties
Channels
Lists
Particle Audio Modifier
Display
Assembly
User Channels
Tags

Name: Particle Audio Modifier

Modifier

Point Source: Particle Simulation

Source Mode: Vertices

Apply: 100.0 %

Position X: 0 m

Y: 0 m

Z: 0 m

Rotation X: 0.0 °

Y: 0.0 °

Z: 0.0 °

Scale X: 100.0 %

Y: 100.0 %

Z: 100.0 %

Audio

Lower Cutoff: 0.0 %

Spectrum Scale

Color

Color Mode: Normal

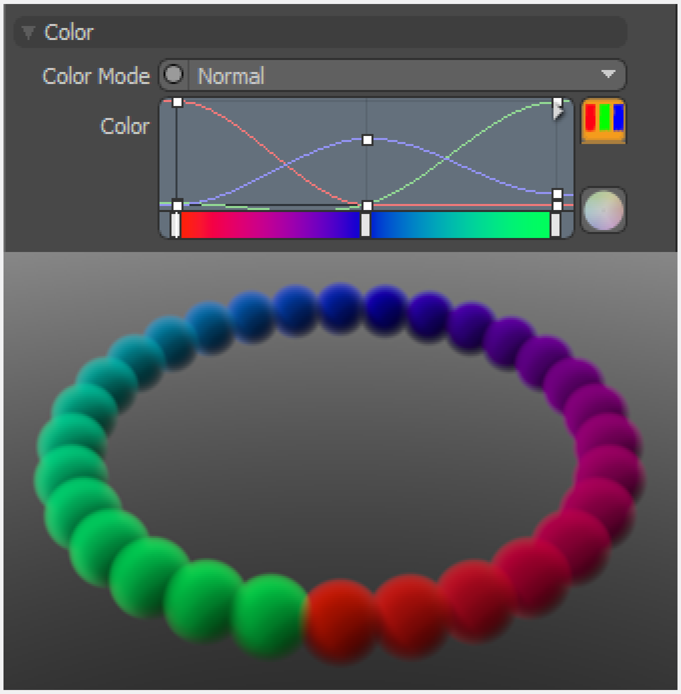
Color

Luminance Mode: Off

Dissolve Mode: Off

Option	Description
Name	Displays the current item name. To change it, click on the field and type in the new

Option	Description
	name.
Modifier	
Point Source	Defines the source of the particles that are affected by the modifier.
Source Mode	Specifies the way the particle modifier affects the source. **Vertices** uses the point positions, **Polygons** uses the polygon centers, and **Detached Vertices** uses the point positions of vertices that are not part of polygons.
Apply	Acts as a multiplier, controlling the overall strength of the transforms applied to the source particles.
PositionX/Y/Z	Determines a position value for each individual axis as an offset from the particle's originating position.
RotationX/Y/Z	Determines a rotation value for each individual axis as an offset from the particle's originating orientation.
Scale X/Y/Z	Determines a scaling value for each individual axis as an increase/decrease from the particle's initial size.
Audio	
Lower Cutoff	Defines a minimum threshold in the frequency range where anything below the defined threshold is ignored. This is useful to eliminate some slight motion that might appear in a noisy audio file.
Spectrum Scale	A multiplier that determines the influence of a given frequency. Lower frequencies are on the left of the gradient and higher ones to the right. This can also be used as a virtual crossover to completely cut out certain frequency ranges, for example, the audio is driving a woofer animation, and a key can be added to cut out all of the high end frequencies.
Color	Allows you to use mini gradient editors to assign and control different attributes for the resulting particles. These attributes are then passed along to the final effect, working with both volumes and sprites. The gradient values are generated sequentially, dividing the gradient itself into the number of generated particles it is assigned to, each receiving the value for that position along the gradient. For example, a red to blue gradient assigned to a linear array of particles produces one red particle at the start. The particle colors then fade toward the blue color in even increments along the length of the array.

Option	Description
	There are three different settings that can be applied to their associated surface attributes. Any of these functions are enabled by selecting any option other than **Off** (the default value that disabled the function). The selected **Mode** determines the blending of the gradient with the base setting of the target element.
Color Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle coloring. The **Color** gradient editor that opens allows you to define the actual gradient values that are applied.
Luminance Mode	Determines how the resulting gradient applies, defining the blending mode of the luminosity value as it applies to the base particle luminosity. The **Luminosity** gradient editor that opens allows you to define the actual gradient values that are applied.
Dissolve Mode	Determines how the resulting gradient applies, defining the blending mode of the colors as they apply to the base particle transparency. The **Dissolve** gradient editor that opens allows you to define the actual gradient values that are determine the transparent amount.  Color Mode: Normal A **Normal** color mode gradient applied to a particle generator's radial array

Point Clouds

Point clouds are procedurally created groups of particles that can be assigned as point sources in the same way any particle simulation can be. They are differentiated from particle simulations in that any animation assigned or applied is either pre-generated, as is the case with the RealFlow particle importer, or kinematic in nature (meaning it is assigned with keyframes, not through an evaluated simulation).

The particle generator allows you to easily create ordered arrays of particles, while the point cloud generates a volume of random particles. The surface particle generator is used to create randomly placed particles over a geometric surface. Once created, you can assign these items as point sources for Replicators, Blobs, Sprites or Volumes.

CSV Point Cache

The CSV point cache item allows particle data to be stored and read from an external series of CSV (comma-separated value) files. CSV files are plain text files and can be easily created by scripts or from many spreadsheet or database programs. You can also easily save simulation data from Modo directly into a CSV file for later reuse.

Usage

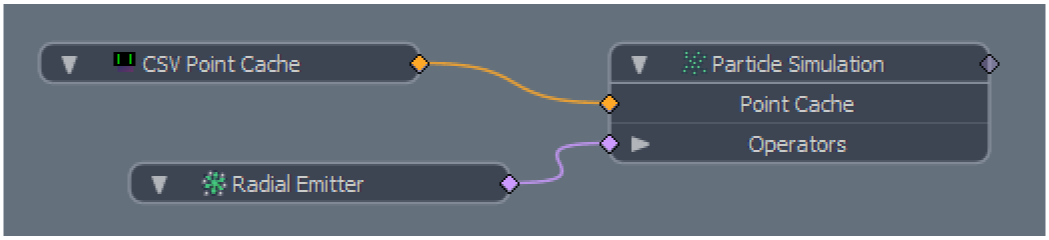
Add a CSV point cache item to the **Items** list for either reading or writing cache data by clicking **Add Item**, and under **Particles** > **Point Clouds**, double-click **CSV Point Cache**. Once added, select the resulting CSV point cache item to make its attributes visible in the **Properties** viewport.

To save a new CSV point cache sequence from an existing particle simulation follow these steps.

1. Create the CSV point cache item and click the **New Sequence** button.

   An OS-specific dialog opens, specifying the file name and location (numerals for a sequence are added automatically).
2. Click **Save** to close the dialog.

The actual connection to the point cache and the particle simulation is done next in the **Schematic** viewport. Use the **Add Selected** button in the viewport to add the proper node, where its output can be connected to the point cache input of the particle simulation item like this.


CSV Point Cache
Particle Simulation
Point Cache
Operators
Radial Emitter

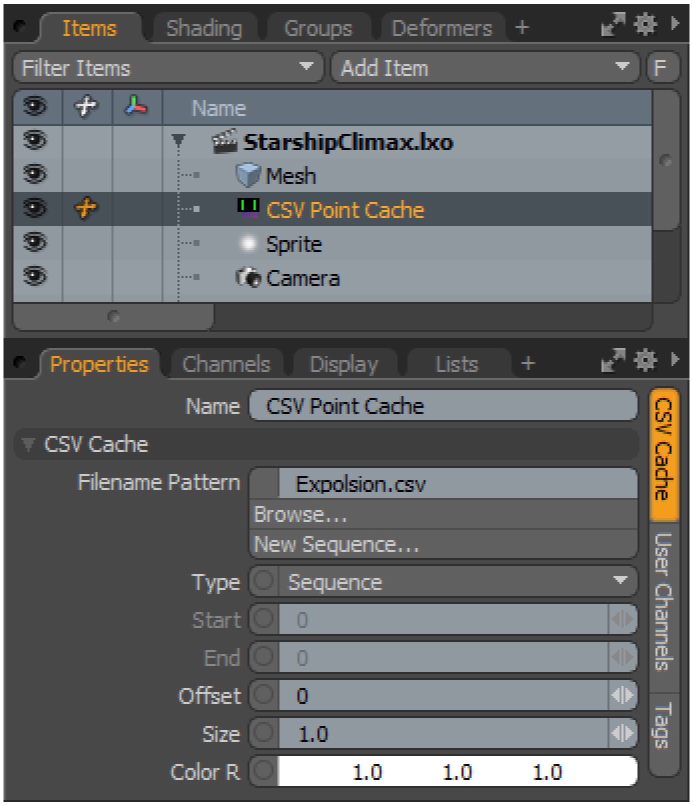
Once connected, each time a simulation is evaluated, the files are written (or overwritten as the case may be. To keep the files form being overwritten, simply disconnect the CSV point cache item from the particle simulation item. Once disconnected, the particles play back normally in the 3D viewport when the Timeline is played or scrubbed.

To load an existing CSV point cache, do the following:

1. Create the CSV point cache item and use the **Browse** button to locate a saved sequence.
2. In the OS-specific dialog, specify the first file in a series for a sequence. Once the file is selected press **Open** to load.

   When the Timeline is played or scrubbed, the particle sequence plays in the 3D viewport.

For static points in a still image, specify the particular file in the series (if necessary) and set the **Type** as **Static Points**. Once the point cache is loaded, it can be assigned as a point source.


Items
Shading
Groups
Deformers
Filter Items
Add Item
F
Name
StarshipClimax.lxo
Mesh
CSV Point Cache
Sprite
Camera
Properties
Channels
Display
Lists
CSV Cache
User Channels
Tags
Browse...
New Sequence...

Name: CSV Point Cache

CSV Cache

Filename Pattern: Expolsion.csv

Type: Sequence

Start: 0

End: 0

Offset: 0

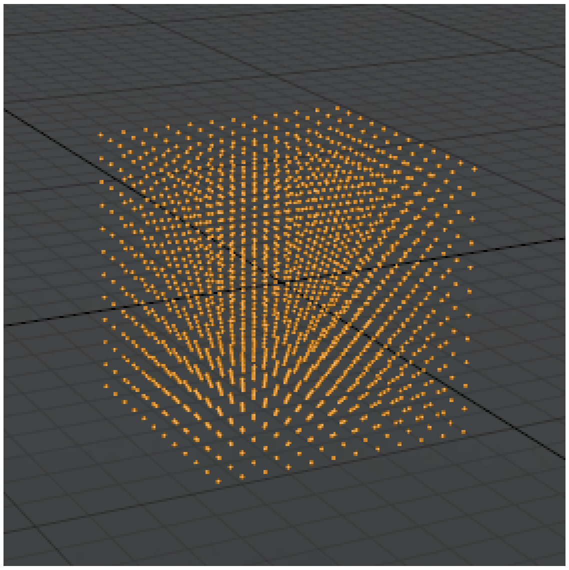
Size: 1.0

Color R: 1.0 1.0 1.0

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
CSV Cache	
Filename Pattern	A file path for a single CSV file, or for a sequence of CSV files. If it's a sequence, then the pattern must contain **[####]**, where the number of **#** characters indicates the number of digits in the frame number.
Browse	Opens an OS-specific dialog allowing you to locate an existing saved CSV file or sequence for playback. Once selected, click the **Open** button to load the cached data.
New Sequence	Allows you to define a new CSV sequence for saving cached data to. The particle simulation item needs to be connected to the CSV item in the **Schematic** view prior to running a simulation.
Type	Indicates how the pattern is used: • **Static Points**: Read the pattern as a single file for all time. • **Sequence**: Substitute a number into the pattern to find the file for a specific time. • **Sequence with Range**: Uses the **Start** and **End** values to limit the range of the sequence.
Start/End	**Start** is the minimum frame number for a range sequence. **End** is the maximum frame number for a range sequence.
Offset	Used to offset the sequence relative to the Timeline. If set to 10, the sequence starts to play back at frame 10.
Size	Determines the pixel display size of the particles within the 3D viewport.
Color	Allows you to define the display color of the particles in the 3D viewport. This option can be very helpful in differentiating particles from those that are part of a different simulation.

Tip: Modo normally saves cached simulation data internally to the saved scene file. CSV point cache files are a great way to externalize particle simulation data, keeping Modo scene files small.

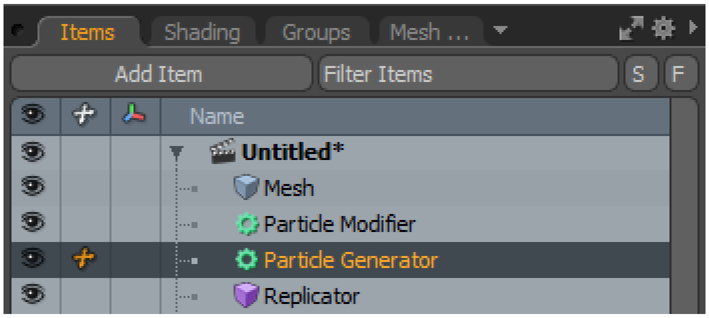
Particle Generator



The particle generator produces arrays of particles that can then be used as a point source for Replicators. They can also be used with Blobs, Sprites, and Volumes (though the results may not be quite as interesting). These arrays can be further modified with the Particle Modifiers that apply further transformations upon the cloned elements.

Adding a Particle Generator

To add a particle generator item, in the **Items** list, click **Add Item** to open the **Preset Browser**, expand **Particles** > **Point Clouds** and double-click **Particle Generator**.


Items
Shading
Groups
Mesh ...
Add Item
Filter Items
S
F
Name
Untitled*
Mesh
Particle Modifier
Particle Generator
Replicator

Once added, you can specify the type of particle array to generate and using the additional options, determine the layout and density of the array (see Generator Properties).

The resulting points are not visible to the user in the 3D viewport or the render output until a Replicator is applied and a source mesh is specified (or another effect that requires a point source is applied). See Replicators for more information on defining point sources.

Transform Properties

The Particle Generator has a built-in locator to position, rotate, and scale the particle array. For a full description of these properties and how to use them, see Locator: Transform Properties.

Generator Properties

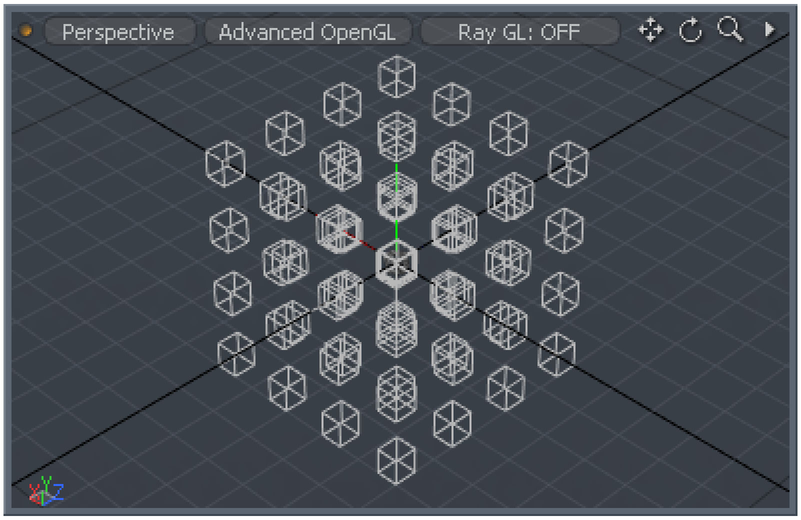
Type: This option determines the style of array that is generated.

- **Array**: produces a 3-D grid of evenly spaced particles.
- **Radial**: produces a circular array of evenly spaced particles.
- **Linear**: produces a straight line of evenly spaced particles.

Center: Check this to center the particle cloud at the origin. If it is unchecked, the particle cloud will start with one corner at the origin.

Particle IDs: Select how to assign the numeric particle IDs. Choose **Sequential** to use incremental values through the array. Choose **Random** to assign random values.

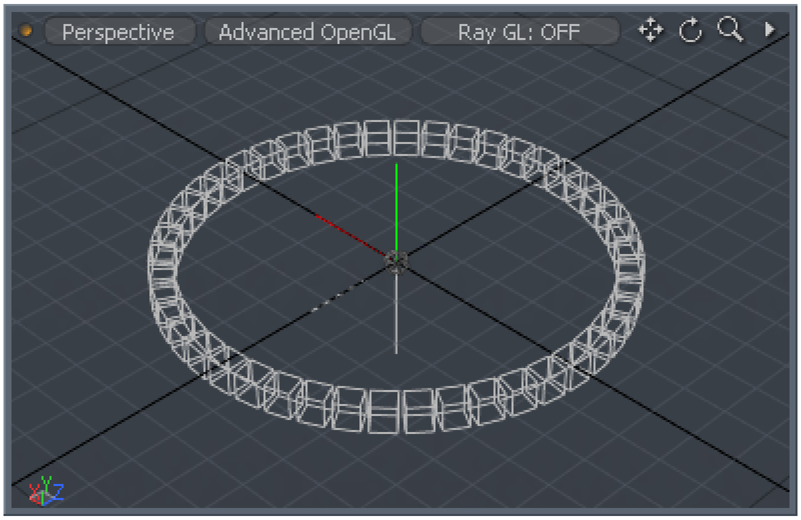
The following options appear when **Type** is set as **Array**.



Option	Description
Count	Determines the number of elements generated along each axis.

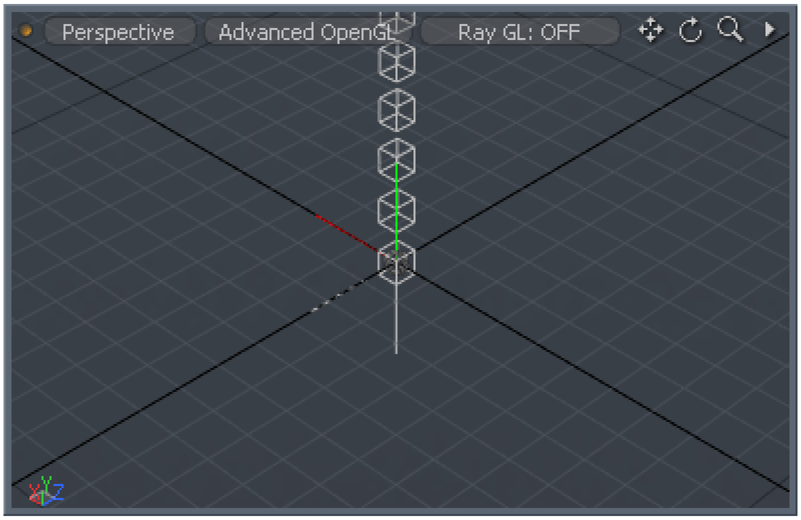
Option	Description
X/Y/Z	
Length X/Y/Z	The three values determine the overall size of the volume where the particles are evenly spaced within.

The following options appear when the **Type** is set as **Radial**.



Option	Description
Count	Determines the number of elements generated around the perimeter.
Align	When enabled, the subsequent cloned elements are rotated to align along the rotation of the circle. When disabled, the elements retain their initial world axis rotation.
Radius	Determines the outside diameter of the circle of particles, calculated as the radius (length from the center to the edge).
Angle Start/End	Instead of having the generated elements evenly spaced around the entire length of the circle, you can limit the distribution to a specific arc segment by defining an **Angle Start** and **Angle End** value.
Offset	This value offsets the initial clone (and all subsequent clones) from the **Angle Start** position the specified amount. Animate this value to get the elements to travel around the circle.

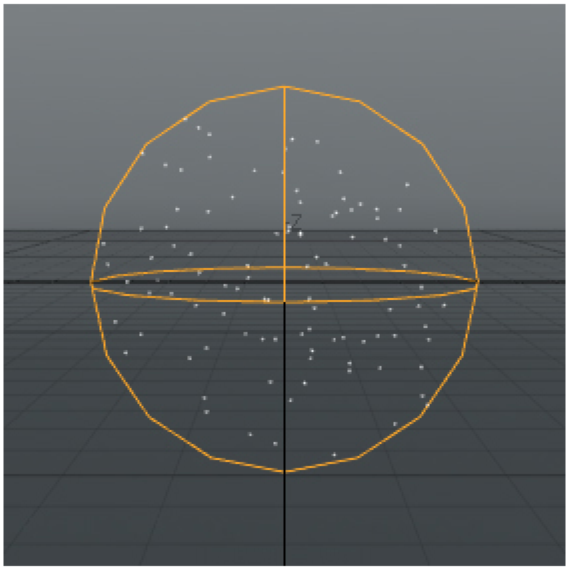
The following options appear when the **Type** is set as **Linear**.



Option	Description
Count X/Y/Z	Determines the number of elements generated along the length of the line.
Length X/Y/Z	Determines the distance between each individual element for each individual axis.
Scale X/Y/Z	Determines a scale value for each individual axis. Assigned as a step, where each subsequent particle inherits the scale of the previous element, producing a slowing scaling up/down across the total length.
Rotate X/Y/Z	Determines a rotation value for each individual axis. Assigned as a step, where each subsequent particle inherits the rotation of the previous element, producing a twisting of the elements across the total length.
Step Rotate X/Y/Z	Different from the **Rotate** value, which rotates individual elements from their center positions, the **Step Rotate** offsets the rotation of each element producing a corkscrew/spring effect.

Note: All the various channel values can be animated over time. When combined with multiple instances of particle arrays, all animating in sync to one another, you can achieve very interesting and unique effects.

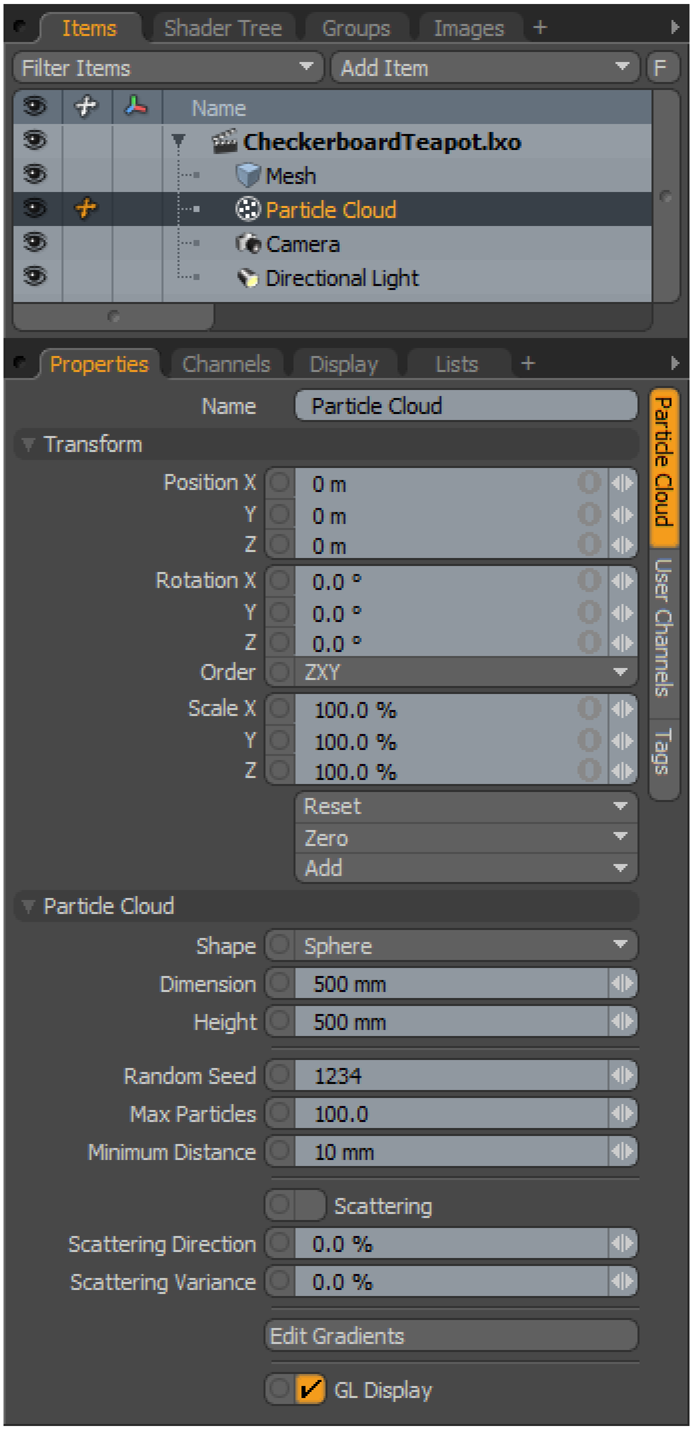
Particle Cloud



The **Particle Cloud** item generates a randomized array of single vertices. Points won't render by themselves, however, they can be used as a point source for a variety of items, such as for Blobs, Sprites and Volumes, or when working with Replicators. Animation can be added to the particles within the cloud by adding gradient controls over times (keyframes) with the **Scattering** function.

Adding a Particle Cloud

You can add a **Particle Cloud** item simply by using the **Add Item** function in the **Items** list. Click on the button to open the menu, and under **Particles** > **Point Clouds** , double-click **Particle Cloud**. Once added, you can then specify the item from the **Point Source**, or **Particle Source** options from within the appropriate properties panel of the related function.


Items
Shader Tree
Groups
Images
+
Filter Items
Add Item
F
Name
CheckerboardTeapot.lxo
Mesh
Particle Cloud
Camera
Directional Light
Properties
Channels
Display
Lists
+
Particle Cloud
User Channels
Tags
Reset
Zero
Add
Edit Gradients

Name: Particle Cloud

Transform

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Particle Cloud

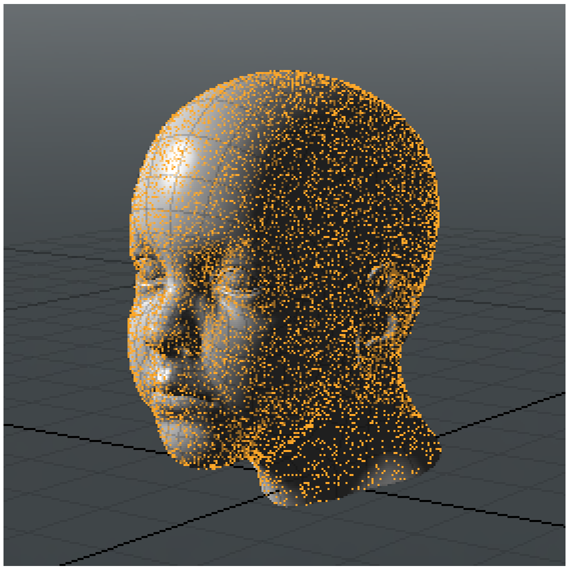
Shape: Sphere
Dimension: 500 mm
Height: 500 mm
Random Seed: 1234
Max Particles: 100.0
Minimum Distance: 10 mm
☐ Scattering
Scattering Direction: 0.0 %
Scattering Variance: 0.0 %
☑ GL Display

Option	Description
Name	This data field displays the current item name. Users may easily change it by left-clicking within the field and typing the new name.

Option	Description
Transform	
Position	An Item transform that allows the user to numerically position the item in XYZ space.
Rotation	An Item transform that allows the user to numerically set the rotation of the item. Rotation transforms originate from the Position value.
Order	Allows the user to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An Item transform that allows the user to numerically set the size of the item. Scale transforms originate from the Position value.
Reset	Resets the selected transform values to (0,0,0) returning the items back to their default state.
Zero	Resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding a negative transform item to the Mesh items channels.
Add	Transform Items are the channel groups associated to an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the Properties panel). This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values input fields). This action causes the particular transform item to be added automatically to the 'Channels' viewport list. The 'Add' function here can also be used to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values (a necessary step for 'Referencing', in order to override the channels, they must first exist).
Particle Cloud	
Shape	Determines the overall shape of the containing volume for the particles, either a **Sphere**, **Cube**, or **Cylinder**.
Dimension	Determines the size of the volume as a radius, so a 500 mm **Dimension** would produce a 1m wide container volume for the particles.

Option	Description
Height	When the **Cylinder** option is selected, the **Height** value determines the overall length of the cylinder shape.
Random Seed	The **Random Seed** value is the initial number used when generating the random procedural values. Each Seed produces different random results.
Max Particles	Determines the maximum number of total particles generated within the volume.
Minimum Distance	Determines the minimum amount of distance between two particles within the volume.
Scattering	The **Scattering** option is provided to allow for simple animation of the particles. When *enabled* each individual particle is scattered in a random direction, based on the **Scattering Direction** and **Scattering Variance** values. In order to see any changes in the 3D Viewports, it is necessary to keyframe some values using the **Edit Gradients** option.
Scattering Direction	The **Scattering Direction** option provides some additional control of the overall random distribution of the particles.
Scattering Variance	The **Scattering Variance** option controls how much variation there is in distance when the particles are scattered. Higher values move some particles further from the volume than others, while smaller values tend to scatter particles in more uniform distances.
Edit Gradients	Clicking the **Edit Gradients** button opens a Gradient Editor window, allowing you to set up gradient controllers for Particle density within the cloud as well as the Scattering Distance.
GL Display	Toggles the visibility of the 3D viewport proxy representation. Enabled (visible) by default.

Surface Particle Generator

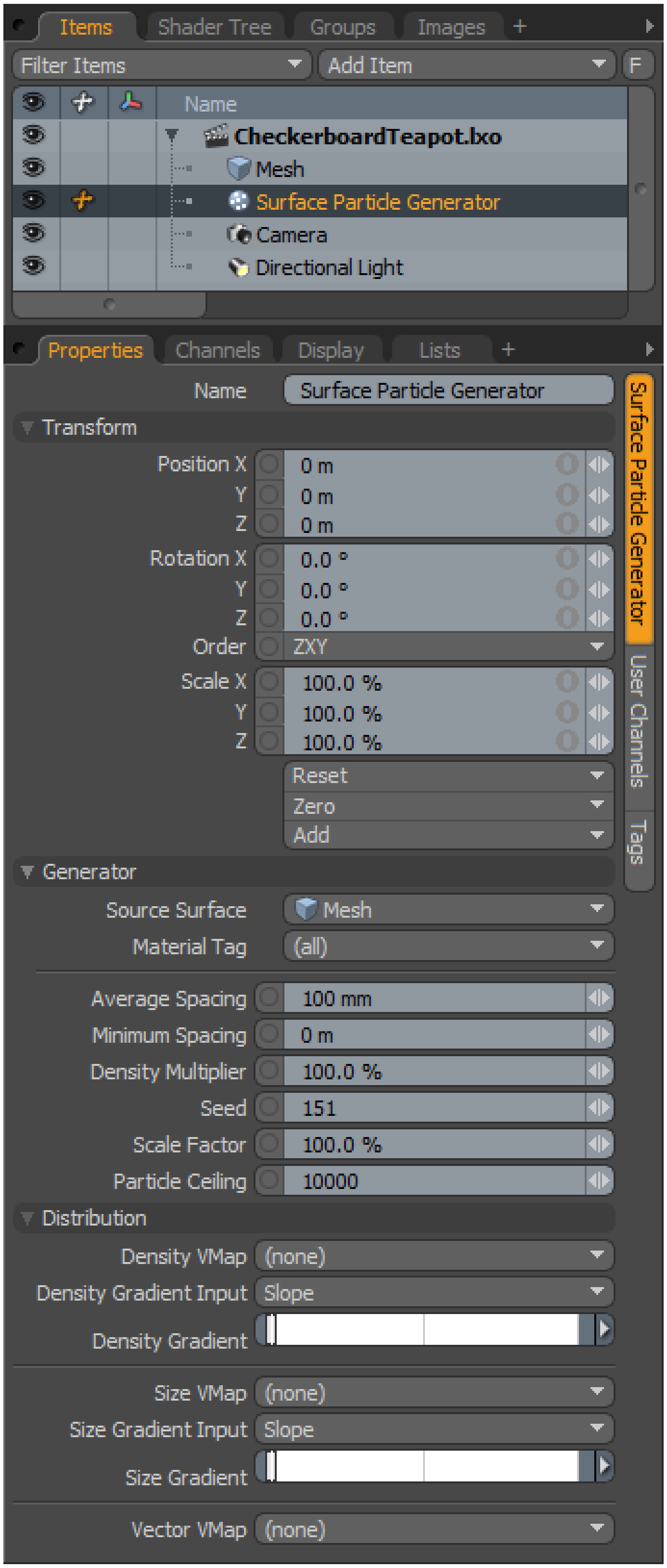


The surface particle generator produces similar results to the **Shader Tree** Surface Generator, in that it is used to scatter vertices (particles) across a specific item or surface. However, the particle generator is evaluated at the scene level making it more flexible and faster. The resulting particles can then be used as a point source for Blobs, Sprites, Volumes,Replicators and when working with Texture Locator items.

> **Note:** Since the surface particle generator is scene-evaluated, it doesn't take rendering effects, such as displacements applied to the surface into consideration; in that case you need to apply a surface generator.

Adding a Surface Particle Generator

You can add a surface particle generator item in the **Items** list, by clicking **Add Item**, and under **Particles**, double-clicking **Surface Particle Generator**. Once added, you can specify the target geometry using the **Source Surface** option. The **Material Tag** option further limits the particles to specific areas of a Mesh Item.


Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
Properties
Channels
Display
Lists
Surface Particle Generator
User Channels
Tags
Transform
Generator
Distribution
Reset
Zero
Add

CheckerboardTeapot.lxo
Mesh
Surface Particle Generator
Camera
Directional Light

Name: Surface Particle Generator

Transform
Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Generator
Source Surface: Mesh
Material Tag: (all)
Average Spacing: 100 mm
Minimum Spacing: 0 m
Density Multiplier: 100.0 %
Seed: 151
Scale Factor: 100.0 %
Particle Ceiling: 10000

Distribution
Density VMap: (none)
Density Gradient Input: Slope
Density Gradient
Size VMap: (none)
Size Gradient Input: Slope
Size Gradient
Vector VMap: (none)

Option	Description
Name	Displays the current force item name. To change it, click on the field and type in the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space.
Rotation	An item transform that allows you to numerically set the rotation of the item. Rotation transforms originate from the Position value.
Order	Allows you to set the order that rotations are applied to the item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. Scale transforms originate from the **Position** value.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and mesh position intact. This is done by adding a negative transform item to the Mesh Item's channels.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values input fields). This action causes the particular transform item to be added automatically to the **Channels** viewport. The **Add** function can also be used to add the selected set of transforms to the **Channel** viewport while keeping the default **0,0,0** values (a necessary step for referencing, in order to override the channels, they must first exist).
Generator	
Source Surface	Specifies an item within the scene to apply the particles over its surface. This is the equivalent of an item mask in the **Shader Tree**.

Option	Description
Material Tag	This option allows you to further limit the application of the particles by specifying a surface definition from the drop-down menu. This is the equivalent of a material mask in the **Shader Tree**.
Average Spacing	Specifies the average distance between particles.
Minimum Spacing	Specifies the minimum amount of space between replicated items, especially useful in eliminating overlaps for neighboring replicas when used as a Replicator source.
Density Multiplier	Offers simplified control over the overall density of replicated items. Lower values reduce the overall number of replicas while higher values increase the overall number of replicas up to the **Particle Ceiling** value.
Seed	An initial setting used when random numbers are generated. Different **Seed** values produce different random variations, in this case, initial particle positions. However, you can use the same **Seed** value when it is necessary that items retain the same variations.
Scale Factor	Controls the scale of replicated objects, eliminating the need to manually scale the base prototype item. With a value of 100%, replicated items are generated at the identical size of the prototype, smaller values decrease each replica's size, while larger values increase each replica's size.
Particle Ceiling	A maximum for the total number of generated particles.
Distribution	
Density Vmap	You can create an embedded Weight Map on the target surface and use its values to control the density of the particle distribution, eliminating the need to create custom image maps to do so. Any available Weight Maps can be selected from this drop-down, assigning it to control the density as a multiplier over the regular density. A weight value of 1 (100%) is the normal density, attenuating toward a weight value of 0 at 0% density. Weight values can also go above 1 for added density control.
Density Gradient Input	Defines an input criterion that is used to additionally control the density of the resulting particles. With the input defined, the actual **Density Gradient** defines the density amount.
Density Gradient	Using a mini-gradient input, the density can be modified with gradient keys that

Option	Description
	represent a percentage of the original density. Much like how a keyframe defines a value in time, these keys represent a value across the selected input parameter (such as over the angle of **Slope**). You can add keys by middle-clicking along the length of the gradient input and then clicking and dragging the key icon up or down, adjusting the value. The grayscale bar directly below gives visual feedback as to how the values fade from one key to the next; white produces 100% density attenuating toward black at 0% density. Opening the **Gradient Editor** allows you to go above 100%.
Size Vmap	You can create an embedded Weight Map on the target surface and use its values to control the scale of the elements connected to the generated particles, eliminating the need to create custom image maps to do so. Any available Weight Maps can be selected from this drop-down, assigning it to control the scale as a multiplier over the regular sizing. A weight value of 1 (100%) is the normal scale, attenuating toward a weight value of 0 at 0% scale. Weight values can also go above 1 for greater than 100% scale control.
Size Gradient Input	Defines an input criterion that is used to additionally control the scale of the elements connected to the generated particles. With the input defined, the actual **Size Gradient** defines the scaling amount.
Size Gradient	Using a mini-gradient input, the scale can be modified with gradient keys that represent a percentage of the original scale. Much like how a keyframe defines a value in time, these keys represent a value across the selected input parameter (such as over the angle of **Slope**). You can add keys by middle-clicking along the length of the gradient input and then clicking and dragging the key icon up or down, adjusting the value. The grayscale bar directly below gives visual feedback as to how the values fade from one key to the next; white produces 100% scale attenuating toward black at 0% scale.
Vector Vmap	You can create an embedded vertex vector map on the target surface and use its values to control the orientation and scale of the elements connected to the generated particles, eliminating the need to create custom image maps to do so. This allows you to use the sculpting tools, like with hair guides, to intuitively control the replicated elements. **Note:** Elements controlled by vector Vertex Maps automatically adapt to surface deformations, as the vector maps are defined in tangent space.

Curve Particle Generator

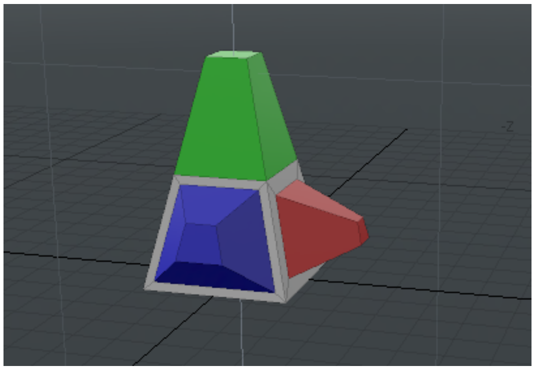
The **Curve Particle Generator** tool is a point source that places particles along the length of a curve.

The **Curve Particle Generator** has an input for a source curve to describe the path of the particles. You can draw a curve in a mesh and assign the mesh to the **Source Curve** property. If you have multiple curves in the source mesh, you can select one using **Curve Index**.

This tool itself does not display the points it creates, to see the results you must use Replicators or Volumes. To use a Replicator, assign the Curve Particle Generator to the **Point Source** setting. For Volumes, use the **Particle Source** setting.

Orienting the Particles

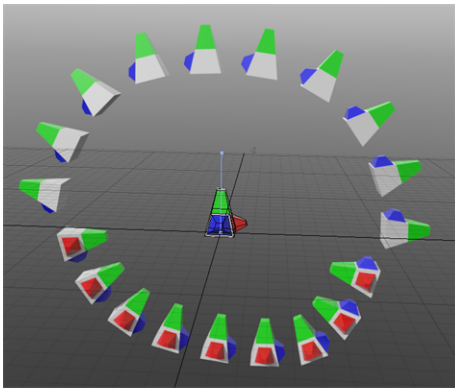
When orienting particles along the curve, the **Up Vector** setting, used in combination with the **Forward Axis**, offers a flexible method of setting the required position. Let's use a simple mesh to show how these settings work:



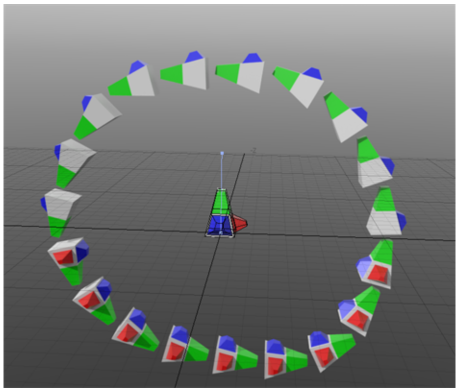
The mesh has colored parts to show its relationship to the world space. The red part points down the positive X-axis, green up the positive Y-axis, and blue down the positive Z-axis. This is our source mesh (prototype) for the replicator.

Using Forward Axis Controls

The **Forward Axis** determines the direction that the particles face along the curve. In the first image below , the **Forward Axis** is set to **Z**. Observe how the blue part points forward along the curve. This is because the blue part points down the Z-axis in the prototype mesh. The image on the right shows the **Forward Axis** set to **Y**. This time the green part leads the way as it is the green section that is pointing up the Y-axis in the source mesh.



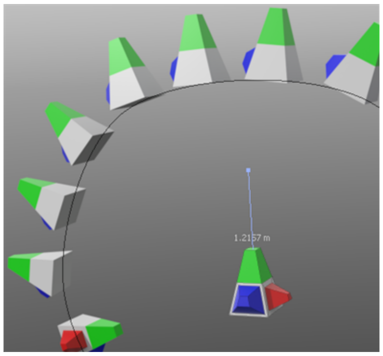
Forward Axis is **Z**.



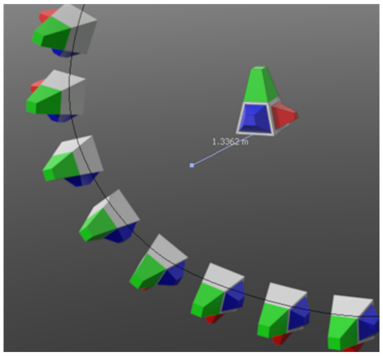
Forward Axis is **Y**.

Using Up Vector Controls

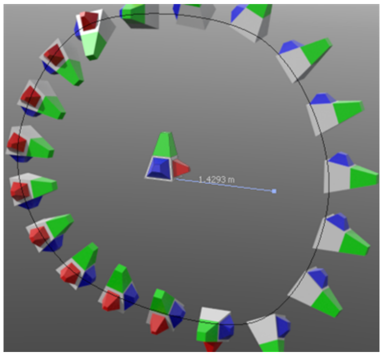
The Up Vector controls the secondary rotation of the particle. It describes the direction that the top of the particles point, when placed along the curve. For example, if the Up Vector is pointing up the Y-axis, the top of each particle attempts to point upwards as can be seen below. The Up Vector is indicated by the blue line, extending from the prototype. In all cases **Forward Axis** is set to **Z**.



The Up Vector is pointing up, in the same direction as the green top. The particles also point upwards.



The Up Vector is pointing along the positive Z-axis. The top of the particles now point the same way.

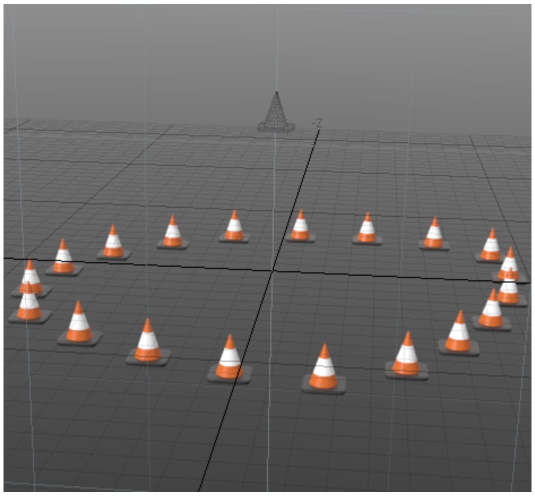


The Up vector is pointing along the positive X-axis.

Establishing the correct orientation depends on several variables, but in most cases it is a good idea to adjust the Up Vector to point through the plane of the curve. You can then rotate the prototype and change the **Forward Axis** to find the particle position that you want.

Note: The final position is subject to additional rotations by the **Forward Axis**, **Particle Rotation**, and **Rotation Shift**.

In the image below, the prototype is inactive and can be seen wireframed towards the center of the image. The Up Vector is (0, 1, 0) , which points perpendicularly through the ground plane, and the **Forward Axis** is set to **Z**. Also, **Align to Curve** is off to keep the particles at the orientation of the prototype. Alternatively, you could adjust the orientation by rotating the prototype in polygon, edge, or vertex mode.



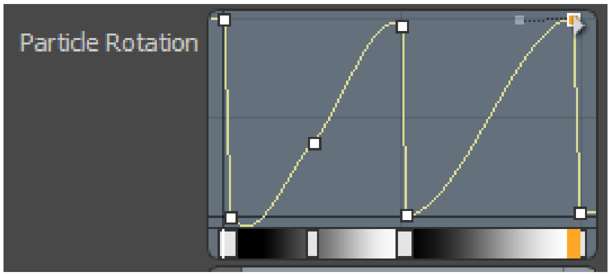
Warning: When rotating the source mesh you must be in polygon, edge, or vertex mode for the rotation to carry through to the particles. This because replicators ignore the item level rotation, position, and scale values from their prototypes.

Using Particle Scale and Particle Rotation

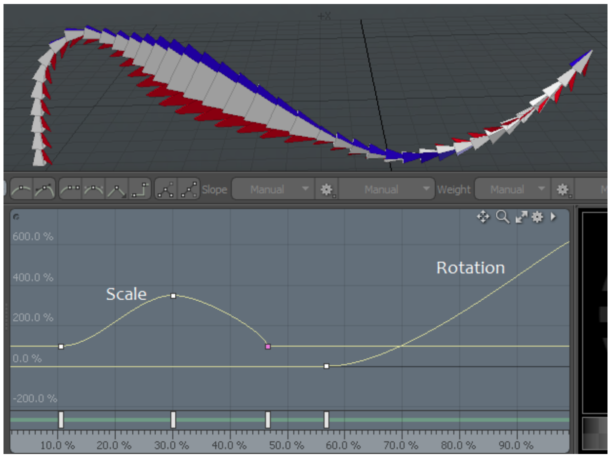
The Curve Particle Generator uses a Gradient Editor to determine variations in particle scale and rotation along the curve. The gradient can be displayed as a gray-scale color bar (where white represents the maximum value) or as a curve.

The image below shows an example for the **Particle Rotation**. The sawtooth pattern of the line indicates that successive particles are increasingly rotated until the maximum value (360^{o}) is reached, the rotation then resets to 0^{o}. This would appear as a smooth gradual increase in rotation along the curve, as a rotation of 360^{o} is equivalent to a rotation of 0^{o}.

Note: The Particle Rotation setting is used to achieve the rotation in the chain, pictured at the top of this topic.


Particle Rotation

The image below shows variation in rotation and scale as depicted in a full-blown Gradient Editor. Each curve is labeled with the name of the property. The result is shown at the top of the image.


Slope
Manual
Manual
Weight
Manual
600.0 %
400.0 %
200.0 %
0.0 %
-200.0 %
Scale
Rotation
10.0 %
20.0 %
30.0 %
40.0 %
50.0 %
60.0 %
70.0 %
80.0 %
90.0 %

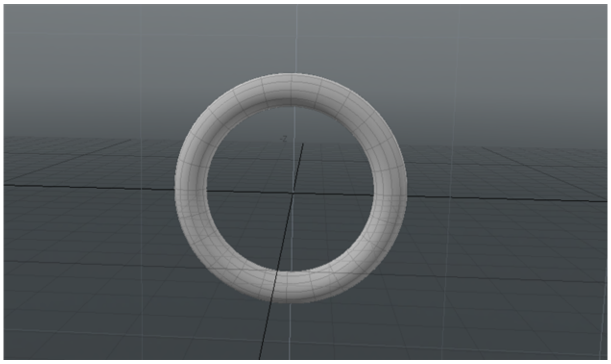
The properties are assigned to gradient channels. Gradients cannot be animated, which is why there is no channel animation button next to the property. However, there are **Scale Shift** and **Rotation Shift** settings, which offsets the gradient and can be animated.

Using the Curve Particle Generator

The following steps show you how to use the Curve Particle Generator to create the wooden chain image shown above.

> **Note:** If you already have a source mesh to replicate, and a curve in a separate mesh layer, skip to step 4.

1. First lets draw the source mesh, in this case let's use the Torus primitive to create the ring that is used as the link in the chain. On the left panel, on the **Basic** tab, click **Torus** and size it to look similar to the one below. The position of the ring won't affect the result.



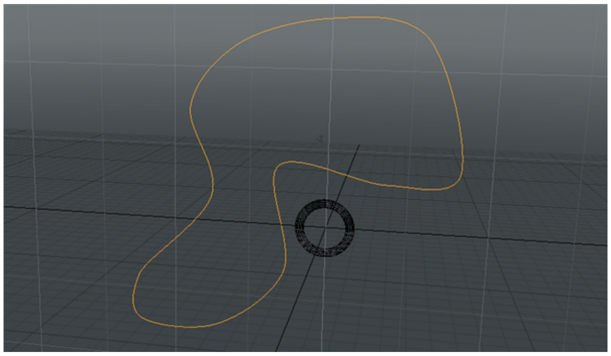
2. On the right panel, open the **Items** tab, click **Add Item**.
3. In the Procedural Preset Browser, expand **Items** on the left column, and double-click **Mesh**.

   A new mesh layer is added to the **Items** list, which will be used to hold the curve.
4. On the left panel, open the **Curve** tab, click Adding Curves, and enable the **Closed** property.

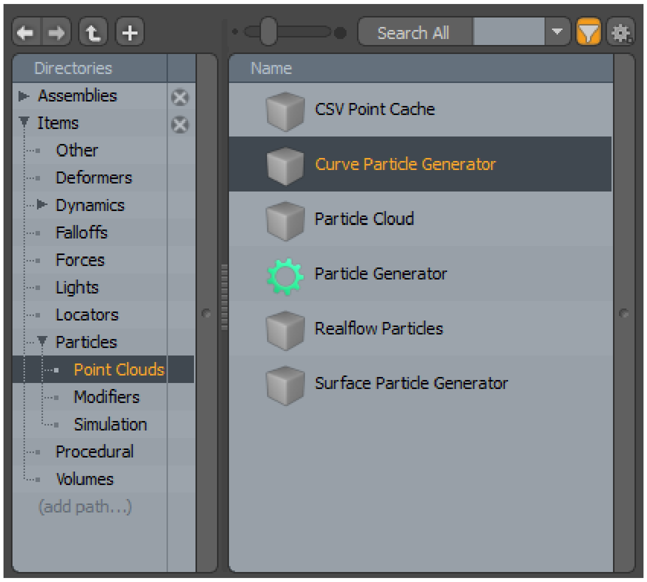
Tip: You can also close the curve by right-clicking on the first control point of the curve.

5. Click several times in the 3D viewport to draw a curve path. Press the **Spacebar** to drop the tool.

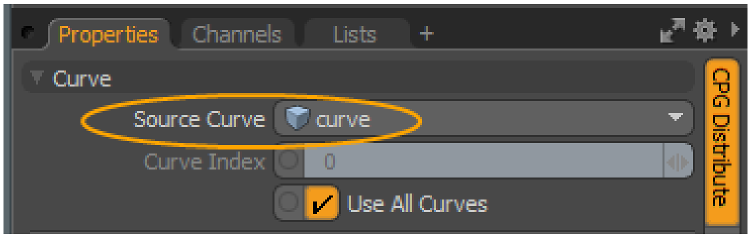
   The image below shows the curve used for the wooden chain (the inactive mesh for the ring can be seen in the background).



6. Add a Curve Particle Generator to the **Item List**: Click **Add Item** and select **Particles** > **Point Clouds**. Double-click **Curve Particle Generator**.


Search All
Directories
Assemblies
Items
Other
Deformers
Dynamics
Falloffs
Forces
Lights
Locators
Particles
Point Clouds
Modifiers
Simulation
Procedural
Volumes
(add path...)
Name
CSV Point Cache
Curve Particle Generator
Particle Cloud
Particle Generator
Realflow Particles
Surface Particle Generator

7. To assign the source mesh (the ring) to the generator, On the right panel, ensure the **Curve Particle Generator** property **Source Curve** is set to the mesh containing the curve.


Properties
Channels
Lists
+
CPG Distribute

Curve

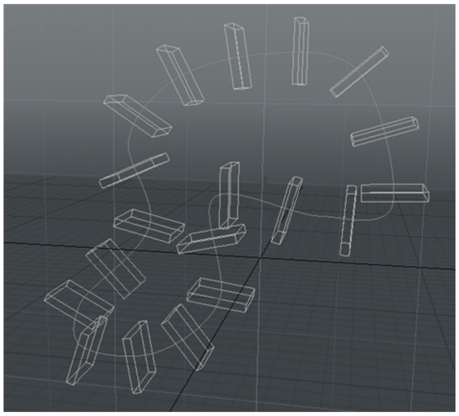
Source Curve: curve

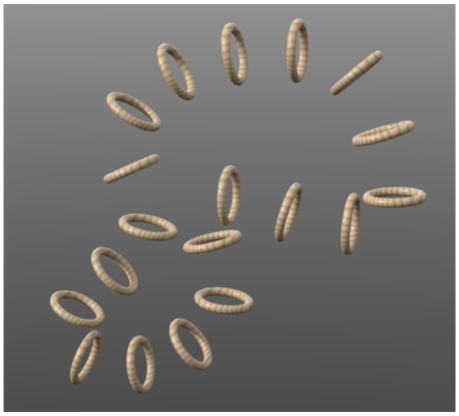
Curve Index: 0

☑ Use All Curves

8. To see the particles add a **Replicator**: Click **Add Item** and select **Items** > **Particles**. Double-click **Replicator**.
9. In the **Replicator** tab in the **Properties** tab, set **Prototype** to the mesh containing the ring. Set **Point Source** to **Curve Particle Generator**.

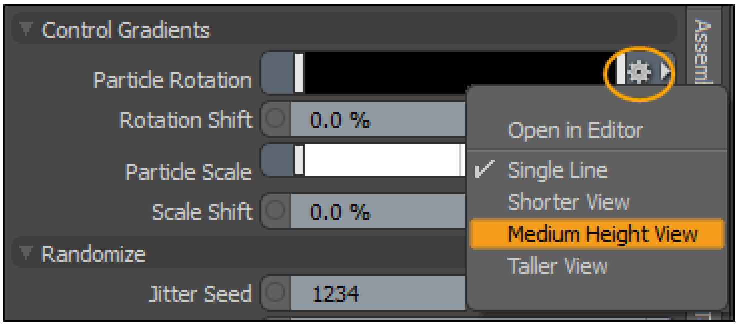
   The images below show the 3D viewport and the rendered output (the source ring layer is hidden).





Tip: To preview the particles without rendering, set **Ray GL** to **Full** or **Fast** in the 3D viewport. To hide the original Toroid item, click the eye icon in the **Items** list.

10. To make the chain look like the one at the top of this topic, make some adjustments to the settings of the Curve Particle Generator as follows:
    - In the **CPG Distribute** tab, increase the **Point Count** to **32**.
    - In the **CPG Control** tab, set the **Forward Axis** to **X**.
    - To make the links appear interlinked, adjust the **Particle Rotation**: Open the **CPG Control** tab, click the gear icon at the right of the **Particle Rotation** bar and select **Medium Height View**. This makes it easier to edit the rotation gradient.



Control Gradients

Particle Rotation

Rotation Shift 0.0 %

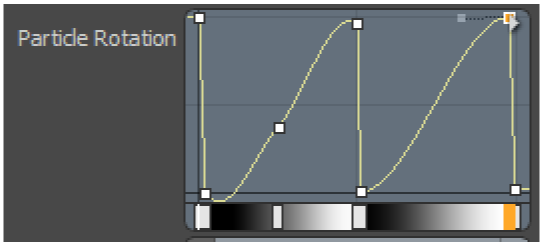
Particle Scale

Scale Shift 0.0 %

Randomize

Jitter Seed 1234

Right-click the key (small square) in the bottom right corner and enter the **500** in the **Value** field to create a linear increase. Add additional keys on the curve by middle-clicking on the curve and then reposition the keys to create a sawtooth type pattern similar to the one below. Right-click on the top keys to set the **Value** to **500** to see the results in the Preview Viewport. This should provide enough increase in rotation along the path to make the links appear interlinked.



11. Finally, add the wood texture: Click the **Shading** tab to open the Shader Tree. Click **Add Layer** and select **Textures** > **Wood**. Ensure the **Wood** layer is above the **Base Material**.
12. Press **F8** to open the Render Preview.

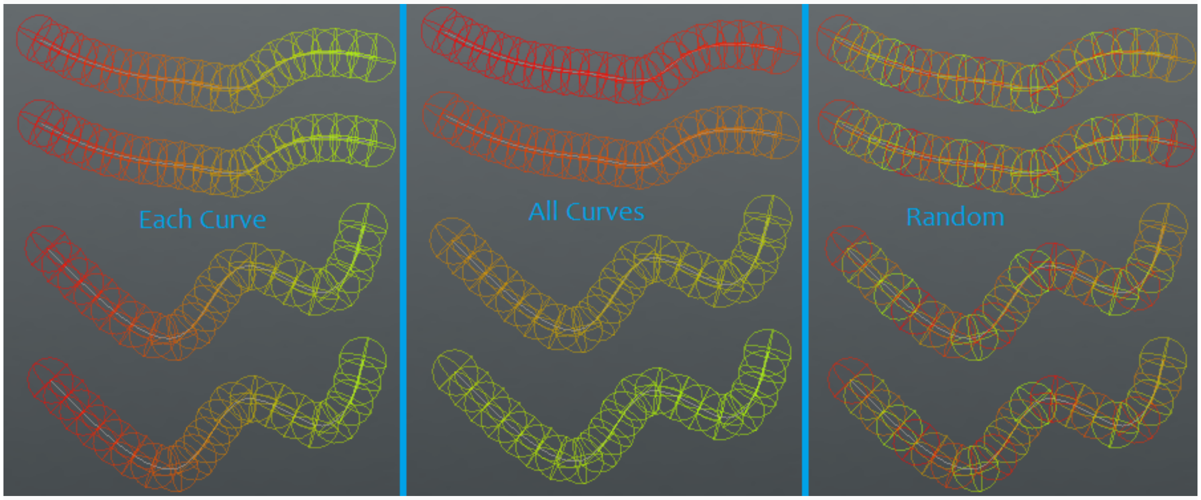
Curve Particle Generator Properties

Edit the following properties to customize the behavior of the Curve Particle Generator.

CPG Distribute Tab

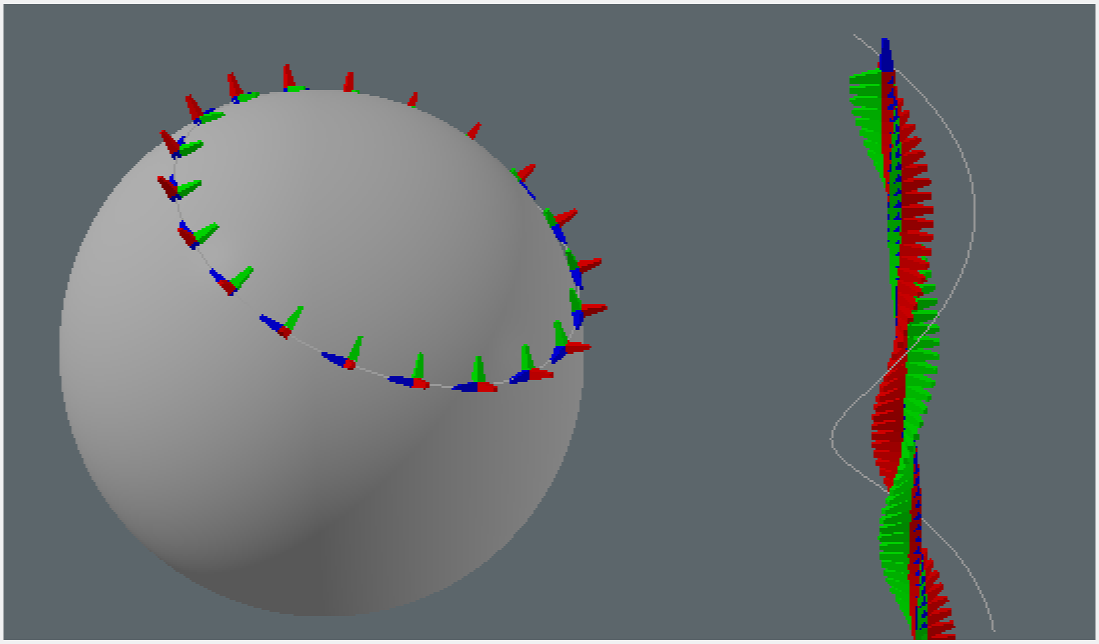
Curve	
Source Curve	Sets the curve to use as the path for the particles.
Curve Index	Selects the index of the curve to use from the source mesh. The index counts up from 0.
Use All Curves	When enabled, uses all curves from the source mesh. This overrides **Curve Index**.
Point Mode	Allows you to choose the particle distribution mode: **Point Count** - Sets the number of particles to place along the curve. This is the default setting. When enabled, set the Point Count value. **Spacing Distance** - Sets the distance between particles. The number of particles is then decided according to this distance. The beginning and ending points are maintained. Once enabled, set the Point Spacing value. **Exact Distance** - Sets the exact distance between particles. Once enabled, set the Point Spacing value. The last point disappears when it pushes off the edge of the curve.

Curve	
Point Count	Specifies the number of particles to place along each curve. This is enabled when **Mode** is set to **Point Count**.
Point Spacing	Specifies the distance between each particle. This is enabled when **Mode** is set to **Spacing Distance** and **Exact Distance**.
End Behavior	Sets how particles at the end of the curve behave. The following options are available: • **Cluster**- Particles stop at the end of the curve. • **Wrap** - Wraps particles from the end of the curve to the start. • **Project** - Particles exceeding the end of the curve are placed on a straight line past the end. • **Remove** - Particles exceeding the end of the curve are removed.
Offset Percentage	Sets the amount by which the particles are pushed up along the curve. At **0 %**, the particles are unmoved. At **100 %** the particles are pushed along by the full distance of the curve. If **Wrap** is unchecked, the particles will bunch up at the end of the curve. If it is checked, the particles will wrap back around and no bunching will occur.
Start Percentage	Sets the amount along the curve length at which to place the first particle.
End Percentage	Sets the amount along the curve length at which to place the last particle.
Particles IDs	
Particles ID Mode	Choose a particle ID mode: • **Span Each Curve** - Particle IDs range from 0-1 for each curve. • **Span All Curves** - Particle IDs range from 0-1 along all curves, , assigning each curve a chunk of the 0-1 range. • **Random** - Particle IDs are randomly assigned.

Curve	
	

CPG Control Tab

Name	Specifies the name of your curve particle generator.
Alignment	
Alignment Mode	Specifies the type of alignment mode to use: • **Automatic** - Particle automatically aligns to the curve, using the up vector of the particles. • **Curve Direction** - Particles align to a specified curve in the scene. Defines that curve in the Alignment Mode option. The particles point towards the chosen curve. Moves that curve and the particles follow. • **Mesh Normal** - Particles align to a selected object in your scene. Defines the object in the Alignment Mode option. The up vector of particles align to the surface normals of that object. • **Mesh Direction** - Particles align to a specified mesh in the scene. Defines the mesh item in the Alignment Mode option. The particles point towards the chosen mesh. Moves that mesh item and the particles follow. • **None** - Uses the forward axis to determine the particle's orientation.

	
Alignment Mesh	Selects the item used for up vector alignment.
Align Index	Specifies the index value. Enabled when **Curve Direction** Alignment Mode is enabled.
Forward Axis	Selects the source mesh axis that is to face forwards along the direction of the curve.
Reverse Axis	Reverses the direction of the particles based on the **Forward Axis** specified.
Up Vector X, Y, Z	Defines the outward facing direction of the curve vector used to determine the particle rotations. Usually, this should be the vector crossing through the plane of your curve.

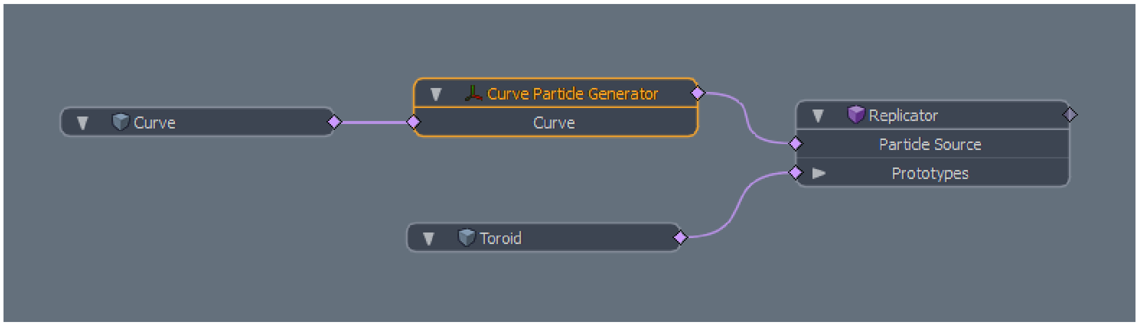
Control Gradients	
Particle Rotation	Sets the change in particle rotation along the curve (through the **Forward Axis**) by adjusting the **Rotation Gradient** channel. The horizontal axis represents the full length of the curve. The **Particle Rotation** can be set using one of five interface options. Click the small triangle on the left to select one of five interface options: • **Open in Editor** - Opens the Gradient Editor. • **Single Line** - Displays a line view using a grayscale gradient to represent the amount of rotation. Black indicates 0 % and white indicates 100 %. You can add a control point by middle-clicking on the

	line. To set the value of the control point, double-click on the control point to open the **Set Key Value** dialog, then edit the **Value** setting. • **Shorter View** - Displays a small graph representation of the gradient. You can drag control points and add new control points as you would in the Gradient Editor. • **Medium Height View** - Displays the same representation as **Shorter View** but with more vertical space. • **Taller View** - Displays the tallest graph view.
Rotation Shift	Specifies the amount along the curve path to shift the effect of the rotation. The gradient is offset by this amount from the start of the curve, though it wraps back from the end of the curve.
Particle Scale	Sets the scale of the particles along the curve by adjusting the **Scale Gradient** channel. The **Particle Scale** is set using the same interface options as the **Particle Rotation**. Click on the small white triangle on the left to reveal the options.
Scale Shift	Sets the amount along the curve to shift the effect of the scale gradient. The gradient is offset by this amount from the start of the curve, though it wraps back from the end of the curve. **Note:** Both the **Rotation Shift** and **Scale Shift** support animation.

Align to Curve	When enabled, orientates the source mesh so that it is aligned to the curve at the point that it is placed. Leave this unchecked to maintain the original orientation.
Randomize	
Jitter Seed	When enabled, generates the random values used by **Position Jitter**. Try varying this to generate alternate random outcomes.
Radial Position Jitter	Sets the degree of random variation in the position of each particle. Using this you can create an uneven distribution of particles.
Linear Position Jitter	Sets the position of the particles along the curve.
Rotation Jitter	Randomizes the rotation of the particles along the curve.
Scale Jitter	Randomizes the scale of the particles along the curve.

Curve Particle Generator Schematic

The schematic node has a single input for the mesh containing the curve(s). You can wire the output to a Replicator or a Volume to draw the particles using a prototype mesh.

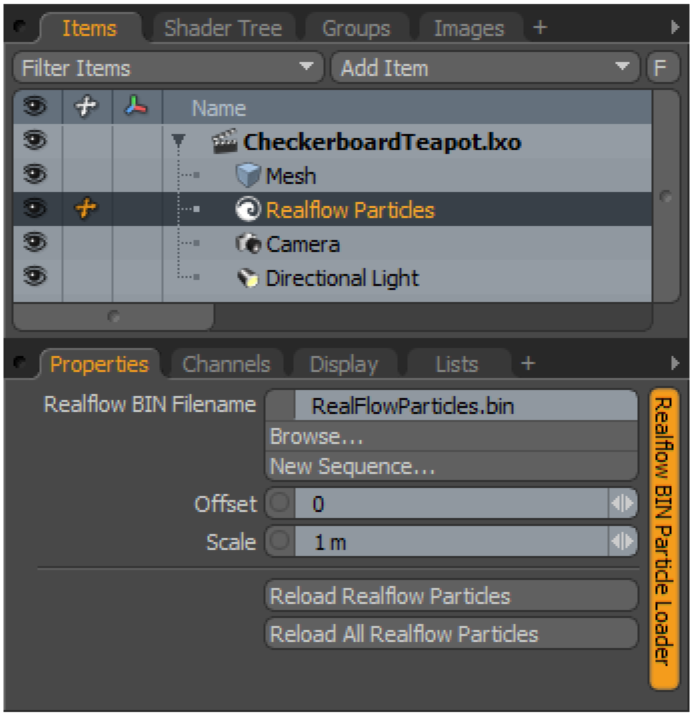


RealFlow Particles

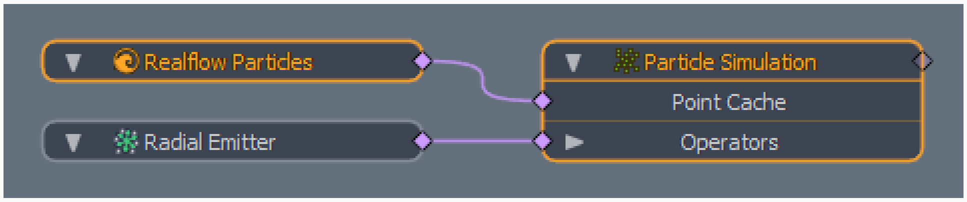
RealFlow is a software tool created and sold by NextLimit Technologies that produces realistic liquid simulations, that include physical interactions with scene elements as well as soft and hard body collisions. The calculated simulations can be saved as binary files that store the motion information for each individual particle within the simulation with additional information related to speed, age, and so on. When properly exported, these **.bin** files can be read into Modo, producing animated particles that can be used as point sources for Blobs, Sprites and Volumes, or when working with Replicators.

Adding the RealFlow Particles

Users can add a RealFlow Particles item by clicking the **Add Item** button of the **Item List**. Under **Particles**, double-click **RealFlow Particles**. Once added, you can specify the **.bin** file's name and location using the **Browse** option found in the items properties panel. Note, however, that the RealFlow **.bin** file is not saved internally to the Modo scene and the external file needs to remain in its same relative location to when it was loaded in order to render properly.


Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
CheckerboardTeapot.lxo
Mesh
Realflow Particles
Camera
Directional Light
Properties
Channels
Display
Lists
Browse...
New Sequence...
Reload Realflow Particles
Reload All Realflow Particles
Realflow BIN Particle Loader

Realflow BIN Filename: RealFlowParticles.bin
Offset: 0
Scale: 1 m

Option	Description
RealFlow BIN Filename	This option displays the name of the file loaded. Clicking the **Browse** button opens an OS specific file browser, where you can specify the name and location of the saved RealFlow cache.
New Sequence	Using the **New Sequence** option, you can define a new, empty **.bin** file that can be used to store the values from a Modo Particle Simulation. Once the file name and location have been defined, this is done by connecting the **Realflow** item into the **Point Cache** of the Particle Simulation in the Schematic View.  Realflow Particles Radial Emitter Particle Simulation Point Cache Operators
Offset	The **Offset** option moves the simulation time forward or backward, depending on value, allowing you to adjust the resulting simulation.
Scale	The **Scale** option allows you to re-size the simulation results. The default **1m** value plays the imported solution at the same scale that it was simulated at.
Reload RealFlow	Useful for when adjustments made in RealFlow are exported over the existing file,

Option	Description
Particles	clicking **Reload...** triggers a reload of the RealFlow simulation, updating the results in Modo.
Reload All RealFlow Particles	Clicking **Reload All...** triggers a reload of all imported RealFlow particle simulations.

Animating

Animation introduces movement to static objects. For many, this is what brings 3D images to life and is often considered the ultimate goal for many 3D projects. Modo has an array of options and functions that help you create animated scenes. The first part of this chapter covers the basic concepts of the animation workflow, and provides details on Modo's animation interface and related viewports. The rest of the topics describe how to use the animation tools in Modo.

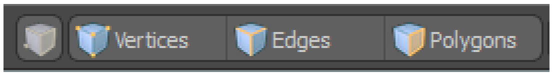
Image courtesy of Madeira3d.

Overview, Timeline, & Hierarchy

Learn about a few basic concepts before creating an animation in Modo.

Model Setup for Animation

Once you model your object and create a surface for it, you set it up in a scene and add lights. The next step is to bring it to life with animation. To ensure the smoothest animation workflow in Modo, you should understand a few basic concepts before proceeding. The first and most important is the difference between components and items. Modo has a number of selection modes for editing meshes: **Vertices**, **Edges**, and **Polygons**. The **Items** selection mode is activated automatically when none of the other selection modes are enabled. In most default layouts, these modes are displayed across the top of the screen, under the menu bar.



Modo Modes

Pressing the space bar on your keyboard toggles through the first three modes: **Vertices**, **Edges**, and **Polygons**. These three are the component modes and how you build the geometry. You edit the raw geometry that makes up the mesh layer when in these modes and apply any type of transform action (position, rotation, or scale). Regardless of where you position the geometry in the layer, the center position for that item in the **Properties** panel always reads as **0** for each axis. This indicates that component edits modify the contents of the layer. If you select **Items mode** and apply the same transforms, you can see the property values change as you transform the layer as a whole. This is an item-level transform and where you animate (and position*) objects. You can use the same tools (**Move**, **Scale**, and **Rotate**) to transform items in both the component and item modes, but, if you keep track of which mode you are in, working with these tools becomes second nature.

Mesh Items as Layers

When modeling objects, think of Mesh Items as layers. (Certain tools, like Booleans, use a background/foreground layering concept for their operations.) You use layers for applying a re-topology with background constraints.) You may want to keep each element in its own layer and use groupings in the **Items** list to keep things organized. Alternatively, you can merge elements to keep everything in a single layer. However you prefer to work, Modo is flexible, but each individual mesh layer is its own object in Modo. Each has its own settings and controls in the **Properties** panel, its own center and pivot points, and its own set of Vertex Maps. (Disabled layers don't show up in the **Lists** viewport.) Anything that needs to move independently from another item needs to have its own item layer. For example, consider a car model. The body needs to be one layer, and the four wheels each need to be their own independent layers. Depending on how detailed you want to be, you could also make layers for the doors, the hood, and trunk objects as well. When these items are independent, you can open them separately from the rest of the model.

Note: When working with very detailed meshes, it might be useful to display certain meshes only as bounding boxes during animation playback, to eliminate lag. To do this, enable **Bounding Box Drawing in Animation** under **System** > **Preferences** > **Display** > **OpenGL**. This option draws surface items as bounding boxes during animation playback, if their polygon count exceeds the **Max GL Polygon Count** value. For more information, see Display Preferences.

Transform Items

Modo uses a layered approach to working with item transforms that has several benefits over a traditional system. Channels define the rotation, position, and scale properties of any item. When you first create a new Mesh Item, before applying any transforms to it, you can switch to the **Channels** viewport and see that there are no transform channels present. The input fields are visible in the **Properties** panel, but until you assign a value, the channels do not exist. What you see are the options: **Add Position**, **Add Rotation**, and **Add Scale**. You can add them manually, but anytime you transform an item, Modo creates the channels automatically.

When you add more than one transform, Modo implements the additional transforms in the order that they are listed in the Channels view from bottom to top. Because of this, you can control how Modo places, orients, and sizes the item in the scene based on the order of its transform items. This system is very flexible, and you can apply complex motions by assigning a few additional transform items.

Modo only adds the channels that are necessary. For scenes with thousands of items, doing this can significantly save in file size and processing time (and power) by not having any unnecessary channel data. With this, Modo keeps the **Items** list clean because all the transforms are under the single item. (In a traditional system, the same result could only be achieved by parenting multiple null items together and animating each one individually, which creates additional unnecessary items.)

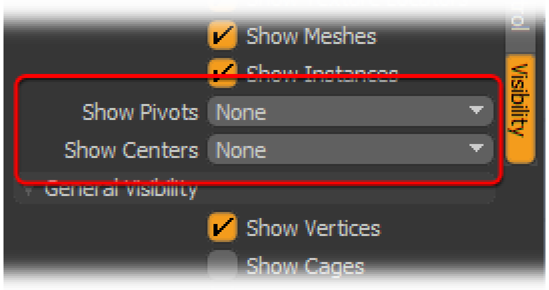
Hierarchies

With the previous car example that has individual Mesh Items, how does one move the whole car around in the scene? It's too cumbersome to select each individual item in the viewport or **Items** list. Also, this method is prone to errors. To ensure all the elements remain together when moving this car, you create a hierarchy. Hierarchies are groups of items that define how each element relates to other elements. For details about creating hierarchies, see the Items (Scenes) List and Animation Setup documentation.

The most basic Modo hierarchy is a group. With the car example, you can select all the elements of the car. By using the keyboard shortcut, **Ctrl**+**G**, you then place all the Mesh Items into a group (seen as a folder icon in the **Items** list). To move the entire car and have its individual parts follow along, select the top Group Item (the Group Locator) and apply a transform to it. All the related items in the group follow along. Another benefit to creating a hierarchy is that it organizes the **Items** list. This is especially important when building up complex scenes.

Centers and Pivots

Modo provides two methods for setting the point around which an item rotates and scales: centers and pivots. In Modo, you can use pivots in the same manner as with other applications. However, Modo also provides centers to use in many situations instead of pivots. The pivots and centers are not visible for all viewports. (This is a per viewport setting.) To see centers and pivots, move the pointer over the target viewport and press **O** to see the **Drawing and Visibility** options for viewports. In the pop-up window, click the **Visibility** sub-tab.



☑ Show Meshes
☑ Show Instances
Show Pivots: None
Show Centers: None
☑ Show Vertices
☐ Show Cages

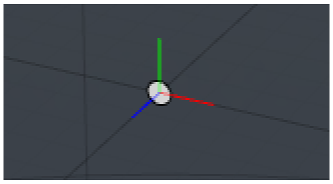
In the **Show Centers** and **Show Pivots** lists, specify **Selected** to show any layers for an associated center or pivot whenever it is selected in the **Items** list. If you select **All**, Modo always displays them regardless of selection, but for a scene with many items, displaying them may be overwhelming.

Center: This is the Local Origin of the model and the default location around which an item's transforms originate. It's the location that sits at the World Origin (0,0,0) when all the item transforms are at their defaults (if you have not moved, rotated, or scaled the item in **Items mode**). You cannot animate centers relative to the element they represent, but you can adjust their location to set the point from where the item rotates and scales.

Pivot: Pivots also provide the ability to specify the location around which an item rotates and scales. A pivot is an offset from the center. (When adjusted, you see a small dotted line draw between the center and the pivot as a reminder.) You can use pivots to animate a location (for example, to have a a cube rolling across the floor where the rotation point jumps to each edge as it rolls) because you cannot animate centers. Once a pivot moves (defining an offset from the center), Modo rotates and scales around this position instead of the center.

The main difference between pivots and centers is that pivots can be animated and centers cannot. (Pivots animate relative to the center because the pivots are an offset from the center.) In most cases, centers are fine for animated items, which is why they are the default method.

Again, using the car example, if the entire object were modeled in a single layer, then you could create the individual Mesh Items by copying and pasting from one layer to a new layer for the tires and doors. In this case, each item has its own center, but the default location for each one is in the same location at the World Origin. If you try to animate the car's tires to roll, you would find they don't rotate as expected. They probably rotate from a point originating somewhere under the center of the vehicle, which sends the tire out into unexpected circles. Centers don't default to the center of the geometrical bounding box. Therefore, you need to position the center, itself, to the correct location relative to the tire. In this case, you could have modeled the tire separate from the car at the origin, and then used an item-level transform to move the tires to the proper position relative to the car body. Modo doesn't care how you create an item, and you can reposition the center point independently without affecting the geometry itself.



If you right-click the **Items selection mode** button, you can view other mode selections. By clicking **Center**, you can edit the location of center positions without affecting the location of the original geometry in any way. So, in the example, by making sure the tire Mesh Item is selected in the **Items** list, you can see a small circle with three lines coming out from it. This circle represents the center of the item's layer. By clicking the center icon, itself, (the white circle), it turns orange. With the center selected, the Move (**W**) and Rotate (**E**) transform tools affect the center point. You could move it into the center of the wheel.

Alternatively, you could use a trick with the Work Plane that is much more precise. Switching back momentarily to **Polygons selection mode**, you should see a selection that could be made that would represent the position from where the wheel should rotate. Select this and apply the **Edit > Work Plane > Align to Selection** command. Now, switch back into **Center mode** and select the center again. In the **Animate** and **Setup** interface tabs, click **Set to Work Plane**. Doing that moves the center to the origin of the Work Plane, which is precisely aligned to the center of the previous polygon selection. Reset the Work Plane by pressing the **End** key. Everything returns to normal with the center precisely positioned. (See the Using the Work Plane topic for more information.) Now, when you animate the tire, it rotates around the proper location.

Just like in modeling there is a process to getting the best results, animation requires a bit of setup. Once you understand the basic setup techniques, it eventually becomes quick and easy.

Animating Workflow

Once you set up the model (also referred to as rigging for more complex setups), you are ready to start animating it. To make anything occur over time, Modo requires a keyframe. A keyframe sets a specific value for elements in time: a position, a surface setting, or other parameter. You can keyframe nearly any setting in

Modo to change over time. In traditional 2D cartoon animation, artists draw the important poses for an animated sequence. Then, another artist, called the in-betweener, draws all the subsequent frames that transition from one pose to the next. Animating in 3D is very similar. You set up the key poses, positions, or settings, but, unlike traditional animation, Modo creates the in-between frames for you.

With Modo's general workflow, you select the type of element you want to work with and then use a tool or apply a command that affects your selection. For example, you can select a vertex, activate the Move tool, and move the vertex. Keyframing is similar in Modo. In this case, you work with channels, which are properties that animate any item.

When you press the **Key** button (or **S** in **Items mode**), Modo determines on which channels to create keys. It does this by using channel selections. Modo runs through each selected channel and creates a keyframe on that channel at the current time. You can select channels in two ways: by clicking them in a Channels Viewport or by clicking the control label on a **Properties** viewport (the small gray circle that precedes an attribute in the **Properties** panel). If you click the dot, it changes from gray to red to signify that a keyframe is set at that point in time for that property.

Constantly selecting channels to add keyframe can be tedious, so Modo provides some auto-key behaviors to help with this. These are set with commands in the Auto Key menu at the bottom of the **Animation** interface layout (next to the transport controls). The default setting is **Animated**. With this on, Modo automatically creates keyframes on channels that are already animated (have at least one key set on the channel previously) whenever a value of the channel changes. You can change the channel by using the Transform Tool, using the mini-slider, or by typing a value on a Forms control. If you make changes to a channel that has not been keyed, Modo still applies the changes but does not make keyframes.

If you select **All**, Modo creates keyframes on any channel at the current time whenever the channel's value changes. When using this setting you must be aware of whether you're in animation mode or not. (It's very easy to create animated values when they may not be wanted.)

The last setting is **None**. With this, Modo never creates keyframes automatically for you. To make keys, you must use the **Key** button whenever you've made changes.

Which setting you use is entirely up to you and how you like to animate your models. Some people like auto-keying; some people don't. The recommended setting is **Animated** because it provides a workflow somewhere between the two extremes of on and off. To use it properly, in advance, you create keys on the channels of the item you want to animate.

The other area that Modo tries to help with keyframing is with automatic channel selection. Auto-keying only works when channel values have changed. When you need to create a set of keys for channels without changing the current values, you use **Key** button (which requires selected channels) or click the individual channel buttons on the **Properties** forms. To help with this, Modo automatically selects channels for you when you use the Transform tools. For example, if you activate the Move tool, Modo selects the position channels.

A general Auto Key workflow

1. Select the camera and go to its **Properties** viewport.
2. Click the channel keying buttons for the six position and rotation channels. (They should turn red.)
3. Activate the transform Move tool (**W** key) and move the camera.
4. Activate the transform Rotate tool (**E** key) and rotate the camera.
5. Move the time slider to a new time.
6. Move or rotate the camera again.
7. Go to a new time.

Once you have made those initial keys, you should not have to do any further manual keying unless you want to key the camera in place. This also works when using viewport navigation to move the camera. This general overview should be enough information to get started with animating.

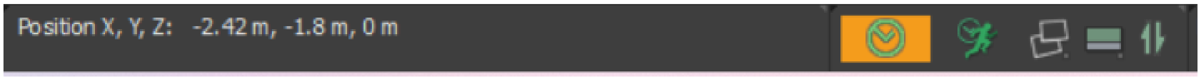
*When working with Instances and Replicators, model the base mesh at the origin. All duplicated items use that position as the transform origin when propagating duplicates. Adjusting the center position afterwards can remedy any peculiarities as well.

Working with the Timeline

In Modo, the chief function of the Timeline is to control time in an animation. The Timeline, which usually appears below the main 3D viewport, looks like a ruler with lines that represent individual frames.

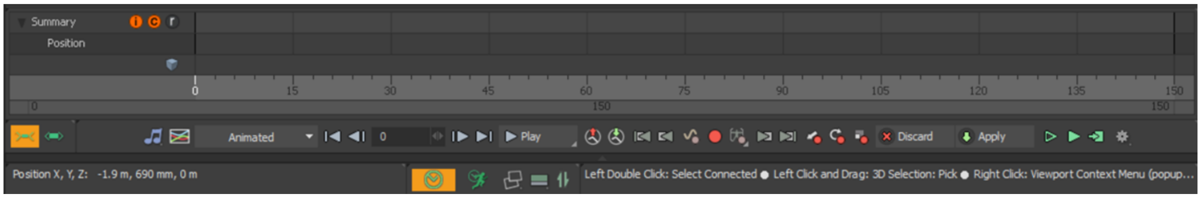
Open the timeline in the Modo layout

To open the timeline in the **Modo** layout, select the clock icon on the bottom bar:



Position X, Y, Z: -2.42 m, -1.8 m, 0 m

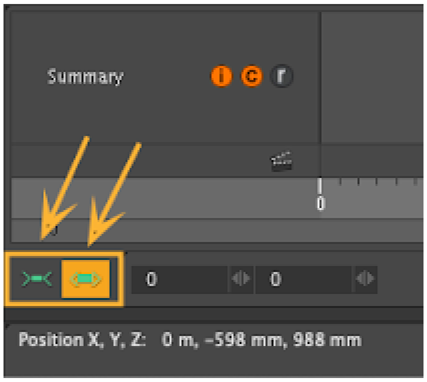
This will then open the **Standard Animation Tools** at the bottom of the viewport.



Position X, Y, Z: -1.9 m, 690 mm, 0 m

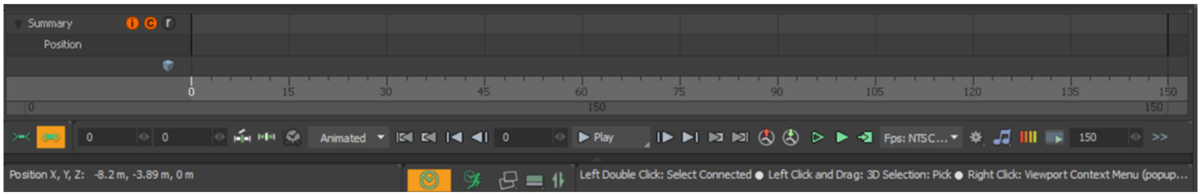
Left Double Click: Select Connected ● Left Click and Drag: 3D Selection: Pick ● Right Click: Viewport Context Menu (popup...

On the left of the timeline, there are now two buttons to toggle between **Standard Animation Tools** and **Extended Animation Tools**:



Position X, Y, Z: 0 m, -598 mm, 988 mm

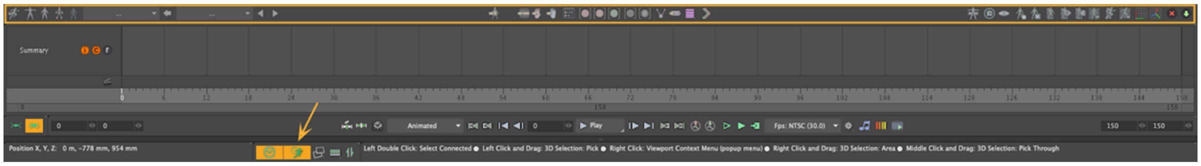
By selecting **Extended Animation Tools**, this will add more options to the toolbar:



Position X, Y, Z: -8.2 m, -3.89 m, 0 m

Left Double Click: Select Connected ● Left Click and Drag: 3D Selection: Pick ● Right Click: Viewport Context Menu (popup...

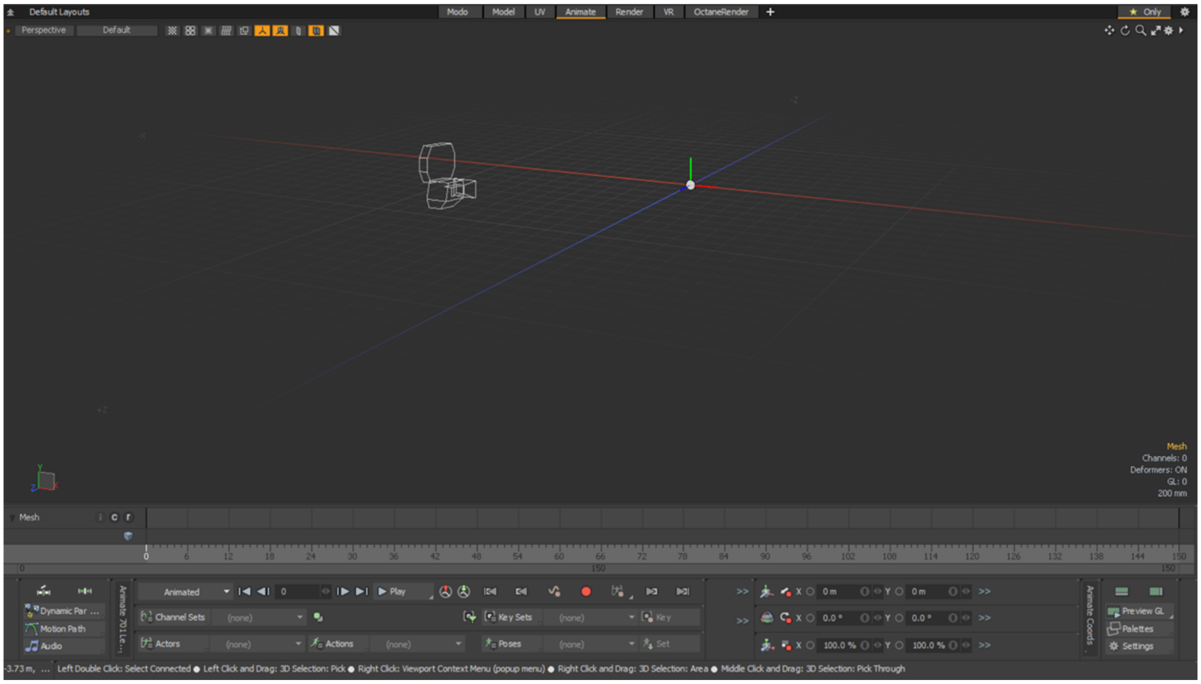
In addition, clicking the **Advanced Animation Tools** button will add a toolbar above the timeline for accessing the **Auto Character System** (ACS) controls:



For more information about Auto Character System controls, the ACS Website and Manual Page includes extensive videos showing how to use ACS.

Open the timeline in the Animate layout

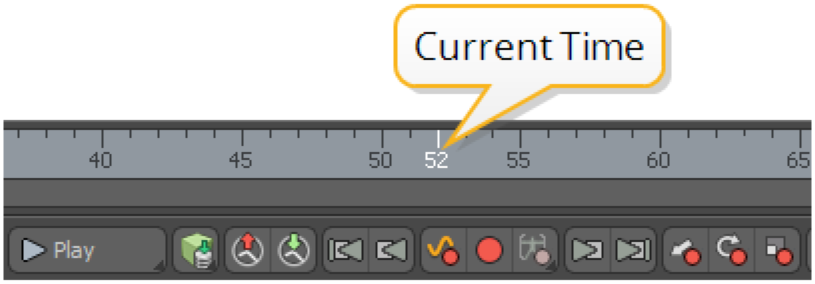
Alternatively, if you open the timeline in the **Animate** layout, a larger timeline with more options will be visible. See Animation Layout and Viewports.


Default Layouts
Modo
Model
UV
Animate
Render
VR
OctaneRender
Only
Perspective
Default
Mesh
Channels: 0
Deformers: ON
GL: 0
200 mm
Animated
Play
Channel Sets
Key Sets
Key
Actors
Actions
Poses
Set
Dynamic Par ...
Motion Path
Audio
Preview GL
Palettes
Settings
Left Double Click: Select Connected ● Left Click and Drag: 3D Selection: Pick ● Right Click: Viewport Context Menu (popup menu) ● Right Click and Drag: 3D Selection: Area ● Middle Click and Drag: 3D Selection: Pick Through

Using the Timeline

Current Time and Keyframes

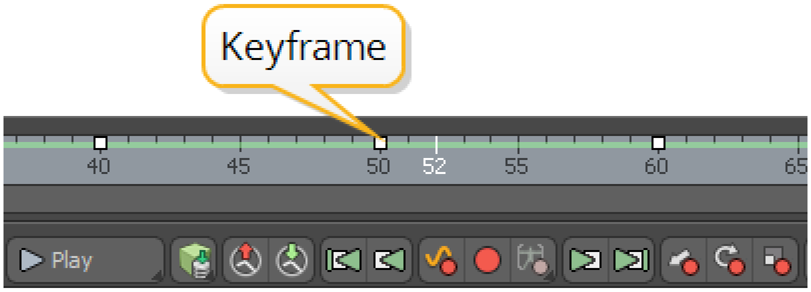
You can modify the current time position, which is represented by the white line and frame number, by dragging the white line left or right within the Timeline's numbered area.


Current Time
40
45
50
52
55
60
65
Play

Note: There is only one current time in an animation and modifying it in one viewport changes it in all viewports so that there can be no confusion.

To animate any item over time, you only need to set a specific value at a point in time (known as a keyframe) and then move to a different point in time to set a new keyframe. Modo smoothly interpolates the in-between values.

You can set keyframes in many ways. Within a **Properties** viewport, each item attribute shows a ● dot that you can click to set a keyframe. When set, the dot is red to signify the presence of a keyframe. You can also set keyframes by clicking any keyframing button under the Timeline. You may also choose **Animate > Auto Key** to have Modo set keys automatically whenever you modify an attribute. Additionally, you can set keyframes by clicking the far right column of the Channels Viewport or by selecting one or more channels in the **Channels** viewport and pressing the **S** key. Keyframe positions show as white dots on the Timeline with the interpolated in-betweens indicated by a green line connecting the various keys.

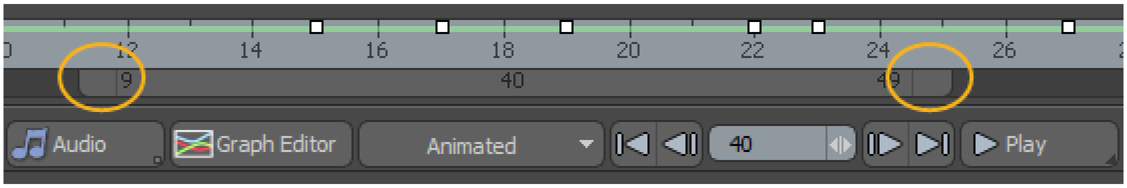


Frame Range

The Timeline itself represents a specified length of time, which is the result of dividing the number of frames by the frame rate. By default, Modo displays 120 frames at the default frame rate of 24 frames per second (FPS) to show 5 seconds of time (120/24 = 5).

For longer or shorter sequences, you can adjust the length several ways. A 12-second sequence requires 288 frames (12 s x 24 fps = 288 frames). The scene's frame range represents the length of the total sequence. You can indicate a different sequence length in the Scene Item's Properties panel as the **Scene Range** or in the Timeline's Settings dialog where you can specify **Scene Start** and **End**.

Modo indicates a fraction of the scene length by using the current frame range. You can adjust the current frame range to concentrate on a specific interval without needing to preview the entire sequence. In the gray bar directly below the Timeline, you see two handles indicating the current frame range. Dragging either of those handles adjusts the current frame range.



Time Fit

You can fit the current Timeline range to the range of the animation belonging to:

- selected channels or items,
- the current Action,
- every item in the scene.

To do this:

1. Select your channels or items, or set the required action to current.

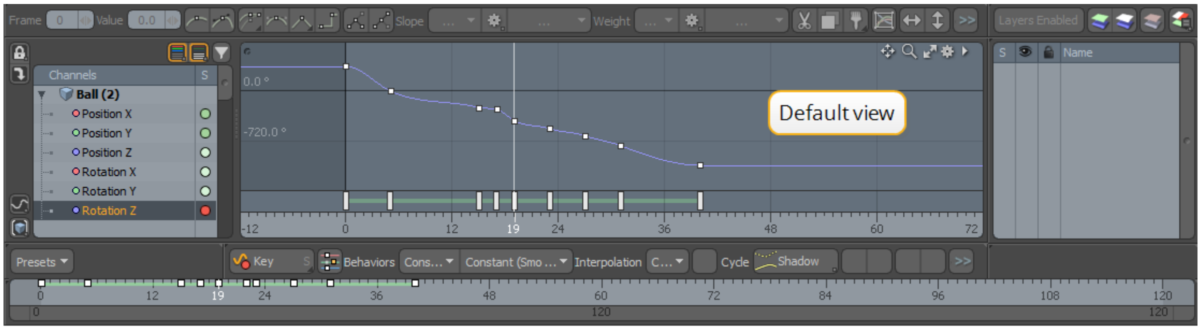
Note: For more information on working with Actions, see Creating Actors, Actions, and Poses.

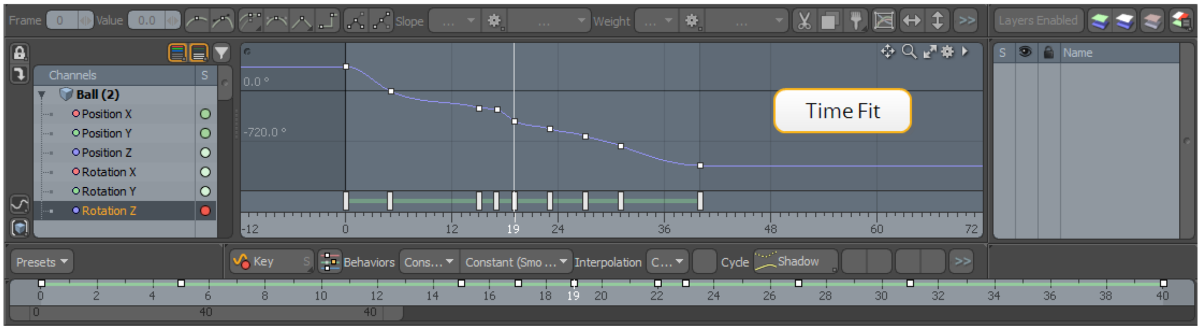
2. Hover the mouse over the Timeline, Marker Row, or Track View (Dopesheet).

Note: To reveal the Track View, drag the top of the timeline upwards.

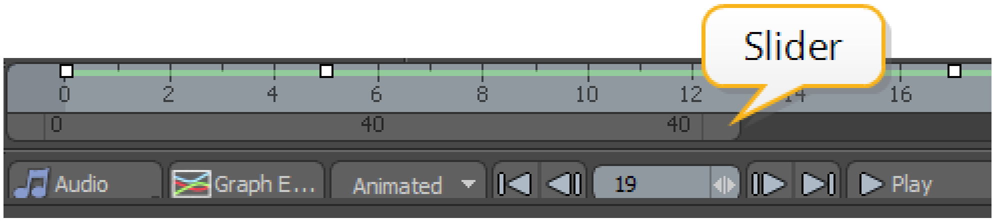
3. Press the **A** key.

   The Timeline range updates from the default frame range to the frame range of the selected channels or items. To set the current range to the animation range of all items in the scene, press **Shift**+**A**.


Default view
Channels
Ball (2)
Position X
Position Y
Position Z
Rotation X
Rotation Y
Rotation Z
Presets
Key
Behaviors
Interpolation
Cycle
Shadow


Time Fit
Channels
Ball (2)
Position X
Position Y
Position Z
Rotation X
Rotation Y
Rotation Z
Presets
Key
Behaviors
Interpolation
Cycle
Shadow

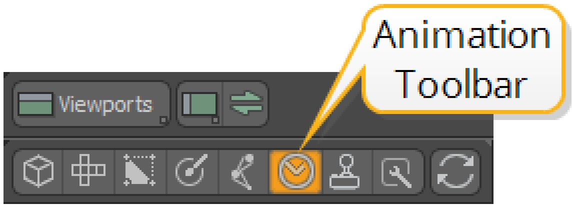
4. To return to full scene range, double-click the slider in the Timeline.


Slider
0
2
4
6
8
10
12
14
16
0
40
40
Audio
Graph E...
Animated
19
Play

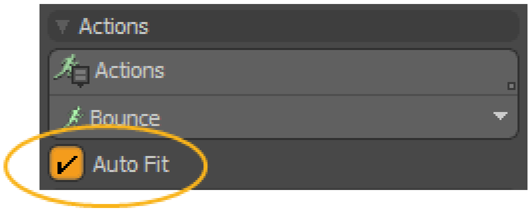
If there are no channels or items selected, the Timeline is scaled to fit the range of the animation of the current Action. If no Action is found, the Timeline is fit to the range of animation found in all items in the scene.

If you want to fit the current Timeline range to the range of the animation in the current Action, you can also do the following:

1. Open the Animation toolbar on the left panel.


Animation Toolbar
Viewports

2. Under **Actions**, select the required Action.
3. Enable **Auto Fit**.

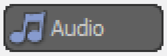


Actions
Bounce
☑ Auto Fit

The Timeline adjusts to the frame range of the currently selected Action.

Timeline Tools

The Timeline has a toolbar directly below it.

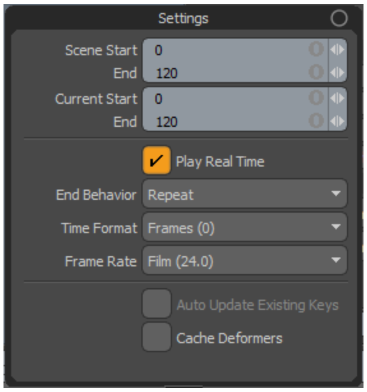
 Audio	Opens the Audio dialog. You can see more information about using audio with animations in the Animation Interface topic.
Graph Editor	Opens the Graph Editor.

Animated ▾	Opens a dialog for setting auto-key behavior. When enabled, Modo generates a keyframe for any modification (position, rotation, or scale) made to an item at the currently selected time. Use this with caution because, although this can save you time from setting keyframes manually, you may find that you can inadvertently modify an item and overwrite previous values. • **All** - Sets a keyframe for any and all changes made to an item. Be aware that with this setting Modo may add keyframes where you may not intend to have them. • **Animated** - Sets a keyframe only for channels that have existing keyframe values. For example, if you set a keyframe only for the position channel an item and you move the time slider and change the position and rotation, Modo only sets a new position keyframe. You need to explicitly set a key for a channel for auto-keying to be effective. • **Off** - Turns off the automatic keyframing function.
⏮	Sets the current time position to the first frame of the current frame range (Keyboard shortcut: **Shift**+**Up Arrow**).
◁‖	Moves the current time position backwards one frame (Keyboard shortcut: **Shift**+**Left Arrow**).
0	Displays the current time position. You can type a value to manually specify the current time.
‖▷	Moves the current time position forward one frame (Keyboard shortcut: **Shift**+**Right Arrow**).
⏭	Sets the current time position to the last frame of the current frame range (Keyboard shortcut: **Shift**+**Down Arrow**).
▷	Plays the animation sequence forward based on the current frame range (Keyboard Shortcut: **/**). Right-clicking this button reveals additional options: • ◁ **Play** - Plays the animation sequence backwards from the current time. • ▷ **Play Keys** - Plays only the keyframes for the current actor. This also supports the Range time markers.
◻	Stops sequence playback while playing. This button replaces the **Play** button (Keyboard Shortcut: **/**).
	Plays through the timeline one frame at a time to populate the cache with meshes resulting from deformers and mesh operations. Right-click to reveal the **Cache**

	Deformers at Current Time option. **Note:** You can enable caching in the **Animation Settings**. To open the settings, click the **Settings** button in the timeline, or in the menu bar, navigate to **Animate** > **Settings**.
	Removes a frame and moves all following keyframes one frame backwards (Keyboard shortcut: **,** [comma]).
	Adds a frame and moves all following keyframes one frame forward (Keyboard shortcut: **.** [period]).
⏮	Sets the current time position to the first keyframe of the currently selected item. For multiple items, Modo uses the time for the first selected item. (Keyboard shortcut: **Alt**+**Up**)
◀	Moves the current time position to the previous keyframe that is nearest the current keyframe (Keyboard shortcut: **Ctrl**+**Left Arrow**).
	Assigns a keyframes to all currently selected channels.
●	Assigns a keyframe to all animated channels belonging to the currently selected items.
	Assigns keyframes to all animated channels belonging to the items in the actor and also any channel members. Right-clicking this button reveals an additional option: - Assigns keyframes to all animated channels belonging to the items in the group.
▶	Moves the current time position to the next keyframe that is nearest the current time (Keyboard shortcut: **Ctrl**+**Right Arrow**).
⏭	Sets the current time position to the last keyframe of the currently selected item. For multiple items, Modo uses the first selected item (Keyboard shortcut: **Ctrl**+**Down**).
	Sets a keyframe for position channels for selected items.
	Sets a keyframe for rotation channels for selected items.

	Sets a keyframe for scale channels for selected items.
	Sets a keyframe for the pivot positions for selected items (Keyboard shortcut: **Alt**).
	Assigns the current World Position value to the local item. (Keyboard shortcut: **Shift**).*
	Assigns the current World Rotation value to the local item. (Keyboard shortcut: **Shift**)*
	Assigns the World Scale transform value to the local item. (Keyboard shortcut: **Shift**)* * These are useful when retaining the position of driven items and when removing a constraint or modifier.
☒ Discard	Discards any changes that have been made to channels since the current time was changed, or when you've applied changes manually.
Apply	Applies any changes that have been made to channels since the current time was changed, or when you've applied changes manually.
▷	Plays the simulation at the current time.
▶	Plays the simulation over the length of the scene.
	Computes channels driven by simulations and caches the result.
⚙ Settings	Opens the **Settings** dialog that has options similar to those found in the Scene Item properties.

Timeline Settings



Scene Start 0
End 120
Current Start 0
End 120
☑ Play Real Time
End Behavior Repeat
Time Format Frames (0)
Frame Rate Film (24.0)
☐ Auto Update Existing Keys
☐ Cache Deformers

Scene Start/End: Sets the sequence length from the starting and ending frame numbers.

Current Start/End: Sets the current length from the starting and ending frame. When animating, you may want to concentrate on a single area of a scene for fine tuning. By adjusting the current range, you can temporarily crop the rest of the scene and concentrate on one section. Double-click the Timeline to reset the current range to the full scene range.

Play Real Time: When enabled, plays the scene in an **OpenGL** viewport using the **Frame Rate** setting as best as can be done. (If necessary, frames may be skipped.) When disabled, Modo plays every frame in the sequence at the fastest speed possible. (Depending on the scene's complexity, this may be slower or faster than the specified frame rate.)

End Behavior: Determines what happens when playback reaches the end of the play range.

- **Hold** - Stops animation at the end of the playback range.
- **Repeat** - The default setting, loops playback indefinitely.
- **Wrap** - Repeats the playback until the current time matches the initial time, and then it stops.

Time Format: Specifies which standard to use when displaying individual frame numbers.

- **Frames** - Indicates frames as sequential numbers by using a standard numbering format.
- **Seconds** - Indicates frames as a passage of time by using the **0.0s** format.
- **SMPTE** - Specifies frames with a shorthand method **00:00:00:00** that represents hours:minutes:seconds:frames.
- **Film Code** - Specifies frames by a linear measurement (the length of film if the animation were filmed) in feet by using **0.0 + 0** format.

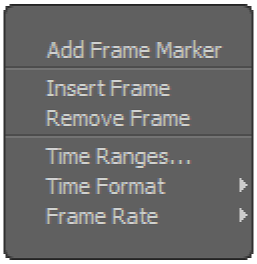
Frame Rate: Specifies a frame rate by standard definitions: **Film (24)**, **PAL (25)**, and **NTSC (29.97)**. The rate is in fames per second (fps), which is the number of still images that playback in sequence in one second of time.

Auto Update Existing Keys: When enabled, updates existing keys automatically. This is only available when **Auto Key** is disabled (**Off**). Normally, when **Auto Key** is off, any edits that you make are discarded when the current time changes unless you have manually keyed them. By enabling this option, you can have Modo update existing keyframes automatically (if changed) and you don't have to manually key them. Modo still discards un-keyed changes when the current time changes.

Cache Deformers: Enable/Disable deformer caching. When enabled, Modo caches the resulting meshes from deformers and mesh operations into memory for each frame. As a result, playback and stepping through frames becomes very fast, even with complex deformers.

Timeline Right-Click Menu

If you right-click the Timeline itself, a menu opens with relevant commands.



Add Frame Marker: Adds a frame marker to the Time Marker area above the Timeline. See **Time Markers** for more information.

Insert Frame: Inserts a single, additional frame for all selected items at the current time positions. Modo pushes subsequent keyframes away from the added frame's position. This is useful in adjusting keyframe positions for selected elements.

Remove Frame: Removes a single frame for all selected items at the current time positions. Modo moves subsequent keyframes toward the removed frame's position. (It does not delete keyframes.) This is useful in adjusting keyframe positions for selected elements.

Time Ranges: Opens a dialog for you to numerically define either the scene range or the current range.

Time Format: Specifies the Timeline's display format. (For details, see Time Format.)

Frame Rate: Specifies the playback frame rate. (For details, see Frame Rate .)

Key Time

Key time represents the time at which you create a keyframe. Some Modo functions that take a time input also use it (for example, the **Insert Frame** and **Remove Frame** commands). By default, the key time is the same as the current time. If you click or drag a marker in the Timeline, you set both the current and key times.

By middle-clicking or dragging with the middle button pressed, you can set the key time independently from the current time. If you do this, Modo shows a green marker in the Timeline. By pressing **S** to keyframe selected channels, Modo reads values from the current time, but it creates keyframes at the key time. This makes it easy when you copy item transform values from one point in time to another time in the scene or when you create a looping animation and the end frame needs to match the first frame.

You can use the **Insert Frame** and **Remove Frame** commands (possibly with the comma and period shortcut keys) with the key time to adjust the timing of animations. The key time does not change when you play an animation, and you can set it by middle-clicking the Timeline during playback. By positioning the key time marker between keys, you can quickly adjust the timing with the **Insert Frame** and **Remove Frame** shortcuts while viewing the results in the 3D viewport.

Keyframing Shortcuts

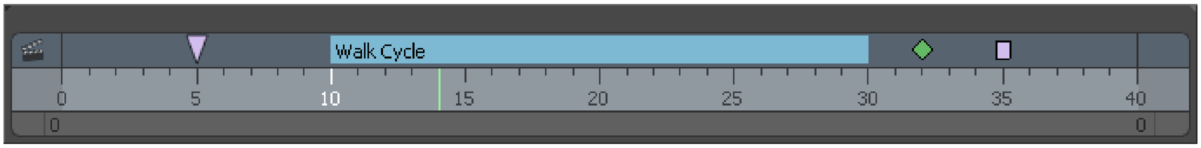
Modo provides a number of keyboard shortcuts to help you set keyframes while working with the Timeline in the Animation view. You can quickly set keyframes on an entire item or specifically on the various transform elements or selected channels. When using the **Key Selected Items** command, Modo creates keys on any channels for an item that already contains at least one keyframe. Modo does not key other channels for that item.

Shift-**Y** = Key transforms
Shift+**E** = Key Rotations
Shift+**W** = Key Positions
Shift+**R** = Key Scale
S = Key Selected Channels
Shift+**S** =Key Selected Items

These two shortcuts quickly slide any keyframes forward or backward that are after the current time. This is useful for quickly retiming portions of your animation.
, (comma) = Time Offset Forward
. (period) = Time Offset Backward

Tip: These Animation keyboard shortcuts are part of the Item mode keymap. When Modo is in a component selection mode (such as Vertices, Edges, or Polygons), many of these keys perform important modeling tasks. Because of this, Modo provides an efficient workflow for both processes.

Time Markers



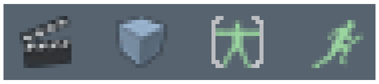
Directly above the Timeline, Modo provides an area where markers can be added to call out specific locations or to indicate certain areas of a sequence. You can also add tags to denote information or reminders about a sequence with special types of markers. By default, the Time Marker is only visible in the Animation layout. In the other layouts that show the Timeline, Modo hides it to maximize the 3D viewport space. By dragging the top edge of the Timeline area upward, you can reveal the Time Marker.

Adding Markers

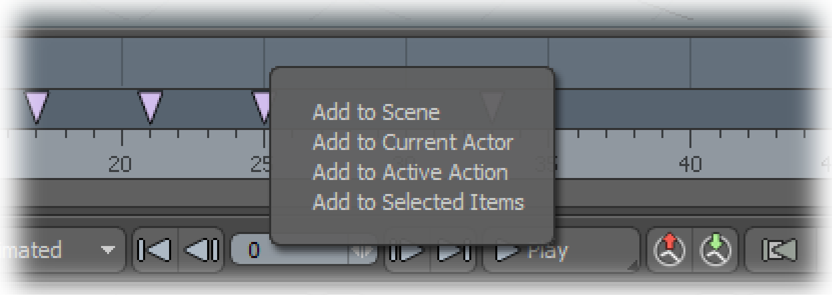
You can add markers anywhere along the Timeline by middle-clicking in the Time Marker to adding a new marker at the nearest whole frame location. You can also add a marker at the current time marker position by pressing **M** on the keyboard when the Timeline, itself, has input focus or by right-clicking the Timeline area and choosing **Add Frame Marker**. Modo attaches time markers to items in the scene, depending on what you selected before you added the marker.

- If you selected a locator-type item (such as mesh, camera, or light items), Modo attaches the marker to it.
- If there is an active actor and action, but you have not selected an item, Modo attaches the marker to the action.
- If there is an active actor but no active action or selected item, Modo attaches the marker to the actor.
- If there is no active actor, action, or selected item, Modo adds the marker to the marker to the scene.

When not a part of the scene, markers only appear when you select the associated elements. Scene markers always appear. The left column of the Time Marker shows an icon that indicates which type of item Modo adds new markers to: a clapper board for scene markers, a cube for locator item markers, a standing actor in brackets for actor markers, and a running actor for action markers.

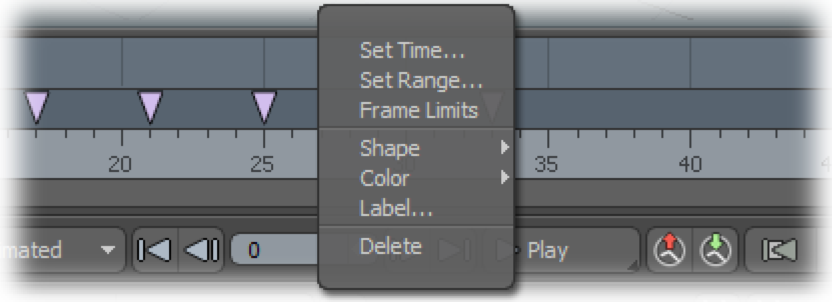


You can also add time markers to specific items by selecting the target markers and right-clicking in an open area of the Time Marker and choosing the relevant command: **Add to Scene**, **Add to Current Actor**, **Add to Active Action**, or **Add to Selected Items**.

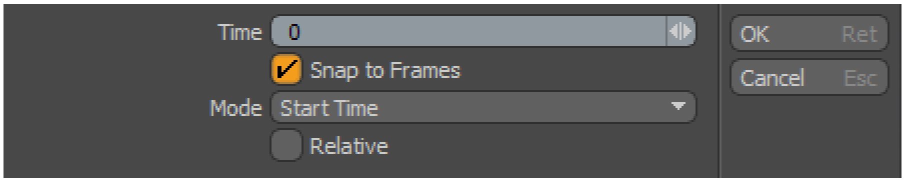

Add to Scene
Add to Current Actor
Add to Active Action
Add to Selected Items
20
25
40
mated
0
Play

Selecting and Editing Markers

You can move time markers in the Time Marker are after you create them by dragging them to a new position. You can select and move multiple markers by pressing the **Shift** key as you select additional markers and then dragging them to the new location. (You can also drag across multiple markers to select the entire range.) Clicking an open area on the Time Marker clears the selection. By right-clicking directly over a marker, you can specify the marker's position as well as customize how the marker looks in the Time Marker.


Set Time...
Set Range...
Frame Limits
Shape
Color
Label...
Delete
20
25
35
40
mated
0
Play

Set Time: Opens the **Time Marker Time** dialog for you to define a new time location for the target marker. (If you have selected multiple markers, then Modo moves all to the specified **Time** value.)


Time 0

☑ Snap to Frames

Mode Start Time

☐ Relative
OK Ret
Cancel Esc

Time: Defines the position on the Timeline that the marker is moved to or defines an offset amount if **Relative** is enabled.

Snap to Frames: Moves all frame markers to the nearest whole frame location.

Mode: Determines what the **Time** value relates to.

- **Start Time**- Defines the marker location or the start position for a range marker.

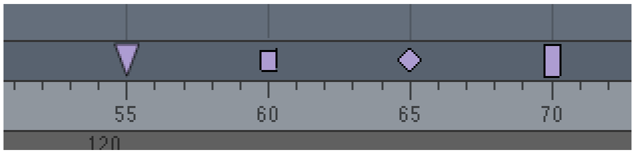
- **End Time**- Defines the end position for a range marker.
- **Shift**- Defines an offset amount from the marker's current location. (The shift can be positive or negative.)

Relative: When enabled, shifts markers by the **Time** value relative to their current position. When disabled, markers move to the absolute time position along the Timeline.

Set Range: Opens the **Time Marker Time** dialog (as does the **Set Time** option), but the dialog defaults to the **End Time** mode with **Relative** enabled.
Frame Limits: When enabled, identifies range markers that have limits by an arrow. (If the marker has a label, then Modo does not draw the graphic.) Modo has some commands that use range markers with limits.

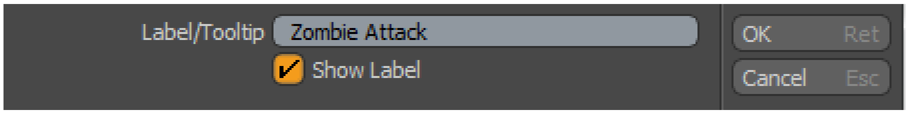
- **Frame Next/Previous**- When stepping through frames the range marker with limits restricts the steps to the range of the marker. If you step through, the stepping loops back to the start or end of the range depending on the direction. This could be useful for rolling poses.
- **Animation Playback**- Animation playback looks for range markers with limits and only plays within the range of the marker. This only applies when the current time is within the range of the marker when playback is first started. If you start playback outside the range of the marker, the animation plays normally.
- **GL Preview**- When generating GL preview, Modo uses range markers with limits assigned if the current time is within the range of the marker when the recording starts. If the current time is outside the marker range, then Modo uses the current, normal scene range.

Shape: Provides several default shape types that you can assign to a given marker or selection of markers. These include: **Default** (a downward triangle), **Square**, **Diamond**, and **Bar**.



Color: Use to color code the marker (or markers) from the list of standard available colors. You can also color bar range markers.

Label: Opens a dialog for you to assign a short text string to the marker. This text string shows as a Tooltip for the marker. If **Show Label** is enabled, the text appears above the marker, itself.


Label/Tooltip Zombie Attack

☑ Show Label

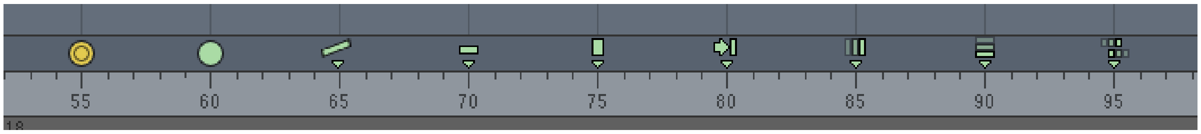
Delete: Removes all selected markers from the Timeline.

Range Markers

Range markers are time markers that have a length of time associated with them. To create a range marker, add a time marker and then drag it with the right mouse button. Modo changes the marker to a range marker and determines the length from how far you drag. You can edit the starting time of a range marker by dragging with the left mouse button and the end time by dragging with the right mouse button. When editing a single range marker, use the left mouse button to change the start or end time by dragging near the start or end of the marker. If you press the **Shift** key as you drag, you move the whole marker. If you move a time marker into the range of a range marker, the time marker shows on top of the range marker. You can adjust the end time of a range marker by right-clicking it and choosing **Set Range**. See the description for the **Time Marker Time** dialog described previously.

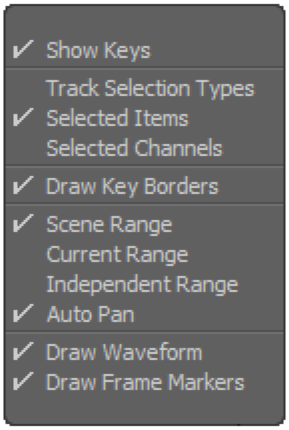
Special Markers

The Spacing Chart tool assigns its own special markers. See the Spacing Chart topic for detailed information.


55
60
65
70
75
80
85
90
95

Viewport Settings

In the **Timeline Viewport Settings** menu, you can adjust the display of the Timeline to customize it for your needs. To view this menu, click the triangular icon in the upper-right corner of the Timeline.


✓ Show Keys
Track Selection Types
✓ Selected Items
Selected Channels
✓ Draw Key Borders
✓ Scene Range
Current Range
Independent Range
✓ Auto Pan
✓ Draw Waveform
✓ Draw Frame Markers

Show Keys: Shows or hides the keyframe markers in the Timeline, itself.

Track Selection Types: When enabled, displays the keyframes for either animated items or animated channels as appropriate.

Selected Items: When enabled, shows Timeline keyframes only when you select an item that is animated.

Selected Channels: When enabled, shows Timeline keyframes only when you select a channel that is animated.

Draw Key Borders: Shows or hides a black outline around the keyframe icons in the Timeline.

The following settings pertain to the Track view when the Timeline is extended upward to display additional keyframe handles for easily adjusting animation timing.

Scene Range: When enabled, the Track view displays the entire range of frames defined by the scene item.

Current Range: When enabled, the Track view displays only the current range as defined in the scene item.

Independent Range: When enabled, the Track view can show an independent frame range from that of the Timeline, itself.

Auto Pan: When enabled, Modo automatically pans the Timeline so that the current time marker stays in the center of the Timeline when you work with a limited range view.

Draw Waveform: Shows or hides a waveform displayed in the background of the Timeline.

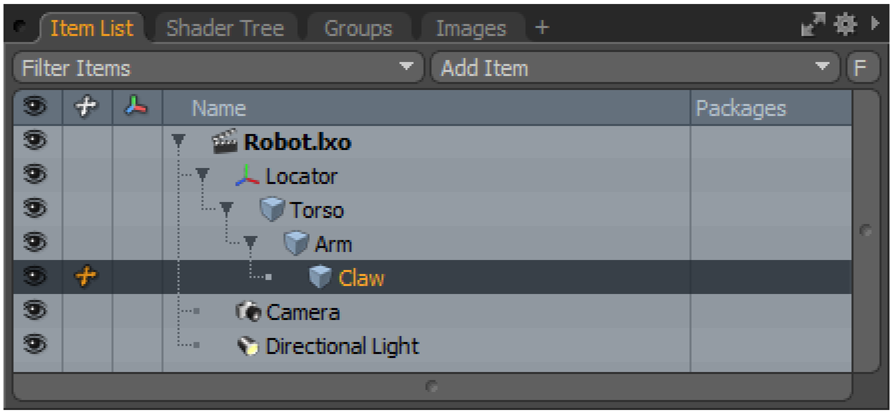
Draw Frame Markers: Shows or hides the Frame Marker area above the Timeline.

Hierarchy

A hierarchy is a series of ordered groupings. In 3D animation, these hierarchies are also known as parent/child relationships, and the term parenting refers to establishing a parent/child relationship. Modo uses hierarchies to define how multicomponent objects relate to each other. For example, if you develop a model for a car with spinning wheels, moving the car around is much easier when the wheels move with the car automatically. Because the wheels move in a different way than the car's body, they need to be separate objects from the car.

Every hierarchy starts with its base item, called the **root**. This is the main parent item, and each subsequent item in the chain is considered a child. The item directly above a child is always considered its parent.

Hierarchies are essential to animation because they define the order of inheritance for transformations in a chain of linked items. For example, with an arm, if you move your shoulder, your arm follows along as does your wrist and hand. So when the shoulder bends it passes its transforms on to the elbow, which can also bend. Both of those transforms are passed on to the hand and so on. Modo visualizes these hierarchical chains as cascading items within the Items (Scenes) List tab.



Defining Hierarchies

In the screenshot, you see that Modo indents each subsequent item in the list. The top layer (called **Locator**) is the parent item. The next item (the **Torso**) is the Locator's child and it is followed by siblings in lower orders. The order of the items in a hierarchy defines how the transforms relate to items further down the chain. You create hierarchies in the **Items List** tab by selecting a layer and then dragging that layer onto another layer. (Modo indicates target layers.) Modo automatically parents the items in place, so the items intelligently inherit the parent item's transforms, if they exist. (It won't move from where it is when parented.) By pressing **Ctrl** before releasing the mouse button, you can have Modo parent the items directly. If the target has any transforms, the child items being parented move as if they were connected to the item all along (for example, if parented before it was moved from the origin at 0,0,0).

The animation **Setup** workspace layout provides additional tools for creating and modifying hierarchies. You can use the hierarchy tools in the default toolbox in the **Setup** sub-tab to assign hierarchy to item selection directly in the 3D viewport.

Compensation

Compensation

When parenting items together that have transforms applied already, you may find that items pop out of position. If this occurs, the child is inheriting the transforms of the parent and sending it off to a position you likely don't want. By enabling the **Compensate** button at the top of the **Setup** layout, the **Parent**/**Unparent** buttons become **Parent in Place**/**Unparent in Place** buttons. If you click these buttons, Modo ignores prior parental transforms; therefore, items retain their initial positions in relation to their parent.

Parent/Unparent

When in item mode, the **Parent** and **Unparent** buttons produce the same results as when using the **Items List** tab, but the buttons work with selections in the 3D viewport. When setting up hierarchies, any item can be parented to any other item (such as cameras, lights, and locators).

Selection order is important when using the **Parent** button as it sets the relationship properly. The last item selected becomes the parent and the first selected item becomes the child of the parent. If you select multiple items, you can't create an entire chain directly. To do that, you need to select the items in pairs and to click the **Parent** button for each pair. To use the **Parent** button, first make sure you are in Item mode. Select an item in the 3D viewport as the child item. Press the **Shift** key as you select the second item. With both items selected, click the **Parent** button. You can see that the first item selected has now become the child of the second item.

With the **Unparent** button, select any item or item(s). Then, click the **Unparent** button to remove the selected item(s) from the hierarchy.

Insert

The **Insert** button is useful for adding items into an existing hierarchy without breaking or upsetting the existing chain. Clicking **Insert** adds the first selection as a child of the second selection. The child item inherits the parent item's transforms and siblings.

Insert at Parent

The **Insert at Parent** button is different from the **Insert** button because it inserts an item as a parent instead of a child. It still inherits transforms and siblings. You may find this useful for inserting an item into the top of a hierarchy.

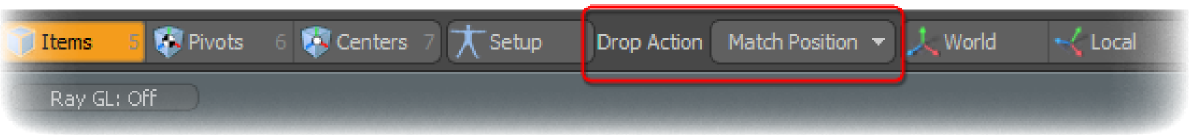
Extract

The **Extract** button removes an item from within a hierarchy without disrupting the rest of the hierarchical chain. Modo automatically parents children of the extracted item to its parent after removal.

3D viewport Parenting Actions

While in Item mode, you can use keyboard shortcuts to quickly set up basic hierarchies. The last item you select always becomes the parent item. When in any 3D viewport, you can select two or more items and use the **P** key to parent in place: **Ctrl**+**P** performs normal parenting, **Shift**+**P** unparents in place, and **Ctrl**+**Shift**+**P** unparents the selected items. For more complex actions, you can use the **Drop Action** menu.

Drop Action



You can also use drag-and-drop actions directly in the 3D viewport to define more complex hierarchies. The **Drop Action** menu in the **Animate** and **Layout** interface tabs defines what action happens when you drag an item onto another. To implement a default action, click **Drop Action** and specify the action. Now, when you drag an item (the Source item) onto another item (the Desination) item, Modo implements the specified action.

You must be in Item mode with no active tools for the actions to work. Click an element and drag it toward the target item. The element in the interface doesn't move until it reaches the target item and you release the mouse button. At that time, Modo implements the specified action.

The **Drop Action** menu has the following commands:

Command	Description
None	Does nothing to either item.
Parent	Makes the Source item a child of the Destination item.
Parent in Place	Parents the Source item as above, but with Compensation applied.
Match	Modifies the Source item to match the position, scale, and rotation of the Destination item.
Match Position	Modifies the Source item to match only the position of the Destination item.
Match Rotation	Modifies the Source item to match onlly the rotation of the Destination item.

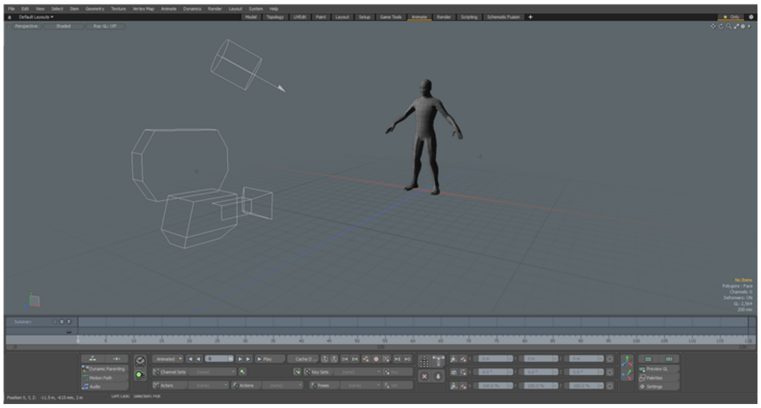
Command	Description
Match Scale	Modifies the Source item to match only the scale of the Destination item.
Insert	Inserts the Source item into the hierarchy of the Destination item. The position and orientation of the Source item matches the Destination item, but it becomes the parent of the Destination item.
Insert at Parent	Inserts the Source item into the hierarchy of the Destination item (as with the **Insert** command), but the Destination item's parent specifies the position and orientation for the Source item (if a parent item is present).
Insert in Place	Inserts the Source item into the hierarchy of the Destination item as the parent of the Destination item, but the Source item retains its current position and orientation.
Place	Inserts the Source item into the hierarchy of the Destination item, but determines the Source item's position based on the intersection of the pointer and the surface.
Place and Align	Inserts the Source item into the hierarchy of the Destination item. With this command, the intersection of the pointer and the surface determine the Source item's position and the surface's normal direction determines its rotation.

Animation Layout and Viewports

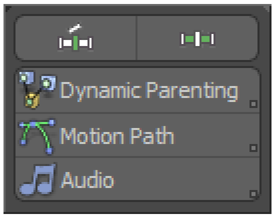
The **Animate** layout is a workspace specifically developed for animation. The workspace has a larger 3D viewport than other workspaces do. With this maximized space, you can concentrate on animating your characters. With the main toolbox across the bottom of the screen, you can set and refine keyframes in an efficient manner.

Directly below the 3D viewport is the Track view (also known as a Dope Sheet), which has details in its own topic. You can access Track view by dragging the top of the frame for the Timeline.

The toolbox below the Timeline has the tools necessary to scrub the Timeline, to control view settings, to add, edit and remove keyframes, and to refine animation timing. Some tools are unique to the **Animate** layout.



With the first group of five buttons, you have quick access to the **Spacing Chart** and **In-between** tools, as well as the **Dynamic Parent**, **Motion Path**, and **Audio** dialogs.



Spacing Chart

You can use the Spacing Chart in a workflow popularized by traditional hand-drawn animation. With this workflow, you concentrate on the poses of the character and treat each keyframe position like a single drawing in an animated sequence. You can define specific key poses for a character before working out the timing. Once the poses are set, then you can modify the timing independently. See the Spacing Chart topic for more information.

In-between Tool

You use the **In-between** tool for quickly creating poses between another pair of poses or for adjusting existing poses. You can use it for building overlap or follow-through, or to create moving holds anything where a pose or parts of a pose need to be biased towards their preceding or following poses (keyframes). The target item needs to be part of an actor, and the items being edited must have at least one key on its transform channels before and after the time at which you apply the tool. Modo supports all action and channel interpolation types. For example, Modo creates an in-between when using stepped curves.

You activate the tool by clicking the button at the right of the toolbar area. When active, **Inbetween** displays in the viewport near the axis widget. To create a new pose, set the Timeline's current time to where you want the pose keyframe. Then, activate the tool and drag with the right mouse button directly in the 3D view. Dragging moves the target item in time as the tool adjusts the inbetween pose. Releasing the button fixes the pose to the now current location. Dragging to the left adjusts the new pose towards the previous pose; dragging to the right adjusts the new pose towards the next pose. If you have not selected an item, but a current actor and action are active, then Modo edits all items in the actor that have surrounding keys at the current time.

To edit an existing pose:

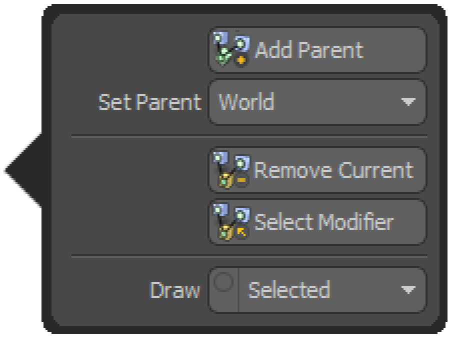
1. Select the target item or hierarchy of items to edit.
2. Position the Timeline indicator at the position to be edited.
3. Right-click and drag in the 3D viewport.

The tool takes the existing transform of the items into account and biases the pose from there. If there are no selected items, then the tool works with the current actor as previously described.

Dynamic Parenting

Parenting is the act of creating a hierarchy and relating the motion of one object to that of another. Standard parenting hierarchies are fixed, and therefore, you cannot animate them. Sometimes, however, you need to change the parent of an item over time. Dynamic parenting gives you the capability to keyframe what an item is parented to and when.

For example, with an assembly line scenario, a product moves on a conveyor belt. Then a robotic arm picks up the product, places it into a package, and moves the package elsewhere. In this case, you parent the item first to the conveyor belt, then to the robotic hand, and finally to the product box. In this way, you only have to animate the conveyor, the robotic arm, and the box. The dynamically parented object goes along with the animated objects.



Set Parent World

Draw Selected

Select the target item to be parented (in **Items mode**). When you click the **Dynamic Parenting** button, the **Dynamic Parent** toolbox opens. Click the **Add Parent** button to initialize the Dynamic Parenting controls for the target. (Assign this when the Timeline is at frame 0.) Modo adds the Dynamic Parent Channel Modifier and provides some in-viewport controls for you to define the parent/child relationships.

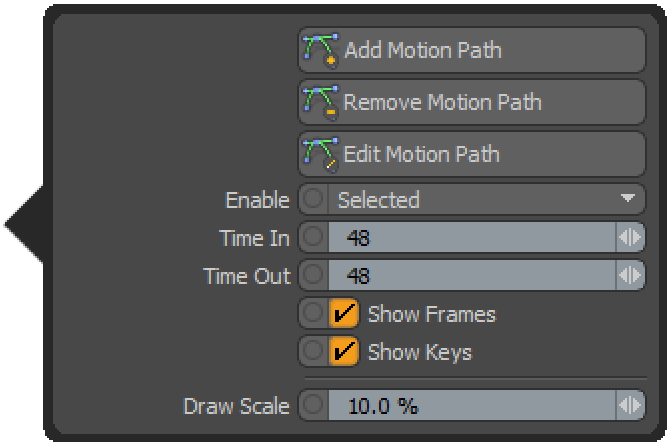
Using the dialog, you only need to move the Timeline to the appropriate position and then use the **Set Parent** option to set the target parent item within the scene defining the parent relationship. Once you assign a parent, that parent remains associated with the item in time until you apply the next assignment. The **World** option is like unparenting the item in a place. The item remains in the position it is in when you select the option.

You can set as many different parent items as required, but you can only assign a single parent per key. If you click the **Remove Current** button, Modo removes the current parent from the target and leaves the other assignments in place. If you click the **Select Modifier** button, Modo selects the actual dynamic parent modifier item in the Items (Scenes) List for you to modify its attribute in the **Properties** panel. The **Draw** control defines the drawing of the control handle in the 3D viewport: with **Off** Modo never draws the handle, with **On** Modo always draws the handle, and with **Selected** Modo draws the handle only after you select the target item.

You can also assign dynamic parenting directly in the 3D viewport using the control handle. After selecting the target item and enabling the dynamic parenting option, you to drag the control dot (the purple dot) to the target parent item to link them. Right-clicking the control unparents the items.

Motion Path

Motion paths are a useful visual aid for seeing how an item moves over time within a scene. With Modo's motion path feature, you can directly edit the curves (such as three-dimensional Bezier curves) in the GL viewport. Additionally, you can adjust the positional easing of the keyframes. Modo identifies each frame step visually as a dot along the path.



Enable: Selected
Time In: 48
Time Out: 48
☑ Show Frames
☑ Show Keys
Draw Scale: 10.0 %

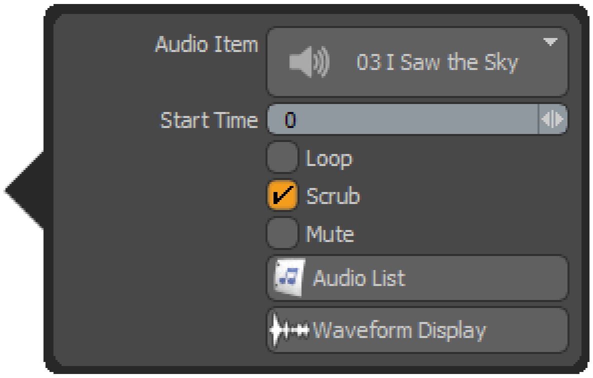
You can view any animation path by first selecting the target item. You may need to assign some position keyframes to the target prior to adding the path. (If there is no motion on an item, there is no path to view.) Once selected, clicking the **Motion Path** button opens the dialog and you can click the **Add Motion Path** button to draw the path. The green section of the path represents the time behind the current item, and the purple path represents time in front of the item.

To modify the path directly in the 3D viewport, enable the **Edit Motion Path** option by clicking the button. Doing this draws the path within the viewport as a ribbon to represent the Z-axis rotation of the item along its length. At each keyframe location, Modo shows small cyan cubes. You can drag these cubes to reposition them. When you reposition these, tangent handles show on the selected key. You can drag the tangent handles to adjust the curvature of the incoming and outgoing curve.

How you edit the curve is dependent on which channels are keyed at the keyframe location. You must key all three (X,Y, and Z) channels at each location for free curvature editing; otherwise, Modo locks the curve to the particular axis that it was keyed to. Right-clicking and dragging the handles adjusts easing in and out of the key without adjusting the path's shape. (**Ctrl**+click has the same effect.) Clicking the **Edit Motion Path** button again disables editing of the path. You can define whether the path displays with the **Enable** option: with **Off**, Modo never draws the path, with **On**, Modo always draws the path, and with **Selected** Modo draws the path only when you've selected the target item.

The **Time In/Out** options control the number of keyframes displayed before and after the current location. This controls the length of the path around the current time. The **Show Frames** and **Show Keys** options control the visibility of the small frame step tick marks on the path and the cyan cube keyframes respectively. The **Draw Scale** value determines the thickness of the ribbon as it relates to the path. A value of **0%** disables the ribbon aspect of the path drawing. You can disable the path display for an item by clicking the **Remove Motion Path** button in the pop-up window with the selected target item.

Audio


Audio Item: 03 I Saw the Sky
Start Time: 0
☐ Loop
☑ Scrub
☐ Mute

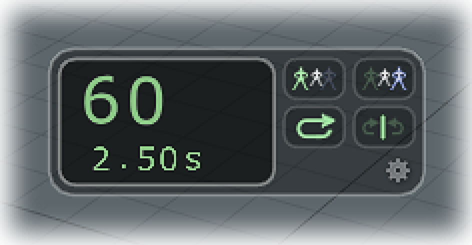
Audio is often an important part of the animation workflow for syncing actions to sound cues or, more importantly, lip sync with a character. Modo can load sound files and play them back with the Timeline. If you click the **Audio** button, you can use the controls in the **Audio** popup window. Once Modo loads the audio, its waveform displays inside the Timeline. Clicking the **Audio Item** opens the **Audio** viewport for you to load and select a variety of audio file formats. Clicking a loaded file selects it and assigns it as the current audio file to the Timeline. Once you load and select the audio file, any subsequent playback of the Timeline plays the audio. (Your computer system must have the proper playback hardware connected and configured.)

The **Start Time** setting controls the frame at which the audio file starts to play. If you select the **Loop** checkbox, Modo plays shorter sound files consecutively and repeats the file until the end of the Timeline. If you select the **Scrub** checkbox, Modo controls playback of audio when you manually shuttle the Timeline.

Selecting **Mute** temporarily disables the audio playback. Clicking the **Audio List** button shows or hides the **Audio List** popup, where you can select audio items and edit their associated attributes in the **Properties** panel. Clicking the **Waveform Display** button displays or hides the audio file's waveform in the Timeline itself. See the Using Audio Overview topic for additional details about using audio in Modo.

Time Tool

With the Time tool, you can shuttle the Timeline indirectly while in the 3D viewport. Doing this can increase efficiency when adjusting items in the viewport and eliminate the need to go directly to the Timeline to manually scrub back and forth. Clicking the **Time Tool** icon activates the function and adds its controls as an overlay to the 3D viewport. The large number in the overlay displays the current Timeline location in the chosen format, and the smaller numbers below show the current time.



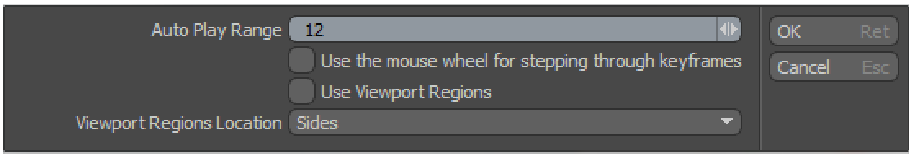
Outside of the display itself, you use the Time tool to control the Timeline with the right mouse button and, possibly, a middle button. When the Time tool is active, you can right-click in the 3D viewport and drag the pointer to scrub the Timeline position. If you hold down the **Ctrl** key while right-clicking and dragging the Time tools jumps between keyframes. With a multi-button mouse, clicking in the viewport with the middle mouse button invokes real-time playback of the Timeline around the current time, holding down control with the middle mouse button will temporarily jump to the previous keyframe. The Time Tool specific overlay provides additional functionality for improving workflow:

A click temporarily jumps to the previous keyframe; a right-click steps the Timeline to the previous keyframe.

A click temporarily jumps to the next keyframe; a right-click steps the Timeline to the next keyframe.

Determines the middle-click playback mode by indicating **Loop** or **Ping-Pong**.

Scrub-mode control, when enabled, returns the Timeline position to the current time when the mouse button is released.



Auto Play Range 12

☐ Use the mouse wheel for stepping through keyframes

☐ Use Viewport Regions

Viewport Regions Location Sides

Clicking the **gear** icon in the overlay opens a dialog box for you to define the **Auto Play Range**, which is the number of steps around the current time that is played when pressing the middle mouse button. You can also select a checkbox to use a scroll wheel to step through keyframes.

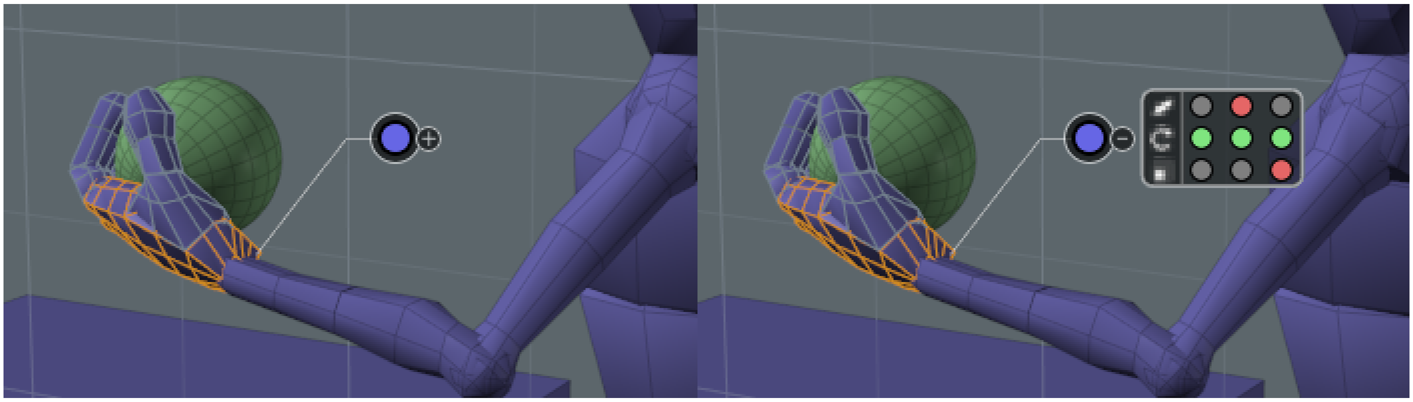
If you select the **Use Viewport Regions** checkbox, Modo places easy-access viewport overlay buttons for the keyframe jump controls. The button regions are invisible until you move the pointer over the designated **Viewport Regions Locations** (**Sides** or **Top Corners**). The buttons appear as transparent gray boxes when visible, and you can click> them to jump to the previous or next keyframes temporarily or right-click to step the Timeline to the next or previous keyframe. (The direction depends on which button you press: left button goes back and right goes forward.)

The Time tool does not operate like a regular tool because you can activate and apply other tools while the Time tool is active. Also, the Time tool uses the middle and right mouse buttons. Therefore, using the Time tool overrides any additional functionality these mouse buttons provide for specific tools. You can only disable the Time tool by clicking the **Time Tool** button again.

Keyframe Tool

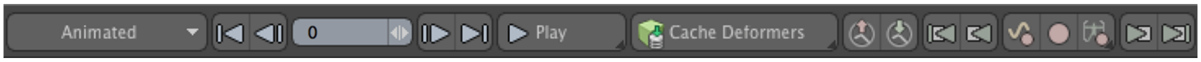
The Keyframe tool adds a control widget to the 3D viewport for you to visualize and modify keyframes directly in the Animation view. Clicking the **Keyframe Tool** button enables the widget. Then, when you select any item that has keyframes associated with it, the widget connects the keyframes to the item's center location. The color of the widget signifies its current state: red for a keyframe, green when animated (but no keyframe), and blue with a mix of keyframes on the item.

Clicking the **+** sign opens the keyframe popup, which shows the keyframe states for all animated transforms of a given item. You can click the dots that represent the X, Y, and Z channels for each transform to toggle the keyframe state for each. Clicking the transform icon, itself, automatically keys all three channels. Clicking a green dot sets a key for that channel and changes the dot to red. Clicking a red dot removes the key. Clicking when the widget dot is blue sets a key for all animated channels.



The tool only works for items that currently have keyframed transform channels. The widget does not draw when you have selected non-animated items.

Animation Timeline Controls



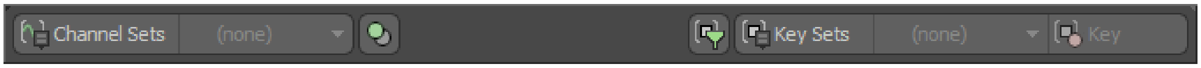
The Timeline controls provide the functions for shuttling the Timeline. Clicking the button activates the function. Some buttons also have alternate functions that can be accessed by right-clicking the button. You can view the keyboard shortcuts for Timeline transport in the Timeline topic.

Buttons	**Description**
Animated ▾	This controls the Auto Keyframe settings. For details, see the Timeline topic.
|◁	Moves the Timeline to the first frame of the **Current Start** position.
◁||	Moves the Timeline backwards one frame.
0 ◁▷	Displays the current Timeline position. You can manually specify the current frame value.
||▷	Moves the Timeline forward one frame.
▷|	Moves the Timeline to the last frame of **Current End** position.
▷	This is the **Play** button. Clicking it plays the Timeline forward. Right-click to access the **Reverse Play** button to play the Timeline backward.

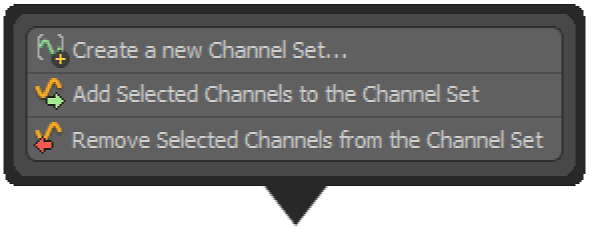
Buttons	Description
▷	(Play Keys) Plays only the keyframes for the current actor. (It also supports the Range Time Markers.)
Cache Deformers	Plays through the timeline one frame at a time to populate the cache with meshes resulting from deformers and mesh operations. Right-click to reveal the **Cache Deformers at Current Time** option.
Cache Deformers at Current Tim	Caches meshes resulting from deformers and mesh operations around the current frame (about +/- half a second).
	Removes a frame and moves all the following keyframes one frame back.*
	Adds a frame and moves all the following keyframes one frame forward.*
⏮	Moves the Timeline to the first keyframe position for the selected item(s).
◁	Moves the Timeline to the previous keyframe position.
●	Assigns a keyframe to all currently selected channels.
●	Assigns a keyframe to all animated channels belonging to the currently selected items.
●	Assigns keyframes to all animated channels belonging to the items in the actor and also to any channel members.
●	Assigns keyframes to all animated channels belonging to the items in the Group.
▷	Moves the Timeline to next the keyframe position.
⏭	Moves the Timeline to the last keyframe position.

* These controls also work with actions when applied while one is active (and there are no items or channels selected).

Channel Sets

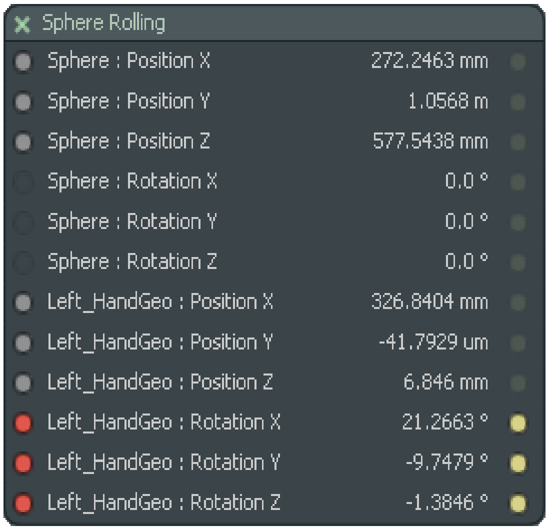


With channel sets you can display a collection of channels from the same or different items in the 3D views with an interface similar to the Channel Haul tool. There are two main differences. First, the interface remains open until you explicitly close it - even when selecting other items in the scene. Second, Modo replaces the hauling controls of the **Channel Haul** tool by keying buttons.


Create a new Channel Set...
Add Selected Channels to the Channel Set
Remove Selected Channels from the Channel Set

You create sets in the **Channel Sets** dialog. By selecting a single item in the **Items** list (or the 3D viewport) and then selecting the target channels in the **Properties** or **Channels** viewport, you can click the **Channel Sets** button to open the dialog. Clicking the **Create new Channel Set** button opens the **New Channel Set** dialog where you can specify a name for the set. Leave the options in the dialog at their default values (**Mode**: **From Selected Channels**) to automatically add the previously selected channels to the set when Modo creates it. After clicking **OK**, Modo creates the set and and activates it, which makes the channel visible in the 3D viewport.

You can add additional channels from other items by clicking **Add Selected Channels** or eliminate target channels from the active set by clicking **Remove Selected Channels**. After you create a channel set, you can use the dialog to specify the current **Channel Set**. This set displays its interface in the 3D views. Setting this to **(none)** closes the interface.


Sphere Rolling

●	Sphere : Position X	272.2463 mm	○
●	Sphere : Position Y	1.0568 m	○
●	Sphere : Position Z	577.5438 mm	○
○	Sphere : Rotation X	0.0 °	○
○	Sphere : Rotation Y	0.0 °	○
○	Sphere : Rotation Z	0.0 °	○
●	Left_HandGeo : Position X	326.8404 mm	○
●	Left_HandGeo : Position Y	-41.7929 um	○
●	Left_HandGeo : Position Z	6.846 mm	○
●	Left_HandGeo : Rotation X	21.2663 °	●
●	Left_HandGeo : Rotation Y	-9.7479 °	●
●	Left_HandGeo : Rotation Z	-1.3846 °	●

The viewport interface works much like the **Channel Haul** tool. You can change channel values by clicking on the channel name and dragging horizontally. By clicking the numeric value of the channel, you can edit the value directly. When a value has been edited, the small dot to the right of the value turns yellow. This allows you to discard the change by simply clicking on the yellow dot.

You move the interface around in the 3D views by dragging the header area that shows the name of the set and close the interface by clicking the **x** button on the upper-left corner. The main difference from the **Channel Haul** tool is that, instead of hauling controls to the left of the channel name, the **Channel Set** interface has controls for keyframing the channels. These work in the same way as the controls in the **Properties** viewport. Clicking them toggles the key state of the channel at the current time (or at the key time set by using the middle mouse button to drag in the Timeline).

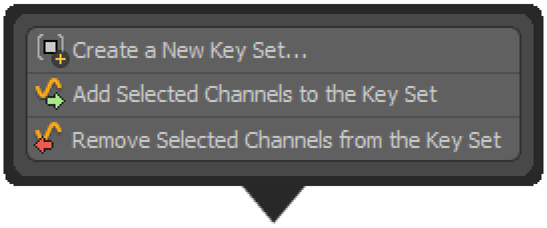
Channel sets are a special type of group, so you can create and edit them in the Groups Viewport. The current channel set is the one that displays in the 3D views and can be specified from the pop-up menu in the toolbar. When you select a Channel Set, the Graph Editor and the Dope Sheet display the contents of the current channel set.

Animated Channel Toggle

To the right of the **Channel Set** controls is the **Animated Channel** button. If no channel sets are active (set to **none**), the **Animated Channel** button, when enabled, displays any animated channels (with keyframes) associated to the current selection in the channel sets' viewport window. This window makes it convenient for you to work with channels by adding keyframes without needing to make specific sets to work with.

Key Sets

Key sets provide a convenient and quick way for adding keys to multiple channels at once. You can create key sets in advance. For example, you might create a key set for all the animated channels in a character: its arms, hands, and so forth. Alternatively, you can create key sets as you work by adding or removing channels as needed. You select the active key set from the popup menu. To set keys on all the channels in the set, you assign them with the **keying** button to the right of the **Key Set** controls.

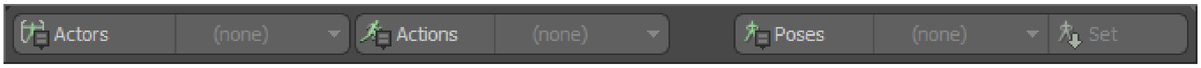

Create a New Key Set...
Add Selected Channels to the Key Set
Remove Selected Channels from the Key Set

You create and modify key sets in the same manner as Channel Sets by selecting an item and the target channels and then clicking the **Create a New Key Set** button. In the **New Key Set** dialog, you can specify the name of the set. After you click **OK**, Modo creates the set and makes it active. When a specific key set is active, pressing the **Key** button defines a keyframe for all items within the key set on the current frame. (The items do not have to be selected.)

Filter Option

Click the **Filter** button to control visibility of certain key sets associated with an actor. Modo parents the Group Items together. You associate these in the Groups Viewport by clicking the key set group and dragging it onto the actor groups. When assigned as such, Modo only displays the key set group in the pop-up menu when that actor is active. This is useful when working with multiple actors that each have their own unique key sets.

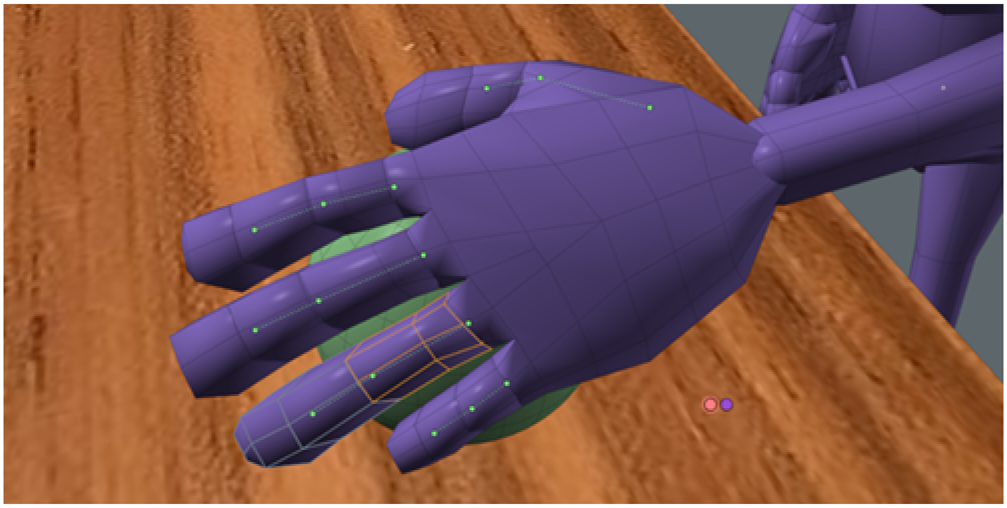
Action, Actors and Poses



The **Actor**, **Action**, and **Pose** menus provide ways to simplify the steps for selecting and keyframing elements in a complex scene. Actors are collections of multiple items that provide ease in selecting the associated elements and setting keyframes. Actions are alternate takes of the same Timeline for an entire actor. Poses are a collection of specific keyframe values that can be applied anywhere along the Timeline for all or part of an actor (such as different hand positions). When working with actors, actions, and poses, you must take many considerations into account. See the Actor, Action, and Pose topic for detailed information.

Keying Hierarchy

The Hierarchy Mode options provide a workflow enhancement for you to set implicit selections of items based on a hierarchy. For instance, you could keyframe all the fingers in a hand while having only a single finger joint actually selected. Generally, you can only adjust selected items. This is the default behavior, but, based on the hierarchy of the elements being animated (such as a bipedal, jointed character), you can change the hierarchy mode to manage associated objects simultaneously. The hierarchy mode also affects the **Graph Editor** and **Dope Sheet** viewports to display the associated items based on the mode type.



Clicking the mode button opens a popup for you to specify an alternate mode. If you select a mode other than **None**, Modo indicates additional items with implied selections in the 3D Viewports by small colored dots at each item's center position. The hierarchy connections show as dotted lines drawn between the items. In the illustration, the **Branch Related** mode is selected so that the single finger joint that is explicitly selected also implicitly selects all the other fingers. The colors for each dot represent the keyframe status (same as with the general controls): green is animated, red is a keyframe position, blue is a mixed state, and orange is for non-animated elements.

Pose Tool

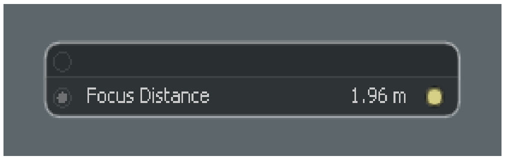
With the **Pose** tool, you can easily pose and animate hierarchies of elements by using a simplified drag and pin workflow. When the **Pose** tool is active, it simulates an IK type of setup between elements. (Inverse Kinematics, or IK, is the ability to animate the position of a goal item and have other related elements automatically follow by reaching toward the goal as it moves.) This helps to simplify the animation workflow greatly and provides an intuitive method to position elements. Although often used with bipedal characters, Modo does not limit the **Pose** tool specifically to these types of setups. You can manipulate any hierarchy of joints, items, or elements with the **Pose** tool, and it is equally effective.

With this workflow, you select the base (or root element) of a hierarchy and then click the **Pose** tool button to activate the tool. When this tool is active, blue dots show on each item's Center in the 3D viewport for each element within the selected hierarchy. From there you can drag any dot to move the target element, which also affects the related elements in the chain. How far the related elements move is dependent on how far you drag the target itself, their position relative to the target, and the weighting value specified in the **Pose** tool's **Properties** panel. When moving an item isn't the desirable behavior, you can also pin specific items to keep them stationary. You do this by right-clicking any dot and changing its color to red to indicate its locked state. Position changes on either side of the locked element keep the elements on the opposing side stationary.

The **Pose** tool offers a simple-to-use, but powerful means of working with characters. See the Pose Tool topic for more detailed information.

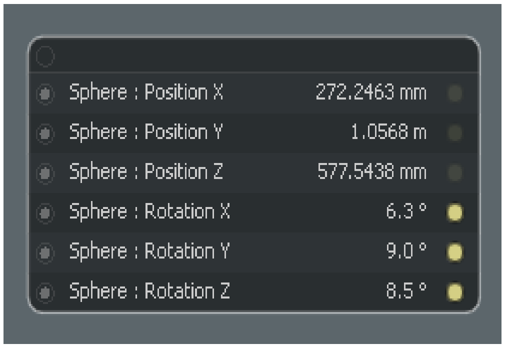
Channel Haul Tool

You use the **Channel Haul** button to adjust any item's selected channels by dragging directly in the 3D viewport.



Focus Distance 1.96 m

The channels visible within the viewport depend on the channels that you selected in either the **Properties** or **Channels** viewport. Clicking the **Channel Haul** button or pressing the **C** key activates the tool and the viewport display in the 3D viewport. Dragging anywhere off the display simultaneously edits all active fields. You can limit this to specific channels by clicking the **Boolean** buttons preceding each channel. The topmost **Boolean** button enables or disables all channels at the same time. You edit values individually clicking the channel name in the display and dragging the pointer to interactively adjust the value. You can also specify a new value by clicking the current value and typing a new one. With the **Channel Haul** tool, you can modify up to 6 specified channels.



Sphere : Position X	272.2463 mm
Sphere : Position Y	1.0568 m
Sphere : Position Z	577.5438 mm
Sphere : Rotation X	6.3 °
Sphere : Rotation Y	9.0 °
Sphere : Rotation Z	8.5 °

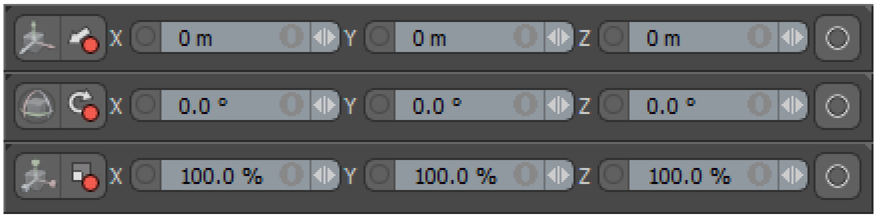
If you have not selected a channel, Modo reminds you that the tool is still active. Click the **Channel Haul** button again (or press the **spacebar**) to disable the tool.

No Channels Selected

Apply/Discard Changes Buttons

Click the **Apply** button to apply channel changes you have made to a scene or action or click the **Discard** button to discard all the changes. Click **Apply** when you want to commit changes without having to scrub the Timeline and return to the frame that was previously being editing, or click **Discard** to erase all changes quickly. The latter is a quick way to return anything that you've edited back to its previous values without having to undo multiple times or to go through all the changed channels and reset them manually. For example, you could adjust a pose or test rigging controls. Then, click **Apply** if you want to retain the many changes to multiple items or click **Discard** if you don't.

Keyframe Control Block



X 0 m Y 0 m Z 0 m

X 0.0 ° Y 0.0 ° Z 0.0 °

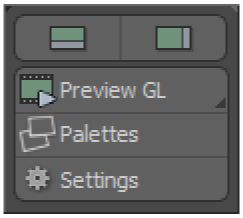
X 100.0 % Y 100.0 % Z 100.0 %

This section of the **Animation** interface layout works like a spreadsheet and shows all the transform values for the current selection. Modo dedicates each row to a specific transform (either **Position**, **Rotation**, or **Scale**, from top to bottom). The first, second, and last columns have function for each row. The first column has the tool buttons that enable or disable the actual transform tool associated with it. You use the second column to automatically set a keyframe for all three axes of the specific transform. The X, Y, and Z input fields represent the current values for the item. In the last column on the right, the buttons with the circles reset the item by adding a keyframe for all three values of the given transform to **0** (zero).

Coordinate Options

The **Animation** layout supports three different types of action centers that automatically orient the Transform Tool's handles when working with items. If you click the **World Coordinates** button (the left icon), Modo orients the handles to the world-based XYZ planes in the scene. If you click the **Local** button (the right icon), Modo orients the handles to match the orientation of the target element that is being modified. (In this case, local is local to the selected item.) When both buttons are disabled, which is the default state, then the orientations of the tool handles match the parent item of the selected element for you to transform items relative to the item that defines its orientation. Clicking the buttons in the interface toggles the modes on or off.

Interface Toggles



The buttons in the last section of the **Animation** layout hide and show viewports in the animation layout, provide a way to convert the GL viewports to video and frames and provide access to a variety of utility viewports as palettes. The top left button shows or hides the Graph Editor or **Dope Sheet** viewport. When you click the button, the viewport opens below the 3D viewport. If you click the button again, Modo hides the viewport. The top right button shows or hides the Items (Scenes) List or Properties Panel cluster of tabs present in the other interfaces. Depending on the amount of screen space available, you can maximize the main 3D viewport workspace when it's necessary to focus on the animation itself.

Preview GL

You use the Preview GL options to convert the GL viewport into a video or frame sequence format for external playback. In complex scenes, converting can reduce hardware limitations to allow for full-speed, full-frame playback. This can also be useful for reviewing animations when someone who doesn't own Modo needs to see an animation for approval purposes. Modo makes video previews easy to create. When you apply any of the commands, Modo temporarily disables any visible HUD elements including the navigation icons, information overlays, tool feedback overlays, and onion skinning to keep the resulting video uncluttered. Modo only displays the frame number. After the recording completes, Modo restores the HUD elements to their previous state.

Modo provides three preview options, which provide ways to make a temporary video that opens in an external player for review. Modo determines the video format based on the **Movie I/O** setting in File I/O Preferences and determines the size by the size of the active 3D viewport when you invoke the **Preview** command. Once Modo creates the video file itself, it calls the operating system for the default playback mechanism for the chosen format. Make sure you have proper video playback support in your operating system before creating a preview. You click the button to invoke the command, and Modo sequentially goes through each frame of the current start and end times to create the preview file. Once Modo creates the preview it opens the default playback application with the video automatically loaded. You can then use the playback software to view the video.

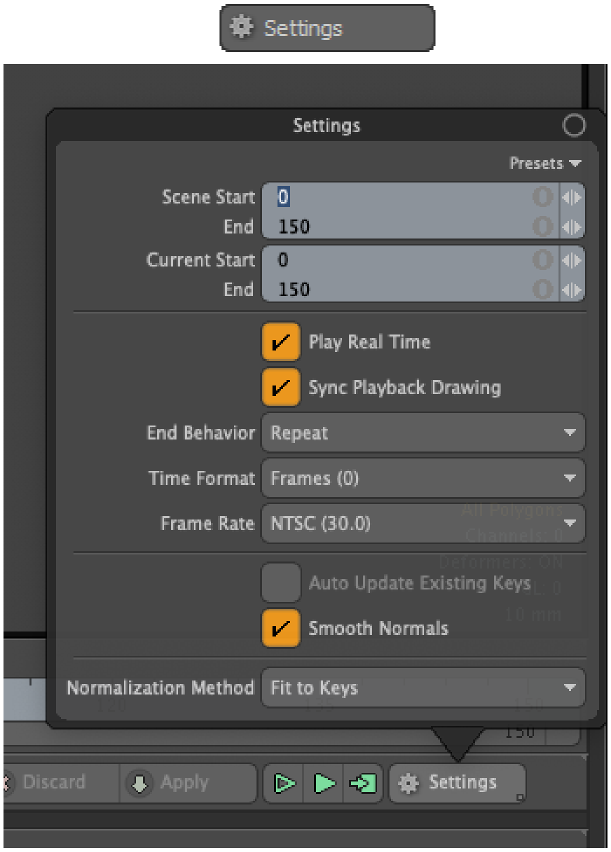
Right-clicking the **Preview GL** button opens the **Preview** menu where you specify alternate options. Specifying **Preview GL 1/2** produces a video at half the size of the 3D viewport and **Preview GL 1/4** produces a video that is one quarter the size of the 3D viewport. The **Record GL** options provide identical video files, with the same 1/2 and 1/4 options; however, by invoking these commands Modo prompts you to specify a **Save** location and **Filename**. Use **Preview GL** for quick, local reviews and **Record GL** when you need to review the video at a later time or send it to another location.

Specifying **Image Sequence** saves a sequence of bitmap images that match the active viewport size to the designated folder. Invoking the command opens an OS-specific folder for you to define the location and provide a filename and format for the preview. Once specified, Modo sequentially steps through the Timeline and saves a file for each frame. You then use a sequence player or video-editing application to gather the still frames into a playable video sequence.

Palettes

Click the **Palettes** button to choose from a variety of useful viewports that open as floating palettes in the **Animation** workspace. Doing this eliminates the need to switch to an alternate workspace layout for simple tasks. The commands act as toggles: clicking opens the named palette; the second click closes the palette. (Also, you can close a palette by clicking the OS-specific close button.) You can arrange open palettes by dragging a palette's header. While the palettes are open, you can hide them temporarily by pressing the grave accent (`) key. This is also the tilde (~) key next to the 1 on American keyboards.

Settings



Scene Start: 0
End: 150
Current Start: 0
End: 150
☑ Play Real Time
☑ Sync Playback Drawing
End Behavior: Repeat
Time Format: Frames (0)
Frame Rate: NTSC (30.0)
☐ Auto Update Existing Keys
☑ Smooth Normals
Normalization Method: Fit to Keys

Click the **Settings** button to open the Scene Options panel to define frame ranges for animations and to set some formats and default behaviors.

Scene Start/End: Sets the sequence length using the start and end frame numbers.

Current Start/End: Sets the current length using the start and end frame. When animating, you may concentrate on a single area of a scene for fine-tuning. With the current range setting, you can temporarily crop out the rest of the scene so you can concentrate on the one section. Double-click the Timeline to reset the **Current** range to the **Scene** range.

Play Real Time: When selected, the sequence plays back in an OpenGL viewport using the **Frame Rate** setting to the best of the host system's abilities (and skipping frames if necessary). When clear, Modo plays every frame in the sequence at the fastest speed possible, which (depending on scene complexity) may be slower or faster than the specified frame rate.

Sync Playback Drawing: Due to Modo's method of processing sequences in OpenGL viewports, mesh drawing may be out-of-sync during playback (i.e. separate meshes, like eyes, lagging behind a character's animated body and head). This playback option is enabled by default. Disabling this option may lead to faster animation playback.

End Behavior: Determines what happens when playback reaches the end of the play range. The following options are available:

- **Hold** - Stops the animation at the end of the playback.
- **Repeat** - The default setting, loops playback indefinitely.
- **Wrap** - Repeats the playback until the current time matches the initial time and then it stops.

Time Format: Use to specify which standard Modo uses when displaying individual frame numbers

- **Frames** - Indicates frames as sequential numbers by using a standard numbering format.
- **Seconds** - Indicates frames as a passage of time by using the *0.0s* format.
- **SMPTE** - Specifies frames by using a shorthand method **00:00:00:00**, which represents hours:minutes:seconds:frames.
- **Film Code** - Specifies frames by a linear measurement (which would be the length of the film in feet if the animation were filmed) using a **0.0 + 0** format.

Frame Rate: You can specify a frame rate by standard definitions, such as **Film (24)**, **PAL (25)**, and **NTSC (29.97)**. (The values shown are in frames per second, or fps, which is the number of still images that play back in sequence in one second of time.)

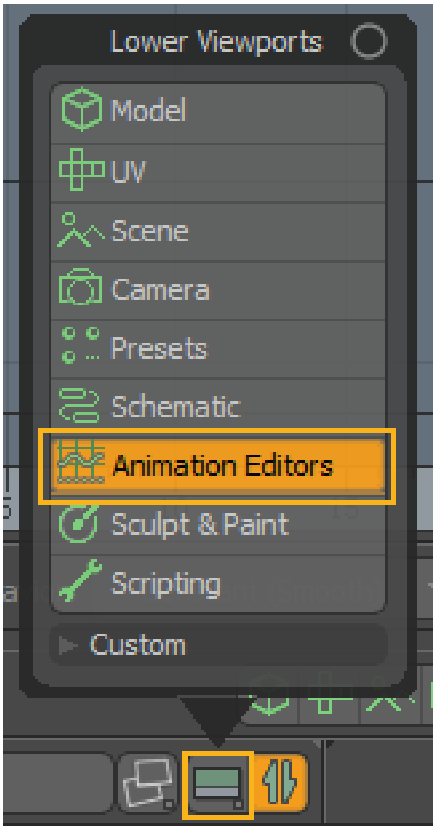
Auto Update Existing Keys: This option is only available when **Auto Key** is set to **Off**. Normally, when **Auto Key** is off, any edits that you make are discarded when the current time is changed unless the edits have been manually keyed. With this option, Modo updates existing keyframes automatically if their values are changed without requiring manual keying. Modo still discards un-keyed changes when the current time changes.

Track View

With the Track view (also called the **Dopesheet**) you can edit keyframes directly in a simplified view that can be accessed by expanding the Timeline in any time-enabled layout. Editing keyframes at this level can improve the process of achieving proper timing for any animated situation (especially in the case of character animation). By clicking the top edge of the Timeline, itself, and dragging upward, you can expose the (normally hidden) viewport. If you have selected items in the scene with keyframes assigned, the viewport indicates this.

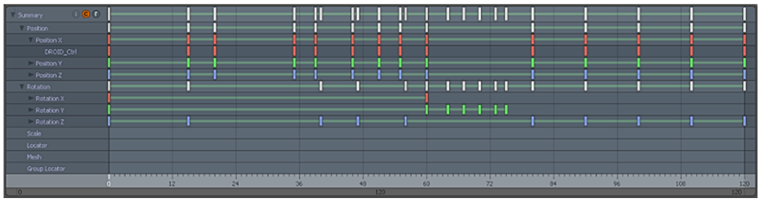
To open the **Dopesheet**, click **Animate** > **Dopesheet**.

You can also view the **Dopesheet** using the **Animation Editors** accessible from the **Lower Viewports** Popover menu at the bottom of the Viewport.


Lower Viewports
Model
UV
Scene
Camera
Presets
Schematic
Animation Editors
Sculpt & Paint
Scripting
Custom

Lower Viewports Popover menu

The **Animation Editors** is a layout containing the **Graph Editor** and **Dopesheet**, as well as the standard **Channels** viewport so you can easily select parts of your rig and channels and make adjustments.


Summary
Position
Position X
DROID_Ctrl
Position Y
Position Z
Rotation
Rotation X
Rotation Y
Rotation Z
Scale
Locator
Mesh
Group Locator

For ease of editing, you can view keyframes in Track view in three different ways. The best view depends on the type of edit you want to do.

- Selecting an item displays its keyframes in a Summary view, which is a streamlined view that has handles to indicate frames that have keyframes assigned to them. With the handles you can easily edit all the keyframes of a specific item or group of items. You access individual channels for items by clicking the triangle to the left of the item's name and, if desired, next to each individual channel's name.

- Alternatively, you can click the small round **C** button in the Track view to change the display so it is based on channel types. Accessing the channel for a specific item requires you to click the triangle next to the channel group's name.
- Finally, selecting a specific channel or group of channels in the **Channels** viewport displays those channels directly in the Track view.

Editing Keyframes

You can select single keys by clicking a keyframe handle, which changes its color when selected to orange. You can select multiple keys by clicking in an open area and dragging over multiple keyframe handles. To add to a selection, press the **Shift** key when selecting the additional handle, and conversely, press the **Ctrl** key as you click a selected handle to remove it from the multiple selection.

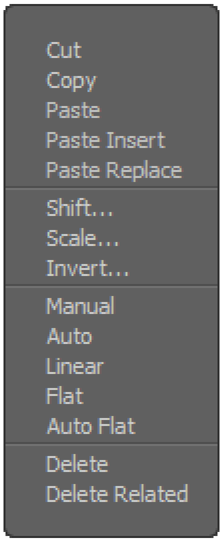
The white keyframe handles in the topmost row are the master keys that provide a convenient way to adjust all handles on a specific frame. Moving a master key moves all keyframe handles below it. Similarly, for example, when moving a key for Position X, the related master key automatically moves along with the Position X keyframe.

Once you select any keyframe, you can drag it to the left or right to adjust its position along the Timeline and move all other selected handles. Right-clicking and dragging over an empty area of the graph moves the selected handles as well without needing to click a specific handle. When you move keyframes, Modo shows the number of frames offset (+/-) from where you started in real time in the upper-left area of the Track view. Positive values from the start frame indicate moving to a later time; negative values indicate an earlier time. To move all frames after a selection (without needing to specifically select them), click the **Ripple Edit** button (the small **r** icon) in the upper-left corner of the Track view.

Note: When in **Ripple Edit** mode, any right-click needs to be over the active key zone (indicated by horizontal green bars between keyframes) for the Ripple Edit mode to be effective.

To scale selected keyframes, you must select at least two handles. You can do this in two ways. By pressing the **Shift** key as you right-click and drag on an open area,Modo scales the selection from the right-click location inward or outward depending on the drag direction. The amount of the scaling appears in the upper-left area of the Track view as you move the pointer. Pressing the **Ctrl** key as you right-click and drag scales the set of keyframes from the position of the last (or the first selected keys) depending on which side of the selection the pointer is closer to. For example, to keep the first key location (in time) at a fixed position and scale the other selected keys relative to it, press **Ctrl** as you right-click and drag down to the right of the last key selection and then scale by dragging to the right or left. Conversely, to scale a selection relative to the last selected key, press **Ctrl** as you right-click and drag down to the left of the first selected handle and then drag to the right or left.

You can create new keyframes by middle-clicking in the Track view. Modo creates a keyframe at that position on the Timeline for all selected channels or items.



In the Track view, a right-click menu provides commands to cut, copy, paste, and delete keyframes and also to scale and shift keyframes by specific amounts. You can also set the slope of keyframes.

Cut: Cuts the selected keys from the view and stores them in a temporary buffer for later access.

Copy: Copies the selected keys from the view and stores them in a temporary buffer for later access.

Paste: Places the keys from the temporary buffer into the Timeline and inserts them at the current time position.

Paste Insert: Places the keys from the temporary buffer into the Timeline, inserts them at the current time position, and moves the existing keyframes beyond the insertion point as needed.

Paste Replace: Places the keys from the temporary buffer into the Timeline, inserts them at the current time position, and removes any overlapping keyframes.

Shift: Opens the **Set Frame** dialog for adjusting the time position.

Scale: Opens the **Scale Keys** dialog for scaling the value of the selected keys.

Invert: Opens the **Key Invert** dialog for adjusting the value of the selected keys.

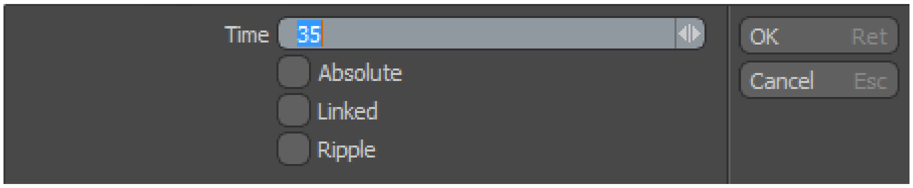
Use the next set of commands to automatically define the slope of the interpolation curve connecting keyframes.
Manual- Gives you full control over adjusting the slope (handle angle).
Auto- Automatically adjusts the slope to produce a smooth ease-in and ease-out from one key value to the next.
Linear In- Angles the incoming slope to point toward the previous key to produce a straight line between

keys.
Linear Out- Angles the outgoing slope to point toward the next key producing a straight line between keys.
Flat- Sets the slope to **0** degrees regardless of the positions of the previous and next keys.
Auto Flat- Acts similar to **Auto** except when the incoming or outgoing keys match the value, then it acts like **Flat**.
For more information on slope and how it relates to keyframes, please see the Graph Editor topic.

Delete: Removes the selected keyframes from the Track view.

Delete Related: Removes the selected keyframes from the Track view, as well as removing any related channel keyframes. (For example, if you delete the **Y** channel, this command also removes the **X** and **Z** channels as related channels.)

In the **Set Frame** dialog you can adjust the time position.


Time 35
☐ Absolute
☐ Linked
☐ Ripple

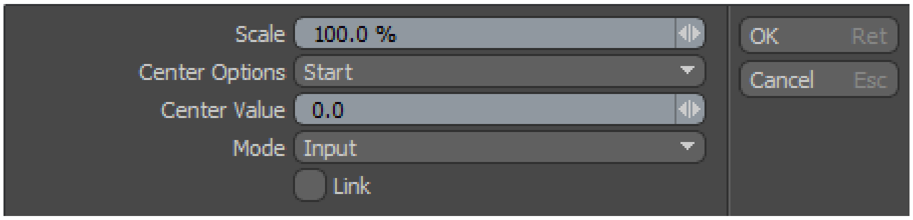
Time: Determines the position in time to move the selected keys to.

Absolute: When enabled, Modo uses an absolute time that you can specify. When disabled, the time is relative to the current time.

Linked: When enabled, linked keys (such as **RGB** channels) remain linked in the same relative position even if only a single channel is edited. When disabled, you can move keys freely.

Ripple: When enabled, all values of those following the selected keys are shifted by the same amount.

In the **Scale Keys** dialog you can scale the value for selected keys.


Scale 100.0 %
Center Options Start
Center Value 0.0
Mode Input
☐ Link

Scale: Defines the amount of scaling applied to the selected keys.

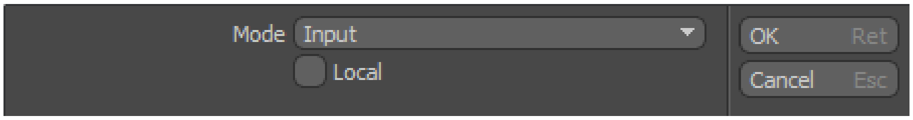
Center Options: Determines the center location to scale the keys from. You can specify: **Start**, **Center**, **End**, **Time**, or **User**. If you select **Time** or **User**, you specify the center in the **Center Value** box.

Center Value: Determines the specific center locations if you select **Time** or **User** as the **Center Options**.

Mode: Determines which direction gets scaled. **Input** scales the interval (horizontal), and **Output** scales the value (vertical).

Link: When enabled, linked keys (such as those **RGB** channels) remain linked in the same relative position even if you only edit a single channel. When disabled, you can move keys freely.

In the **Key Invert** dialog you can adjust the value for selected keys.



Mode: Input

☐ Local

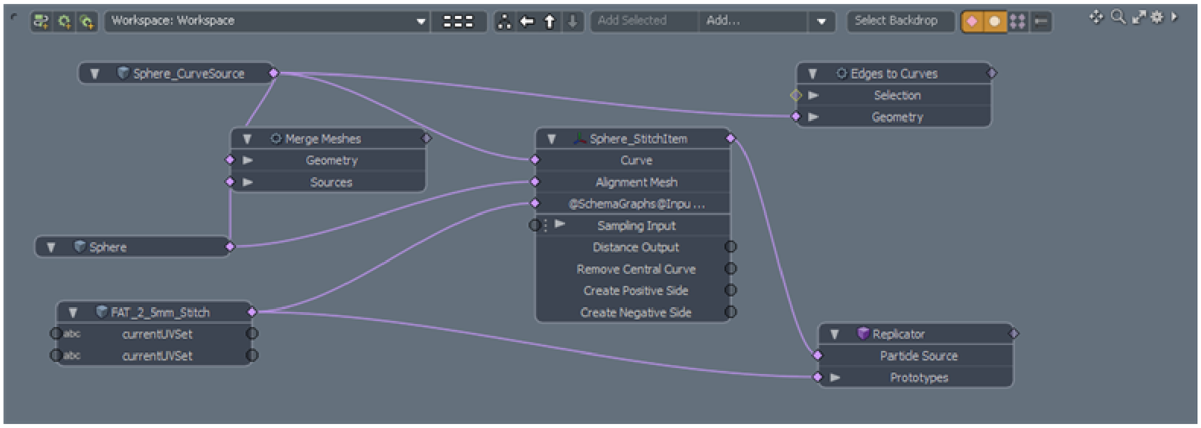
Mode: Determines which direction gets inverted. **Input** inverts along the interval (horizontal), and **Output** inverts along the value (vertical).

Local: When enabled, inversion happens individually for each channel. When disabled, inversion happens as a group for all selected channels.

Schematic Viewport

The 3D viewports give you a means to visually edit your scenes in a fully 3D environment. While great for laying out models, it's not as effective at setting up symbolic relationships between items and their channels. This particular activity is better suited to the **Schematic** viewport, a 2D environment where elements are visually linked, providing a more intuitive and powerful way to define relationships between elements; building networks of operations that combine to produce a specific result. This is, in a way, is like a graphic programming paradigm, where you define a flow of information through the network of elements visually.

In the **Schematic** viewport, you work with nodes, which are simple 2D representations of items that exist only within the **Schematic**. Each node signifies an item, an operation, or a function and its associated channels. The combinations of these nodes and their symbolic links create networks known as graphs. You use nodal graphs commonly in rigging (creating simple controls for complex actions). Modo extends this workflow by using the same functionality for dynamics, particles, and shading as well. Mainly, you use the **Schematic** viewport with the Channels Viewport and link channels together with a drag-and-drop process.


Workspace: Workspace
Add Selected
Add...
Select Backdrop
Sphere_CurveSource
Edges to Curves
Selection
Geometry
Merge Meshes
Geometry
Sources
Sphere_StitchItem
Curve
Alignment Mesh
@SchemaGraphs@Inpu ...
Sampling Input
Distance Output
Remove Central Curve
Create Positive Side
Create Negative Side
Sphere
FAT_2_5mm_Stitch
abc currentUVSet
abc currentUVSet
Replicator
Particle Source
Prototypes

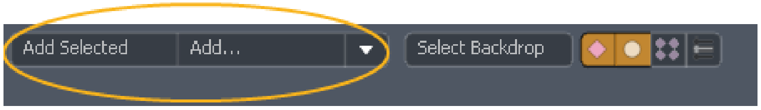
Navigating in the Viewport

The default **Schematic** view is similar to the UV view, where you view and work with flat, two-dimensional representations of information. You can navigate the view of complex node graphs using the navigation icons in the upper-right corner of the viewport. For example, drag the arrow icon to pan the view, or click the magnifying glass icon to zoom into the view. You can also use keyboard shortcuts to navigate: **Alt**+click and drag to pan and **Ctrl**+**Alt**+click and drag to zoom. Alternately, with a scroll wheel and multi-button mouse, you can middle-click and drag to pan and roll the scroll wheel to zoom.

Tip: The **Schematic** view, as other viewports, can be split to show multiple views, or even cloned to show a workspace in one view and the contents of an assembly in another view for simultaneous editing.

Adding and Arranging Nodes

When first opening the **Schematic** view (in the **Setup** layout), you see an empty workspace. Modo needs to first know what elements you want to work with. To add the elements, you can drag any item from the 3D (OpenGL) Viewport, the Items (Scenes) List, any attribute in the **Properties** panel, or the Channels Viewport directly into the Schematic view. You can add multiple items by pressing the **Shift** key as you select multiple elements before dragging them. You can also click the **Add Selected** button at the top of the viewport to automatically add any selected items or channels. You can add additional components, such as modifiers or constraints, using the **Add** menu at the top of the viewport, which lists all the available item types. Click an item to add it to the current **Schematic** workspace and the **Items** list.



When adding channels, you can drag them anywhere in the viewport and they are automatically added to their respective items. Doing this adds the associated item node as well if it wasn't already there. On a drag-and-drop operation, if you pause before dropping the item or channels into the viewport, a drop action menu appears with a number of contextually appropriate options for the drop operation.

Once you add an element, Modo displays it as a gray rectangular box, referred to as a node. The node is a bar with an icon and title (the same name as its representation in the **Items** list). As you add related channels, the box expands to accommodate them.

To duplicate a node, right-click it and click **Node Duplicate** > **Duplicate**. You can also select it, then press **Ctrl**+**D** on the keyboard. To create an instance of the node, right-click it and click **Node Duplicate** > **Instance**. You can also select it, then press **Shift**+**D** on the keyboard.

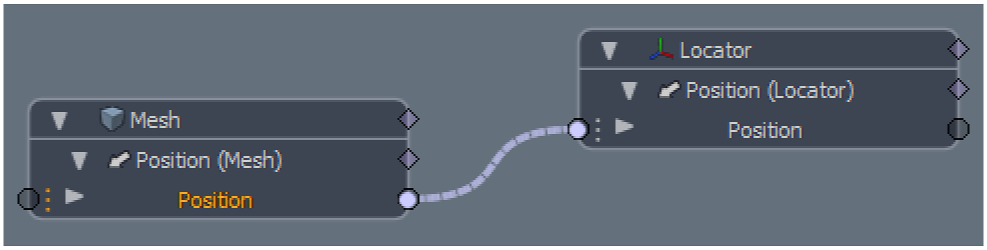
You may find it easier to work with the nodes by organizing them and arranging them in the view. To do this, click the node header (the name or title bar) and drag it to a new position. Modo snaps the nodes into positions in relation to surrounding nodes to place and align nodes for an orderly **Schematic**.

Tip: You can customize the node snapping behavior in **Preferences** > **Display** > **OpenGL** > **Schematic Grid Snapping**. For more information, see OpenGL.

To remove a node, right-click the node header and choose **Remove Node** or select the node and press the **Backspace** key. Pressing **Ctrl**+**Backspace** removes all related nodes (if you have applied the **Separate Channels** command previously). Pressing **Shift**+**Backspace** removes all nodes in the current workspace.

Tip: When adding items to the **Schematic** view, if channel links or custom user channels are present, Modo automatically adds the related channels to the node.

Defining and Editing Node Links



You create links between node connectors with a line representing their association. Once connected, node layouts are referred to as a node graph. There are two types of connectors that can be linked in the **Schematic** view between nodes: channel links, represented by circles, and relationship links, represented by diamonds.

Channel links are similar to pipes with data flowing from the Output connector of a node (the right-side connection) to the Input connector of the Target (the left-side connection). Relationship connections are different in that they represent a link between the two nodes. Channels can only be connected to channels, and relationships can only be connected to relationships. When you're connecting two channels with different data types, for instance, an **Angle** channel to a **String**, Modo converts the source data to the target data type automatically. Modo only allows correct connections to reduce errors.

To create a link between connectors, click any connector and drag to its intended target connector to create a line (often referred to as a 'noodle'). If the link is legal, the line turns green when you move the pointer over the target. Illegal connections change to red. For all legal connections, releasing the mouse button creates the link between connectors. If the line turns yellow, that line replaces the connection. You can create multiple node links simultaneously by pressing the **Shift** key to select multiple channels and then pressing **Shift** again while dragging one of the selected channels to its target.

You can move existing node links by clicking either end of a link and dragging the line to a new target. Selecting any link before creating a new item automatically links the element to the selected link.

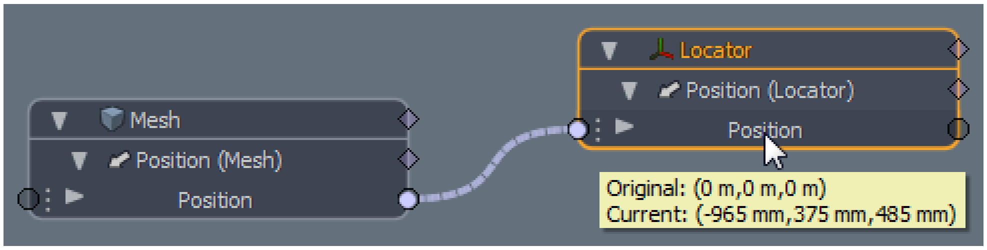
To remove a link, hold **Shift** and click the link, or select the link and press the **Delete** key. To remove all links at a connection, hold **Shift** and click the connection.

You can disable links by selecting them and pressing the **H** key. The link turns red, marking it inactive. To enable it again, press the **U** key. To disable all links connected to a node, select the node and press **H**. If you press **H** with no selection, all links are disabled.

You can select a node and all its incoming nodes by right-clicking the node, then in the context-menu, clicking **Select Inputs**.

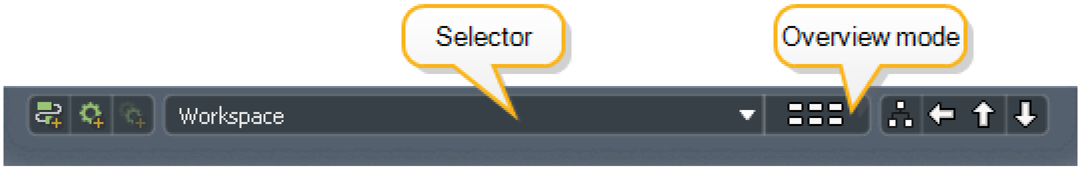
Often, Modo generates a link automatically, for example when creating simple particle simulations. When you add any pre-linked elements to the **Schematic** viewport, connectors with linked elements not present in the view show with a yellow outline signifying the hidden (minimized) connection. You can make the connected element visible by double-clicking the outlined connector.

When working with existing links, you can move the pointer over a channel row to see the original and current (driven) values for that connector. By doing this, you can see the current value for the channel itself and see the computed value produced by any linked elements. Moving the pointer over an existing link's line displays it in the foreground with a tooltip showing the connected node's icons on channel row for the channel's type (scalar, vector, matrix, or string). You can also edit node names and channel values by clicking the name or value and typing new information inline.



Original: (0 m,0 m,0 m)
Current: (-965 mm,375 mm,485 mm)

Workspace and Assembly Navigation



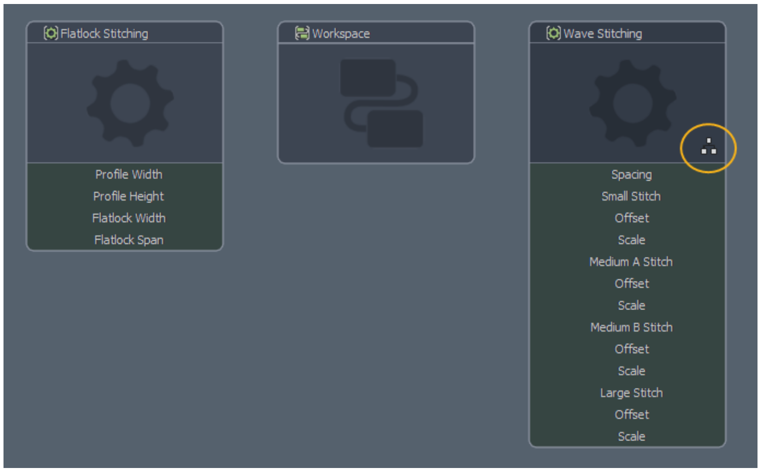
There are two view types in the **Schematic** viewport: workspaces and assemblies. A workspace is the main area where you create and work on node graphs. You can also experiment and examine existing scene setups. Only a single node graph can be visible at any time in a **Schematic** viewport. The visibility is determined by the view navigation buttons in the upper part of the viewport. You can create and view as many separate workspaces as desired and compartmentalize them in hierarchies as necessary for organizing scenes. You can extract any node graph or portion of a node graph created within a workspace into an individual assembly. You can view assemblies and navigate them in the same way as you do with workspaces.

You can open the Selector list (the wide button) to select from the currently available workspaces and assemblies. When opened, the list shows all the root level workspaces and assemblies defined in the scene.

Note: Child assemblies or workspaces are only visible within their own hierarchies.

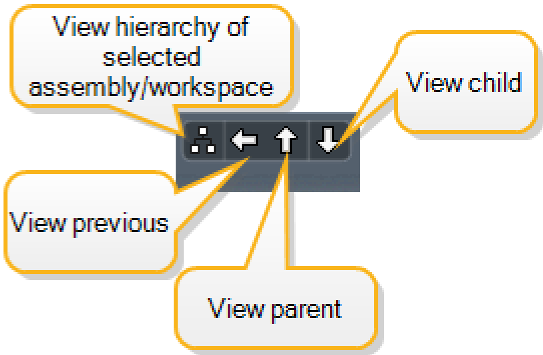
By selecting the desired workspace or assembly, you see the node graph as the active graph in the viewport. You can create new workspaces and assemblies and navigate and define hierarchies for multi-workspace and assembly setups by using the buttons next to the Selector list.

Clicking the button with the six small white boxes switches the view to Overview mode. This view shows all the workspaces and assemblies in the current scene. Here you can define hierarchies by dragging any workspace or assembly onto that of another to make it a child of the item it is dragged to. Assemblies and workspaces that have a hierarchy beneath them show a small hierarchy badge icon in the lower-right corner of their Overview node graphic. This icon indicates that it has child workspaces or assemblies.

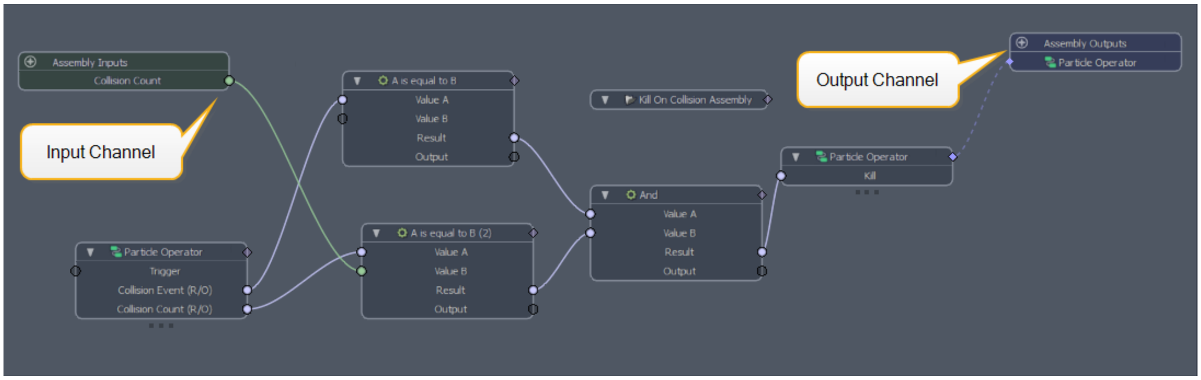

Flatlock Stitching
Profile Width
Profile Height
Flatlock Width
Flatlock Span
Workspace
Wave Stitching
Spacing
Small Stitch
Offset
Scale
Medium A Stitch
Offset
Scale
Medium B Stitch
Offset
Scale
Large Stitch
Offset
Scale

You can also navigate to any existing workspace or assembly from this view by double-clicking a workspace or assembly node to make that workspace (or assembly) the active workspace.

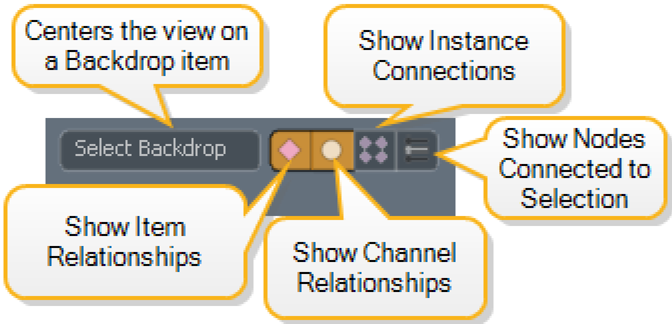
You can navigate hierarchies of workspaces or assemblies directly with the arrow buttons to the right of the Selector list. The first button displays a hierarchy layout of the current active workspace or assembly. With this view, you can see how workspaces and assemblies relate and can double-click to make any node active in the viewport. Clicking the **Left arrow** navigates back to the last view. Clicking the **Up arrow** shows the parent item and the **Down arrow** shows the child item. (You can also use keyboard shortcuts: **Alt**+**Up**/**Alt**+**Down** arrow keys and **Alt**+**Left**/**Alt**+**Right** arrow keys to navigate between sibling nodes in a hierarchy.)


View hierarchy of selected assembly/workspace
View child
View previous
View parent

You can use assemblies to package up parts of a graph into a single node and to simplify a node layout graph or as a way to create re-usable assets within a scene (or as a preset to place in other scenes). You can view the contents of any assembly (or sub-assembly) by double-clicking it within the workspace or the Overview view. Doing this shows the contents of the assembly, which is identified by the darker background and the input/output channels. The input/output channels are located on either side of the graph and are color coded: green for input and purple for output. Clicking the **+** (plus) icon in these areas creates a new user channel on the assembly as additional inputs or outputs to the assembly's contents.


Assembly Inputs
Collision Count
Input Channel
A is equal to B
Value A
Value B
Result
Output
Kill On Collision Assembly
Output Channel
Assembly Outputs
Particle Operator
Particle Operator
Kill
And
Value A
Value B
Result
Output
A is equal to B (2)
Value A
Value B
Result
Output
Particle Operator
Trigger
Collision Event (R/O)
Collision Count (R/O)

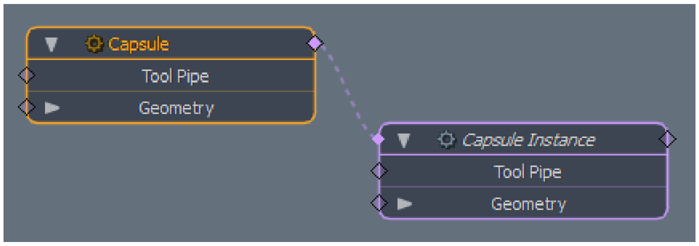
The buttons on the top right of the viewport allow you to manipulate the visibility of different connections.


Centers the view on a Backdrop item
Show Instance Connections
Select Backdrop
Show Nodes Connected to Selection
Show Item Relationships
Show Channel Relationships

Clicking **Select Backdrop** focuses in on the selected Backdrop node. For more information on Backdrops, see Backdrop Nodes.

By default, the Schematic viewport shows connections between both items and channels (the links to diamond and circle shaped connectors, respectively). See Defining and Editing Node Links for more information on connections. Clicking the Show Item Relationships button hides or displays connections between items, and clicking Show Channel Relationships hides or displays connections between channels.

The Show Instance Connections button displays connections between an instance and its source when the instance is selected. For more information on using instances, see Instances.


Capsule
Tool Pipe
Geometry
Capsule Instance
Tool Pipe
Geometry

In complex layouts it can be difficult to see a specific network. In such cases you can use the Show Nodes Connected to Selection button to hide everything that is not directly connected to a selected node.

Note: If no node is selected, Modo displays the whole node graph.

Instancing and Mirroring Assemblies

By instancing assemblies you create multiple copies of a setup (or rig) without needing to duplicate the entire contents of the assembly or without having to manually update all the copies if the original assembly changes. You identify which items are inputs and which are outputs in the original assembly. For example, a suspension rig might have a root item that's parented to the chassis as an input and the hub or wheel as an output. When you instance an assembly, Modo only instances the items marked as inputs or outputs. (Creating replicas for certain items is an option.). You can also mirror an assembly. (For example, you could create a suspension setup and have the other side of a car automatically mirrored. Modo propagates any changes to the original setup to the mirrored Instance.)

To instance an assembly, either right-click the assembly node within a workspace or within the Overview and choose **Instance**. This creates an instanced duplicate of the items in the **Items** list, but the node itself has a bracketed number reflecting the number of instances (such as MyAssembly [+3] for three instances). Double-clicking the assembly opens it and, along the bottom of the viewport, the instances show as numbered assembly nodes. To specify how to mirror the assembly, right-click the specific instance and choose **Mirror > how to mirror the assembly**. Modo mirrors the assembly as directed, such as which axis to use and whether to mirror over the Local (center position) axis or the World position (origin) axis.

Assembly Inputs and Outputs

Inputs are the items or channels that drive the inner workings of the assembly. Most often these are the items that you manipulate to control the rig and the items that you use to position and orient the rig within the scene (such as an assembly root item). An instanced assembly that has no defined inputs is a direct copy of the Source assembly, and it 'sits' directly on top of its source in the 3D views. Any motion on the Source assembly is mimicked in its instances. You use the inputs to establish independent behavior with the instanced assemblies. For example, by assigning the root item of an assembly as an input, you can have all the assembly's instances be positioned independently in the scene.

Another example would be assigning the IK goal of an IK chain as an input and, therefore, allowing all the instances to be controlled independently. You can assign items as an input by right-clicking the item node header and choosing **Assign Item as Input**. You can also assign individual channels as inputs for an assembly by right-clicking the channel and choosing **Expose Channel**. This creates a new user channel of the same type, adds it to the assembly Group Item, and assigns a channel link made to the selected channel. Also, by doing this, you can have assembly groups used as sub-assemblies (or black boxes), where internal processes are controlled by a few input/output channels. Alternately, channels can be exposed as inputs by dragging and dropping one or more channels directly over the plus sign icon of the Assembly.

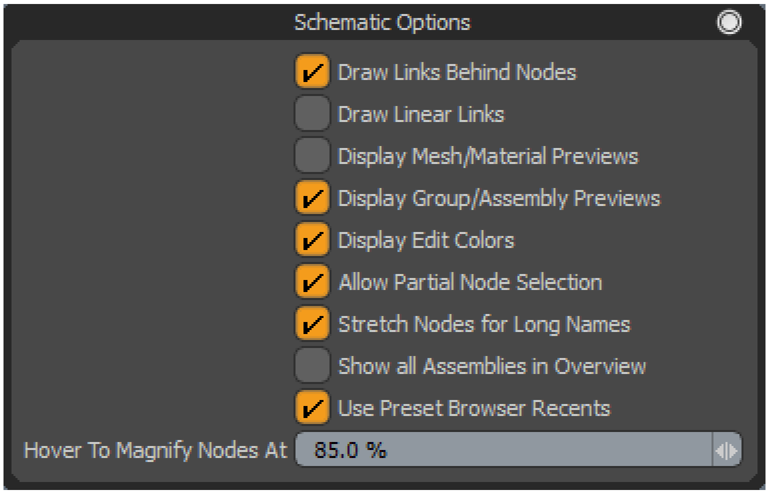
The outputs in an assembly are items that are driven (either directly or indirectly) by the assembly, and are instanced and visible in the 3D views. Mesh items default as outputs in an assembly; you can assign other items to outputs by right-clicking the item node header and choosing **Assign Item as Output**. Doing this adds an instance of that item to each of the assembly instances in the scene. You cannot manipulate output items in the instanced assemblies directly (for example, by using the Transform tool); however, output items inherit animation from their Source items.

An item cannot be both an input and output for an assembly. Items assigned as outputs cannot be manipulated in any instances of the assembly; however, if you add one or more of the transform items for the output as inputs to the assembly, then FK manipulation, channel linking, and so forth do work.

Tip: Assembly root items are special locators to help keep the items in an assembly organized and that make the assembly easier to manipulate in the acene (especially when working with instanced and mirrored assemblies). Parent any items within the assembly to this root element. You can create an assembly root item within a workspace or an assembly by right-clicking and choosing **Create Assembly Root Item** or by right-clicking the node header for an existing locator and choosing **Assign as Assembly Root Item**. Assembly root items are colored, so they are easy to identify within the **Schematic** view.

Viewport Options

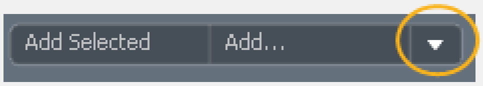
You can find the options that control what elements are displayed in the viewport by clicking the **Options** icon (the **gear** icon in the upper-right corner of the viewport). In the **Options** panel, all options are toggles. A checkmark indicates that the function is enabled, no checkmark indicates that the function is disabled.



Schematic Options

☑ Draw Links Behind Nodes
☐ Draw Linear Links
☐ Display Mesh/Material Previews
☑ Display Group/Assembly Previews
☑ Display Edit Colors
☑ Allow Partial Node Selection
☑ Stretch Nodes for Long Names
☐ Show all Assemblies in Overview
☑ Use Preset Browser Recents

Hover To Magnify Nodes At: 85.0 %

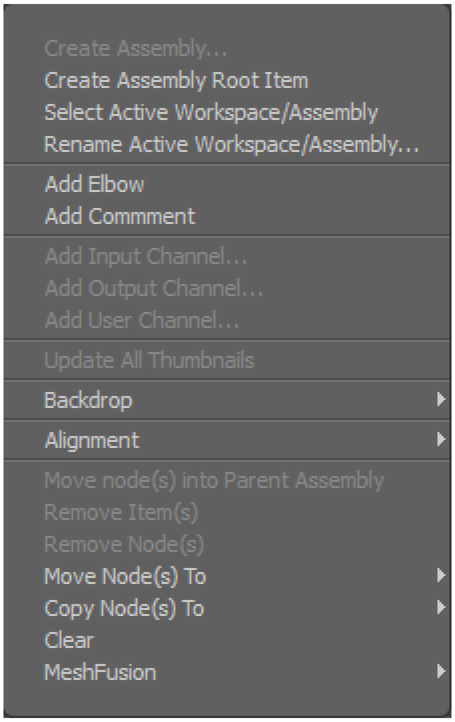
Draw Links Behind Nodes	When enabled, the resulting link lines are drawn behind the node panels. You can move the pointer directly over an individual connection line to temporarily make it visible in front of the node. When disabled, the lines always appear in front of the nodes.
Draw Linear Links	When enabled, links are drawn as straight lines rather than curves. This aids visibility in more complex scenes.
Display Mesh/Material Preview	When enabled, a quick preview icon displays in the header of the node panel for meshes and materials. When disabled, no icon displays.
Display Group Assembly Preview	When enabled, a quick preview icon displays in the header of the node panel for groups and assemblies. When disabled, no icon displays.
Display Edit Colors	When enabled, editing colors for items display as node panel outlines. When disabled, the outline colors do not display, but the **Items** list still shows colored item layers.
Allow Partial Node Selection	When enabled and doing a marquee selection of nodes in the **Schematic** view, the marquee need only touch the node to select it. When disabled, the marquee must completely surround the node to select it.
Stretch Nodes for Long Names	When enabled, the the node panels widen to accommodate long names. When disabled, node names are truncated to fit the standard node width. The long name is always visible in the tooltip.
Show all Assemblies in Overview	When enabled, all assemblies in the scene are visible in the Overview and not just the assemblies at the root level. When disabled, only the root level assemblies are visible in the Overview.
Use Preset Browser Recents	When enabled, the Add Items menu at the top of the Schematic viewport is replaced by a menu containing items recently added from the Preset Browser. 
Hover to Magnify Nodes At	Controls the zoom level of the auto-magnify function to make it easier to work on complex node graphs. When the viewport is scaled at or below **85%** (the default value), moving the pointer over the scaled node temporarily enlarges it to **100%**. This is useful when dragging connectors from one side of a graph to another. Zoom out and move the pointer over the Source node. When scaled, click the channel hub to initiate the

	link and then drag to the Target node. When the pointer is over the Target node, it too scales for you to see a full-sized view to connect the channels. (This also works with multiple links if you press the **Shift** key to select the channels.)

Shortcut Menus

The **Schematic** viewport has a variety of shortcut menus to use when you right-click a specific element. Depending on what element you click, you see a different menu with commands specific to that task.

Right-clicking an empty area of the workspace or assembly background opens this menu.


Create Assembly...
Create Assembly Root Item
Select Active Workspace/Assembly
Rename Active Workspace/Assembly...
Add Elbow
Add Comment
Add Input Channel...
Add Output Channel...
Add User Channel...
Update All Thumbnails
Backdrop
Alignment
Move node(s) into Parent Assembly
Remove Item(s)
Remove Node(s)
Move Node(s) To
Copy Node(s) To
Clear
MeshFusion

Create Assembly: Opens the Schematic Create Assembly dialog to package the selected node(s) into a single assembly node that simplifies the graph and makes the setting more portable for re-use elsewhere. You can double-click the resulting assembly to view and edit its contents.

Create Assembly Root Item: Creates a special locator item that you can assign as the parent in a hierarchy to the items in an assembly. It becomes the root of all the elements within. This makes the elements easier to manipulate in the 3D viewport — especially when working with instanced and mirrored assemblies. This must be assigned as an input item. Modo draws the assembly root item in the **Schematic** viewport with a green indicator and in the 3D viewport as a green box for easy identification.

Select Active Workspace/Assembly: Selects the active workspace or assembly in the **Groups** palette. Modo indicates all related elements to denote its selected state.

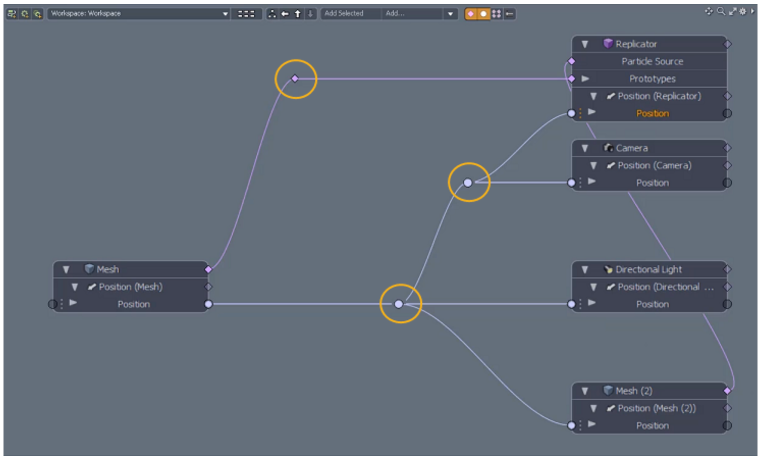
Rename Active Workspace/Assembly: Opens a dialog for you to rename the current active workspace or assembly. Once changed, the name is visible in the various menus and in the Overview view.

Add Elbow: Adds an elbow node to the scene. Elbow nodes can be placed on a link to reduce visual clutter in complex scenes. You can add elbows in a number of other ways:

- Right-click a link and then click **Add Elbow**.
- Double-click a link.
- Select the link and press **.** (period) on the keyboard.

In addition to changing the shape of a link, you can also use them to group several links together. Select the links to add, and press **.** (period) on the keyboard. You can drag additional links to an elbow.

To delete an elbow, select it and press **Delete** on the keyboard.


Workspace: Workspace
Add Selected
Add...
Replicator
Particle Source
Prototypes
Position (Replicator)
Position
Camera
Position (Camera)
Position
Mesh
Position (Mesh)
Position
Directional Light
Position (Directional ...
Position
Mesh (2)
Position (Mesh (2))
Position

Add Comment: Adds a Comment node that allows you to leave notes anywhere in the Schematic. You can also add a comment by pressing **Alt**+**n** on the keyboard. For more information, see Adding Comments.

Add Input/Output Channel: Opens the Add User Channel dialog and connects the resulting channel as an exposed input or output to the assembly.

Add User Channel: Opens the **Add User Channel** dialog for you to create custom channel types that can be connected for various rigging purposes. See the Add User Channel topic for more details.

Update All Thumbnails: Forces Modo to refresh all node icons and to update them to their current states if they have changed. You must have **Display Mesh/Material Preview** enabled as a viewport option to see icons on nodes.

Backdrop: Adds a backdrop node. Backdrop nodes help you to organize complex scenes by grouping several nodes together. For more information, see Backdrop Nodes.

Alignment: You can use various alignment options to arrange the nodes in your scene. In the viewport, select the nodes to align, right-click, and choose one of the alignment options. You can also use keyboard shortcuts. For more information, see Aligning Nodes.

Move Node(s) into Parent Assembly: Moves one or more nodes from the current assembly to its parent.

Remove Item(s): Removes the selected item(s) from the scene.

Remove Node(s): Removes the selected nodes(s) from the **Schematic** viewport.

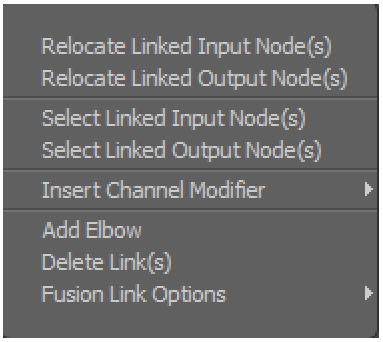
Move Node(s) To: Moves all selected elements to the specified location in the sub-menu. The sub-menu lists all the current workspaces and assemblies.

Copy Node(s) To: Copies the selected node(s) to the specified location in the sub-menu. The sub-menu lists all the current workspaces and assemblies. The copied nodes still reference the same initial Source item and changes made in the copy are reflected in the original node because both nodes reference the channels from the same source item. (The item itself is not being cloned, only its representation.)

Clear: Removes all nodes from the workspace or assembly. This command essentially resets the viewport to its initial, empty state.

MeshFusion: Options for manipulation MeshFusion nodes.

Right-clicking a link connection line opens this menu.


Relocate Linked Input Node(s)
Relocate Linked Output Node(s)
Select Linked Input Node(s)
Select Linked Output Node(s)
Insert Channel Modifier
Add Elbow
Delete Link(s)
Fusion Link Options

Relocate Linked Input/Output Nodes(s): Autopositions the node within the **Schematic** viewport for the incoming or outgoing link(s).

Select Linked Input/Output Node(s): Automatically selects the nodes of the incoming or outgoing links.

Insert Channel Modifier: Adds the specified channel modifer to the schematic graph. The submenu lists all the Channel Modifiers that can be added.

Add Elbow: Adds an elbow node to the scene. Elbow nodes can be placed on a link to reduce visual clutter in complex scenes. You can add elbows in a number of other ways:

- Right-click in the Schematic viewport, then click **Add Elbow**. This adds an elbow that's not part of any link.
- Double-click a link.
- Select the link and press **.** (period) on the keyboard.

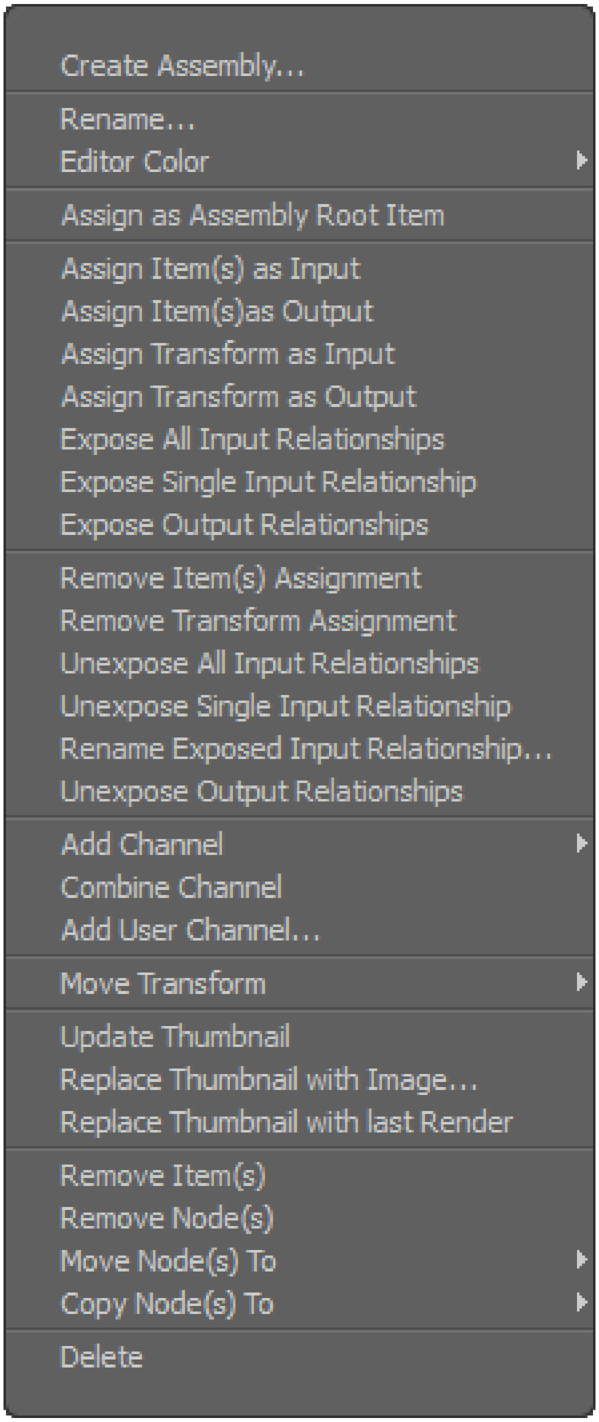
In addition to changing the shape of a link, you can also use them to group several links together. Select the links to add, and press **.** (period) on the keyboard. You can drag additional links to an elbow.

To delete an elbow, select it and press **Delete** on the keyboard.

Delete Link(s): Removes the connector line between nodes and eliminates the link.

Fusion Link Options: Options to manipulate MeshFusion links.

Right-clicking a node header (title) opens this menu.


Create Assembly...
Rename...
Editor Color
Assign as Assembly Root Item
Assign Item(s) as Input
Assign Item(s)as Output
Assign Transform as Input
Assign Transform as Output
Expose All Input Relationships
Expose Single Input Relationship
Expose Output Relationships
Remove Item(s) Assignment
Remove Transform Assignment
Unexpose All Input Relationships
Unexpose Single Input Relationship
Rename Exposed Input Relationship...
Unexpose Output Relationships
Add Channel
Combine Channel
Add User Channel...
Move Transform
Update Thumbnail
Replace Thumbnail with Image...
Replace Thumbnail with last Render
Remove Item(s)
Remove Node(s)
Move Node(s) To
Copy Node(s) To
Delete

Create Assembly: Opens the Schematic Create Assembly dialog to package the selected node(s) into a single assembly node that simplifies the graph and makes the setting more portable for re-use elsewhere. You can double-click the resulting assembly to view and edit its contents.

Rename: Makes the name of the Target assembly editable inline.

Editor Color: Defines a layer color for identifying and organizing items. This shows in the **Schematic** viewport as a colored outline around the node panel.

Assign as Assembly Root Item: Converts an existing locator item into an assembly root item locator. This is a special kind of locator that you can assign as the parent in a hierarchy to the items in an assembly. Therefore, it becomes the root of all the elements within. This makes the elements easier to manipulate in the 3D viewport — especially when working with instanced and mirrored assemblies. This must be assigned as an input item. Modo draws the assembly root item in the **Schematic** viewport with a green indicator and in the 3D viewport as a green box for easy identification.

Assign Item(s) as Input: Assigns one or more of the items within an assembly as an input for you to control which of the items within the assembly are instanced with the assembly (appear in the **Items** list as instances). You can independently transform items designated as inputs as instances (relative to their root). If you defined the assembly root item as the input, by using this command you can position the instanced assembly independently of its source. If you define additional items as inputs, they appear in the **Items** list, and you can select them and transform them independently as well (relative to their root).

Assign Item(s) as Output: Assigns one or more of the items within an assembly as an output for you to control which of the items within the assembly are instanced with the assembly (appear in the **Items** list as instances). Mesh items default to being outputs. Items designated as outputs cannot be independently transformed in the resulting instances, but are driven by the Source item and duplicate its transforms. If parented to a hierarchy under an assembly root item, the instanced item transforms relative to the root's position (if the root is defined as an input).

Assign Transform as Input/Output: Allows selected transform channels (such as position, rotate, and scale) to be designated as inputs or outputs independently of the item's designation.

Expose Input/Output Relationship: Exposes a relationship connector on a node within the assembly so it can drive or be driven from outside the assembly.

Remove Item Assignments: Removes any input or output assignment applied to an item.

Remove Transform Assignments: Removes any transform input or output assignment applied to an item's transform channels.

Unexpose Input/Output Relationships: Removes an exposed relationship and prevents the channel from driving or being driven from outside the assembly.

Add Channel: Opens a menu with all the un-added channels for the associated item for you to add additional channels to the node without having to drag them or select them in the **Channels** viewport.

Combine Channel: Moves all selected channels for the selected item from their original nodes into the recently selected node.

Add User Channel: Opens the **Add User Channel** dialog for you to create custom channel types that can be connected for various purposes. See the **Add User Channel** topic for more details.

Move Transform: Changes the order of the selected Target transform channel (for items that have related transforms) as indicated in the the submenu.

Update Thumbnail: Forces Modo to refresh the thumbnail image. You must have the **Display Mesh/Material Preview** viewport option enabled to see icons on nodes.

Replace Thumbnail with Image: Opens an OS-specific dialog for you to select a saved image to replace the current icon representing the node. If the selected image is larger than 128x128, Modo resamples it to that size.

Replace Thumbnail with last Render: Replaces the current icon image with the last render in the Render list of the **Render Display** viewport. Modo must have created the render during the current session. If the rendered image is larger than 128x128, Modo resamples it to that size.

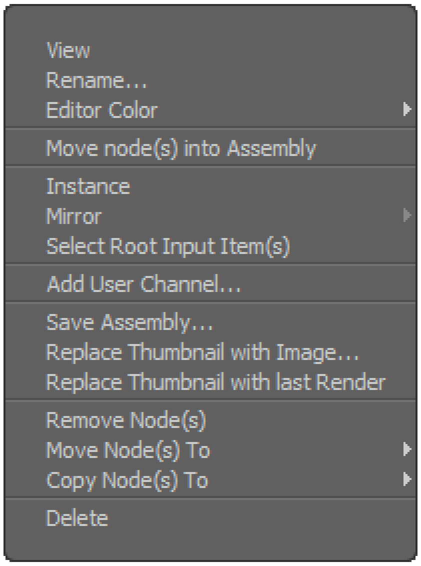
Remove Node(s): Removes the selected nodes(s) from the **Schematic** viewport.

Move Node(s) To: Moves all the selected nodes to the specified workspace or assembly. The submenu lists all the current workspaces and assemblies.

Copy Node(s) To: Copies all the selected elements to the specified location. The submenu lists all the current workspaces and assemblies.

Delete: Removes the node and its associated item from the scene.

Right-clicking a node header (title) for an assembly opens this menu.


View
Rename...
Editor Color
Move node(s) into Assembly
Instance
Mirror
Select Root Input Item(s)
Add User Channel...
Save Assembly...
Replace Thumbnail with Image...
Replace Thumbnail with last Render
Remove Node(s)
Move Node(s) To
Copy Node(s) To
Delete

View: Switches to view any selected assembly.

Rename: Makes the name of the Target assembly editable inline.

Editor Color: Defines a layer color for identifying and organizing items that display in the **Schematic** viewport as a colored outline around the node panel.

Move node(s) into Assembly: Move the selected node(s) into the selected assembly.

Instance: Creates a virtual copy of an assembly (which references its master assembly) that can be used elsewhere in a schematic graph, workspace, or assembly. You can only apply this command to an assembly node from the workspace or in the Overview view.

Mirror: Inverts an instanced assembly across the user-defined axis specified in the submenu. This is useful for making multiple identical items that mirror (such as the wheels of a car or the IK leg setup of a spider). The submenu lists the Local axis (the root item's center point) or the World axis (World Origin). If there is an assembly root item in the assembly, then Modo uses its location and orientation as the mirror axis.
To mirror an assembly, you must instance it first. Then, by opening the root assembly (double-click the assembly in the Overview view), the individual instances appear in a new section toward the bottom of the viewport. By right-clicking the Target instance, you can then choose **Mirror**. Once you assign mirroring, you can position the resulting mirrored assembly by using the exposed channels or by positioning the root locator.

Select Root Input Item(s): Selects all locator items in the assembly that are assigned as inputs and that have no parent present in the assembly.

Add User Channel: Opens the **Add User Channel** dialog for you to create custom channel types that can be connected for various purposes. See the Add User Channel topic for more information.

Save Assembly: Opens an OS-specific dialog for you to save the assembly as a preset. If saved to the Presets directly, you can then later drag the resulting assembly preset from the Preset Browser directly into a new scene.

Replace Thumbnail with Image: Opens an OS-specific dialog for you to select a saved image to replace the current icon representing the node. If the selected image is larger than 128x128, Modo resamples it to that size.

Replace Thumbnail with last Render: Replaces the current icon image with the last render in the Render list of the **Render Display** viewport. Modo must have created the render during the current session. If the rendered image is larger than 128x128, Modo resamples it to that size.

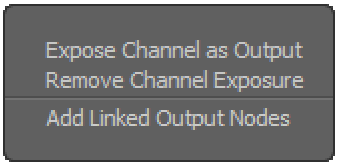
Remove Node(s): Removes the selected nodes(s) from the **Schematic** viewport.

Move Node(s) To: Moves all the selected nodes to the specified workspace or assembly. The submenu lists all the current workspaces and assemblies.

Copy Node(s) To: Copies all the selected elements to the specified location. The submenu lists all the current workspaces and assemblies.

Delete: Removes the assembly from the scene and all items contained in it.

Right-clicking a node connector dot opens this menu (for both input and outputs).

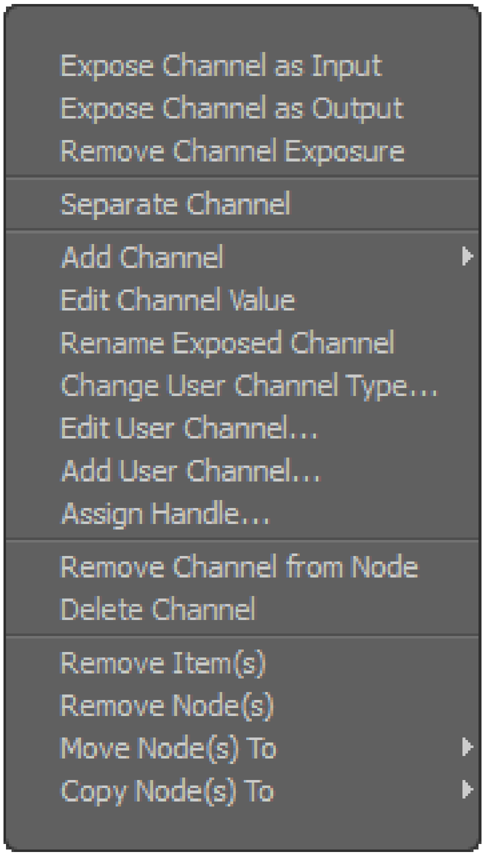

Expose Channel as Output
Remove Channel Exposure
Add Linked Output Nodes

Expose Channel as Input/Output: Makes the selected channel visible on the assembly node so you can link other external nodes to it.

Remove Channel Exposure: Hides exposed channels so they are no longer visible on the assembly node.

Add Linked Output Nodes: Finds any nodes that are linked to the selected channels or relationships and brings them into the current workspace or assembly.

Right-clicking any channel inside a node panel opens this menu.


Expose Channel as Input
Expose Channel as Output
Remove Channel Exposure
Separate Channel
Add Channel
Edit Channel Value
Rename Exposed Channel
Change User Channel Type...
Edit User Channel...
Add User Channel...
Assign Handle...
Remove Channel from Node
Delete Channel
Remove Item(s)
Remove Node(s)
Move Node(s) To
Copy Node(s) To

Expose Channel as Input/Output: Makes the selected channel visible on the assembly node so you can link it to other external nodes.

Remove Channel Exposure: Hides exposed channels so they are no longer visible on the assembly node.

Separate Channel: Separates the channels out from a node to their own node panel that can be positioned independently for easier graph construction. Modo draws three small dots under both node panels to indicate the separated state.

Add Channel: Opens a menu that allows you to select from any of the items available channels and add them to the node directly when selected.

Edit Channel Value: Makes the channel value of editable inline.

Rename Exposed Channel: Makes the user channel name editable inline.

Change User Channel Types: Opens the **Change Channel Type** dialog to change the type of a user channel (for example, changing a **Scalar** channel to a **Boolean** channel).

Edit User Channel: Opens a form for editing the properties of a user channel's definition.

Add User Channel: Opens the **Add User Channel** dialog for creating a custom channel type that you can connect for various purposes. See the Add User Channel topic for more information.

Assign Handle: Works in conjunction with the Channel Handles function to assign the selected channel to the appropriately created handle. See the Channel Handles page for more information.

Remove Channels from Node: Removes the channels from the channel row of the node panel, but the channels are still active within the **Channels** viewport of the related Item.

Delete Channel: Removes the user channels for the selected item.

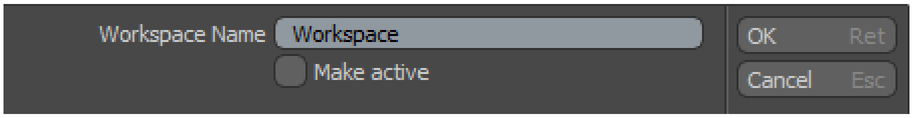
Remove Item(s): Removes the selected item(s) from the scene.

Remove Node(s): Removes the selected nodes(s) from the **Schematic** viewport.

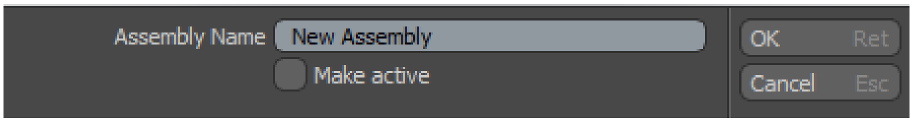
Move Node(s) To: Moves all the selected nodes to the specified workspace or assembly. The submenu lists all the current workspaces and assemblies.

Copy Node(s) To: Copies all the selected nodes to the specified workspace or assembly. The submenu lists all the current workspaces and assemblies.

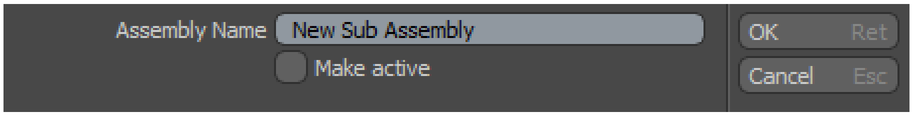
Create New Workspace: Clicking this button opens the **New Schematic Workspace** dialog where you can create and name the new empty workspace. You can also define whether you want it to become the currently active workspace. In a new scene, if there is nothing in the active workspace when you create a new workspace (an empty workspace), the newly created workspace becomes the initial workspace. All other new workspaces thereafter are in addition to the initial workspace.


Workspace Name: Workspace
☐ Make active

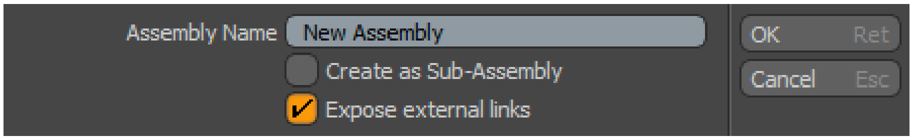
Create New Assembly: Clicking this button opens the **New Schematic Assembly** dialog where you can create and name a new empty assembly and specify it to be the currently active assembly for editing.


Assembly Name: New Assembly
☐ Make active

Create New Sub-Assembly: Clicking this button opens the **New Schematic Sub-Assembly** dialog where you can create and name a new empty sub-assembly (a child in the hierarchy of the currently active assembly). You can define whether you want it to be the currently active assembly for editing.

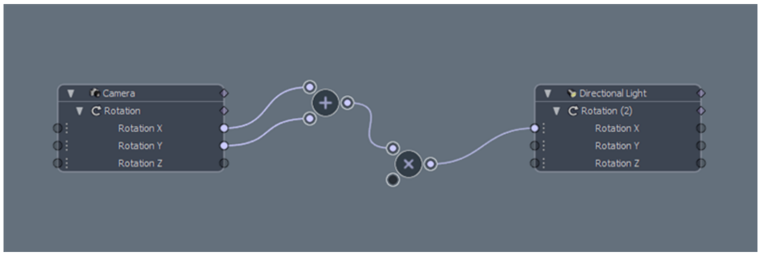

Assembly Name: New Sub Assembly
☐ Make active

Creating Assemblies from a Graph

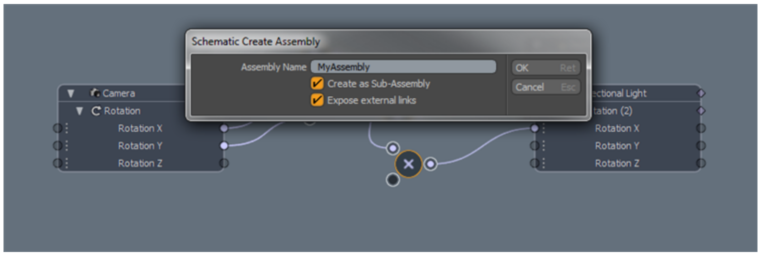

Assembly Name: New Assembly
☐ Create as Sub-Assembly
☑ Expose external links

Once you have a working graph or even a portion of a graph set up, you can create an assembly from it. Assemblies are useful to simplify very complex graph layouts by compartmentalizing multiple nodes into a single node. You can also create assemblies to separate parts of a graph for organization to be able to work on each area independently. You can also treat assemblies as assets to save and re-use later. To create an assembly, in the graph select each of the nodes that you want to include. Then right-click a blank area of the background and choose **Create Assembly**. In the **Schematic Create Assembly** dialog, provide a name. You can create an assembly as a normal assembly or as a sub-assembly. To create a sub-assembly, select the **Create as Sub-Assembly** checkbox. (The new sub-assembly becomes part of the current graph; otherwise, Modo creates a standalone root level assembly.) In both cases, Modo removes the selected nodes from the graph and adds them to the new assembly. To create channels on the new group for each channel link to or from a channel within the group, select the **Expose External Channels** checkbox. (It is only applicable to sub-groups and is useful for creating assemblies that can be re-used elsewhere — a black box that you can connect without knowing about the internal processes.) Click **OK** to create the new assembly. If it was a standalone assembly, the view changes into the Assembly view to show its contents.

Creating Sub-Assembly Walk-through


Camera
Rotation
Rotation X
Rotation Y
Rotation Z
Directional Light
Rotation (2)
Rotation X
Rotation Y
Rotation Z

Select the elements that make up the sub-assembly. This shows the + (add) and **x** (multiply) nodes.



Schematic Create Assembly

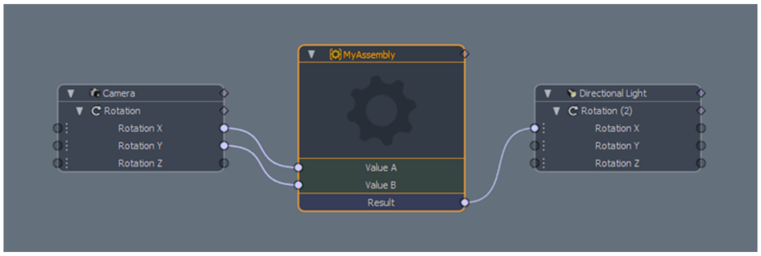
Assembly Name: MyAssembly

☑ Create as Sub-Assembly

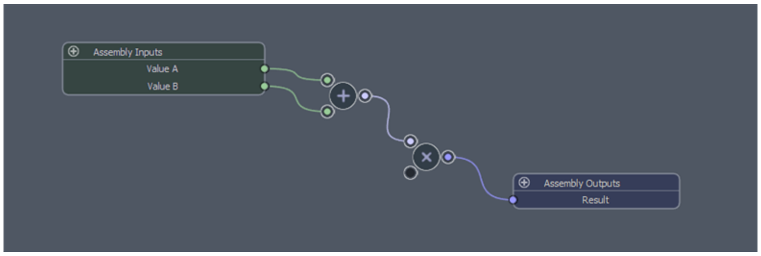
☑ Expose external links

OK Ret
Cancel Esc
Camera
Rotation
Rotation X
Rotation Y
Rotation Z
ectional Light
tation (2)
Rotation X
Rotation Y
Rotation Z

Right-click an open area of the workspace and choose **Create Assembly**.


MyAssembly
Value A
Value B
Result
Camera
Rotation
Rotation X
Rotation Y
Rotation Z
Directional Light
Rotation (2)
Rotation X
Rotation Y
Rotation Z

Clicking **OK** changes the selected elements into the sub-assembly.


Assembly Inputs
Value A
Value B
Assembly Outputs
Result

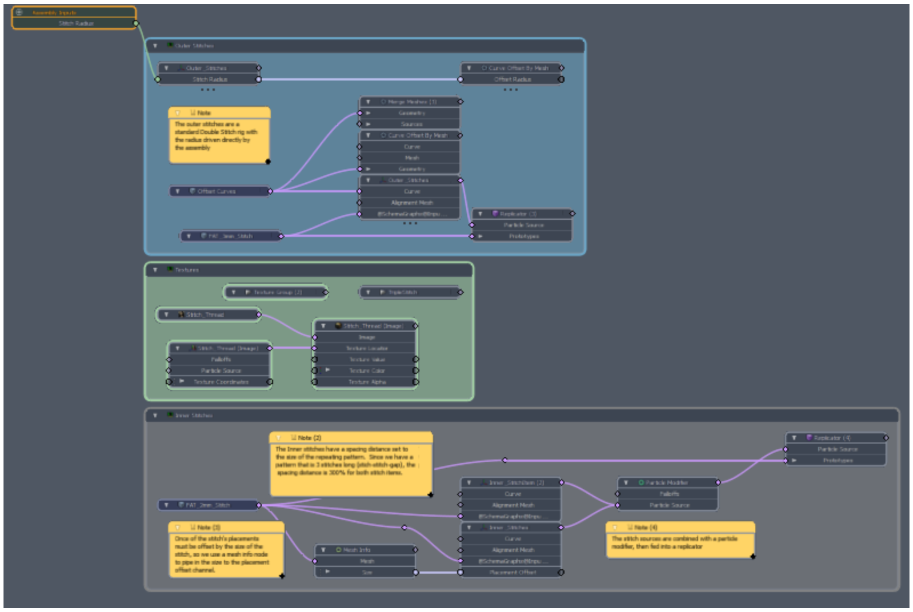
You can view the contents of any sub-assembly by double-clicking the named assembly node.

Note: Within the scene's file structure, Modo stores the actual data for schematic graphs in a special group type in the **Groups** viewport. Within that viewport you can see the options for assemblies. These are the same groups that show in the Assembly mode of the **Schematic** viewport. You can create new assemblies by using the **New Group** button. Using this button creates a new blank display in the schematic. The schematic always shows the most recently selected assembly. (You cannot create workspaces, although viewable, within the **Groups** viewport.).

Note: The concept of using the **Schematic** viewport is easy enough to understand for creating links from one node to another, but the concept of rigging and animation control building overall is very complex.

Backdrop Nodes

Backdrop nodes help you to organize complex Schematic networks by grouping several nodes together.



To add a Backdrop, select your nodes, right-click in the viewport and click **Backdrop** > **Create Backdrop**. Alternatively, select your nodes and press **B** on the keyboard.

To add additional nodes to the group, select the node and the Backdrop, right-click in the viewport and click **Backdrop** > **Add Item(s) to Backdrop**. Alternatively, select the node, hold **Shift** and drag it into the Backdrop.

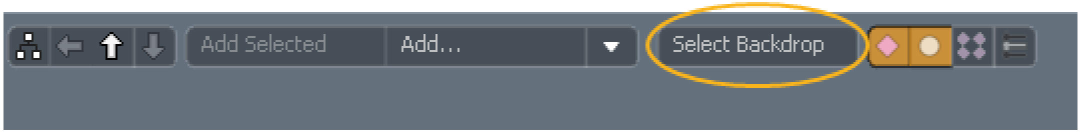
To remove a node from the Backdrop, do either of the following, select the node, right-click in the viewport and click **Backdrop** > **Remove Item(s) from the Backdrop**, or select the node, hold **Shift** and drag it out of the Backdrop.

Tip: You can also create Backdrops within Backdrops. This can be useful for example in procedural modeling, when you want to isolate a set of mesh operations.

To move a Backdrop, simply click on its header and drag it to the required position in the viewport.

Backdrops can be colored too, to make them easier to see. To assign color to a Backdrop node, right-click the node and in the context menu under **Editor Color**, click the color you need.

To focus the view on a single Backdrop node, click **Select Backdrop** at the top of the viewport and select the Backdrop from the list.



Aligning Nodes

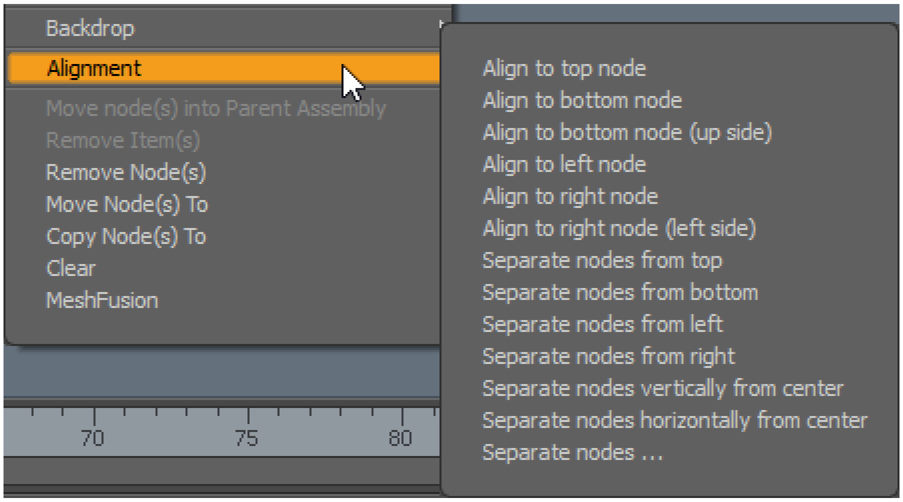
With complex scenes, you may need to arrange your nodes more clearly. Modo's alignment and separation commands can help you to create a better-organized scene.

In the Schematic, select the nodes to align, then use the required keyboard shortcut:

Right-click in the viewport, and in the context menu, select **Alignment**, then click the option you need.

Align Commands	
Align to top node	Up Arrow
Align to bottom node	Down Arrow
Align to bottom node (up side)	Shift+Down Arrow

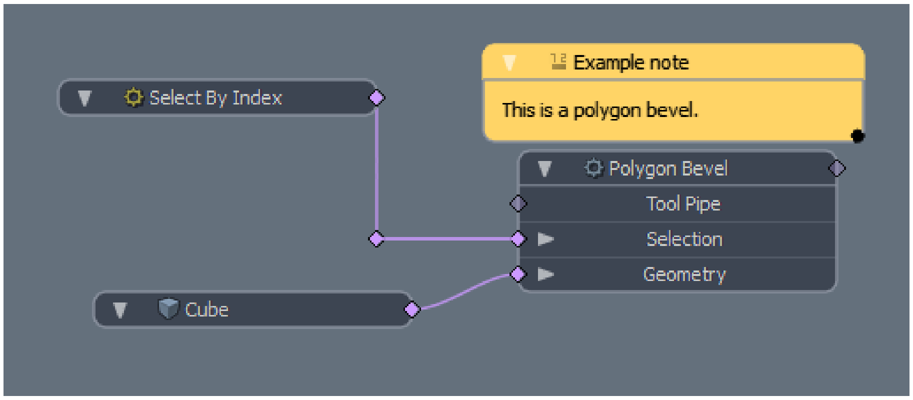
Align to left node	Left Arrow
Align to right node	Right Arrow
Align to right node (left side)	Shift+Right Arrow
Separate Commands	
Separate nodes from top	Ctrl/Cmd+Alt+Down Arrow
Separate nodes from bottom	Ctrl/Cmd+Alt+Up Arrow
Separate nodes from left	Ctrl/Cmd+Alt+Right Arrow
Separate nodes from right	Ctrl/Cmd+Alt+Left Arrow
Separate nodes vertically from center	Ctrl/Cmd+Alt+Shift+Up/Down Arrow
Separate nodes horizontally from center	Ctrl/Cmd+Alt+Shift+Left/Right Arrow


Backdrop
Alignment
Move node(s) into Parent Assembly
Remove Item(s)
Remove Node(s)
Move Node(s) To
Copy Node(s) To
Clear
MeshFusion
70
75
80
Align to top node
Align to bottom node
Align to bottom node (up side)
Align to left node
Align to right node
Align to right node (left side)
Separate nodes from top
Separate nodes from bottom
Separate nodes from left
Separate nodes from right
Separate nodes vertically from center
Separate nodes horizontally from center
Separate nodes ...

Adding Comments

Note: The keyboard shortcut for adding a comment node has been changed to **Alt**+**N**.

You can add comments in the Schematic using the Comment node.

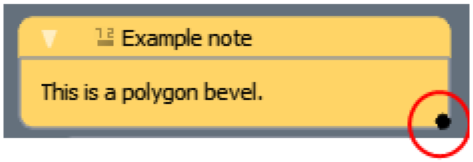

Select By Index
Example note
This is a polygon bevel.
Polygon Bevel
Tool Pipe
Selection
Geometry
Cube

You can add a comment in the following ways:

- Right-click in the Schematic viewport and click **Add Comment**.
- Press **Alt**+**n** on the keyboard.

To rename your Comment node, click the title. The field becomes editable.

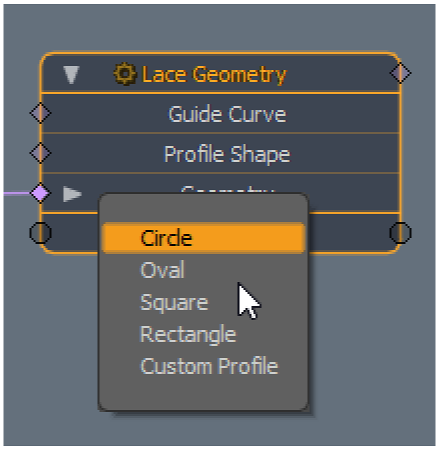
To resize the Comment node to accommodate your text, click the black circle in lower right corner and drag the box to the required size.


Example note
This is a polygon bevel.

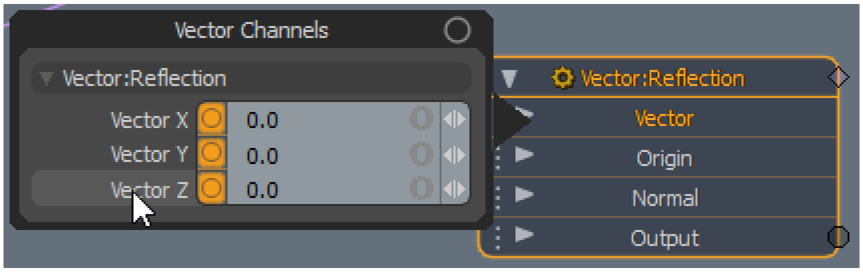
Editing Channel Values on a Node

You can edit a channel's value on a node by clicking on it in the Schematic. Depending on the type of channel, you have different ways of editing the channel:

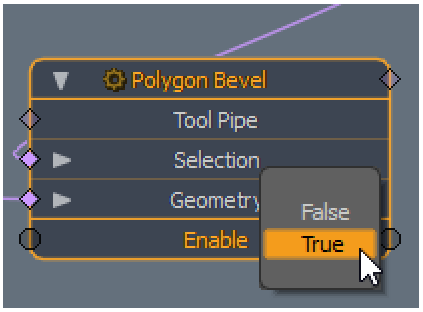
- When different text options are available, you can select the option you need from a dropdown.


Lace Geometry
Guide Curve
Profile Shape
Circle
Oval
Square
Rectangle
Custom Profile

- For colors, this opens Modo's default Color Picker.
- For vectors, this opens the **Vector Channels** form.


Vector Channels
Vector:Reflection
Vector X
0.0
Vector Y
0.0
Vector Z
0.0
Vector:Reflection
Vector
Origin
Normal
Output

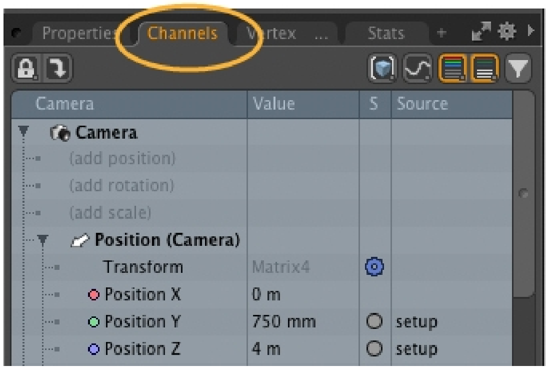
- For gradients, this opens the Gradient Editor.
- For Boolean toggles, such as the **Enable** channel on a Mesh Operation, use a simple True/False popup.


Polygon Bevel
Tool Pipe
Selection
Geometry
Enable
False
True

- For other channel types, you can use the inline editor to type in a value.

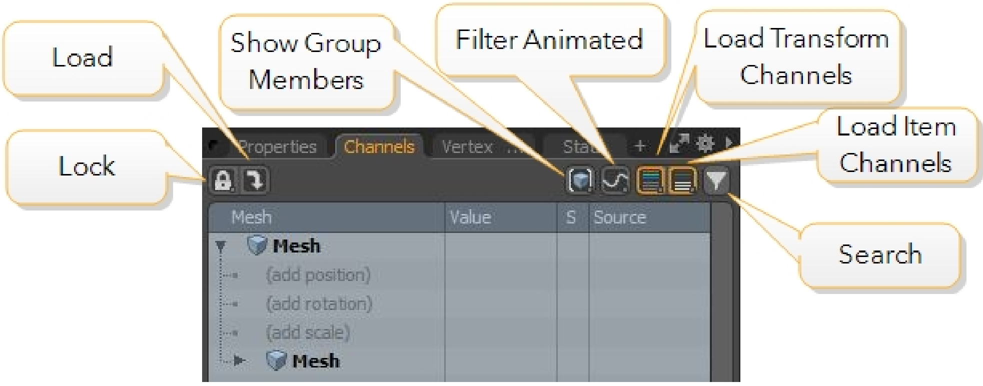
Channels Viewport

The **Channels** viewport provides a direct method for selecting channels as well as creating, navigating, and managing any keyframes on channels. Channels are any animateable attributes of an item (identical to the attributes in the Properties Panel). You can view all channels of any item in the **Channels** viewport.



Camera	Value	S	Source
Camera			
(add position)			
(add rotation)			
(add scale)			
Position (Camera)			
Transform	Matrix4	◎	
Position X	0 m		
Position Y	750 mm	○	setup
Position Z	4 m	○	setup

At the top of the viewport, you can find a few controls.



Mesh	Value	S	Source
Mesh			
(add position)			
(add rotation)			
(add scale)			
Mesh			

From left to right, these controls are:

- **Lock** - Locks the contents of the Channels viewport allowing item selection to be changed without affecting the contents of the viewport.
- **Load** - When the viewport contents are locked this button will add any selected items to the viewport.
- **Show Group Members** - When a Group is selected this button will load the Group member items and/or channels into the viewport.
- **Filter Animated** - When active only channels that are animated will be loaded into the viewport.
- **Load Transform Channels** - Allows you to add or remove an item's transform channels (position, rotation, and scale).
- **Load Item Channels** - Adds or removes an item's non-transform channels.
- You can search the list of channels by clicking the **Search** button in the top-right corner of the viewport. This opens a **Search** field, allowing you to enter a search string. Clicking the arrow on the right of the field allows you to do a **Simple Search** or search for a pattern using **Pattern Matching**. Click **Match Case** to do a case-sensitive search. Click **Flat** on the left to only display the list of channels without the hierarchy.

The **Channels** viewport has four columns: **Channel Name**, **Value**, Status (**S**), and **Source**. The first column displays the channel names for the currently selected item, and the second column displays the value of the channel at the current scene time. The Status (**S**) column shows the state of the channel (using the same dots or icons used with the channel keyframing control): animated, static, or on a keyframe. The last column displays the internal **Source** location of the current value. These locations are important when you define Render Passes and when using the skeleton tools to define rigs for animation.

Tip: To improve your workflow while working with User Channels, use the **Channel Haul** tool to toggle the visibility of all channels available for a selected item. For more information, see Displaying All Available User Channels in the 3D Viewport

Interacting with Channels

Similar to the **Properties** viewport, the **Channels** viewport updates to reflect the currently selected item — whether items are within the **Item** list, the Shader Tree, or even the **Groups** palette. Once you select an item, all the animateable attributes appear in the **Channels** viewport. To change an attribute's value, you click the current value in the **Value** column. For boolean values, clicking changes the value from **True** to **False** or vice versa. For attributes with properties, a list opens to show the possible options. For numerical values, you can click, pause, and click again to edit the value inline. Press **Enter** to set the value or just click elsewere than on the value.

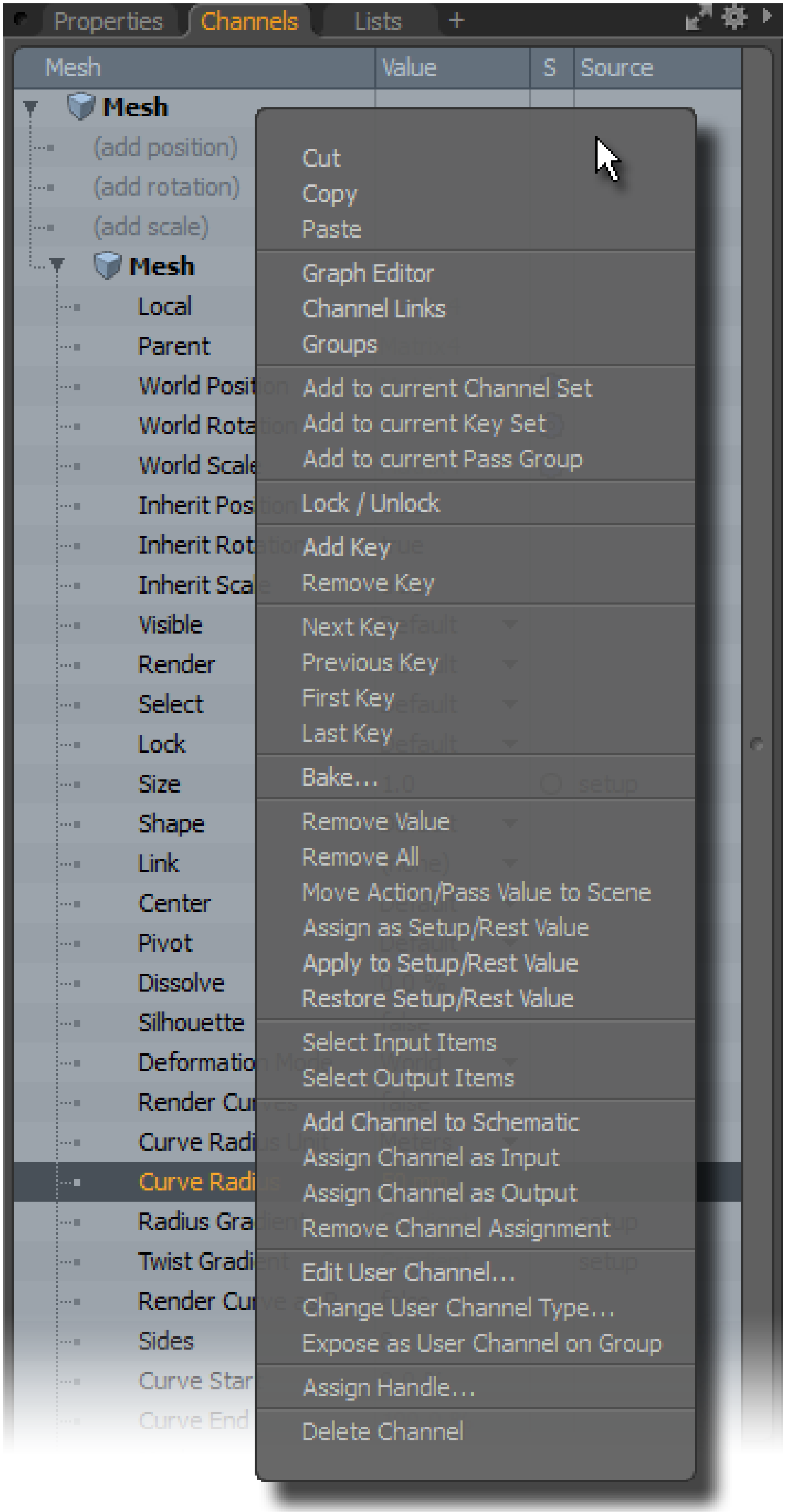
Setting Keyframes

You can set animation keyframes by using the keyframe buttons, pressing the **S** key, setting in the **Properties** viewport, or setting within the Graph Editor. To set a keyframe for a value directly in the **Channels** viewport, right-click the target value and choose **Add Key**. You may also remove keyframes this way. When any attribute has a keyframe, colored circles appear next to the value in the **S** (Status) column that indicate the attribute's current state. The possible states are:
○ **None**- The channel is at the default value.
● **Constant**- The channel has a constant value.
🟢 **Animated**- The channel has no keyframe at the current time position.
🔴 **Animated**- The channel has a keyframe at the current time position.
🔵 **Mixed**- The channel has differing values or states.
🟣 **Driven**- The channel is driven by an external source.
🟡 **Undefined**- The channel is not defined.
An arrow over the dot represents that channel as driving another with a channel link.
➲ **Driver**- The channel is linked to another.
A gear icon represents a channel that is being controlled by another with a channel link.
⚙ **Driven**- The channel is linked to another.

You can automatically select input or output items by selecting the appropriate channel and pressing **i** to select the input, and **o** to select the output.

Right-click Menu

Right-clicking in the **Channels** viewport opens a menu with commands for modifying the keyframes of the various channels.


Properties
Channels
Lists
+
Mesh
Value
S
Source
Mesh
(add position)
(add rotation)
(add scale)
Mesh
Local
Parent
World Posit
World Rota
World Scale
Inherit Pos
Inherit Rot
Inherit Sca
Visible
Render
Select
Lock
Size
Shape
Link
Center
Pivot
Dissolve
Silhouette
Deformatio
Render Cu
Curve Radi
Curve Radi
Radius Gra
Twist Gradi
Render Cu
Sides
Curve Star
Curve End
Cut
Copy
Paste
Graph Editor
Channel Links
Groups
Add to current Channel Set
Add to current Key Set
Add to current Pass Group
Lock / Unlock
Add Key
Remove Key
Next Key
Previous Key
First Key
Last Key
Bake...
Remove Value
Remove All
Move Action/Pass Value to Scene
Assign as Setup/Rest Value
Apply to Setup/Rest Value
Restore Setup/Rest Value
Select Input Items
Select Output Items
Add Channel to Schematic
Assign Channel as Input
Assign Channel as Output
Remove Channel Assignment
Edit User Channel...
Change User Channel Type...
Expose as User Channel on Group
Assign Handle...
Delete Channel

Cut: Cuts the channel values and sets them back to their defaults.

Copy: Copies the channel values.

Note: Only channel values are copied, not the properties of the channel itself, such as its lock state or any minimum/maximum range settings.

Paste: Pastes channel values. When values are pasted, the entire target channel is replaced by the source values.

Note: The above operations require the source and target channels to be compatible. Such channels are found by matching the types of the source channels to the types of the destination channels. For a channel to match it must be of the same numerical type, for example, integer or floating point, and also the same data type, such as distance, angle, or percentage. If the channel has integer hints, such as the camera **Projection Type** channel, then the hints must also match.

Graph Editor: Opens the Graph Editor, which represents the interpolations of keyframes (in-betweens).

Channel Links: Opens the Channels Links palette, which is useful in having one channel control another automatically. In current versions of Modo, use the Schematic View instead.

Groups: Opens the Groups viewport as a pop-up palette.

Add to current Channel Set: This command adds the selected channels to the currently active (or selected in the Groups viewport) Channel Set.

Add to Current Key Set: This command adds the selected channels to the currently active (or selected in the Groups viewport) Key Set.

Add to Current Pass Group: This command adds the selected channels to the currently active (or selected in the Groups viewport) Pass Group.

Lock/Unlock: This command toggles the edit state of the selected channel(s), as either locked, (uneditable) or unlocked (editable). Locked values are grayed out.

Add Key: Sets a keyframe at the current time with the current value.

Remove Key: Removes the keyframe at the current time.

Next Key/Previous Key/First Key/Last Key: Moves the Timeline to the appropriate time.
Bake: Bakes (or freezes) the current animation to individual keys at every frame. This is useful when exporting animations of driven items.

Remove Value: Removes all keyframes and retains the value of the initial key.
Remove All: Removes all keyframes and returns the default value.

Move Action/Pass Value to Scene: Takes the current channel value and moves it to the **Scene** level. This is the default base level value used when working with Render Passes.

Assign as Setup/Rest Value: Tells Modo that the values for this channel should be kept separate between **Setup** and **Scene** actions. (Only used in **Setup** mode.) When you edit a channel out of **Setup** mode, it won't affect the **Setup** value. (The channel has a value in **Setup** and a different one in the **Scene** action.) For example, you can use this to hide certain rig elements when animating but have them visible in **Setup** mode without having to toggle visibility all the time.

Apply to Setup/Rest Value: Takes the current channel value and sets it as the default **Setup** value. This is the resting value of an item prior to being deformed.

Restore to Setup/Rest Value: Removes the existing value for the channel and reverts it to its **Setup** state.

Select Input Items: Selects the driver item or channel for driven channels.

Select Output Items: Selects any items or channels being driven by the current channel for driven channels.

Add Channel to Schematic: Inserts the selected channel(s) as a schematic node into the Schematic viewport.

Assign Channel as Input/Output: You can define various User Channels for Assembly nodes, which adds them to the node's channels list. The resulting channels, once created, can be selected and defined as either inputs or outputs of the appropriate Assembly node, allowing you to connect any elements external to the Assembly.

Remove Channel Assignment: Removes any channel assigned as an input or output from the Assembly node.

Edit User Channel: Opens a dialog that allows you to modify the settings of the Users Channel, such as Name, Default Value, and the minimum and maximum values.

Change User Channel Type: Opens a dialog to change the User Channel type. The Type determines the formatting of the resulting value, such as a scalar, float, or Boolean.

Expose as User Channel on Group: When making presets for surfacing, often times the author only wants to show certain channels for you to edit, instead of giving you everything. This command allows preset creators to control which channels are exposed. By selecting specific Shader Tree item channels, the **Expose as User Channel on Group** command links the channels to a properly-defined User Channel on the Material Group Item where they can be edited directly (under the **User Channels** sub-tab). The right-click context menu for the Material Group Item allows authors to also lock the preset, making it easier to direct the preset end-user to the appropriate controls. See Preset Browser for more information on creating presets.

Assign Handle: Works in conjunction with the **Channel Handles** function to assign the selected channel to the appropriately created handle. See the Channel Handles page for more information.

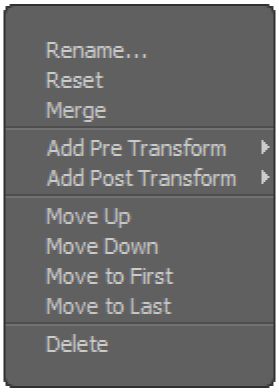
Delete Channel: Removes the selected user channel(s) from the item.

Transform Items

Modo has a very powerful animation feature with its transform items, which are individual transform operations (such as move, rotate, and scale transforms) that control the position, orientation, and size of an item and packaged into their own container inside an associated Item. Each operation, with its three axis coordinates, is its own entity within the item it is controlling. You can add multiple transform items and change their stacking order to control the order that Modo evaluates them. This provides a flexible and powerful means of controlling and animating items. (Modo applies transforms from the bottom up.)

The primary transform's name appears in **Bold** in the **Channels** list to indicate its primary status. These are the transforms that Modo directly modifies using any Transform tools in the viewport. Modo designates the first transform created for any type as the primary transform.

You can add transform items to any item by using the various **Add** operations in the channels display area: **Add Position**, **Add Rotation**, and **Add Scale**. Once added, you can right-click the title of the transform operation to use menu commands and edit transforms in a variety of ways.



Rename: Specifies an alternate name for the selected transform item.

Reset: Resets the selected transform values to zero.

Merge: Evaluates the values of two similar, selected transform items and combines them. Modo clears the channel values for the source transform and removes it without affecting the position, orientation, or scale of the item. Modo transfers the merged values to the primary transform when it's selected, but if you've selected two auxiliary Transforms, Modo merges the values into the last item selected. This command is useful for removing zeroed transforms.

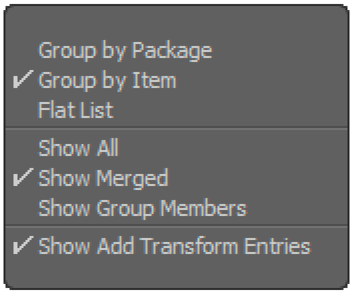
Note: Do not use the **Merge** command for combining transforms with animations as Modo combines the values at the current time only.

Add Pre Transform/Add Post Transform: Defines the position of the new transform item when added to the stack. **Add Pre Transform** adds the item below the selected transform, and **Add Post Transform** adds it after.

Move Up/Move Down/Move to First/Move to Last: Changes the stack location of the selected transform item. **Up** or **Down** moves the selected item one position the specified direction. **First** or **Last** moves the selected item to the bottom or top of the stack respectively.

Delete: Removes the selected transform item from the layer stack.

Right-click Menu—



Right-clicking the **Channels** tab opens the viewport menu. Click **Viewport Settings** to open the menu for the **Channels** viewport.

Grouping options: Define how channels are grouped in the **Channel** list.
Group By Package- Organizes channels by package, which are sub-groupings of related elements by individual items.
Group By Item- Organizes channels strictly by items.
Flat List- Organizes channels as a straight list with no grouping.

Show All: Shows all channels on an item. When disabled (no checkmark), Modo hides certain less-used channels (such as matrix channels).

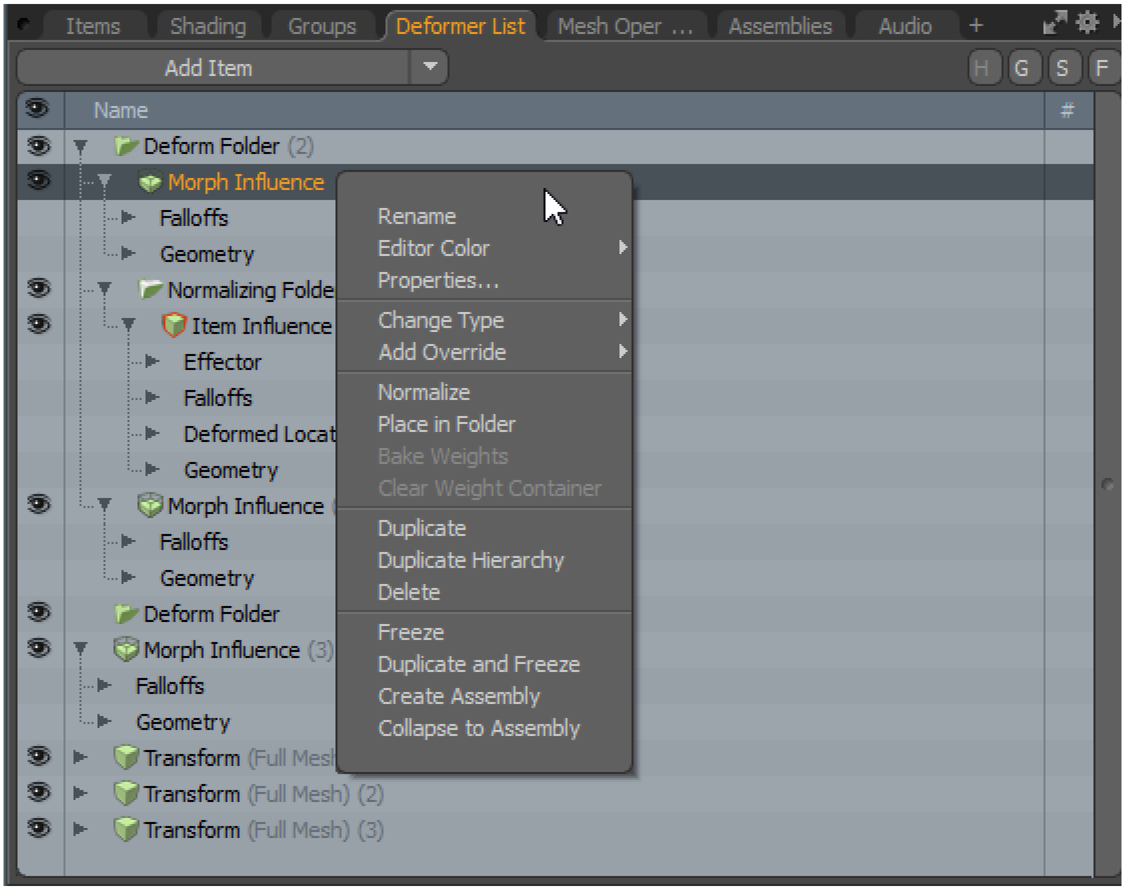
Show Merged: Combines all the channel values for all selected items instead of listing each item's channel one after another. Useful when editing multiple items at once.

Show Group Members: Shows the members of any selected Group.

Show Add Transform Entries: Displays the **Add Position/Add Rotation/Add Scale** entries in the Channel list. These are shortcuts for adding additional transform channels, but for smaller monitors, you may want to remove them to save space on screen.

Deformers Viewport

The **Deformers** viewport provides a view of all the deformers in a scene. In this viewport, you can view, select, reorder, rename, and remove effector and influence items.


Items
Shading
Groups
Deformer List
Mesh Oper ...
Assemblies
Audio
+
Add Item
H
G
S
F
Name
#
Deform Folder (2)
Morph Influence
Falloffs
Geometry
Normalizing Folde
Item Influence
Effector
Falloffs
Deformed Locat
Geometry
Morph Influence
Falloffs
Geometry
Deform Folder
Morph Influence (3)
Falloffs
Geometry
Transform (Full Mesh
Transform (Full Mesh) (2)
Transform (Full Mesh) (3)
Rename
Editor Color
Properties...
Change Type
Add Override
Normalize
Place in Folder
Bake Weights
Clear Weight Container
Duplicate
Duplicate Hierarchy
Delete
Freeze
Duplicate and Freeze
Create Assembly
Collapse to Assembly

Each deformation in Modo, to work properly, has both an effector (the deformer, itself) and an influence. The effector describes how something is modified, and the influence defines what gets modified. When combined, these two parts are referred to as a **Deformer**. When you apply multiple deformers to an item, Modo computes the deformations based on an order of operation that is determined by the item's position within this list. Modo computes and applies items lowest in the list first, then the next item up and so on — moving from the bottom of the list upwards.

The **Deformers** viewport is in the **Setup** and **Animate** interface tabs of the default interface and located in the upper-right side panel. Selecting any item in the list displays its associated attributes in the **Properties** viewport. You can group together multiple elements into a folder by right-clicking one or more selected elements and choose **Place in Folder** (you right-click an item layer to open it).

To select any item in the list, click the item's layer name. Modo indicates a selected layer by accentuating the row and changing the text's color in the viewport. You can add to or remove layers from an existing selection by pressing the **Ctrl** key and clicking the additional layer. You can drag selected layers to reposition them within the list or can delete them by pressing the **Delete** key. If you click, pause, and then click again, you can edit layer names inline.

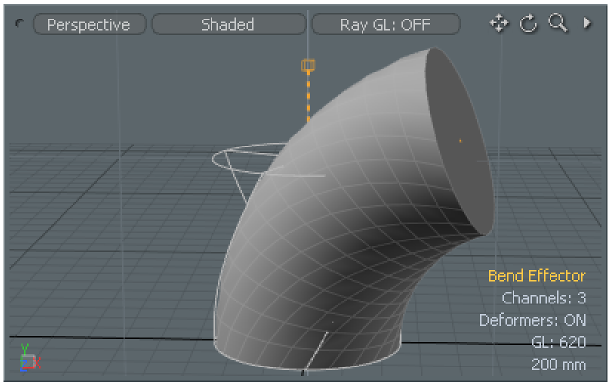
When you right-click any layer within the list, a menu provides additional commands related to the target.

- **Rename** - Opens a dialog box to assign a new name to the selected item.
- **Properties** - Opens a popup window with the associated attributes of the target item. You can adjust the values directly. To dismiss the window automatically, move the pointer away from it.
- **Change Type** - By using this command, you can change similar items from one type to another and retain associations. For example, you can change a Slack deformer to a Lag deformer and keep its association with the original deformer.
- **Add Override** - By using this command, you can add an additional operator on an existing deformer: either an effector or a falloff.
- **Normalize** - Creates a Normalizing folder and moves all currently selected items into the folder. See below for more information about the Normalizing folder.
- **Place in Folder** - Creates a folder and moves all currently selected items into the folder. You can use this to organize multiple items within the **Deformers** viewport.
- **Bake Weights** - Pushes computed weight values down to Weight Maps. (This command only applies to a Normalizing folder.) Modo (on the fly) computes the options of the Normalizing folder on the weights during normalization, and these can be changed on the fly, too, with this command. When you invoke this command, Modo disables the Weight Normalization options.
- **Clear Weight Container** - Removes the weight values associated with the selected item when applied to a **Weight Container** item.
- **Duplicate** - Duplicates the currently selected item.
- **Duplicate Hierarchy** - Creates a duplicate of the current parent item and all child items while retaining the overall relationships of the hierarchy.
- **Delete** - Deletes the currently selected item(s) layer.
- **Freeze** - This freezes the effect of the Deformer into the mesh, resulting in an updated mesh. The Deformer is then removed from the Deformer List as it is no longer required. For more information on freezing, see Freezing Geometry.
- **Duplicate and Freeze** - This creates a new mesh containing a frozen version of the selected mesh. This allows you to produce a new mesh without the Deformers, whilst preserving the original for future editing.
- **Create Assembly** - Creates a new assembly containing the selected Deformers. For more information, see Assembly Presets

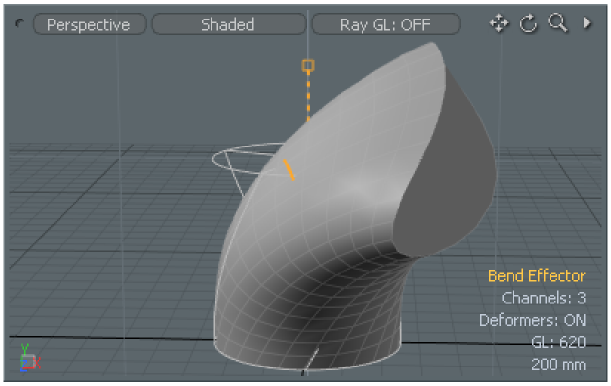
- **Collapse to Assembly** - Creates a new assembly but hides the contents from the scene.

Order of Operations

When Modo applies multiple deformers to a single mesh, the order of operations can have a dramatic effect on the final deformed outcome. For example, you may want to apply a twist and a bend to a cylinder. If Modo twists and then bends the cylinder, the results are very different from having Modo bend and then twist the cylinder. In Modo, the order of operations is determined by the layer order starting at the bottom of the list and moving up. Modo calculates lower deformers, and then it passes the values 'up' to the next deformer in the list, and so on.


Perspective
Shaded
Ray GL: OFF
Bend Effector
Channels: 3
Deformers: ON
GL: 620
200 mm

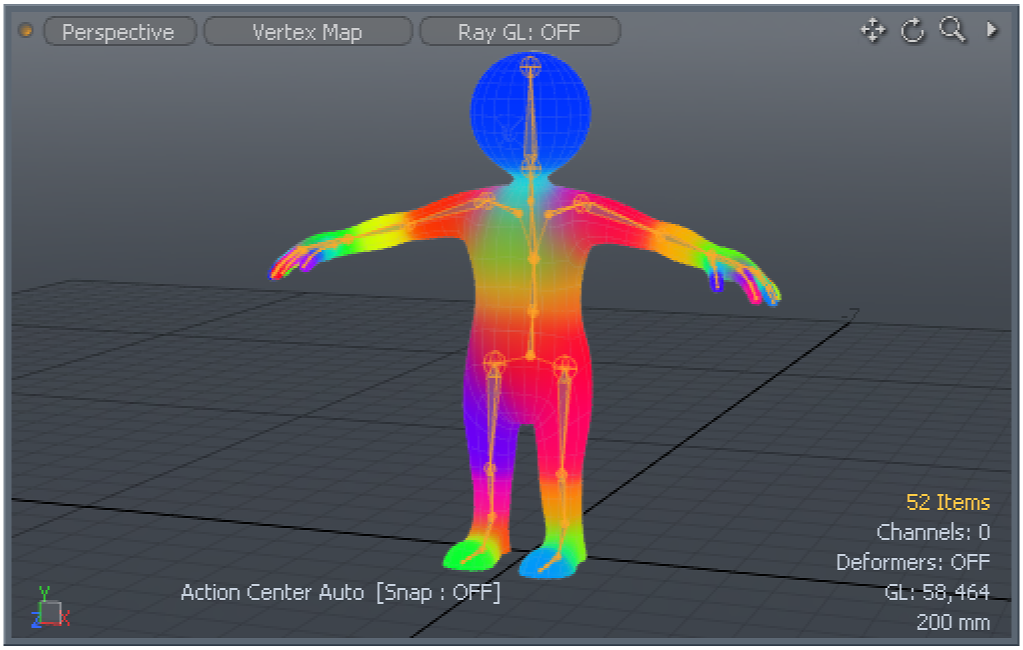
The twist applied first and then the bend


Perspective
Shaded
Ray GL: OFF
Bend Effector
Channels: 3
Deformers: ON
GL: 620
200 mm

The bend applied first and then the twist

Normalization

If you select the **Normalizing Folder** directly, Modo enables a special weight drawing mode for you to more easily see how overlapping weights affect each other. You must have the Vertex Map display style applied to the viewport to see the weights.

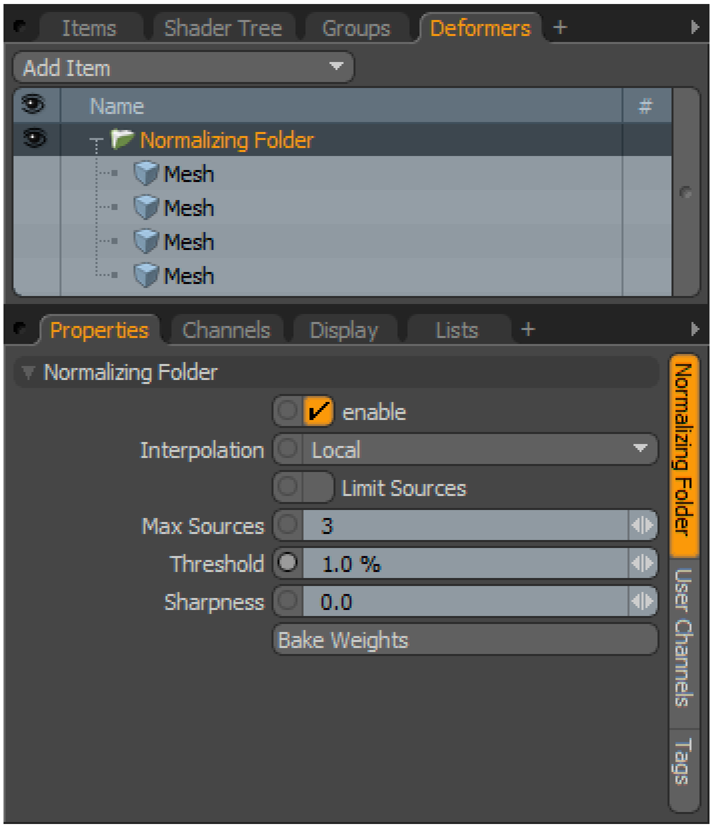


Normalizing Folder

Sometimes you won't want Modo to apply an order of operation, but you want Modo to calculate everything at once (such as with joints in a character). Mathematically, if a vertex between two bones is weighted at 50% for each one's influence, then one applied right after the other doesn't add together to produce 100% of the deformation. To resolve this, you can add a Normalizing folder and place any deformers and influences into it. Modo normalizes the weight values for any items within the folder. To normalize a weight value, Modo determines whether multiple Weight Map values influence a single vertex and adjusts all the values so that their combined values add up to a 1.0 weight value (100%). By doing this, Modo assures that no vertex gets more than a 100% deformation — no matter how many items influence it. Additionally, by using the Normalizing folder Modo makes sure that it calculates deformers concurrently, so that two 50% weighted influences actually produce a 100% deformation. You typically apply normalization to deformations generated using a skeleton joint, where multiple weight-mapped deformers influence a single vertex.

Note: You may never have to consider this issue because Modo adds the normalization automatically during the Bind operation. For more information, see the topic about the Bind & Setup process.

Normalizing Folder—


Items
Shader Tree
Groups
Deformers
+
Add Item
Name
#
Normalizing Folder
Mesh
Mesh
Mesh
Mesh
Properties
Channels
Display
Lists
+
Normalizing Folder
Bake Weights
Normalizing Folder
User Channels
Tags

☑ enable
Interpolation: Local
☐ Limit Sources
Max Sources: 3
Threshold: 1.0 %
Sharpness: 0.0

Enable: Toggles the effect that the **Normalizing Folder** has on its contents. When cleared (disabled), the layer has no effect on the Influences. However, Modo saves disabled layers with the scene, and the values are persistent across Modo sessions.

Interpolation: Determines how Modo interpolates each influence during normalization.
Local- Applies the interpolation setting for each influence — whether it's **Parametric** or **Linear**.
Linear- Overrides each influence's setting and does a **Linear** interpolation for all. This is the same as if the Influences were all individually set for a **Linear** interpolation.
Quaternion- Implements dual-quaternion skinning as the method of interpolation. This only works if all the deformers are transform effectors. (For example, Modo does not use this for a Bend or Lag effector.)

Limit Sources: Specifies whether Modo sets an upper bound on the number of influences that can affect a single vertex. When selected, Modo determines the **X** amount of the highest weights and ignores the rest. (This is good for **Heat** binding, which creates many very small weights, and for **Unity**, which has a maximum of four joints per vertex.)

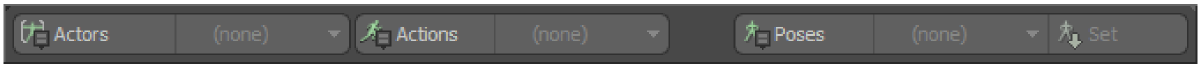
Threshold: Ignores any weight values below this defined threshold. This is good for cleaning up **Heat** binding.

Sharpness: Fine-tunes the strength of the binding. Positive values make vertices bind more strongly to nearby bones, and negative values make them more permissive and bind to weaker bones.

Bake Weights: Pushes the computed weights down to Weight Maps. (Normally, Modo computes the weights on the fly during normalization based on the three previous options and can change the weights on the fly.)

Creating Actors, Actions, and Poses

The **Actors**, **Actions**, and **Poses** menus appear in a bar at the bottom of the **Animate** layout. This bar is under the keyframing and timeline transport controls. You use these menus to animate items. An actor is a special type of Groups Viewport that acts as a container for storing items or channels.



You can quickly assign keyframes for all associated channels, and then save the values as alternate layers with the **Actions** menu. Each action is an alternate of the entire timeline. You can set the current action as a driven action to re-time actions, create time warping affects, and also see multiple actions together in the timeline.

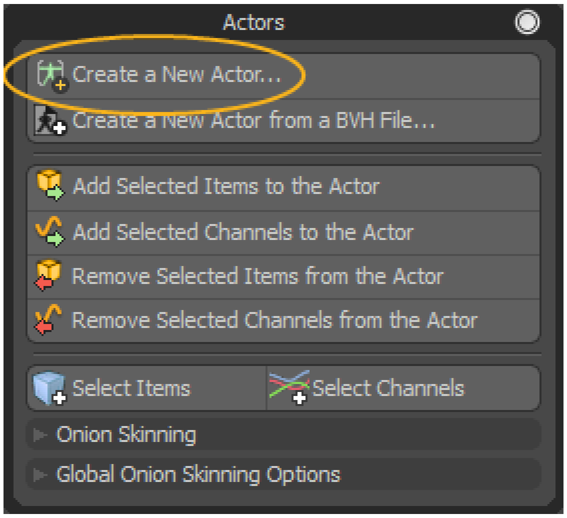
Additionally, the **Poses** menu stores specific keyframe values that you can then easily apply at any point along the timeline.

Creating an Actor

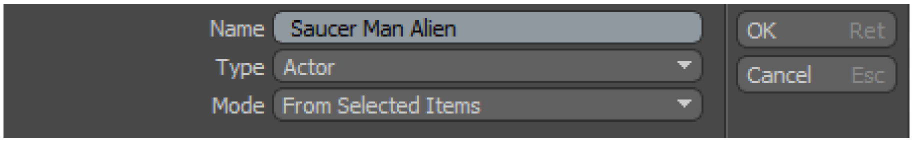
An actor is a container that groups related items together. This could be an entire hierarchy of joints for a character, or it could be all the parts to a dump truck (an actor doesn't need to be a character). By creating an actor you can apply keyframes to all associated items.

Note: Before you create an actor, select all the target items that you want to include in the **Items** list and then click the **Actors** menu.

1. In the **Animate** layout, select the mesh item you wish to use as the target for your animation.
2. At the bottom of the interface, click the **Actors** button.
3. In the **Actors** menu, click **Create a New Actor...**.


Actors
Create a New Actor...
Create a New Actor from a BVH File...
Add Selected Items to the Actor
Add Selected Channels to the Actor
Remove Selected Items from the Actor
Remove Selected Channels from the Actor
Select Items
Select Channels
Onion Skinning
Global Onion Skinning Options

This opens the **New Actor** dialog where you create the actor group.


Name Saucer Man Alien

Type Actor

Mode From Selected Items

OK Ret
Cancel Esc

4. Type a **Name** and ensure **Type** is set to **Actor**.

   This defines how Modo treats the group.

5. In the **Mode** dropdown menu, select which items or channels should be added automatically to the new actor. The following options are available:
   - **Empty** - Creates an actor group with no automatically assigned items. This is useful when you are planning the different actors that are needed. You can assign items later by clicking **Add Selected Items to the Actor** in the **Actions** popup menu.
   - **From Selected Items**- Automatically adds all the selected items in the scene.
   - **From Selected Channels** - Automatically adds all the selected channels in the scene.
   - **From Item Edits** - Adds any items from the current selection that have been changed from their rest state (modified from the **Setup** action state).
   - **From Channel Edits** - Adds any channels from the current selection that have been changed from their rest state (modified from the **Setup** action state).

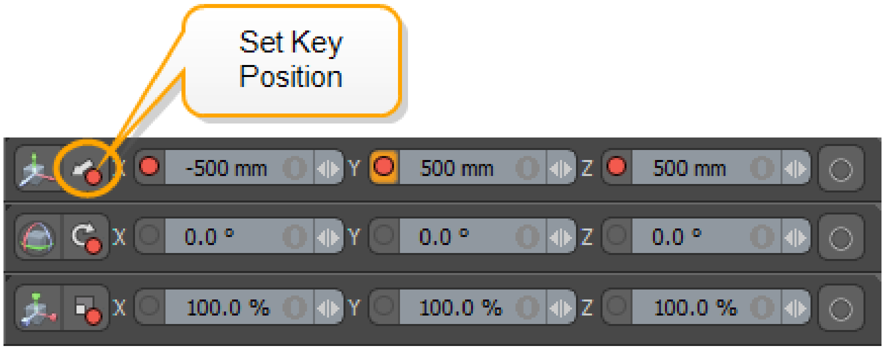
Tip: In most cases, use **From Selected Items** so that Modo adds all selected items automatically to the actor (group).

6. Click **OK** to create the actor.

Tip: If you need to add/remove more items or channels, use the **Add Selected Items to the Actor** and **Remove Selected Items from the Actor** commands in the **Actors** menu after selecting the target element.

Setting Keyframes

When an actor is defined and the item is selected, you can click the **Key** button to define keyframes for every animated channel belonging to each item in the actor. Define at least one keyframe for each channel. (For characters, this would typically be a position or rotation channel.)



For information on creating keyframes, see Working with the Timeline.

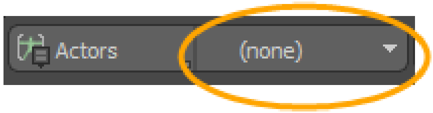
Creating Actions

Actions store keyframes of the entire timeline for each actor. Each separate action can be a complete set of keyframes for the entire length of a given scene within a particular actor group. You can create each action either as a completely separate animation (such as a run cycle in one and a crouch action in another) or a subtle reworking of an existing animation (such as non-destructively modifying a character's motions). In movie terms, each action is its own take of the scene.

To create a new Action layer:

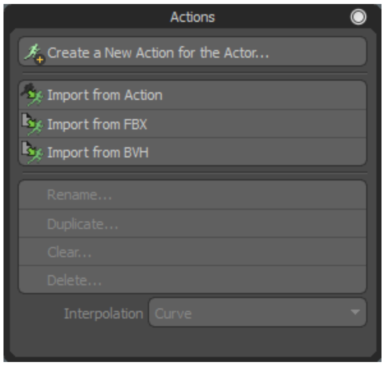
1. Ensure an actor is selected in the **Actors** menu for the new animation. Click the dropdown menu and select an actor.

Note: Each Action has to belong to a specific actor, so you need to have an actor chosen before you can create a new action.

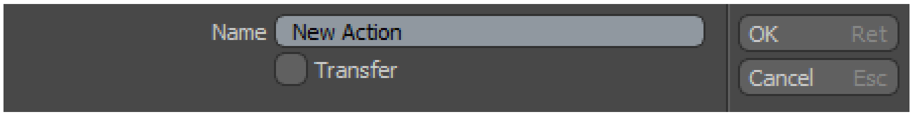

Actors
(none)

2. At the bottom of the **Animate** layout, click the **Actions** button.

   This opens the Actions dialog.


Actions
Create a New Action for the Actor...
Import from Action
Import from FBX
Import from BVH
Rename...
Duplicate...
Clear...
Delete...
Interpolation
Curve

3. Click **Create a New Action for the Actor...**.



Name: New Action

☐ Transfer

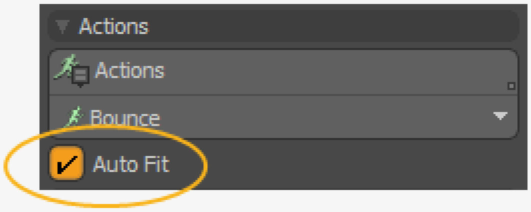
OK Ret
Cancel Esc

4. Type a name and click **OK**.

   Now, when animating items belonging to the current actor, Modo stores the current action in defined keyframes. You can create any number of alternate animations for an actor by repeating the above steps. If you select **Transfer** in the **New Action** dialog, Modo copies any keyframes assigned to the item at the scene level into the new action. You can choose which action to view or edit by clicking it in the **Actions** menu. You can edit actions with the functions in the menu by specifying the action and clicking the appropriate command.

Note: When animating a scene with multiple actors, the current actor switches automatically when you select any item belonging to another actor. That actor's last active action becomes the current action.

Tip: You can set the current Timeline range to the range of the animation in the current Action. To do this, enable **Auto Fit** in the Animation toolbar:



Creating Driven Actions

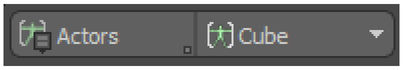
You can apply driven actions to key an action through a single channel. This allows you to re-time actions, create time warping affects, and also see multiple actions together in the timeline. Actions can be played multiple times and with different speeds or directions. This is useful when deciding how fast an animation is happening and where it is happening in a scene.

In this example, we are using a cube which has actions applied to animate it tipping forwards and backwards. Once done, we'll apply a driven action to change how fast that animation is happening and when it is happening in the scene. This driven action controls the entire animation. We'll set the driven action keys to control when the animation starts, ends, and the speed it is played.

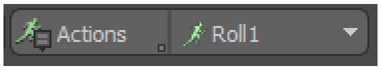
To apply a driven action:

1. Download our example scenes and open **driven_action-roll_model.lxo** in the **Animate** layout.
2. In the bottom panel of the Animate layout, select the actor from the Actors dropdown menu. In this example, ensure **Cube** is selected.

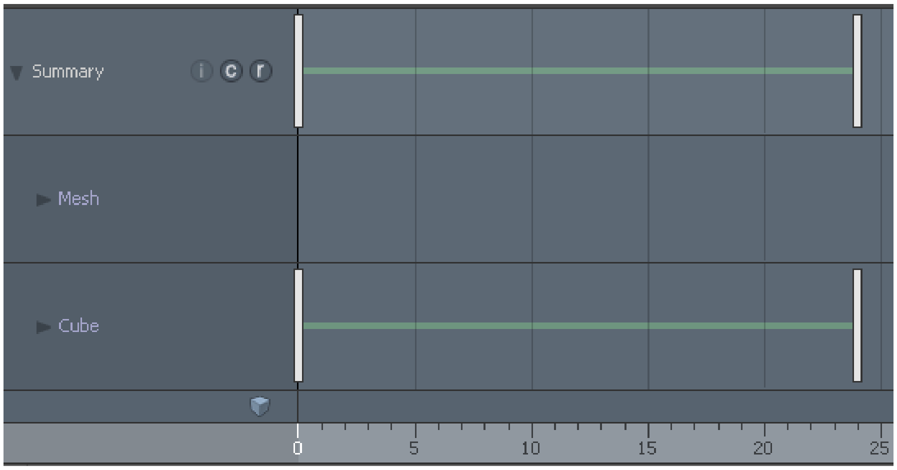
   The Cube actor is a Cube mesh item, which is used later in this procedure for the animation.



3. Select the action you'd like to 'drive' from the Actions dropdown menu. In this example, select **Roll1**.

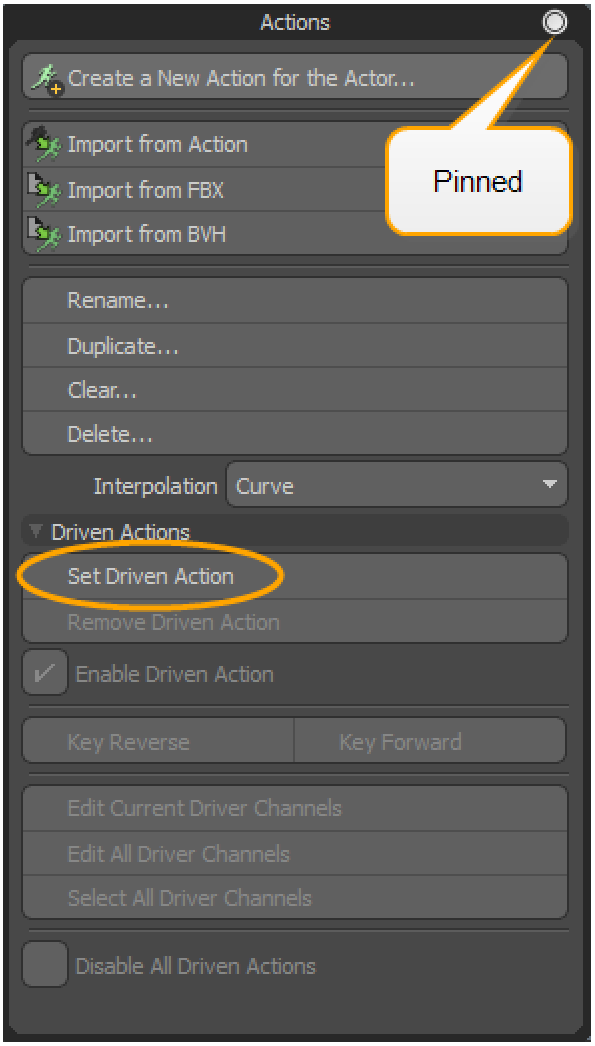


   The timeline updates and shows the summary of the overall animation and the cube animation; starting at 0 and ending at 24 seconds.


Summary
i c r
Mesh
Cube
0
5
10
15
20
25

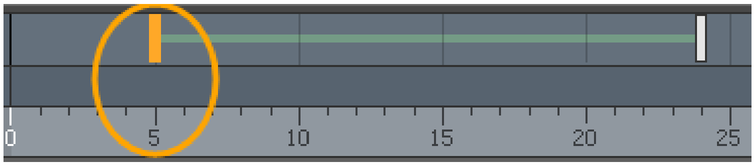
4. At the bottom of the Animate layout, click **Actions** to display the Actions dialog, move it to pin it open, and click **Set Driven Action**.

   The current action is set as a driven action and added as an initial empty driver channel. There will be no visible change until this driver channel is keyed.


Actions
Create a New Action for the Actor...
Import from Action
Import from FBX
Import from BVH
Pinned
Rename...
Duplicate...
Clear...
Delete...
Interpolation
Curve
Driven Actions
Set Driven Action
Remove Driven Action
Enable Driven Action
Key Reverse
Key Forward
Edit Current Driver Channels
Edit All Driver Channels
Select All Driver Channels
Disable All Driven Actions

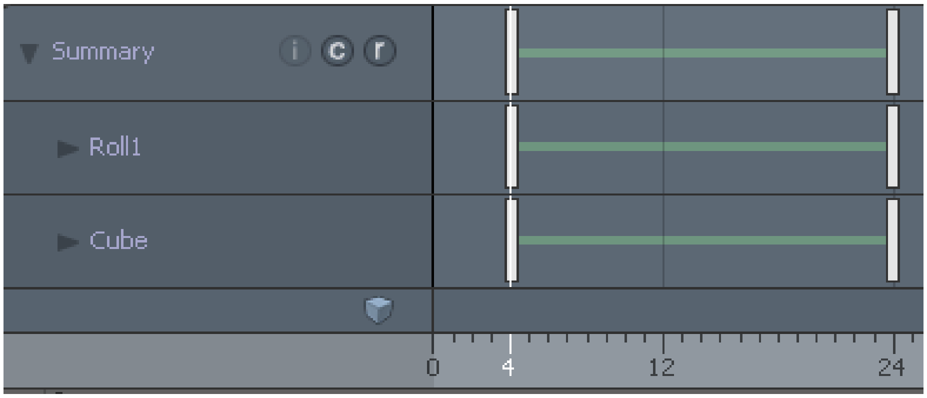
5. In the timeline, click+drag the first key, and position it to 5 seconds.

   This key will be used as the starting point for the Key Forward animation in our next step.



6. In the Actions dialog, click on **Key Forward**.

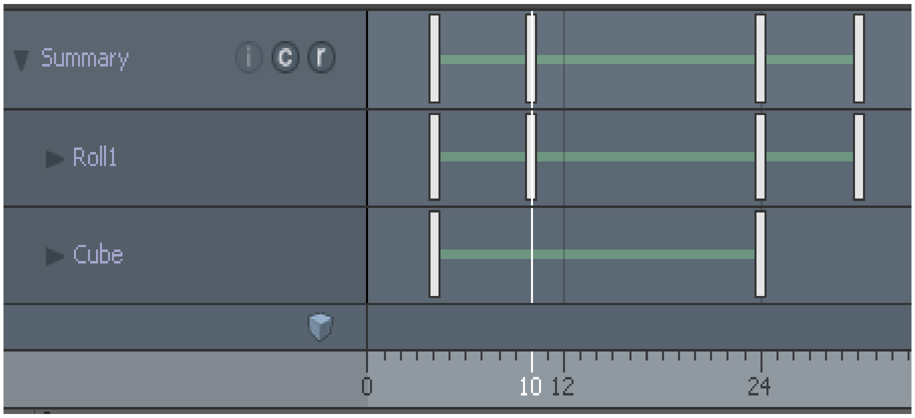
   This sets the initial key and the end key value for the action **Roll1**.



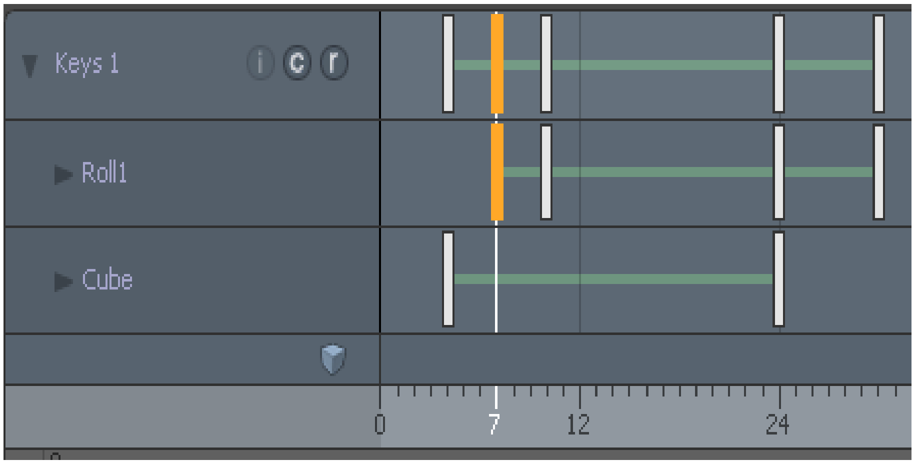
7. Scrub in the timeline to 10 seconds and click **Key Reverse** in the Actions dialog.

   This creates a new percentage driver channel and adds a key at the current time for the start, and a key for the end.

   When you scrub to 10 seconds in the timeline, notice the cube is animated. Once you have set **Key Reverse** at 10 seconds, the timeline displays new keys.



8. Click+drag the first key for the **Roll1** item and reposition it to **7** seconds.

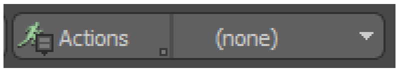


9. Scrub in the timeline.

   The action starts playing from where you keyed it at 7 seconds and ends at 29 seconds.

10. At the bottom of the Animate layout, set **Actions** to **None**.

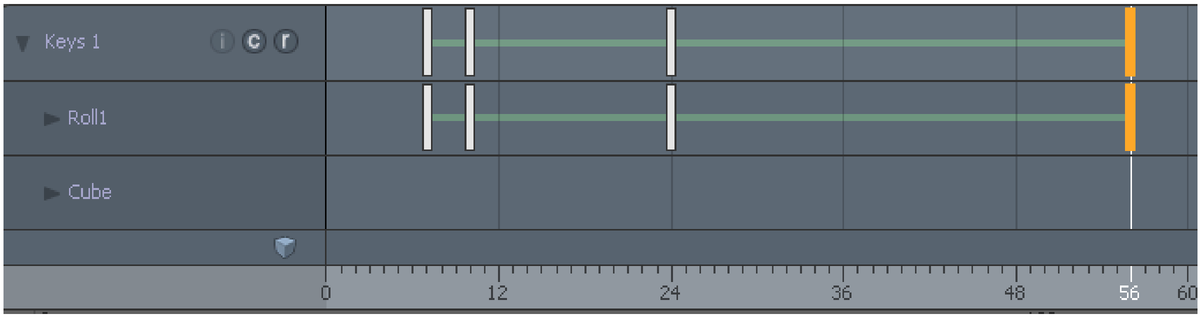
    Only the driven action is displayed in the timeline. We'll change the location of when the animation starts and how fast it runs later in this procedure.



11. Scrub in the timeline.

    The action starts playing from where you keyed the driven action at 10 seconds and ends at 29 seconds.

12. Change the end key position to 56 and scrub in the timeline.



Notice how this driven action controls the entire animation. Also, notice that the speed of the cube rotating at the end of the animation is much slower because we have increased the time.

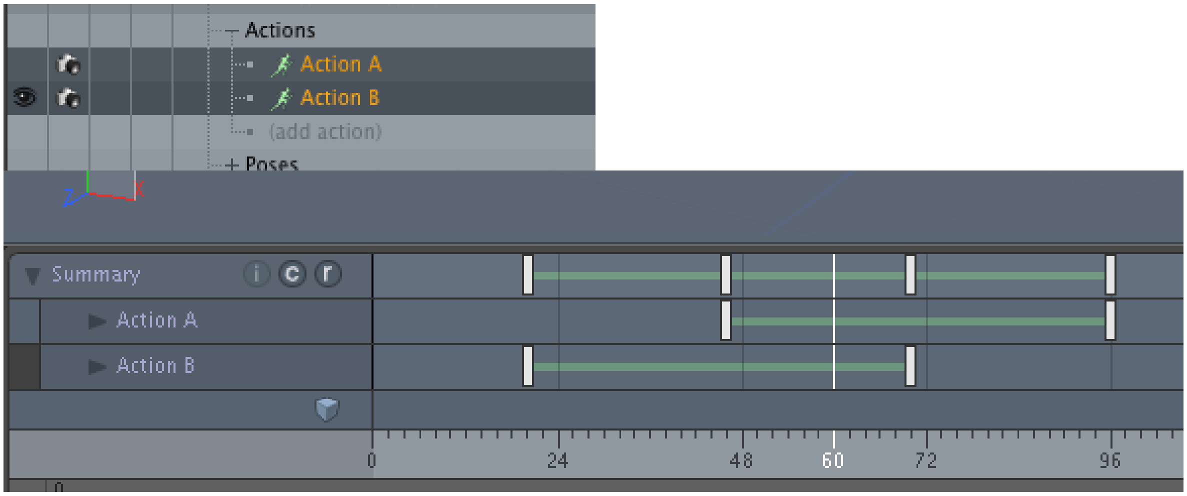
Advanced Driven Actions

Driver channels are user channels on the Action Clip items, which can be found in the Groups Viewport under the actor they belong to. Percentage type user channels may be added manually and are treated by the modifier as driver channels. User channels of others types are ignored. For more information about channels types, see Add User Channels.

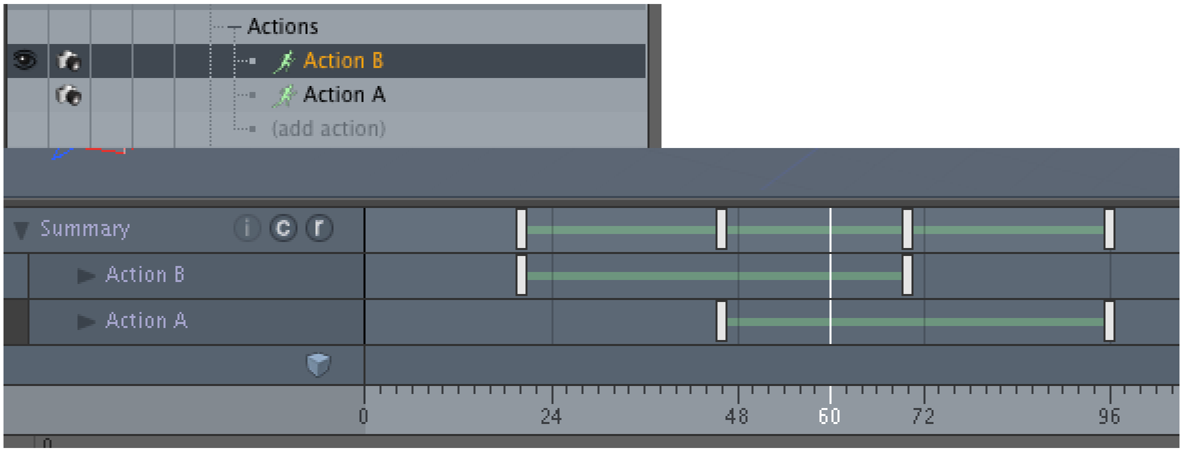
Actions can be prioritized when they overlap by changing their order in the Groups viewport. Actions higher up take priority over those below.

Driver channels can be driven by Channel Links, they affect the whole range of the scene unless there is another driven action with a higher priority that has a keyed Driver channel.

In the images below **Action A** takes priority over **Action B** at frame 60. **Action B** plays up until the start of **Action A**.

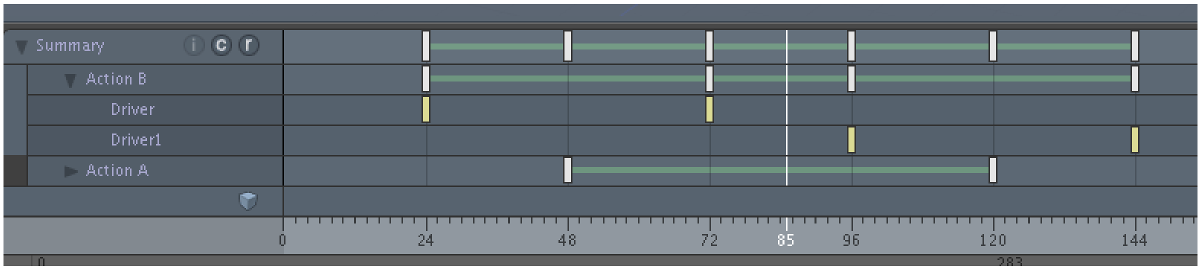

Actions
Action A
Action B
(add action)
Poses
Summary
Action A
Action B
0
24
48
60
72
96

Dragging **Action B** above **Action A** in the Groups viewport gives it priority. **Action B** plays until it ends after which **Action A** plays.


Actions
Action B
Action A
(add action)
Summary
Action B
Action A
0
24
48
60
72
96

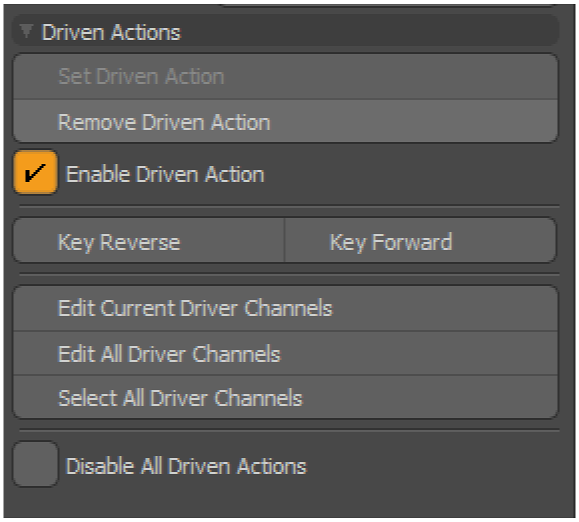
The first and last keys on the driver channels define the range over which the channel affects the animation. Underlying actions are not considered inside this range, even if the driver channel has a zero percentage set.

Multiple Driver channels should be used to play an Action at multiple times in the scene and have any other Actions play in between.


Summary
Action B
Driver
Driver1
Action A
0
24
48
72
85
96
120
144

In the image above **Action B** plays between frames 24 and 72, **Action A** plays between frames 73 and 95, and the **Action B** plays again between frames 96 and 144.

Driven Actions Controls


Driven Actions
Set Driven Action
Remove Driven Action
Enable Driven Action
Key Reverse
Key Forward
Edit Current Driver Channels
Edit All Driver Channels
Select All Driver Channels
Disable All Driven Actions

- **Set Driven Action** - Sets the current Action (as chosen in the Actions popup) as a driven action and adds an initial empty driver channel. There will be no visible change until this Driver channel is keyed.
- **Remove Driven Action** - Removes the driven state from the current action and deletes all the driver channels.
- **Enable Driven Action** - Used to enable or disable the driven state of the current action. The driver channels are preserved when disabled.
- **Key Reverse** - Creates a new percentage driver channel and adds a key at the current time with a value of 0% for the start, and a key at the current time plus the length of the animation in the Action with a value of 100% for the end. The channel has it's limits set to clamp key values between zero and 100%.

  The channel **Curve Interpolation** is set to **Linear** so the animation plays back at it's correct rate. This can be changed if you'd like to vary the speed of the animation between the keys. Additional keys can be added for warping the playback speed.
- **Key Forward** - Sets the initial key values to 100% at the start and 0% at the end so the animation plays backwards.

- **Edit Current Driver Channels** - Selects the current driven action and opens the Graph Editor for editing the driver channels.
- **Edit All Driver Channels** - As above but selects all the driven actions belonging to the current actor.
- **Select All Driver Channels** - Selects all the driven actions belonging to the current actor. This is useful to load the driver channels into the Dope Sheet for example.
- **Disable All Driver Channels** - As the name suggests this will disable all the driven actions belonging to the current actor. This is useful if you need to edit or view the actions without the driven action modifier affecting them. This does not affect the individual enable state of driven actions.

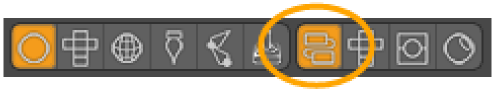
Driven Actions in the Schematic View

Driven actions are regular channels found in the **Items** list **Channels** tab, which you can drag-and-drop into the Schematic Viewport to link to the actor in your rigging.

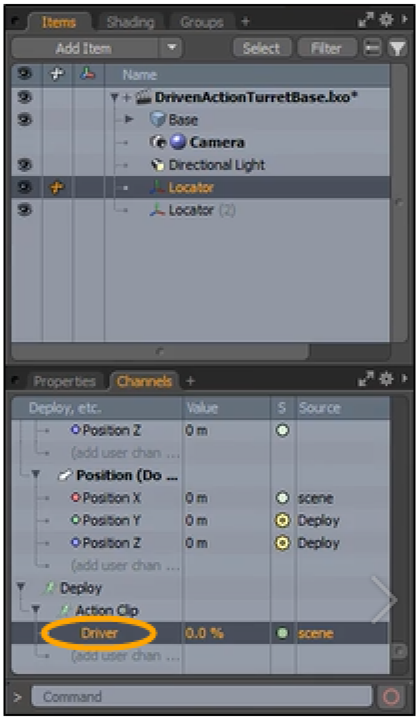
The following procedure assumes that you have created your rigging, which includes a locator. This example walks you through the process of dragging-and-dropping the driven action into the Schematic view and moving the Locator Item to update your animation.

To add and relocate a locator in the Schematic view:

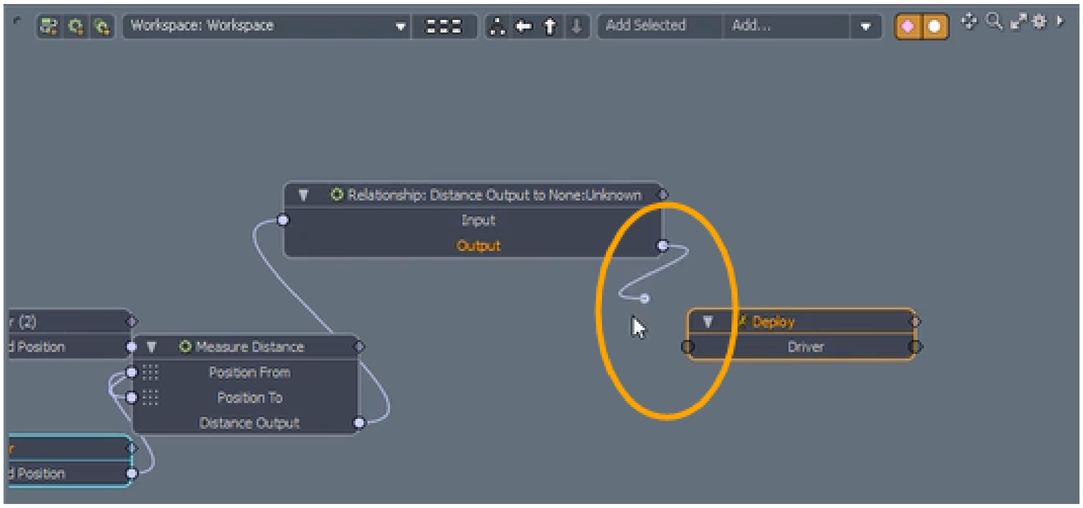
1. At the top of the left side of the Modo interface, click **Open the Schematic Palette**.



2. On the right panel of the **Model** layout, open the **Items** list, and then open the **Channels** tab.


Items
Shading
Groups
Add Item
Select
Filter
Name
DrivenActionTurretBase.lxo*
Base
Camera
Directional Light
Locator
Locator (2)
Properties
Channels
Deploy, etc.
Value
S
Source
Position Z
0 m
(add user chan ...
Position (Do ...
Position X
0 m
scene
Position Y
0 m
Deploy
Position Z
0 m
Deploy
(add user chan ...
Deploy
Action Clip
Driver
0.0 %
scene
(add user chan ...
Command

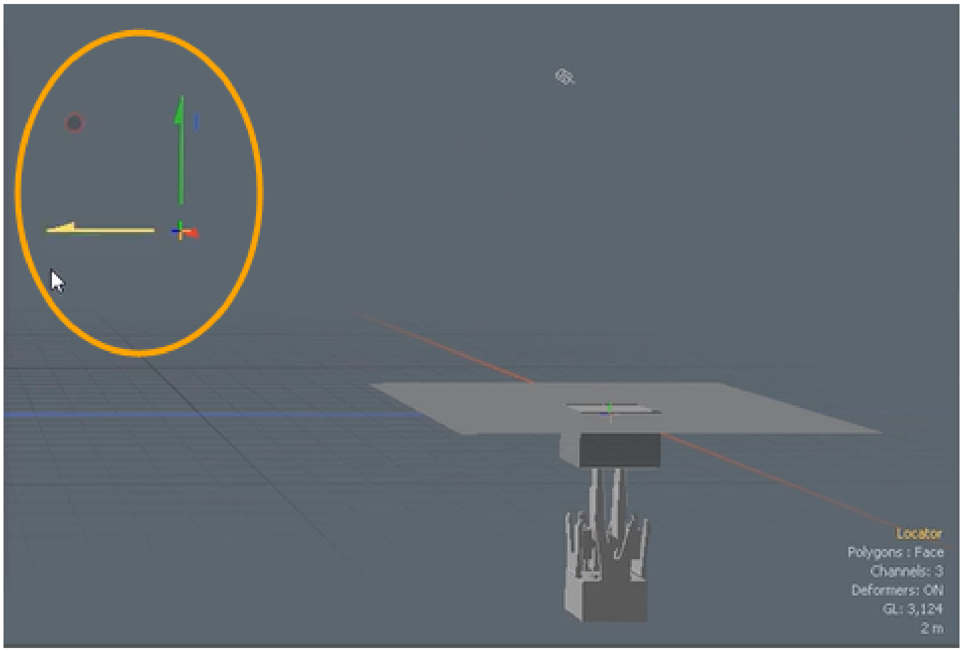
3. Drag the **Driver** from the **Items** list onto the **Schematic** view and hook the driver to the actor.


Workspace: Workspace
Add Selected
Add...
Relationship: Distance Output to None:Unknown
Input
Output
r (2)
d Position
Measure Distance
Position From
Position To
Distance Output
d Position
Deploy
Driver

4. In the 3D viewport, select the **Locator** and move it toward the mesh item.

   The animation plays forward.

5. Move the **Locator** away from the mesh item to play the animation in reverse.

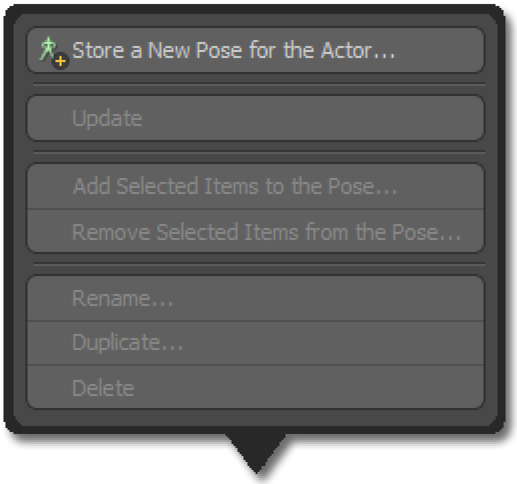


Creating Poses

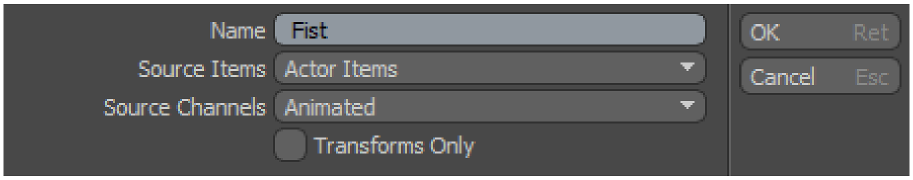
A pose stores a static snapshot of the values for the current keyframe of the active actor. You can then re-apply these values to the actor at another point in time. Poses don't need to be for the full character. They can be limited to just part of an actor or even a single joint. For example, you may want to create poses for a character's hand positions (such as a fist or a pointing finger). You can then apply these poses to the actor at any place in the timeline.

To create a pose:

1. First position the target elements as desired.
2. Define keyframes for the target elements.
   For example, with a hand, this could be making a fist by bending the fingers appropriately.
3. Select all the target items. For example, double-clicking the wrist joint selects the item and all its children.
4. Next, click the **Poses** menu.


Store a New Pose for the Actor...
Update
Add Selected Items to the Pose...
Remove Selected Items from the Pose...
Rename...
Duplicate...
Delete

5. In the **Poses** menu, click **Store a New Pose for the Actor**.

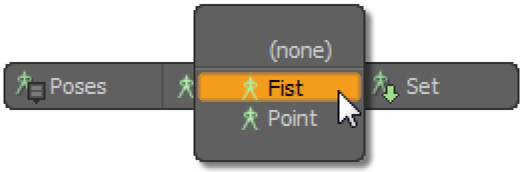

Name: Fist
Source Items: Actor Items
Source Channels: Animated
☐ Transforms Only

6. In the **Store Pose** dialog, type a **Name** for the pose.
7. Select **Selected Actor Items** in the **Source Items** list.
8. Click **OK**.

Modo stores the pose for you to apply later. In the **Store Pose** dialog, you specify an item from the **Source Items** list to indicate which items belonging to the actor get stored in the pose: **Actor Items** stores values for all elements in the actor and disregards any current selection, and **Selected Actor Items** limits the values stored to the current item selection. The item that you select in the **Source Channels** list controls which channels from the source are stored: **Edited** stores channel values from the source elements only for channels that are different from their rest state (modified from the **Setup** action state), **Animated** stores only channels from the source elements that have keyframes applied to them, and **All** stores all related channels, animated or not, from the source elements in the pose. If you select the **Transforms Only** checkbox, Modo only stores transform-specific channels: **Position**, **Rotation**, and **Scale**.

Tip: When you create a pose, the items in the actor are not affected. Their channel values are copied (not cut). You can create more poses with the current values as a starting point.

Applying a Pose

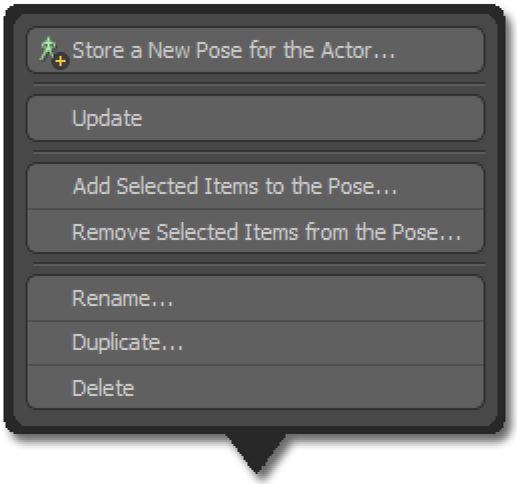

Poses
(none)
Fist
Point
Set

To apply a stored pose:

1. Go to the position on the timeline where you want the pose.
2. Specify one of the stored poses from the list next to the **Poses** menu.
3. With the pose now active, click the **Set** button next to the pose to assign it.

Modo assigns all the stored channel values for the elements within the pose as keyframes. Once the values are set, they can be treated like any other keyframe value.

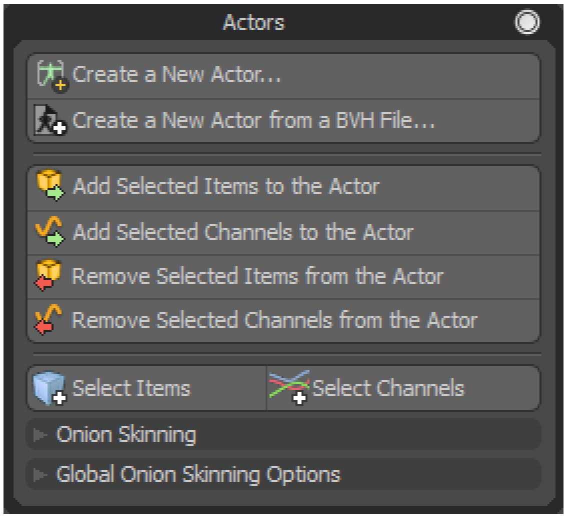
Pose Controls


Store a New Pose for the Actor...
Update
Add Selected Items to the Pose...
Remove Selected Items from the Pose...
Rename...
Duplicate...
Delete

- **Store a New Pose for the Actor** - Opens the Store Pose dialog for saving collections of keyframe values to reuse.
- **Update** - Updates the keyframe values in an existing pose. Modo stores keyframe values when it creates a pose. With the target pose selected, make adjustments to the posed elements and then click this command to save the value in the current pose.
- **Add Selected Items to the Pose** - Opens the **Edit Pose** dialog for you to add additional items to those already included in a specific pose. The dialog also has options to control which channels from the added elements are stored in the pose when added. You can also limit the included channels to just the transform channels (**Position**, **Scale**, and **Rotate**).

- **Remove Selected Items from the Pose** - Opens the **Edit Pose** dialog, too, but defaults to **Remove** options, instead of **Add** options.
- **Rename** - Opens the **Pose Name** dialog where you can change the existing name.
- **Duplicate** - Opens the **Pose Duplicate** dialog where you can clone an existing pose to a new one with a different name.
- **Delete** - Removes the action from the scene entirely. Modo opens a confirmation dialog for you to verify the removal.

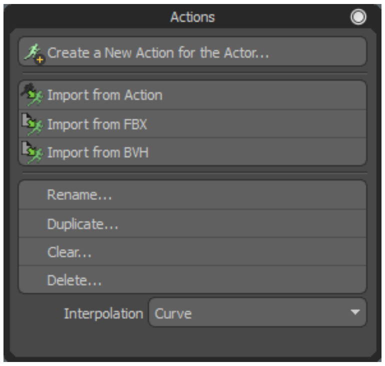
Actor Controls


Actors
Create a New Actor...
Create a New Actor from a BVH File...
Add Selected Items to the Actor
Add Selected Channels to the Actor
Remove Selected Items from the Actor
Remove Selected Channels from the Actor
Select Items
Select Channels
Onion Skinning
Global Onion Skinning Options

- **Add Selected Items to the Actor** - Adds additional items to the current actor. To use, first select the items to add and then click this button. Modo adds the items to the actor.
- **Add Selected Channels to the Actor** - Adds additional channels to the current actor. To use, first select the channels to add and then click this button. Modo adds the selected channels to the actor.
- **Remove Selected Items from the Actor** - Removes items associated with the current actor. To use, first select the items to remove and then click this button. Modo removes the element from the actor.
- **Remove Selected Channels from the Actor** - Removes items associated with the current actor. To use, first select the channels to remove and then click this button. Modo removes the element from the actor.
- **Select Items** - Selects all items related to the current actor.
- **Select Channels** - Selects all the animated channels related to all the items in the actor.

See below for information about Onion Skinning.

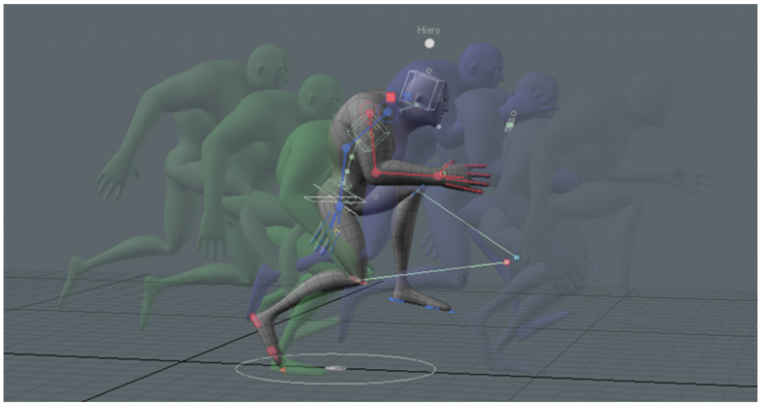
Action Controls



- **Create a New Action for the Actor** - Opens the New Action dialog for creating alternate versions of the entire timeline.
- **Import from Action** - Allows you to import an existing animation into the existing actor.
- **Import from FBX** - Allows you to import motion data from **.fbx** files. For more information on importing, see FBX Format Files.
- **Import from BVH** - Allows you to import motion data from **.bvh** files. For more information, see BVH Format Motions.
- **Rename** - Opens the **Action Name** dialog where you can edit the current name.
- **Duplicate** - Opens the **Action Duplicate** dialog where you can clone an existing action into a new one with a different name. You can make the new action the current, active action.
- **Clear** - Removes all keyframes from the current action, but retains the empty action itself. When used, Modo shows a confirmation dialog for you to verify that you want the keyframes removed.
- **Delete** - Removes the action from the scene entirely. Modo shows a confirmation dialog for you to verify the deletion.
- **Interpolation** - Defines the default method of interpolation between keyframes. You can select **Curve**, **Linear**, or **Stepped** to modify keyframes in the Graph Editor.

Onion Skinning

Onion skinning is a term related to traditional hand-drawn animation when the initial rough motion studies were drawn on thin, translucent paper for the artist to see the previous frames in a sequence through the paper. A drawing would seem more faded the lower it was in the stack. This way of working makes it easier for the artist to visualize the overall motion of an action and adjust the timing as necessary. The onion skinning feature in Modo mimics this same method of working, but offers additional controls not available to paper-based artists.



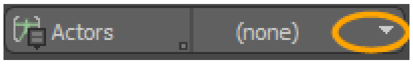
You find onion skinning options in the **Actors** menu. Actors determine the items that are drawn with onion skinning in the 3D viewport, and Modo takes the animated motion from the active action for the actor. When you set the active action to **(none)**, Modo does not display onion skinning. Also, Modo does not display onion skinning during playback or while scrubbing the timeline.

You can use the onion skinning options to assign and modify common onion skinning settings for the current actor as well as general options for the onion skinning drawing.

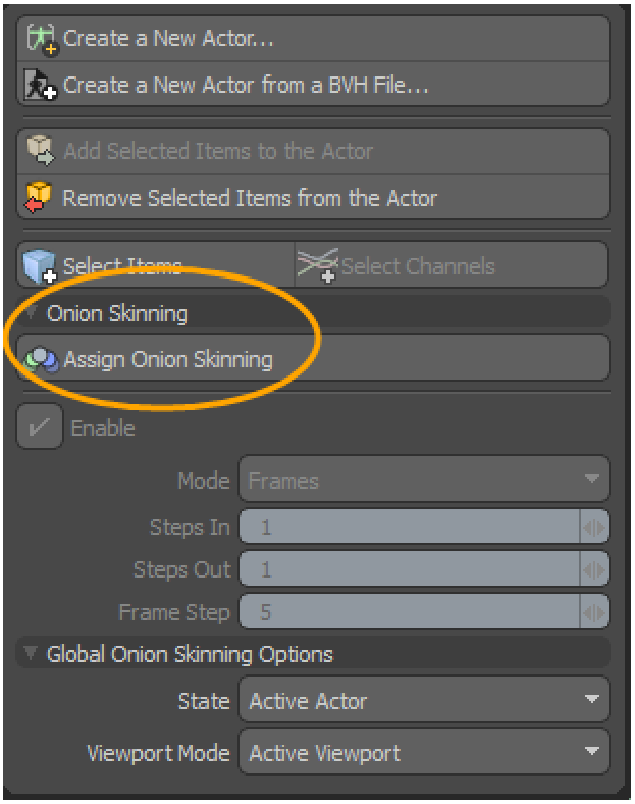
Note: You must first create an actor to have something to assign the onion skinning options to. You must have some animation applied to the elements in the actor to give the onion skinning function something to display.
You'll also need an Action for the actor which has the **Transfer** option enabled for Modo to copy any keyframes assigned to the item at the scene level into the new Action.

To assign onion skinning:

1. In the bottom panel of the **Animate** layout, click the dropdown menu beside the **Actors** button, and click + select the target actor.



2. Click to **Actors** button, expand **Onion Skinning** and click **Assign Onion Skinning**.


Create a New Actor...
Create a New Actor from a BVH File...
Add Selected Items to the Actor
Remove Selected Items from the Actor
Select Items
Select Channels
Onion Skinning
Assign Onion Skinning
Global Onion Skinning Options

☑ Enable

Mode: Frames

Steps In: 1

Steps Out: 1

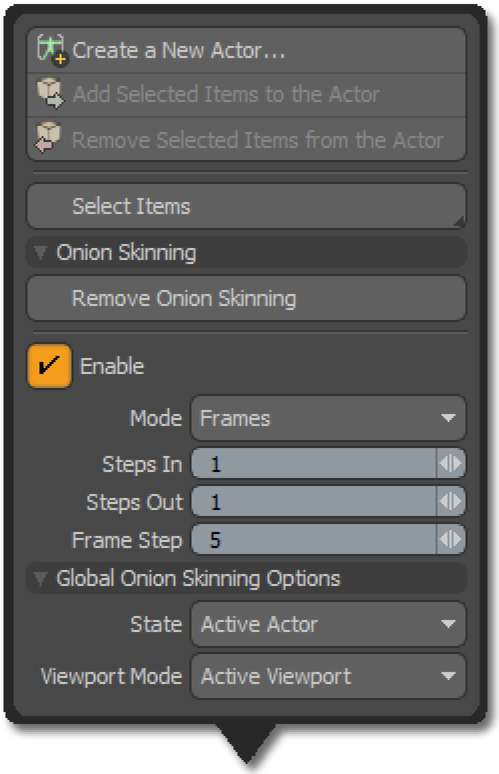
Frame Step: 5

State: Active Actor

Viewport Mode: Active Viewport

The preceding and following frames from the current point in time with the current time position is displayed most prominently.

Onion Skinning Controls


Create a New Actor...
Add Selected Items to the Actor
Remove Selected Items from the Actor
Select Items
Onion Skinning
Remove Onion Skinning
Global Onion Skinning Options

☑ Enable

Mode: Frames

Steps In: 1

Steps Out: 1

Frame Step: 5

State: Active Actor

Viewport Mode: Active Viewport

- **Assign/Remove Onion Skinning** - Turns onion skinning on or off for the currently selected actor.
- **Enable** - Use this to temporarily enable or disable onion skinning while retaining the existing settings.
- **Mode** - Determines the intervals used to display the onion skinning:
  - **Frames** - Uses the whole frame values of the timeline.
  - **Keys** - Takes the displayed intervals based on defined keyframes for the selected actor.
  - **Time Markers** - Uses the time markers associated with the selected actor.
- **Steps In** - Controls the number of steps to show prior to the current time (maximum of 5).
- **Steps Out** - Controls the number of steps to show after the current time (maximum of 5).
- **Frame Step** - Controls the number of frames between each drawn step and only applies when the interval **Mode** is set to **Frames**

The following **Global Onion Skinning Options** are available:

- **State** - Provides choices to control globally how Modo displays the Onion Skinning function:
  - **Off** - All settings remain intact for each actor. Modo only disables the display.
  - **Active Actor**- Modo only draws onion skinning for the current actor. This is the default setting.
  - **All Actors**- Modo draws onion skinning for all actors that have onion skinning assigned and enabled.
- **Viewport Mode** - Provides choices to control how Modo displays onion skinning:

- **Active Viewport** - Displays onion skinning only in the current active viewport.
- **All Viewports** - Displays onion skinning in all visible 3D Viewports.

Tip: You can also apply onion skinning to cameras and other locator-based items (such as lights) to include custom drawing features. To do this, create an actor that includes these types of items. (Locator items do not fade because of how Modo draws them in the viewport.)

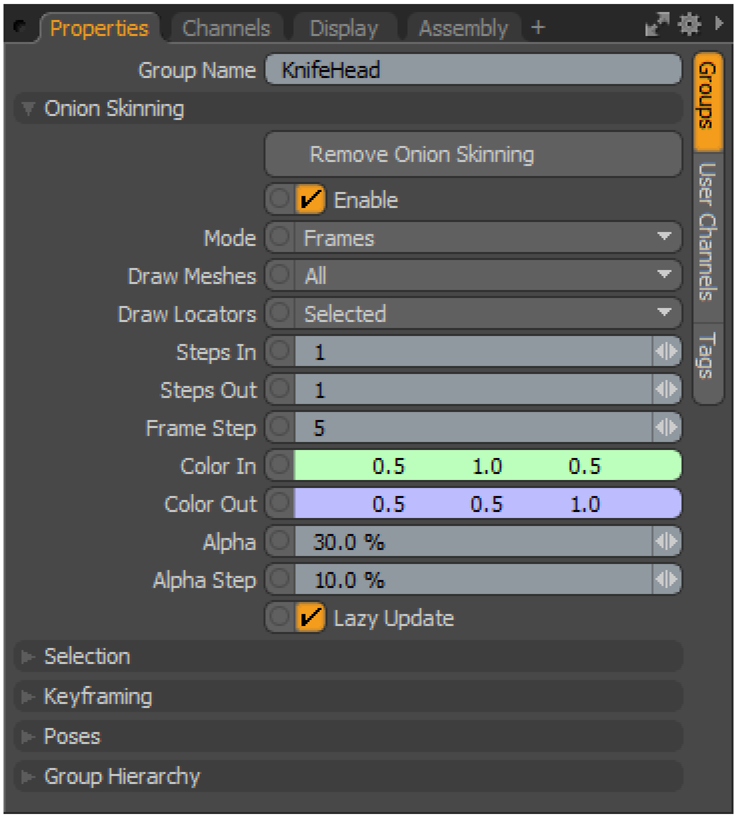
Group Onion Skinning Properties

When you select an actor, you can also activate Onion Skinning options in the **Properties** viewport. The **Properties** panel also has some additional options for controlling how Modo displays onion skinning.

To view the Group Onion Skinning properties:

1. Open your scene in the **Model** layout and then open the **Groups** tab on the right panel.
2. In the properties **Groups** tab, under **Onion Skinning**, click **Assign Onion Skinning**.

   The following properties display.


Properties
Channels
Display
Assembly
+
Groups
User Channels
Tags
Remove Onion Skinning
Selection
Keyframing
Poses
Group Hierarchy

Group Name: KnifeHead

Onion Skinning

☑ Enable

Mode: Frames

Draw Meshes: All

Draw Locators: Selected

Steps In: 1

Steps Out: 1

Frame Step: 5

Color In: 0.5 1.0 0.5

Color Out: 0.5 0.5 1.0

Alpha: 30.0 %

Alpha Step: 10.0 %

☑ Lazy Update

- **Assign/Remove Onion Skinning** - Enables or disables the Onion Skinning function for the currently selected actor.
- **Enable** - Use this to temporarily enable or disable onion skinning while retaining all the existing settings.
- **Mode** - Determines the intervals used to display the onion skinning: **Frames** uses the whole frame values of the timeline, **Keys** takes the displayed intervals based on defined keyframes for the selected actor, and **Time Markers** uses the time markers associated to the selected actor.
- **Draw Meshes** - Provides fine control over the elements of the actor group that are drawn as part of the onion skinning:
  - **None** - Modo does not draw meshes that are members of the actor.
  - **Selected** - Modo only draws selected meshes belonging to the actor.
  - **All** - Modo draws all meshes belonging to the actor.
- **Draw Locators** - This option provides fine control over what locator-type elements (such as locators, cameras, and lights) of the actor group that Modo draws as part of the onion skinning:
  - **None** - Modo does not draw any locator-type items belonging to the actor.
  - **Selected** - Modo draws only selected locator-type items. (This is the default.)
  - **All** - Modo draws all locator-type items belonging to the actor.
- **Steps In** - Controls the number of steps Modo draws prior to the current time (maximum of 5).
- **Steps Out** - Controls the number of steps Modo draws after the current time (maximum of 5).
- **Frame Step** - Controls the number of frames between each drawn step and only applies when the interval **Mode** is set to **Frames**.
- **Color In** - The display colors for steps prior to the current frame.
- **Color Out** - The display colors for steps following the current frame.
- **Alpha** - The transparency amount for the frames directly before and after the current frame.
- **Alpha Steps** - Defines the additional amount of transparency for subsequent steps.
- **Lazy Update** - Displays onion skinning only after you apply an edit. This is the default to increase the interactivity speed. For simple scenes, you can disable this option to preview all edits in real time.

Note: Sometimes it may not be desirable to have the mesh of a given character in the actor group along with rig control items. In this situation you can create a standard Groups Viewport with the character mesh and then parent it to the actor group (in the **Groups** viewport) to keep it separate. Modo applies the onion skinning settings to the actor as well as its child item.

Animation Layers

Animation Layers stack animations on top of one another, which allows you to non-destructively add or change elements of your animation sequence without editing one continuous, complicated sequence. Artists with experience in image editing applications are familiar with the principle of using multiple image layers and masks; this is the same principle, only using animations instead of still images.

Before creating animation layers, Modo needs an Actor and Action for that Actor to be enabled. For more information, see Creating Actors, Actions, and Poses.

You also need a base keyframe in the Action. For example, if you want to use an additive layer for moving an object along the Y axis, the Position Y channel needs a keyframe within the Action. Additive layers need a starting reference point for the channel.

There are two types of Animation Layers you can create: Additive Layers and Override layers.

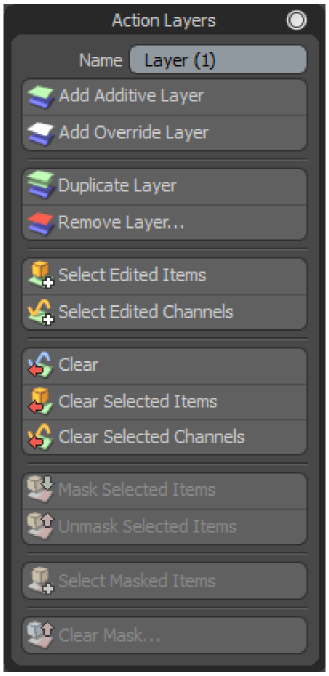
Additive Layers

An Additive Layer takes any animation that you put within a layer and adds it to an existing animation. For example, an animation of a moving ball might be your base animation.

To create an additive layer:

1. Open the **Modo** layout.
2. Click on the **Time** Viewport and **Graph Editor** Viewport buttons. These are located at the bottom of the Modo layout and control animation playback and edit keyframes respectively.
3. Click the **Animation** button in the left panel to open the Animation toolbar, then click on **Action Layers**.

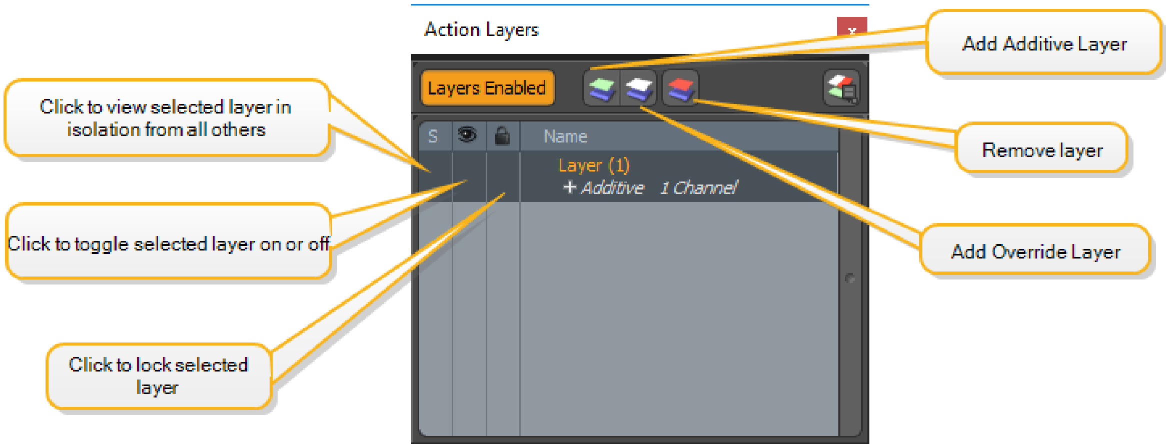
   The **Action Layers** panel opens.


Action Layers
Name
Layer (1)
Add Additive Layer
Add Override Layer
Duplicate Layer
Remove Layer...
Select Edited Items
Select Edited Channels
Clear
Clear Selected Items
Clear Selected Channels
Mask Selected Items
Unmask Selected Items
Select Masked Items
Clear Mask...

> **Tip:** If you prefer to work in the **Animate** layout, you can access the Action Layers tools by clicking the Action Layers button on the **Graph Editor**.

4. Click **Add Additive Layer**.

   Your new additive layer appears in the **Action Layers** palette, to the right of the **Graph Editor**. To remove a layer, click on the Remove Layer button or click on the ☒ button on a selected layer in the palette. To duplicate a layer, right-click on the layer and select **Duplicate.**


Action Layers
Layers Enabled
S
Name
Layer (1)
+ Additive 1 Channel
Add Additive Layer
Remove layer
Add Override Layer
Click to view selected layer in isolation from all others
Click to toggle selected layer on or off
Click to lock selected layer

5. As a test, make a keyframe animation change, such as a change in an object's position over time.

6. Click the **Play** button below the graph editor to play through your animation.

   You can add as many additive layers as you like.

Note: The layer stack hierarchy is rigid, meaning once layers are added they cannot be rearranged in the list.

Override layers

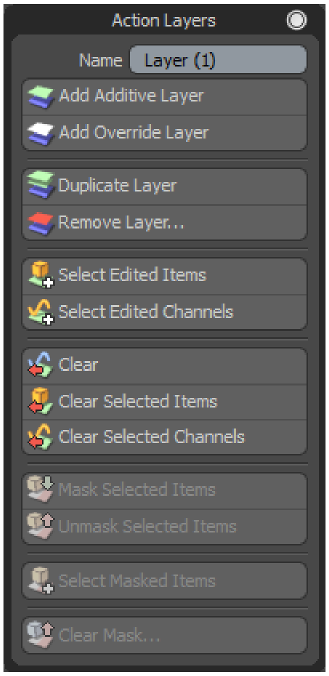
An override layer replaces the channel values in the active Action with the channel's setup value. In other words, an override layer resets an animation to its rest state without destroying the original animation. Override layers sit on top of any existing layers so you can create whole new animations within them.

Override layers are designed so you can add masks to individual sections of an actor, which allows you to create alternative animations of that actor. Because the new animation is held in an override layer, your original animation remains unaffected.

To create an override layer:

1. Select Action Layers in the left panel.

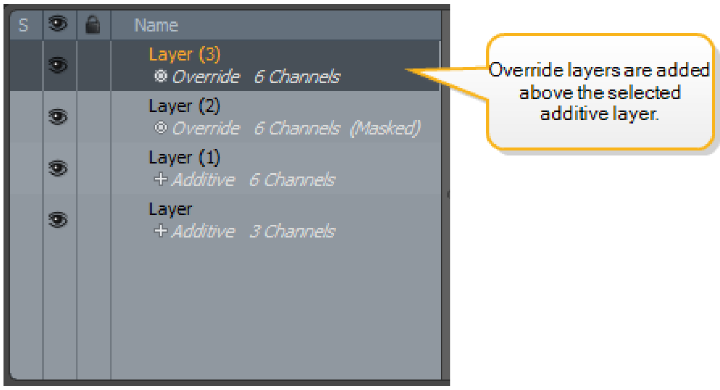
   The **Action Layers** panel opens.


Action Layers
Name
Layer (1)
Add Additive Layer
Add Override Layer
Duplicate Layer
Remove Layer...
Select Edited Items
Select Edited Channels
Clear
Clear Selected Items
Clear Selected Channels
Mask Selected Items
Unmask Selected Items
Select Masked Items
Clear Mask...

Tip: If you prefer to work in the **Animate** layout, you can access the Action Layers tools by clicking the Action Layers button on the **Graph Editor**.

2. Click **Add Override Layer**.

Your new override layer is added directly above whichever layer is selected in the **Action Layers** palette.



Override layers are added above the selected additive layer.

Using Masks with Override Layers

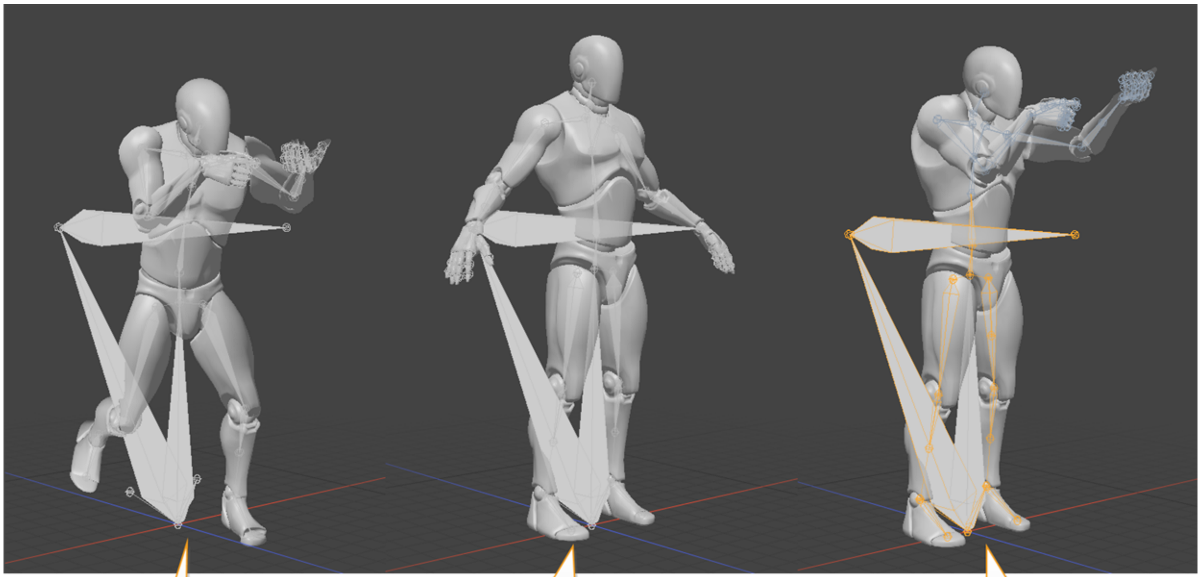
Let's imagine you have a character model of a space marine. In your original animation, he is running with his rifle aimed. Perhaps you want to change his motion so instead of running he is advancing slowly. You can use an override layer to effect this change while retaining the underlying animation.

1. Create an override layer.

   Your character model resets to its setup position.

2. Select the parts of your character model you want to animate separately. For this example, it would be the legs and lower body.
3. Click on the **Masks** button to bring up the mask menu.
4. Select **Mask Selected Items**.
5. As a test, click the **Play** button below the graph editor to play through your animation. Notice that the original animation is retained except the selected items are now static.
6. Create new keyframes to animate the selected items. In this example, you would change the motion of the legs.

   Any changes you make now only affect the selected item and your original animation remains underneath. However, you can select parts of the masked items and expose the original animation underneath by clicking on the **Masks** button and selecting **Unmask Selected Items**. You may do this if you want to see your original foot animation along with the new animation you've created for the legs.



Base animation.

With an override layer applied, all channels in the action reset to their setup values.

Lower half of the body selected with the override layer masking only the selected joints. This makes the underlying animation of the arms reappear.

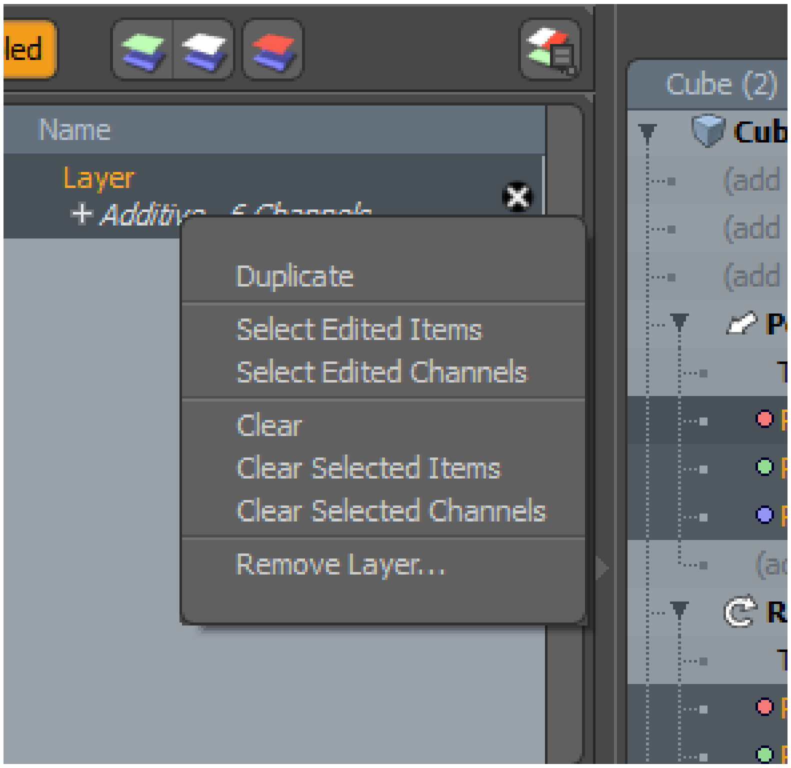
Selecting Edited Items and Channels

Items that have edited channels in a layer can be selected by right-clicking on the layer and choosing **Select Edited Items**. This automatically selects any channels within that item that have been edited. To select edited channels along with the items they belong to, right-click on the layer and choose **Select Edited Channels**.

Clearing Edited Channels from Layers

To remove edits within a layer:

1. Select the item for which you want clear channels.
2. In the Channels panel, select the channels you want to clear. Hold **Ctrl**/**Cmd** and click to select multiple channels.
3. Right-click on the layer in the layers palette and select **Clear Selected Channels** from the contextual menu. To remove all channels belonging to particular items, select the items and choose **Clear Selected Items**.


led
Name
Layer
+ Additive 6 Channels
Duplicate
Select Edited Items
Select Edited Channels
Clear
Clear Selected Items
Clear Selected Channels
Remove Layer...
Cube (2)

Channels can also be cleared from a layer from the Channel List viewport. Right click on the channels and select **Remove Current Values.**

Baking Actions with Layers

Animation Layers can be baked down into the base Action by selecting the item and clicking **Animate** > **Bake**. Channels belonging to the items that have been baked are removed from all non-muted layers.

Exporting Actions with Layers

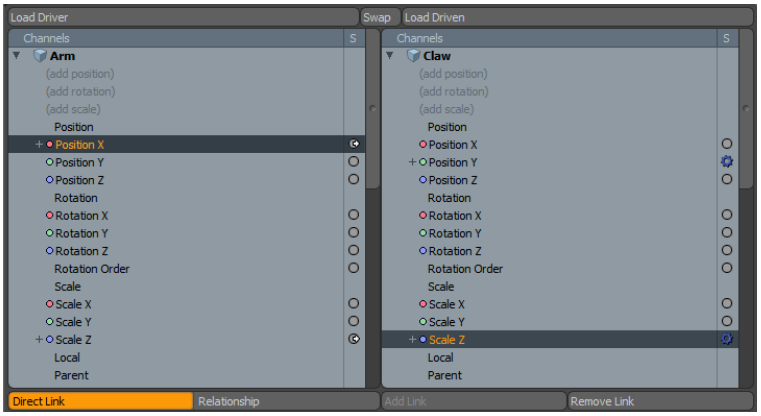
Animation Layers can be exported with an Action without the prerequisite that the layers have been baked. Only non-muted layers are taken into account during export. This makes it easy to export multiple variations of an Action by muting and unmuting layers before each export. See FBX Export for more information.

Note: When an Action is duplicated, any layers on the Action are also be duplicated.

Using Channels

With the **Channel Links** viewport you can define and manage linked channels. In Modo, channel links are a way for the property of one item to control the property of another.

If you select two items before opening the viewport, Modo automatically populates the **Driver**/**Driven** channels. The selection order is important because the first item selected becomes the driven channel, and the second item becomes the driver. You can also load an item into a specific column by selecting the item in any 3D viewport (or the **Item** list) and clicking the **Load Driver** or **Load Driven** button at the top of the **Channel Links** viewport. Clicking the **Swap** button switches the driven item with the driver.


Load Driver
Swap
Load Driven
Channels
S
Arm
(add position)
(add rotation)
(add scale)
Position
Position X
Position Y
Position Z
Rotation
Rotation X
Rotation Y
Rotation Z
Rotation Order
Scale
Scale X
Scale Y
Scale Z
Local
Parent
Channels
S
Claw
(add position)
(add rotation)
(add scale)
Position
Position X
Position Y
Position Z
Rotation
Rotation X
Rotation Y
Rotation Z
Rotation Order
Scale
Scale X
Scale Y
Scale Z
Local
Parent
Direct Link
Relationship
Add Link
Remove Link

To define a link between two channels, select any two channels (one from each column), specify the type of link to define (**Direct Link** or **Relationship**), and click the **Add Link** button. The **S** (Status) column updates to reflect the link. The driver channel displays a tiny arrow over the keyframe indicator dot ◉ to indicate its controlling state. The driven channel displays a small gear ⚙ to indicate that it is being controlled by another channel. To remove a link, select any driven or driver channel and click **Remove Link**.

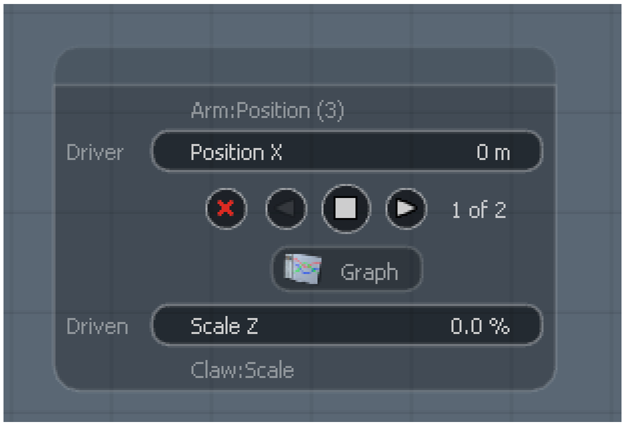
Direct Link

When you define a **Direct Link**, Modo uses the value of the driving channel to directly control the value of the driven channel. For example, if the **Position X** of the locator drives the **Scale Z** of the Mesh Item, positioning the locator item at 1 meter on the X axis scales the Z axis on the Mesh Item to 100%. At this position, it appears to do nothing, but if you move the X position of the locator to 2 meters, the Mesh Item scales to 200%. Once you specify the link, you only need to keyframe the driving item.

Relationship

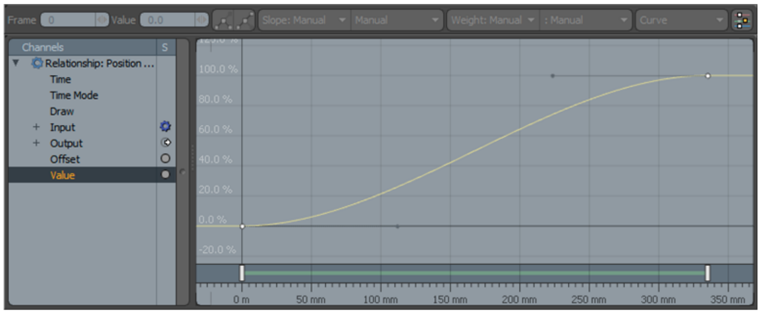
A **Relationship** is similar to a **Direct Link**; however, the link itself can be non-linear and defined by a graph. With a **Direct Link**, any modification to the driver instantly affects the driven item, but once you link a **Relationship**, you must define the relationship, itself, numerically in the **Relationship** interface window that appears in the 3D viewport.

> **Note:** If you don't see the **Relationship** interface window right away, you may need to select the Relationship Modifier item in the **Items** list under the scene item. If it is not initially visible, click the **+** to expand the item.



The **Relationship** interface window provides all the necessary information about the channel relationship. Modo lists the **Driver** channel on top; it lists the **Driven** channel on the bottom. The linked channel names appear in the darker area. Clicking and then dragging over the channel's name adjusts the value interactively. By clicking the value itself you can type a value. You can click the four round buttons in the middle to define the value relationship with set values called keys. The red **X** deletes a key, and the white square creates a key. Click the two small arrows to jump between keyframes.

The graphic shows the **X Position** channel linked to the **Scale Z** channel. If you set a value keyframe at 0 meters for the **X Position** and 0% for **Scale Z**, when the driver is at 0 meters, there is no scale change on the Z axis for the Mesh Item. Change the scale value to 100% and the **Position X** value to 10 meters, and then click the white square to set a keyframe. That effectively sets the relationship. Now, when the first item moves to 10 meters in the X direction, the Z scale of the Mesh Item automatically scales 100%. To visually see the relationship curve, click the **Graph** button to open the **Graph Editor**.



You can see that Modo places the generated keyframes into the **Value** channel of the relationship modifier item. The driver values appear along the length and the driven values, the height. The yellow line graphs the relationship between the two channels and smoothly interpolates between values. Modifying the slope and weight of the keyframes modifies the in-between interpolation values. For more on modifying graph editor keyframes, see the Graph Editor topic. For more information on the Relationship modifier and its usage, see Channel Modifiers.

Add User Channels

User-created channels are useful when animating. You may want to create controllers for an animation rig. (A rig takes a complex action and creates a simplified control for it.) By having user channels, you can add additional channels to an item (such as to a locator).

For example, if a machine had several gears that needed to synchronize with each other, you could add a user channel, called gear spin, to the locator. Then you could link the channel to all the gears in such a way that the gears would spin in proper relation to one another. You could animate the single channel of the locator to simplify the overall process and make it easy to find the item that is needed to control the gears.

There are a several ways you can add custom channels to an item.

- You can drag and drop channels onto mesh items in the viewport or on mesh items in the **Items** list to turn them into users channels. For more information, see Drag and Drop Workflow.
- With the target item selected in the **Items** list, use **Animate > Channels > Add User Channel**.
- In the **Setup** layout, open the **Modifiers** sub-tab of the toolbox, and click **Add User Channel**.
- In the Channels Viewport, select an mesh item in the list, and click **Add Channel to Schematic**.

Drag and Drop Workflow

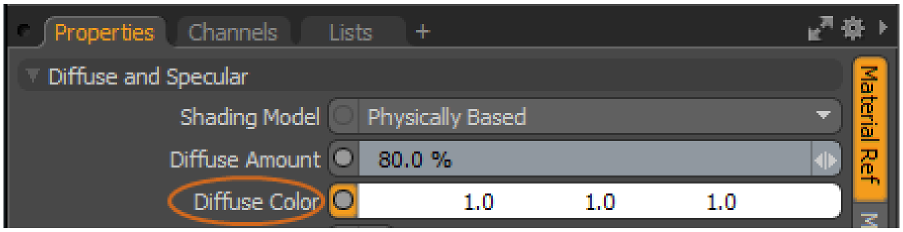
In the following example, we will demonstrate creating a user channel and changing the color of a mesh item.

To drag-and-drop a user channel:

1. Add a **Sphere** to your scene. In **Model** layout, on the **Basics** tab in the Toolbar, hold **Shift** and click **Sphere**.

   The Sphere displays in the center of the scene.
2. Open the **Shading** tab and select the **Base Material** item.
3. On the right panel in the **Properties** panel, open the **Material Ref** tab.
4. Click on the text **Diffuse Color**.

   The channel state is activated and the button turns orange.



Diffuse and Specular

Shading Model: Physically Based

Diffuse Amount: 80.0 %

Diffuse Color: 1.0 1.0 1.0

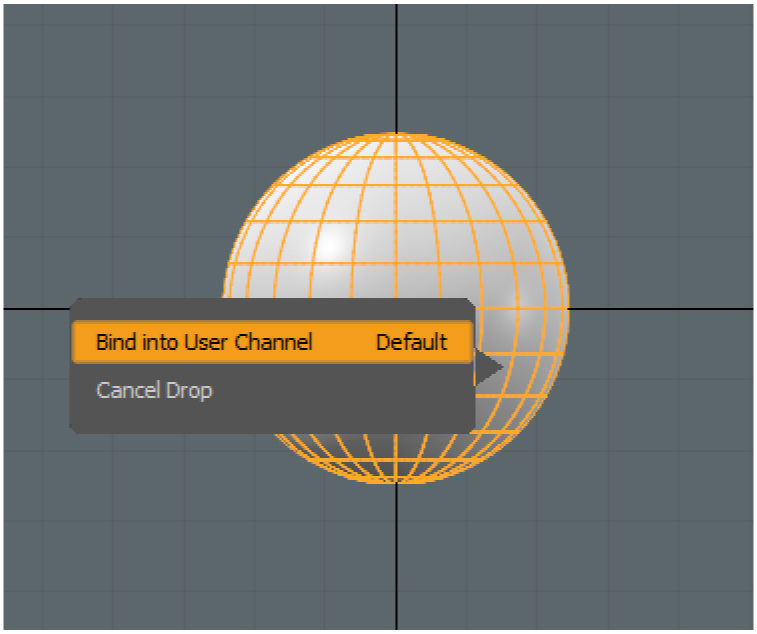
5. Click-and-drag the **Diffuse Color** text onto the **Sphere** in the viewport.

Tip: Alternatively, you can drag the **Diffuse Color** text onto a mesh item in your **Items** list.

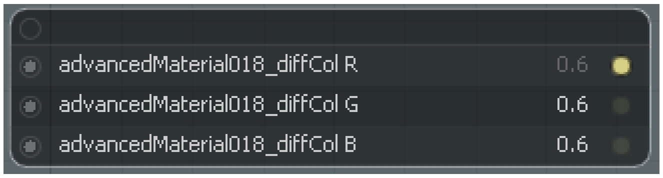
   A menu is displayed.

6. Click **Bind into User Channel**.

   This is the default. To cancel, select **Cancel Drop**.



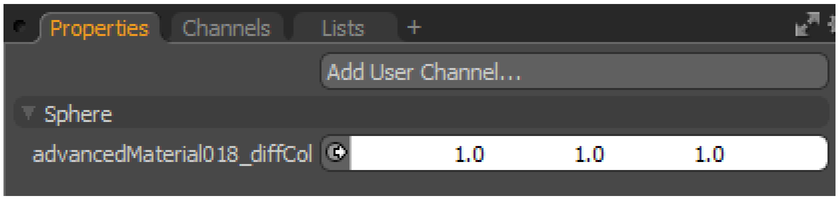
7. RGB dialog displays in the viewport. Click on a value to change the color.



advancedMaterial018_diffCol R 0.6
advancedMaterial018_diffCol G 0.6
advancedMaterial018_diffCol B 0.6

Alternatively, on the right panel, open the **User Channels** tab.

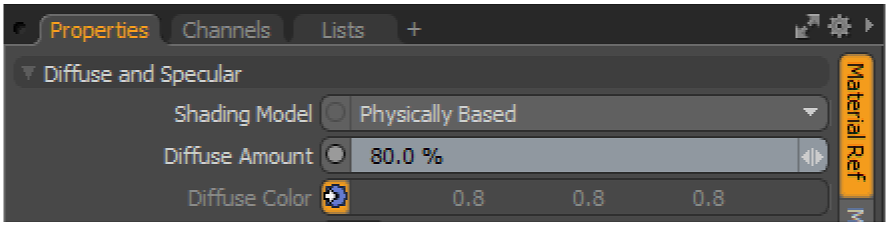
An advanced material is listed.


Properties
Channels
Lists
+
Add User Channel...
Sphere

advancedMaterial018_diffCol 1.0 1.0 1.0

8. Click on the color field to open the Color Picker, select a new color, and press **Esc** to close the window.
9. Open the **Properties** tab and **Material Ref** tab.

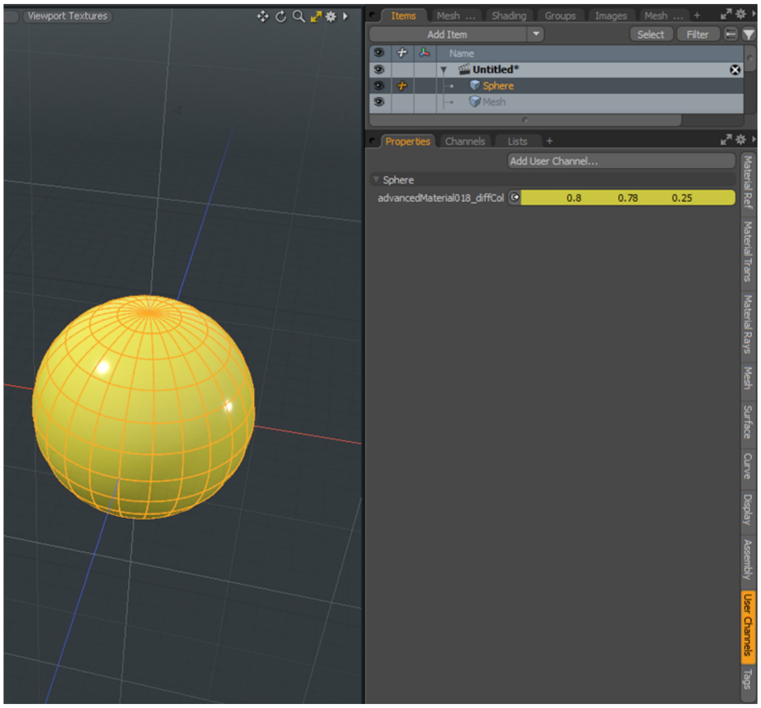
   Notice, the Diffused Color channel is disabled and button color is updated to blue (Driven). This channel is driven by a Modifier and cannot be edited. It can only be edited in the **User Channels** tab.


Properties
Channels
Lists
+
Diffuse and Specular
Material Ref

Shading Model Physically Based
Diffuse Amount 80.0 %
Diffuse Color 0.8 0.8 0.8

10. Open the **User Channels** tab and change the color.

    The 3D Viewport displays the updated color on the mesh.

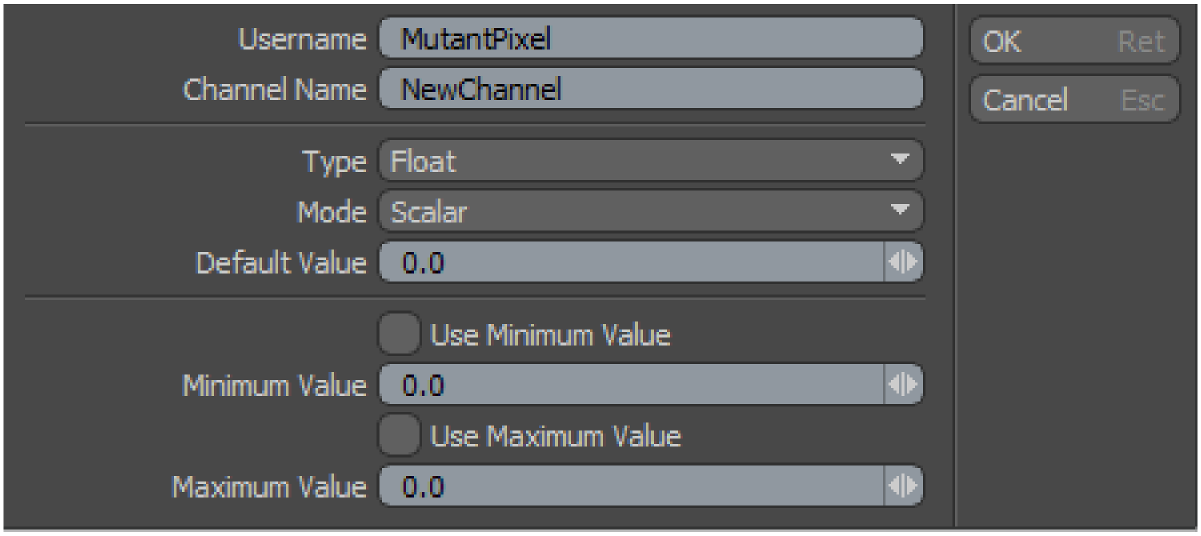

Viewport Textures
Items
Mesh ...
Shading
Groups
Images
Mesh ...
Add Item
Select
Filter
Name
Untitled*
Sphere
Mesh
Properties
Channels
Lists
Add User Channel...
Sphere
Material Ref
Material Trans
Material Rays
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

advancedMaterial018_diffCol 0.8 0.78 0.25

Tip: Best practices: Give your user channel a meaningful name. Right-click on the user channel and select **Edit User Channel**. In the **Name** field, type a new name.

Create User Channel Options

This following options are available:


OK Ret
Cancel Esc

Username MutantPixel
Channel Name NewChannel
Type Float
Mode Scalar
Default Value 0.0
☐ Use Minimum Value
Minimum Value 0.0
☐ Use Maximum Value
Maximum Value 0.0

Option	Description
Username	This is an optional identifier when creating channel sets. If you do not define a username, Modo displays the internal name.
Channel Name	Defines a name that identifies the new channel.
Type	Channels are places to store values. Depending on their use, the values in a channel determine how they're used. For example, an **Angle** channel is appropriate for rotations and a **Distance** channel is necessary for transform info. The channel **Type** defines what format the values take. The available options are: • **Float** - Produces a floating point value (a decimal typically between 0 and 1). • **Distance** - Produces a distance output value. • **Angle** - Produces an angle output value in degrees. • **Percentage** - Produces a percentage value (0% - 100%). • **Light Intensity** - Produces standard units of Watts per volumetric meter. • **Time** - Produces time values. (The default is in frames.) • **Speed** - Produces speed values in meters per second (m/s). • **Force** - Produces force values in newtons (N). • **Mass** - Produces mass values in kilograms. • **Acceleration** - Produces acceleration values in meters per second squared. • **Pixel (integer)** - Produces whole pixel values. • **Pixel (fractional)** - Produces sub-pixel values that are fractions of pixels. • **Integer** - Produces a whole number value (not a fraction, for example, 1,2, or 3). • **Boolean** - Produces only a True or False value. • **String** - Stores multiple values as a string of characters. • **Matrix** - Stores multiple values as a matrix. • **Quaternion** - Stores a single rotation around a specific axis in 3D space. • **Divider** - Creates a named, collapsible divider in the **Properties** panel. It is useful as an organizational device when dealing with multiple channels. These can be moved and renamed but have no value. • **Axis** - Creates an Axis channel. • **Focal Length** - Creates a Focal Length channel. • **Int Range** - Creates an integer range channel. • **Pattern** - Creates a pattern channel, which takes the values 'True' and 'False'.

Option	Description
	• **Gradient** - Creates a gradient channel.
Mode	By default, Modo creates a single channel with the given name, but you can also have Modo create multiple channels by using the **Mode** options: • **Scalar**- Creates a single channel (such as MyChannel). • **Vector XY**- Creates two channels (such as MyChannel.X and MyChannel.Y). • **Vector XYZ**- Creates 3 channels (such as MyChannel.X, MyChannel.Y, and MyChannel.Z). • **Vector RGB**- Creates 3 channels (such as MyChannel.R, MyChannel.G, and MyChannel.B). • **Vector RGBA**- Creates 4 channels (such as MyChannel.R, MyChannel.G, MyChannel.B and MyChannel.A). • **Vector UV** - Creates U and V channels. • **Vector UVW** - Creates U, V, and W channels.
Default Value	Use to specify the default value for the channel.
Use Minimum Value	Use to specify a minimum value.
Use Maximum Value	Use to specify a maximum value.

Modifying Gradient Channels

Modo allows you to dynamically generate and modify gradients, rig their values and vary them over time. They enable you to carry out simple tasks, such as generating a new gradient from a waveform, as well as complex ones, like introducing noise or offsetting the value of a gradient.

Gradient Modifiers and Gradient Layers

Depending on whether you want to generate a new gradient from source gradients, or apply an effect on an existing gradient, you can use **Gradient Modifiers** or **Gradient Layers**.

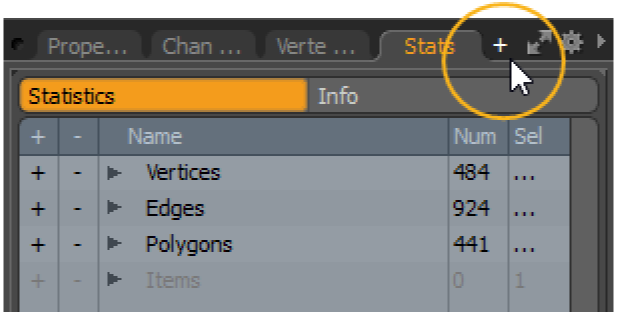
You can use **Gradient Modifiers** to read an incoming gradient, perform an operation on them and output a new gradient. They allow you to blend gradients, generate a gradient from a mesh layer, perform mathematical operations on gradients, and so on. For more information, see Gradient Modifiers.

Similarly to Gradient Modifiers, **Gradient Layers** allow you to modify a gradient. However, instead of using a series of channel links to compute a new gradient, you can modify a gradient channel in place, by evaluating a stack of layers from the bottom up to modify the gradient curve. It works similarly to the **Mesh Operations** list and the Shader Tree. The base gradient is evaluated, and the value is passed up the stack to be modified by subsequent layers, until the final computed value is output from the channel. Gradient Layers can be added to a gradient channel using the **Gradient Layers** viewport.

Accessing the Gradient Layers Viewport

To open the **Gradient Layers** viewport you can add a new Data List tab to the right panel, or open a new window.

To open a tab on the upper or lower right panel, click the + sign to the right of the tab names, and in the dropdown, click **Data Lists** > **Gradient Layers**.



+	-	Name	Num	Sel
+	-	Vertices	484	...
+	-	Edges	924	...
+	-	Polygons	441	...
+	-	Items	0	1

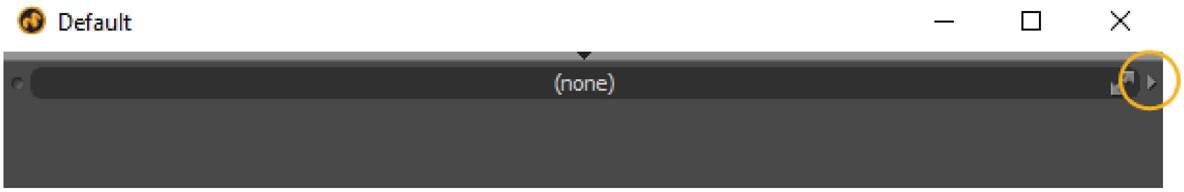
Note: If your right panel is narrow, a triangle is displayed instead of a + sign. In that case, click the triangle, and in the dropdown, click **New Tab** > **Data Lists** > **Gradient Layers**.

To open the **Gradient Layers** viewport in a new window:

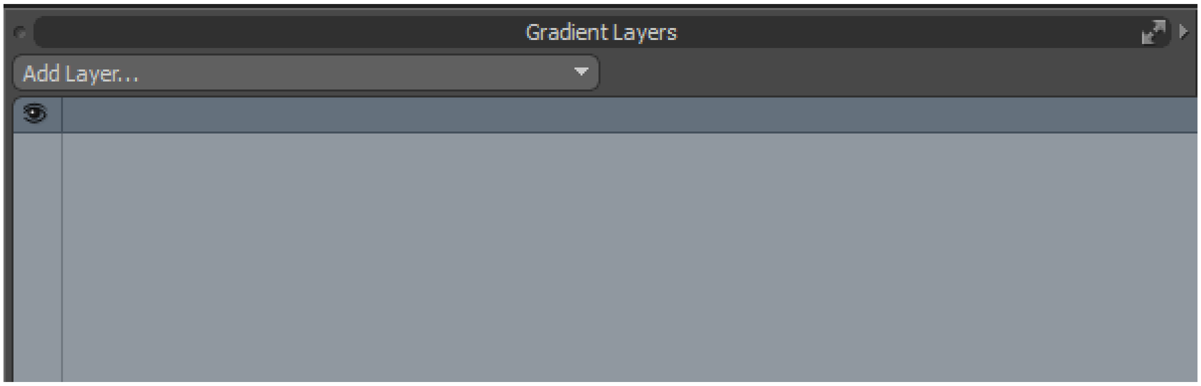
1. In the menu bar, click **Layout** > **Windows** > **New Window** > **Data Lists**.

   This opens a new window.

2. Click the triangle in the top right corner and select **Data Lists** > **Gradient Layers**.

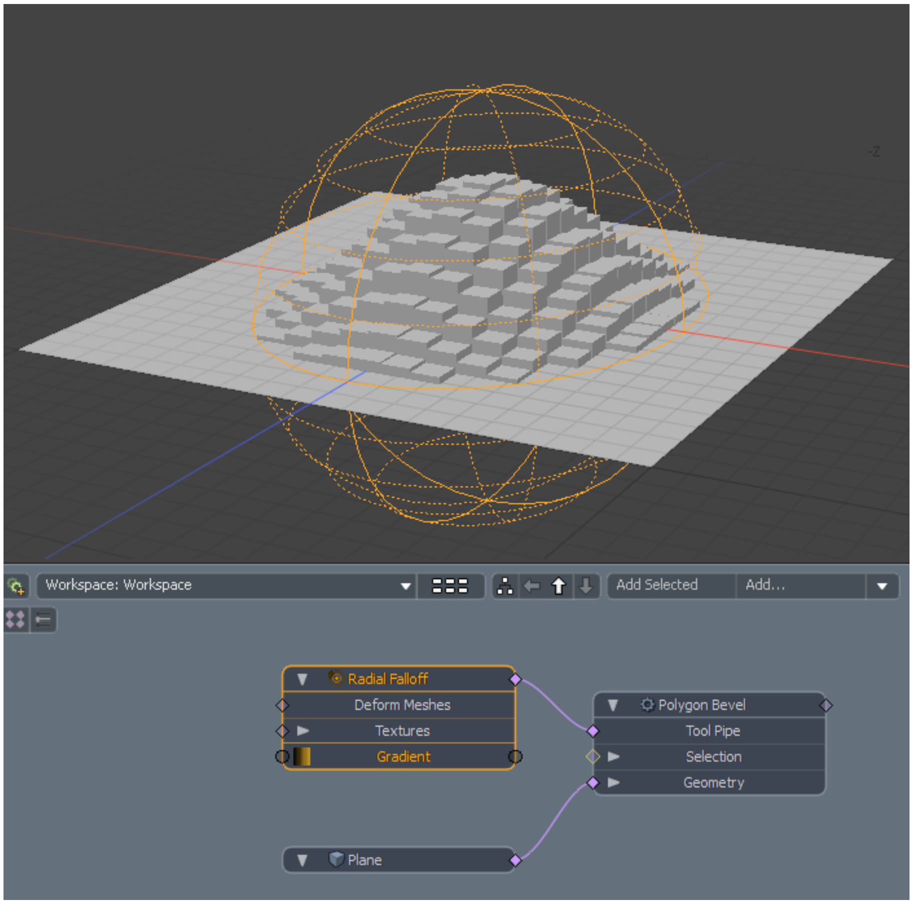


This opens the **Gradient Layers** viewport.

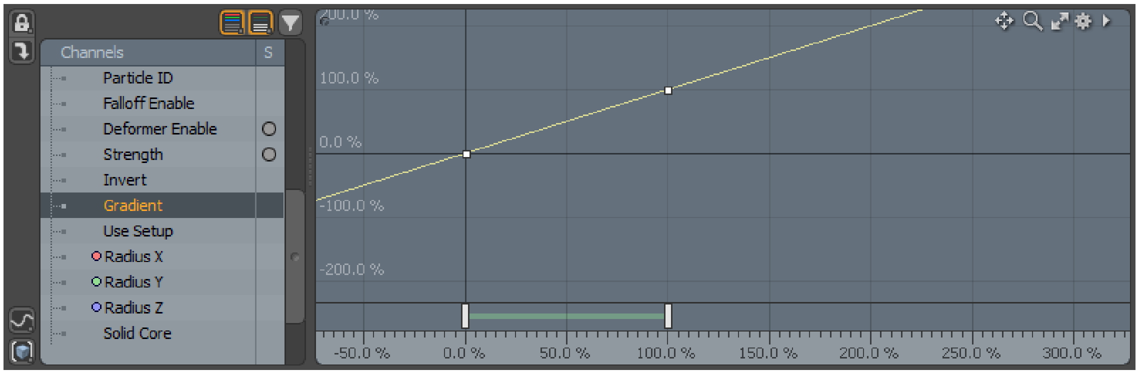

Gradient Layers
Add Layer...

Using Gradient Layers

1. Select a Gradient channel you want to modify. In the following example, we have a Polygon Bevel applied to a plane, modulated by a Radial Falloff. We're using the Radial Falloff's Gradient channel.

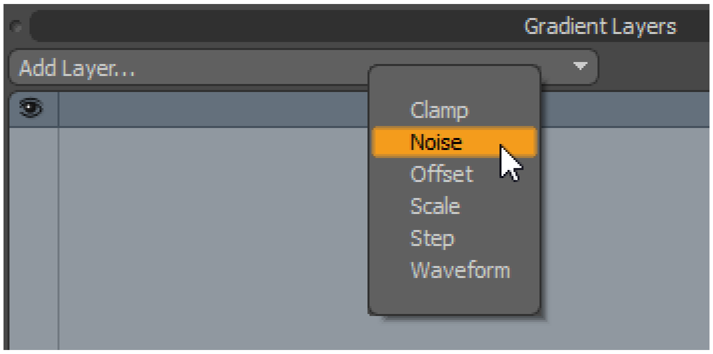

Workspace: Workspace
Add Selected
Add...
Radial Falloff
Deform Meshes
Textures
Gradient
Polygon Bevel
Tool Pipe
Selection
Geometry
Plane

You can double-click the **Gradient** channel on the node to open it in the Gradient Editor:


Channels
S
Particle ID
Falloff Enable
Deformer Enable
Strength
Invert
Gradient
Use Setup
Radius X
Radius Y
Radius Z
Solid Core
200.0 %
100.0 %
0.0 %
-100.0 %
-200.0 %
-50.0 %
0.0 %
50.0 %
100.0 %
150.0 %
200.0 %
250.0 %
300.0 %

2. In the **Gradient Layers** viewport, click **Add Layer** and select the Gradient Layer you want to use, for example, **Noise**.

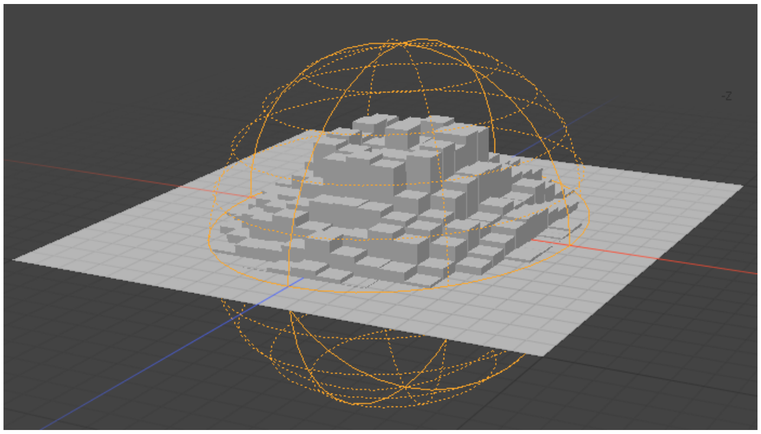
Note: For more information on the different gradient layers, see Gradient Layer Types.


Gradient Layers
Add Layer...
Clamp
Noise
Offset
Scale
Step
Waveform

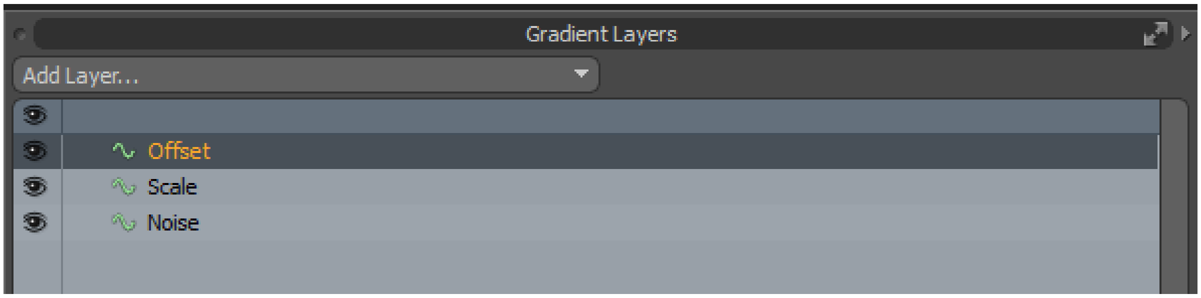
Your source gradient is modified by the new layer.

3. To edit the Gradient Layer's properties, select it in the stack and open the **Properties** panel on the lower right. Adjust the properties as needed.

The result of the Noise layer applied with the **Amplitude** set to 7.0 looks like this:



Adding multiple layers on top of each other and reordering them by dragging them around in the viewport allows you to produce a wide range of effects.



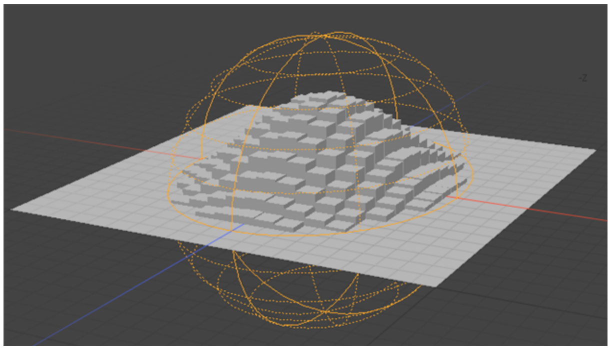
To hide a layer, toggle the Visibility 👁 button on the left. To delete a layer, select it and press **Delete** on the keyboard.

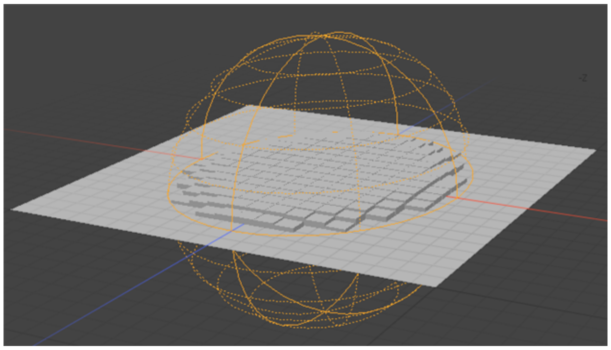
Gradient Layer Types

The following Gradient Layers are available.

Clamp

Gradient Clamp clamps the gradient output value, ensuring that the value never exceeds the specified minimum and maximum range. You can edit the minimum and maximum values in the Gradient Layer's properties.



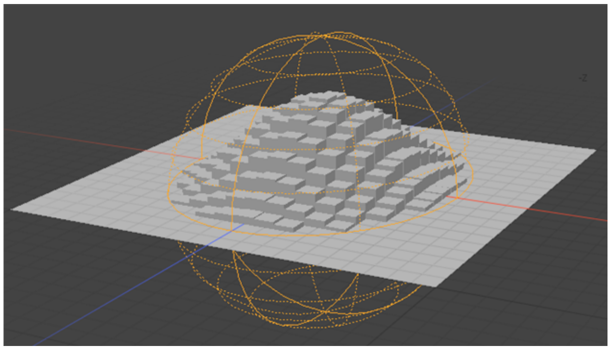


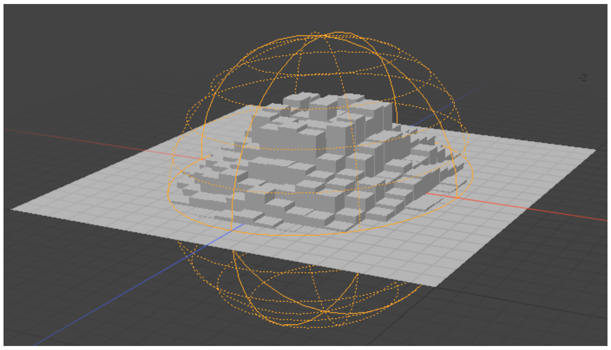
Gradient Clamp Layer Properties

Minimum Value	The lower end of the clamped range.
Maximum Value	The upper end of the clamped range.

Noise

The Noise layer applies Perlin Noise to an existing gradient. You can control both noise amplitude and frequency in the layer's properties.



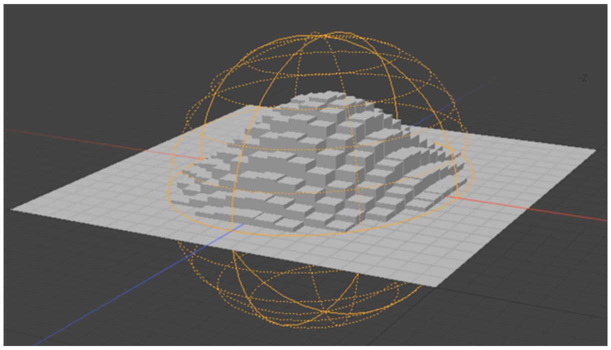


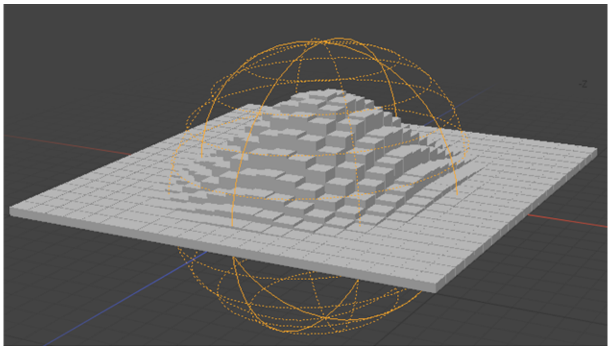
Gradient Noise Layer Properties

Frequency	Determines how regularly the noise occurs per second.
Amplitude	Controls how strong the noise offsets the original base gradient.
Seed	Seeds the randomness of the noise generator. Animating this value animates the noise.

Offset

The Offset layer offsets all the values in a gradient equally, either by offsetting the value in Y, and/or the time the value is sampled in X.



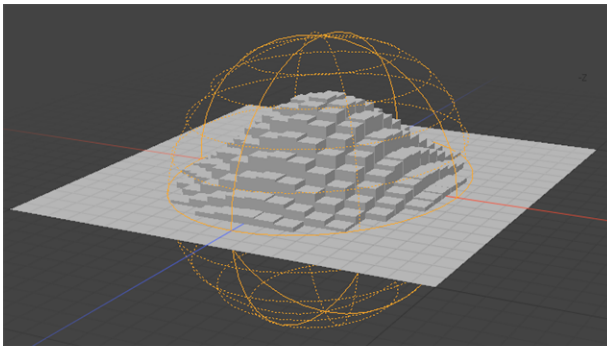


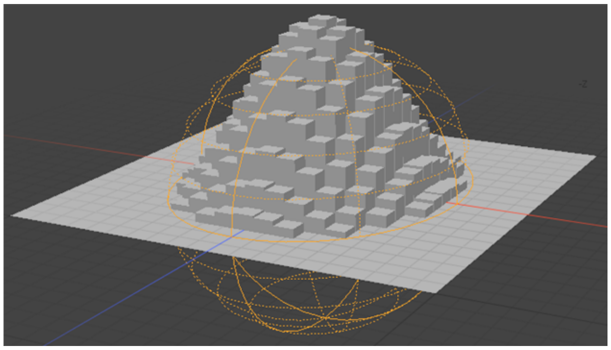
Gradient Offset Layer Properties

Offset X	Offsets the gradient value in X. This offsets the sample time, so reading the gradient at a specific time sample remaps the time sample to read the input gradient at a different time.
Offset Y	Remaps the gradient value, adding the Offset Y value to the input gradient sampled value.

Scale

The Scale layer can be used to scale a gradient either in the Y axis, modifying the gradient's value, or the X axis, modifying the time of each value. You can also control the pivot point for each axis the scaling is performed in.



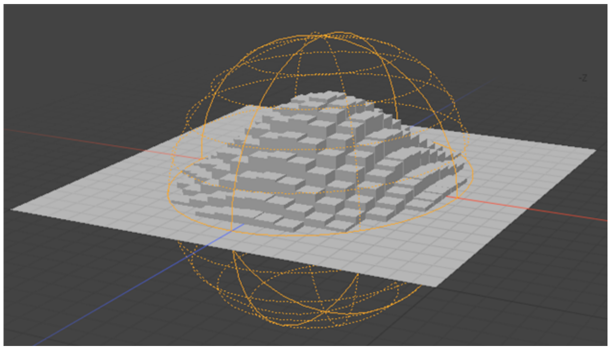


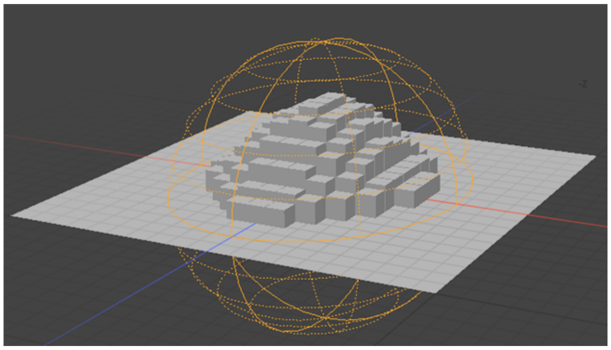
Gradient Scale Layer Properties

Amount X/Y	The amount the input gradient is scaled in the X/Y axis. Changing X modifies the sample time of each value in the gradient. Changing Y increases or decreases the gradient values.
Center X/Y	The pivot point to perform the scale in the X/Y axis. The gradient is scaled from this position.

Step

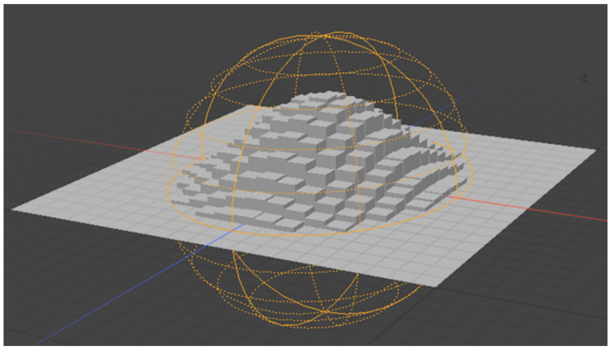
The Step layer rounds values in a floating point gradient to their nearest whole number. For example, 0.25 becomes 0.0, and 0.6 becomes 1.0. This can introduce a step pattern to gradients that cover a large range of floating point values.

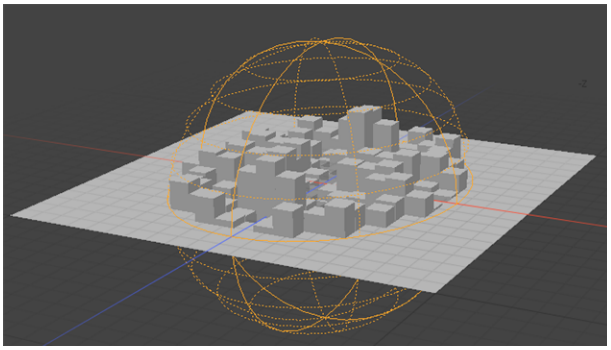




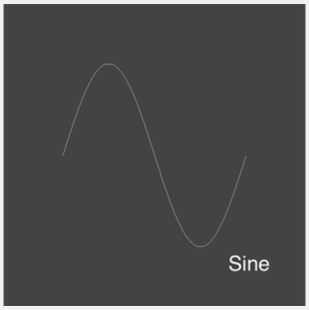
Waveform

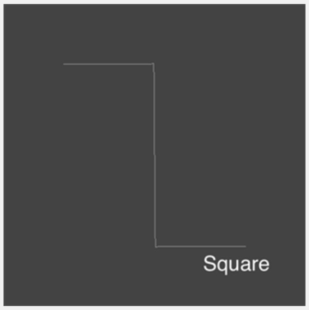
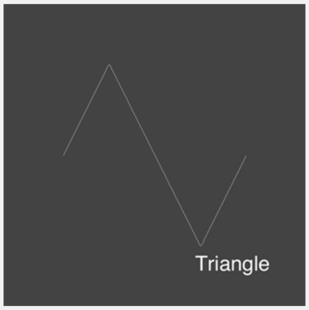
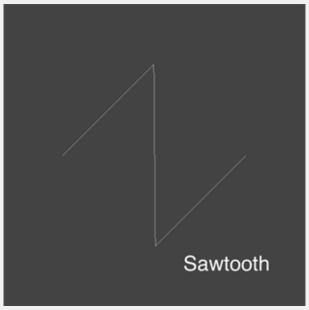
The Waveform layer generates a gradient from four standard waveform presets: Sine, Square, Triangle, and Sawtooth. You can use this to create regular repeated gradients.





Gradient Waveform Layer Properties

Type	The type of waveform to generate:  A sinusoidal waveform that describes a smooth periodic oscillation from one extreme of the amplitude to the other.

	 A square wavefrom that rounds to the nearest amplitude value.  A triangle waveform that blends in linearly from one extreme of the amplitude to the other.  A sawtooth waveform that blends linearly to the positive extreme of the amplitude, and then snaps back to the negative extreme of the amplitude.
Amplitude	The highest and lowest peak of the waveform. A value of 1.0 produces a low of -1.0, and a high of 1.0.
Frequency	Determines how regularly the oscillating motion occurs per second. A value of 1.0 produces one up and down cycle per 1.0 time step.
Offset	Offsets the gradient output value in the Y axis.

Using Modifiers & Constraints

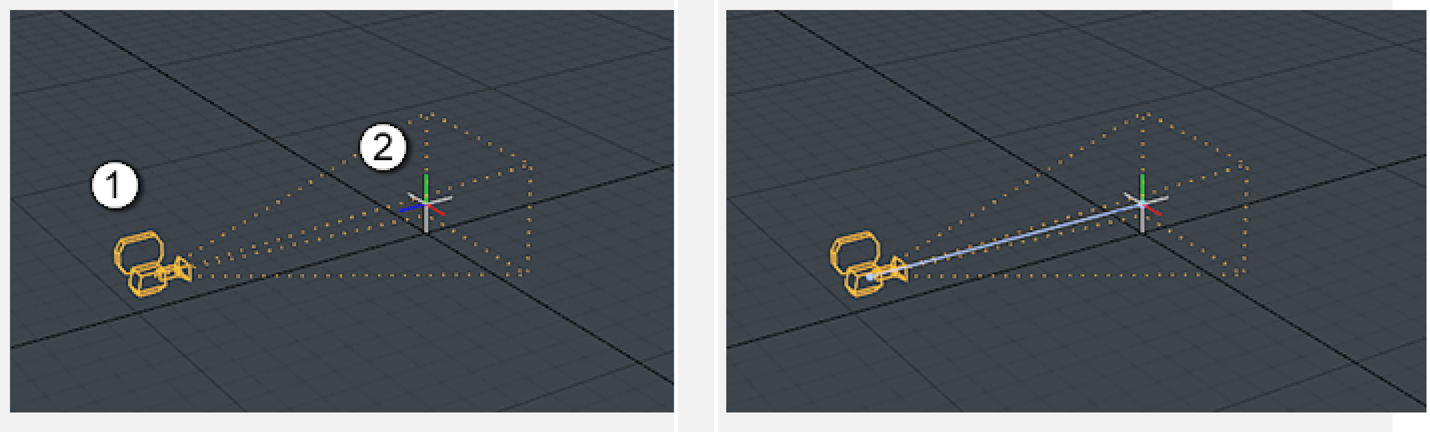
Constraints, at their most basic, are controllers that regulate how one item relates to another. Constraints can serve many purposes and, when used in combination with other constraints, can create very complex controllers and motions.

Modo's constraints are weighted in that a weight value can determine the influence a constraint has over the item it's controlling. Once you apply a constraint, you can select it by clicking the small **+** sign that appears next to the constrained item in the Items (Scenes) List. Once you select a constraint, you can edit its attributes in the Properties Panel or within the Channels Viewport. You can add individual constraint items to a scene by using menus in the **Schematic** viewport and other locations, but only by using the buttons found in the **Setup** interfaces toolbox in the **Modifiers** sub-tab can you have Modo automatically connect the selected elements when assigned.

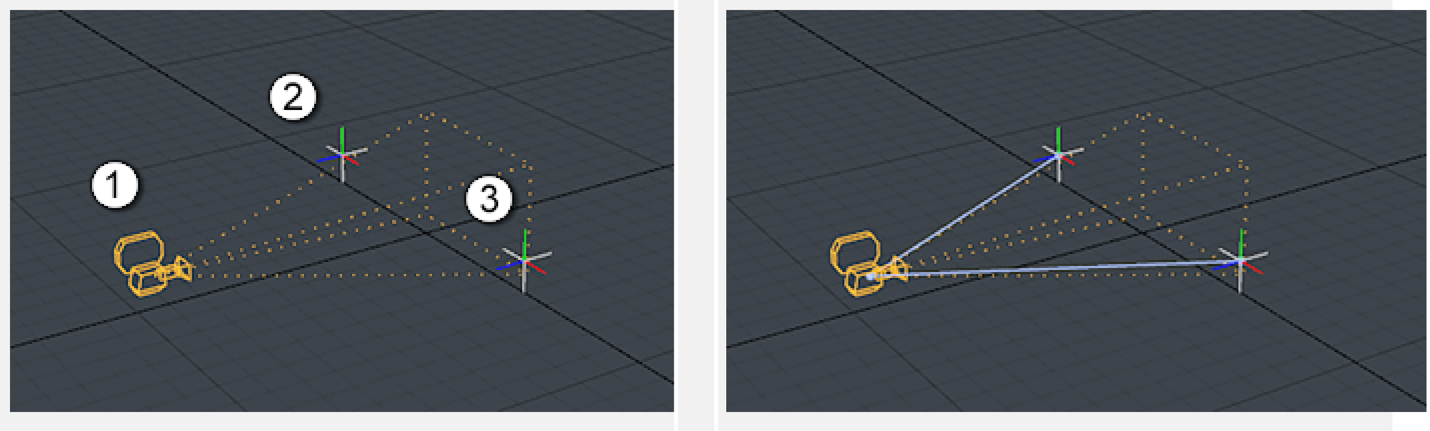
Automatically Assigning Constraints

There are three basic methods for applying constraints:

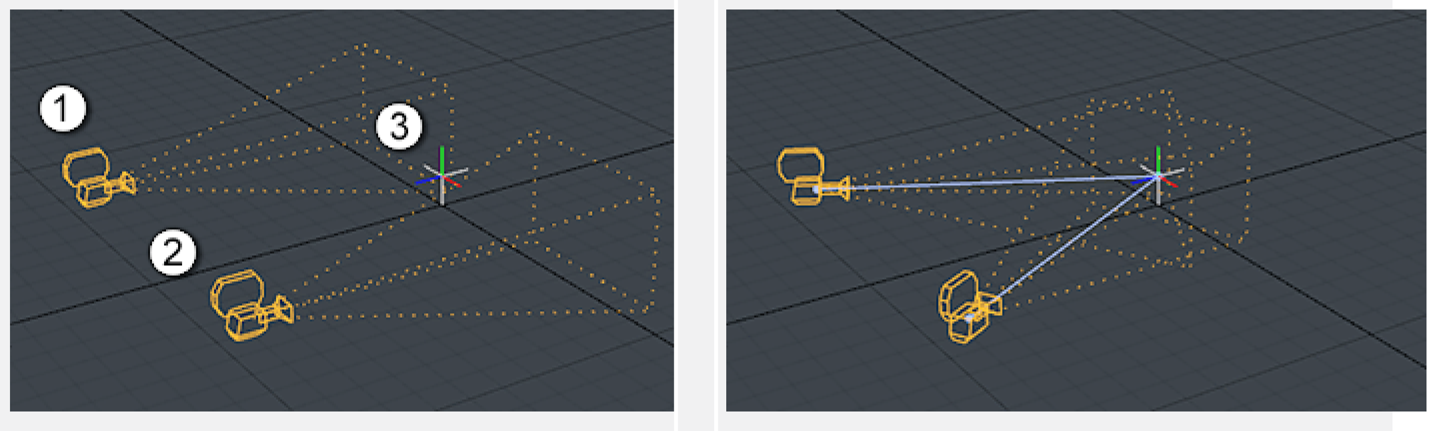
- **Default method, creates the appropriate constraint item(s) from item selections** - The simplest constraint is between two items. In the example, the camera was selected first and then the locator item. (Once you select an item, press the **Shift** key as you select subsequent items.) Now, click the **Direction** constraint button to apply the constraint. Once applied, the locator controls the direction of the camera: wherever the locator moves, the camera points directly at it.



- **Multiple items control another item** - In this example, the camera was selected first and then the two locators. Click the **Direction** constraint button to apply the constraint. Modo uses the position of both locators to determine the orientation of the camera. In this case, if you aim the camera towards a point half-way between the two locators to average the two positions. You can control the influence of one item over the other by adjusting the weight amount. Modo would face the camera more toward the item with the higher weight (influence). You can adjust the weight amount in the **Constraints Properties** viewport or the **Channel** list.



- **Multiple items are controlled by a single item** - In this example, the two camera items were selected in sequence and then the locator item. With this method you apply the constraint by pressing the **Ctrl** key. When you do so, the button indicates **Direction (multiple)** to signify that the one controller affects multiple items. Click the button to apply the constraint. Once applied, you can see that each camera now independently orients toward the one locator.



Constraint Weights

When you constrain an item to multiple items, Modo adds a weight channel to the constrained item: one for each of the constraining items. The weights control how much influence a particular item has on the constrained item. You can adjust weights and animate them for the constraint's effect to be biased towards (or away from) a particular item's input. By setting a weight channel to **100%**, the constraining item exerts full influence on the constrained item. With a value of **0%**, the constraining item has no influence on the constrained item. To access the weight channels for a constraint, select the constraint item in the **Item List**. Modo lists the weight channels at the end of the constraint item's channels in the **Channels** viewport. You can type values here or use the **Channel Haul** tool to interactively adjust the values in the 3D views. (Select the weight channels and press the **C** key or the button in the **Modifiers** tab.)

Manually Assigning Constraints

You can also manually add constraints to a scene by using the menus in the **Item List** and in the **Schematic** viewport. (To open the **Item List**, click the **Add Items** button in the menu bar under **Animate** > **Channel Modifiers** > **Constraint**.) This method of adding the constraint to the scene only adds the element itself and won't have the same effect of connecting the constraint to any selected elements. Once added, you must properly link the element to the constraining item. You do this in the **Schematic** viewport. For more information about adding elements and assigning links between them, see the Schematic viewport topic.

Removing Constraints

You can easily remove constraints from any item. First, select the constrained item. When you press the **Shift** key, the **Constrain** buttons change. For example, the **Direction** constrain button now reads **Remove Direction**. As you press the **Shift** key, click the button to remove the constraint from the selected item. Additionally, you can select the constraint below the constrained item. (If you can't see the constraint, click the **+** sign next to the constrained item.) When you right-click the constraint itself, choose **Delete** in the menu.

Constraints

Constraints, at their most basic, are controllers that control how one item relates to another. Constraints can serve many purposes and, when you use them in combination with other constraints, you can create very complex controllers and motions. Modo's constraints are weighted. A weight value determines the influence a constraint has over the item it's controlling.

Click any link below to view the topic for a particular constraint.

Transform Constraints

Position

Rotation

Scale

Direction

Distance

Geometry Constraint

Vertex Position

Vertex Normal

Edge Position

Edge Normal

Polygon Position

Polygon Normal

Particle Constraint

Particle Constraint: Position

Particle Constraint: Rotation

Path Constraint

Path

Path Position

Path Normal

Intersection

Line 2 Items

Linear Curve

Plane

Surface

UV Constraint

UV Constraint

Camera Constraints

Camera Match

Camera Constraints

Tools

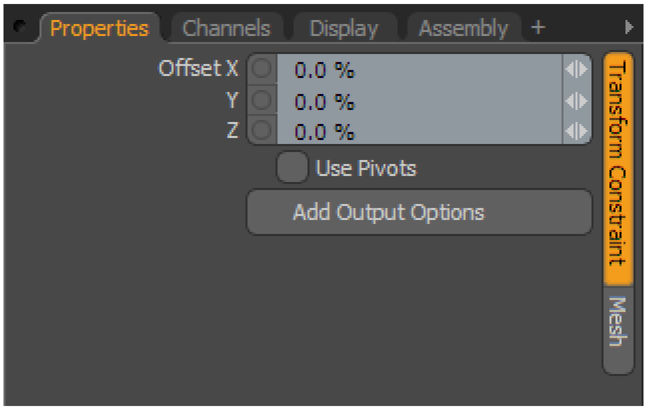
Assign Up Vector

Assign Axis Up Vector

Transform Constraints

Transform Constraint Types

- **Position** - With this constraint, the item follows the World Position of another item or follows the weighted average of the positions of multiple items.
- **Rotation** - With this constraint, the item follows the rotation of another item or follows the weighted average of the rotations of multiple items.
- **Scale** - With this constraint, the item follows the scale of another item or follows the weighted average of the scale of multiple items.



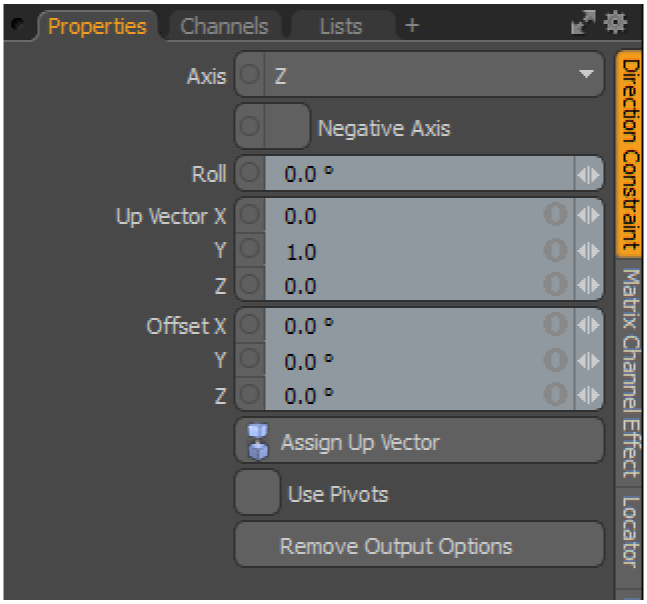
Offset X 0.0 %
Y 0.0 %
Z 0.0 %
☐ Use Pivots

Transform Constraint

- **Offset** - These values define an offset amount that Modo applies after you indicate the constraint: an XYZ value for position, a number of degrees for rotation, and a percentage for scale. If you apply the constraint with **Compensation** enabled, these are set to values that maintain the item's original transform values, but you can also use them to adjust the constrained item's transform, if required.
- **Use Pivots** - Directs Modo to use an item's Pivot position, instead of the default Center position.
- **Add Output Options** - This adds a Channel Effect modifier to each of the constraint's outputs. With these options, you can blend the effect of the constraint with the item's underlying Forward Kinematics or restrict the constraint to one or more axes. See Channel Modifiers for details.

Note: If the item being constrained is parented, then the Channel Effect modifier requires that you add a channel link from the parent's World Position or rotation channel (depending on the type of the constraint) into the Channel Effect's local channel.

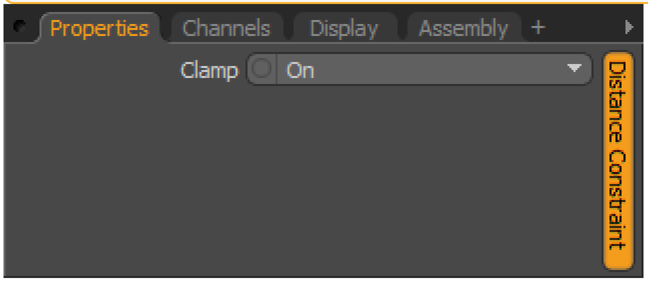
Direction Constraint


Properties
Channels
Lists
+
Axis
Z
Negative Axis
Roll
0.0 °
Up Vector X
0.0
Y
1.0
Z
0.0
Offset X
0.0 °
Y
0.0 °
Z
0.0 °
Assign Up Vector
Use Pivots
Remove Output Options
Direction Constraint
Matrix Channel Effect
Locator

A Direction constraint changes the item's rotation so that it is always pointing toward the World Position of another item.

- **Axis** - Determines the axis that Modo uses to rotate the item for the constrained item to follow its intended target.
- **Negative Axis** - The forward-facing direction is reversed from what is expected. (Modo assumes the positive direction to be forward.) This option flips the orientation to the negative direction of the defined axis.
- **Roll** - Provides some banking for the constrained item to lean to the left or right.
- **Up Vector** - Defines a position to determine the upward direction for the constraint.
- **Offset** - Sets a rotational offset for the item.
- **Use Pivots** - Directs Modo to use an item's Pivot position instead of the default Center position.
- **Assign Up Vector** - This changes the up direction of the constraint (and therefore the item). This is useful if the upward direction needs to change during an animation. You can assign an item as an Up Vector and its current position will represent up for the constraint. With the Constraint item selected in the **Items** list, hold down the **Ctrl** key, select the item intended as the Up Vector target, and click **Assign Up Vector**. Once assigned, any movement of the Up Vector item will influence the **Roll** of the constrained item.
- **Remove Up Vector** - Removes and resets the Up Vector.
- **Add Output Options** - Clicking this button adds a Channel Effect modifier to each of the constraint's outputs. With these, you can blend the effect of the constraint with the item's underlying Forward Kinematics or restrict the constraint to one or more axes. See Channel Modifiers for details.

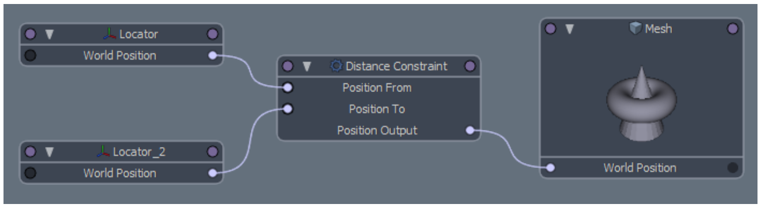
Note: If the item being constrained is parented, then the Channel Effect modifier requires that you add a channel link from the parent's World Position or rotation channel (depending on the type of the constraint) into the Channel Effect's local channel.


Properties
Channels
Display
Assembly
+
Distance Constraint

Clamp On

Distance Constraint

- With a Distance constraint, Modo sets a fixed, minimum or maximum distance away from another object that an item can travel. This constraint is one of the few that has no attributes and needs to be assigned manually in the **Schematic** viewport (see Schematic Viewport). In the Channels viewport, you can adjust the attributes by setting the **Distance** value, which determines how far the constrained item (**Position Output** link), can travel based on the **Clamp** setting.
- **Clamp**: When Clamp is set to **On**, Modo constrains the target items to the **Distance** channel value. When set to **Minimum**, the **Distance** determines how close an item can get to the target. **Maximum** determines how far from the target it can travel.


Locator
World Position
Locator_2
World Position
Distance Constraint
Position From
Position To
Position Output
Mesh
World Position

Geometry Constraints

Geometry Constraints Types

Vertex Position: This constraint attaches an item's center to a deformed vertex position of a Mesh Item.

Vertex Normal: This constraint matches an item's orientation to that of a specified vertex's normal direction. A vertex derives its normal direction as an average of all adjoining polygons.

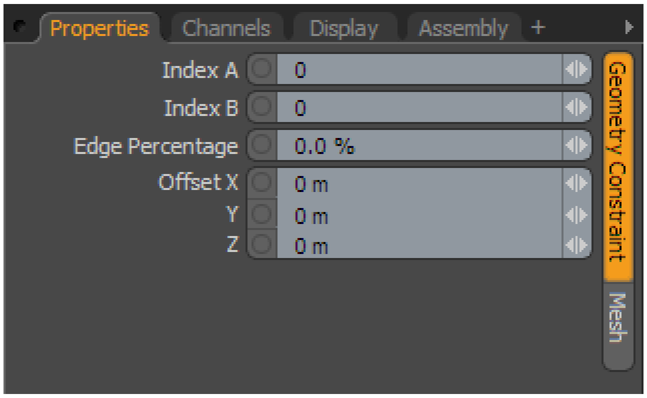
Edge Position: This constraint attaches an item's center to a deformed edge of a Mesh Item.

Edge Normal: This constraint matches an item's orientation to that of a specified edge's normal direction. An edge derives its normal direction from an average of its two bordering polygons.

Polygon Position: This constraint attaches an item's center to the center of a deformed polygon of a Mesh

Item.
Polygon Normal: This constraint matches an item's orientation to that of a specified polygon's normal direction.

Geometry Constraints



Index A: 0
Index B: 0
Edge Percentage: 0.0 %
Offset X: 0 m
Y: 0 m
Z: 0 m

Index A: For vertices and polygons, the constraint uses this component index to get position and normal information. For edges, this is the index of the first vertex that defines the edge. By default, Modo uses the component index that is closest to the center of the item being constrained. To find the component index value for a particular component (vertex, edge, or polygon), make the appropriate selection of the item you wish to constrain to. The value shows in the **info** form within the Info and Statistics panel.

Index B: This index is used only for edges. It specifies the second vertex to use for the edge. This can be any vertex in the same mesh; the vertices do not need to be connected.

Edge Percentage: This is only used for Edge constraints. With this specified, the constraint can use an interpolated position (or normal) between the first and second vertices that define the edge. You can use this to slide the constrained item along the edge (for example, with 0% as the start of the edge and 100% as the end of the edge).

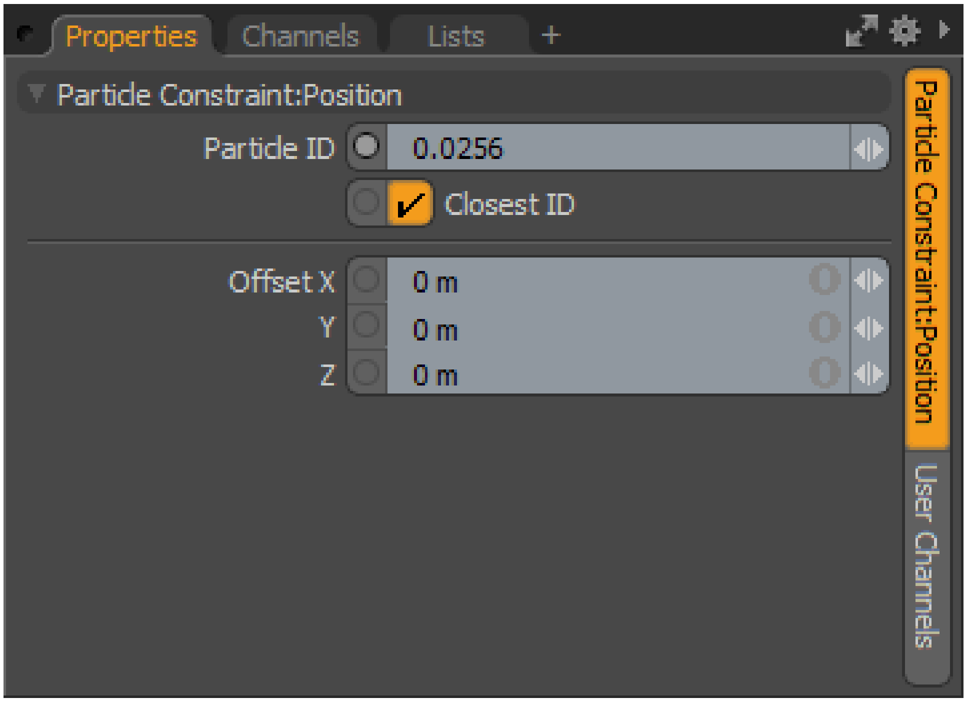
Offset: This is a world offset usually set when the constraint is applied with compensation on but can be used for further adjustment if required.

Particle Constraints

Particle Constraint Types

- **Position**: With this constraint, the item follows the World Position of the specified particle.

- **Rotation**: With this constraint, the item follows the rotation of the specified particle.



Particle Constraint:Position

Particle ID 0.0256

☑ Closest ID

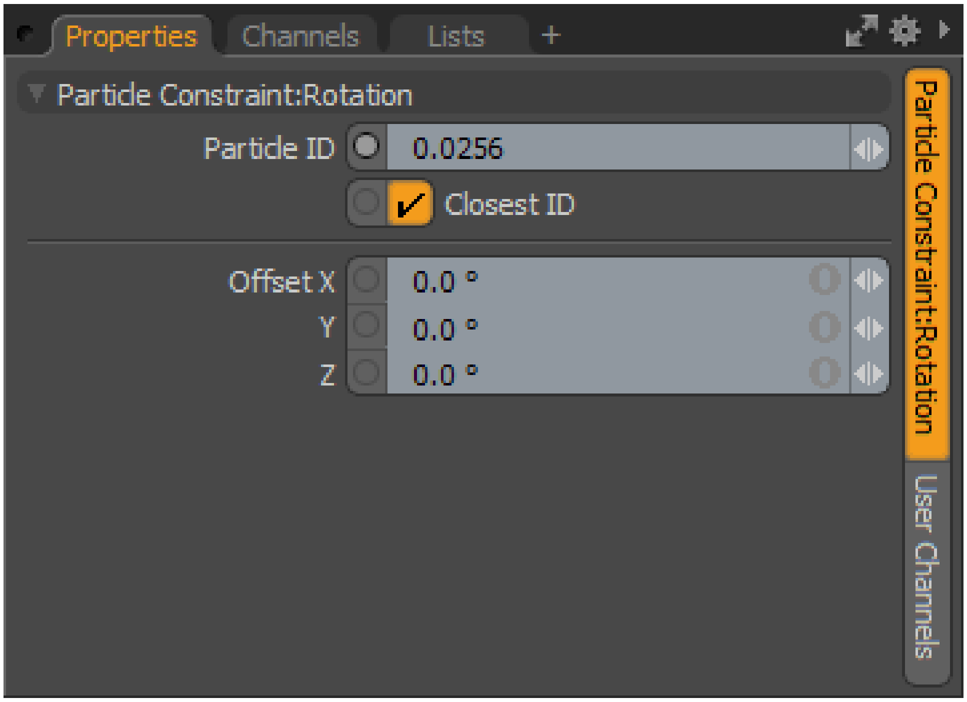
Offset X 0 m

Y 0 m

Z 0 m

Particle Constraint: Position

- **Particle ID**: This value specifies which particle within an item layer is used as the constraint source. Particle IDs are always specified as a value between 0.0 and 1.0.
- **Closest ID**: When this option is enabled, Modo rounds the ID values to the nearest actual value, so that an exact ID match is not necessary.
- **Offset XYZ**: Defines an additional distance away from the constrained position, offsetting the constrained item.



Particle Constraint:Rotation

Particle ID ● 0.0256

☑ Closest ID

Offset X 0.0 °

Y 0.0 °

Z 0.0 °

Particle Constraint: Rotation

- **Particle ID**: This value specifies which particle within an item layer is used as the constraint source. Particle IDs are always specified as a value between 0.0 and 1.0.
- **Closest ID**: When this option is enabled, Modo rounds the ID values to the nearest actual value, so that an exact ID match is not necessary.
- **Offset XYZ**: Defines an additional rotation away from the constrained rotation angle, offsetting the constrained item.

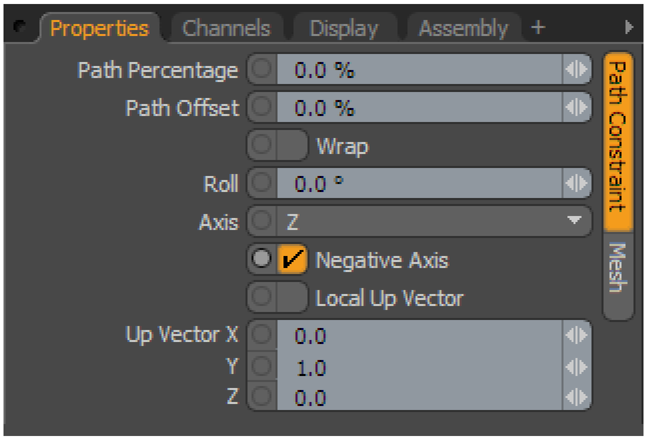
Path Constraints

Path Constraints Types

Path: This constraint binds an item to a user-generated curve or to the weighted average of multiple curves and affects both the item's position and its rotation values orienting the item to face along the curve's normal direction. Once constrained, the **Percent** amount controls the item's position and rotation along the length of the curve. You can use this setting to animate items moving along the path.
Path Position: This constraint binds an item to a user-generated curve or the weighted average of multiple curves and affects its position values only. Once constrained, the **Percent** amount controls the item's position along the length of the curve. You can use this setting to animate items moving along the path.
Path Normal: This constraint matches an item's rotation to the normal direction of a particular position

along the curve. Once constrained, the **Percent** amount defines the item's rotation based on that position's normal direction.

Path Constraints



Path Percentage 0.0 %
Path Offset 0.0 %
☐ Wrap
Roll 0.0 °
Axis Z
☑ Negative Axis
☐ Local Up Vector
Up Vector X 0.0
Y 1.0
Z 0.0

Path Percentage: This value defines a position along the path to which the item is constrained with 0% being the start of the curve (shown by a small circle around the curve's first vertex) and 100% the end of the curve. Paths can be open or closed.

Path Offset: You use this value to offset the point along the curve from the position specified by the **Path Percentage** value. With this value, for example, you can specify a starting offset along the curve or apply some variation to the constraint position.

Wrap: By default, constrained items stop when they reach either end of the curve. If you enable **Wrap**, then the item can continue its motion from the start of the curve. For example, if the **Percentage** channel is set to animate from 0% to 200%, then the constrained item travels along the curve twice.

Up Vector: When you enable **Local Up Vector**, Modo uses the XYZ value to specify the Up Vector direction as used in the Path and Path Normal constraints.

Note: The Path Position constraint does not use the following options.

Roll: This value specifies a rotation around the axis that's aligned to the curve.

Axis: This is the axis of the constrained item that is orientated along the curve.

Negative Axis: Selecting this inverts the defined **Axis** from the default positive direction to the negative direction.

Local Up Vector: By default the Path and Path Normal constraints use the values in the **Up Vector** options as world coordinates. However, with this checkbox, you can specify that these should be interpreted as local coordinates relative to the transforms of the Curve Mesh Item. This is useful if the Curve mesh is animated or transformed in any way.

Intersection Constraints

Intersect Constraints Types

The Intersect constraints determine the closest point and ray intersections with lines, planes, and surfaces. These constraints are different from other constraints in that Modo creates a new constrained item when the constraint is applied, rather than being specified in advance from the selected items.

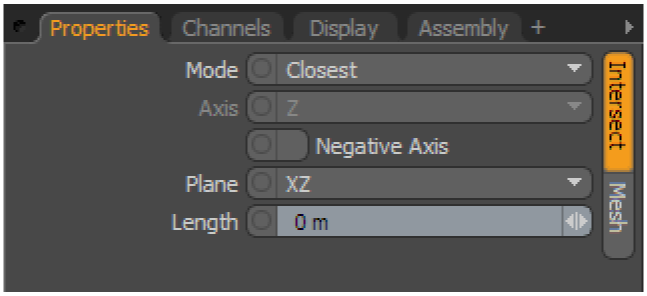
Intersect Line-Two Items: This creates a new locator at the closest point along an infinite line that runs through the world center positions of two selected items.

Intersect Line-Linear Curve: This creates a new locator at the closest position along a user-generated linear curve, which is a curve defined by only two vertices.

Intersect Plane: This constraint creates a new locator at the closest point (by default) from the first selected item to an infinite plane derived from the world axes of the second selected item. You can change the default axis plane by adjusting the **Plane** setting.

Intersect Surface: You can use this constraint to project a closest point or ray intersection from an item to a mesh surface. Select the item (as the source of the intersection) and then a Mesh Item. Modo creates a new locator at the closest point on the surface from the driver item.

Intersection Constraint



Mode: Closest
Axis: Z
Negative Axis
Plane: XZ
Length: 0 m

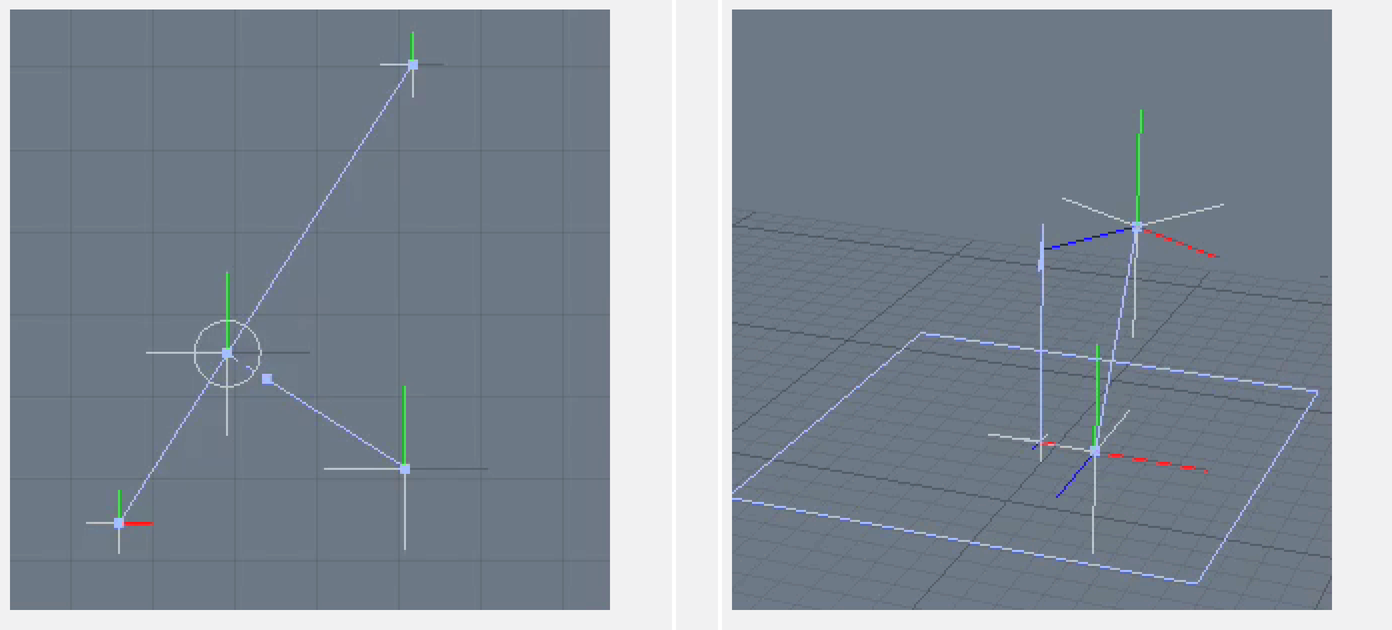
Mode: Specifies whether Modo performs a **Closest** point or **Ray** intersection to determine the item's position along the line. The **Closest** point intersection indicates the closest point along the line to the constrained item. A **Ray** intersection intersects the line along the chosen axis and takes in to account the item's position and rotation transform values.

Axis: When the **Mode** is set to **Ray**, this defines which axis channel on the driver item Modo uses to project the ray.

Negative Axis: Selecting this, inverts the defined **Axis** from the default positive direction to a negative direction.

Plane: For a Plane intersection, this selection (such as **XZ**) defines the plane.

Length: This value is different when used with a line intersection or plane intersection. When used with a line intersection, the **Length** value specifies a minimum distance between the driving item and the intersection point. If the driving item is located further away from the line than the distance set in this channel, then Modo continues to use the closest distance. However, if the distance is less than the **Length** value, then Modo pushes the intersection along the line to maintain the specified length between the driver and the intersection point. When used with a Plane intersection, Modo determines the direction to push from the positive Z axis of the driving item. See below for an example of each.



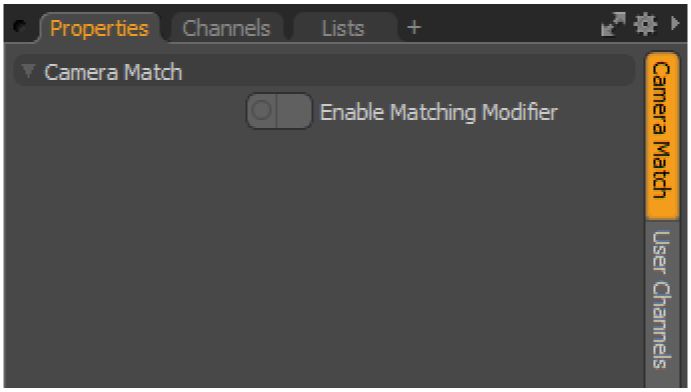
Camera Constraints

Camera Constraint Types

- **Camera Match**: Constrains a camera by position, rotation and focal length.
- **Camera Plane**: Adjusts the size, position and rotation of an item to match the image plane projected from the camera at a given depth.

Camera Match

- **Enable Matching Modifier**: Check this to activate the matching modifier.


Properties
Channels
Lists
+
Camera Match
User Channels

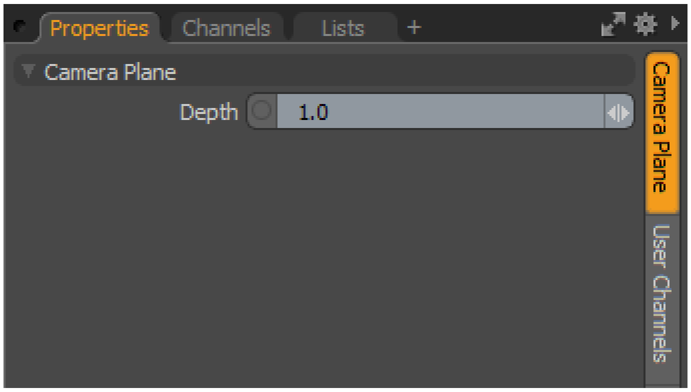
Camera Match

○ Enable Matching Modifier

Camera Plane

The Camera Plane Modifier adjusts the size, position and rotation of an item so that it matches the image plane projected from the camera at a specified depth. This is particularly useful with image-mapped quads, which are sized to fit the screen exactly.

- **Depth**: The distance of the plane from the camera viewpoint.

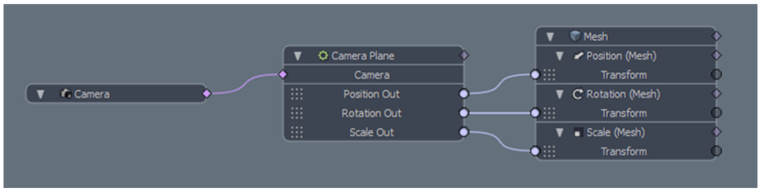

Properties
Channels
Lists
+
Camera Plane
User Channels

Camera Plane

Depth 1.0

To use:

1. Add the modifier in the schematic view.
2. Connect the camera to the input.
3. Connect the output position, rotation, and scale transforms to the corresponding inputs.


Camera
Camera Plane
Camera
Position Out
Rotation Out
Scale Out
Mesh
Position (Mesh)
Transform
Rotation (Mesh)
Transform
Scale (Mesh)
Transform

UV Constraint

UV Constraint is a constraint type that provides a 3D rotation position matrix for a polygon, to a specific UV coordinate. Two matrices are output from the modifier: **World Position** and **World Rotation**. **World Position** represents the 3D position of the UV coordinate, and **World Rotation** represents a rotation on the 3D surface, constructed from the Normal, dPdu and dPdv.

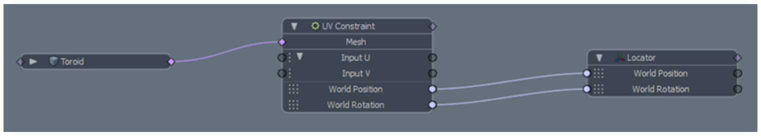
To use the UV Constraint, you need a mesh and a Locator in your scene.

You can set up the UV Constraint in the **Schematic** viewport. To do this, follow the steps below:

Make sure you're in the **Setup** layout, to have access to the **Schematic**.

1. Click **Add...** at the top of the **Schematic**.
2. Select **Channel Modifiers** > **Constraints** > **UV Constraint**, and double-click to add it to the viewport.
3. Click your mesh in the 3D viewport and drag it into the **Schematic**.
4. Connect the mesh to the **Mesh** input of the UV Constraint, and connect the **World Position** and **World Rotation** channels of the Locator to the **World Position** and **World Rotation** outputs of the UV Constraint, as illustrated below:

Note: The **World Position** and **World Rotation** channels of the Locator should be visible in the **Schematic**. You can drag them in from the **Channels** list on the right side of the interface.



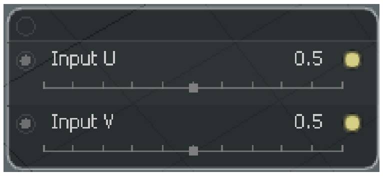
The Locator snaps to the 0.0 U and V positions on the mesh.

Note: If the Locator doesn't snap to the set U and V positions automatically, make sure that the correct UV Map is selected in the **Properties** panel of the UV Constraint.

To move the UV Constraint to a different position, select its **Input U** and **Input V** channels in the **Schematic** and press **C** on the keyboard to activate Channel Haul.

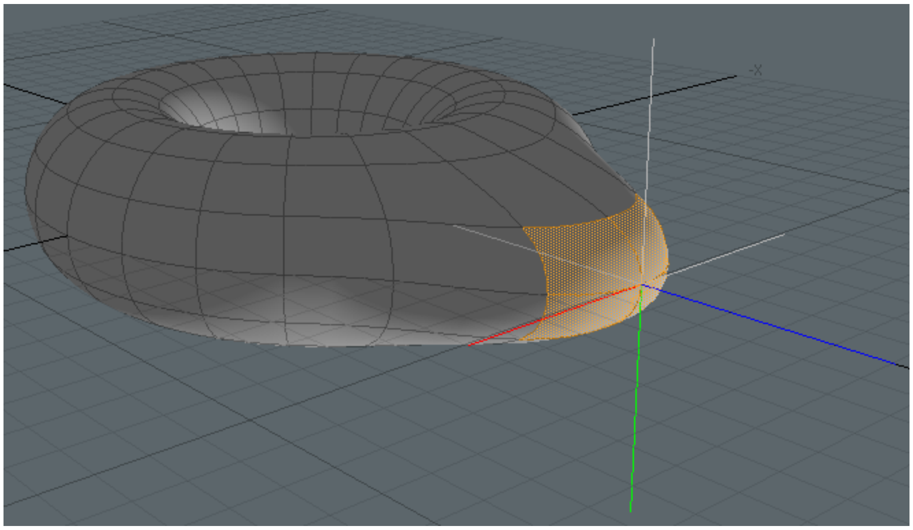
Note: Make sure the selection mode is set to **Items**.

The Channel Haul controls appear:



5. Use the sliders to specify the vertex to which you'd like to place the UV Constraint.

   As you drag the sliders, the Locator moves to the selected vertex in the viewport. Any edits you make to the geometry are reflected on the UV Constraint.



UV Constraint Properties

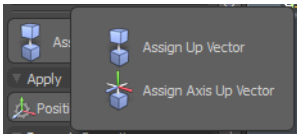
The following options are available in the **Properties** panel of the UV Constraint.

UV Constraint Properties	
UV Map	Specifies the UV Map that is searched for polygons at the matching UV coordinate.
Input U/V	The UV coordinate to constrain to. You can set a value between 0.0 and 1.0.
UDIM	This UDIM transforms the UV coordinate to determine the final UV coordinate. Set a value between 1001 and 9999.
Roll	The rotation matrix output is transformed by this value to roll the matrix around the surface normal.

Constraint Up Vectors

By default, the 'up' direction of an item is up the positive Y-axis. However, it may be useful to alter this property and transform the item (particularly when animating) around an alternate up direction, also known as the Up Vector. Modo offers two ways to modify the Up Vector by constraining it to another item or another item's axis.

You can find these vectors in the **Setup** layout, in the **Modifiers** sub-tab on the left of the layout. Click and hold the **Assign Up Vector** button to reveal the **Assign Axis Up Vector** option.



Assign Up Vector

This changes the up direction of the constraint (and therefore the item). This is useful if the upward direction needs to change during an animation. You can assign an item as an Up Vector and its current position will represent up for the constraint. With the Constraint item selected in the **Items** list, hold down the **Ctrl** key, select the item intended as the Up Vector target, and click **Assign Up Vector**. Once assigned, any movement of the Up Vector item will influence the **Roll** of the constrained item.

Assign Axis Up Vector

This lets you use the X, Y, or Z axis of another item as the direction for the up vector.

Channel Modifiers

Channel Modifiers provide you with different ways to procedurally control elements by overriding or modifying channel values as they are fed into other nodes in a schematic graph. Channel Modifiers adjust raw channel values in various ways and can be added to a scene in the **Modifiers** toolbox sub-tab of the **Setup** interface layout. You can also use the **Add** button in the Schematic view.

Click any link below to view a particular topic. You can find additional Channel Modifiers specific to shading in the Shading Nodes topic.

Arrays

Array Blend

Array Count

Array Element By Index

Array Element Type

Array Operator

Curves to Array

Filter Array

Locators to Array

Merge Arrays

Mesh Data Array

Offset Array

Particles to Array

Sort Array

String to Array

Channels

Channel Noise

Channel Oscillator

Channel Relationship

Channel Sound

Channel Waveform

Conditional

Logic

Gradient

Gradient Blend

Gradient Clamp

Gradient from Mesh

Gradient Math

Gradient Noise

Gradient Offset

Gradient Scale

Gradient Step

Gradient Waveform

Geometry

Mesh Info Modifier

Int Range

Int Range Bounds

Int Range Clamp

Int Range Merge

Int Range Test

Math

Basic Math

Math Multiple

Math

Trigonometry

Matrix

Matrix Blend

Matrix Compose

Matrix Construct

Matrix Euler

Matrix Invert

Matrix Transpose

Matrix Vector

Matrix Vector Multiply

Measure

Measure Angle

Measure Distance

Measure Speed

Measure Velocity

Particle

Particle Constraint:Position

Particle Constraint:Rotation

Probe

Curve Probe

Falloff Probe

Quaternion

Quaternion Conjugate

Quaternion From Axis-Angle

Quaternion From Euler

Quaternion From Matrix

Quaternion Get Value

Quaternion Math

Quaternion Normalize

Quaternion Set Value

Quaternion Slerp

Quaternion To Axis-Angle

Quaternion To Euler

Quaternion To Matrix

Quaternion Vector Multiply

Simulation

ADSR Envelope

Simulation Flow-Through

Simulation Modifiers

Simulation Latch

String

Encode String

String Compose

String Constant

String Find and Replace

String Switch

Time

Cycler

Time

Time Offset

Time Warp

Vector

Vector

Vec. Magnitude

Vector Maths

Vector Orthogonalize

Vector Reflection

Vector Multiply by Scalar

Other

Clamp

Dual Joint Planar IK

Dynamic Parent

Expression

Generate ID

Linear Blend

Random

Randomize ID

Revolve

Simple Kinematics

Smooth

Switch

Array Modifiers

Arrays are a multi-dimensional value type that can store multiple values in a single value. They can take any value type supported by Modo, such as integer, float, and matrix. However, all values within the array should be of the same type.

You can create a complex hierarchy of arrays, each of which can store a different value type. For example you can have an array that stores three other arrays: one that contains integers, another that contains floats, and a third one that contains strings.

When arrays are evaluated, they behave like any other channel type. They are constructed by sampling a data set, manipulated using modifiers, and used by various operations, such as the procedural modeling system.

You can use array modifiers to to generate, merge, blend, sort arrays, and so on. When you add a modifier to the Schematic, some of its channels are exposed on the node. You can open the **Channels** viewport on the right panel and drag additional channels onto the node, if you need them. Most of them also have properties that you can edit in the Properties panel on the right.

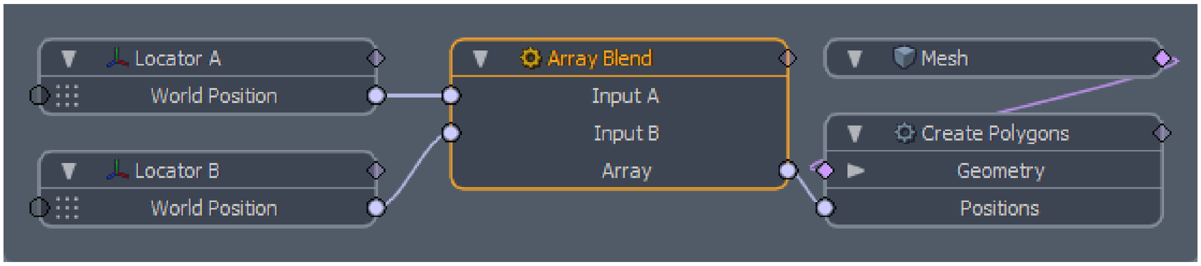
Note: For more information on using the **Channels** viewport, see Channels Viewport.

Array Blend

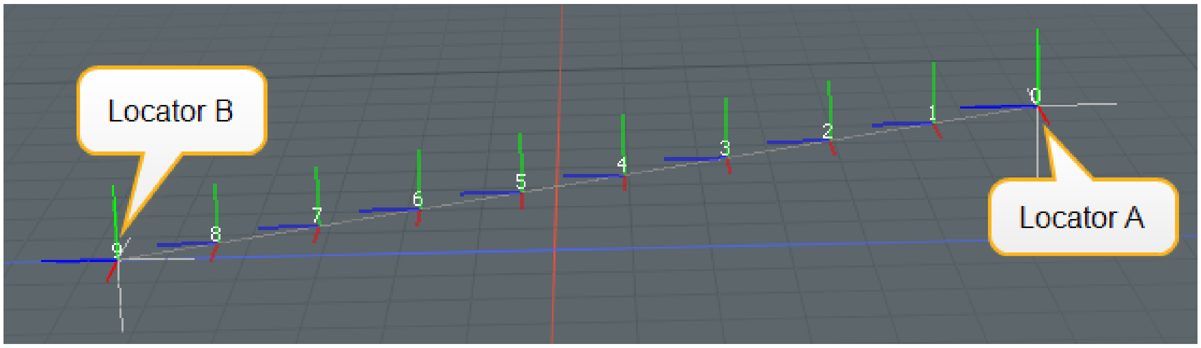
The Array Blend modifier interpolates between two input values and generates an output array containing values blending from the first value to the last value, over a specified number of steps. For example, if the input values are 0.0 and 1.0, and the step count is 5, the output array contains 0.0, 0.25, 0.5, 0.75, 1.0.

Note: Only value types supporting Value Math are correctly interpolated, any other value type is stepped from the first value to the second.

The example below takes the world positions of two locators (**Locator A** and **Locator B**) to create an output array. The **Count** property is set to 10, so the modifier creates eight additional points between the locators.


Locator A
World Position
Locator B
World Position
Array Blend
Input A
Input B
Array
Mesh
Create Polygons
Geometry
Positions

The points are then used by a Create Polygons node to produce a curve:


Locator B
Locator A
0
1
2
3
4
5
6
7
8
9

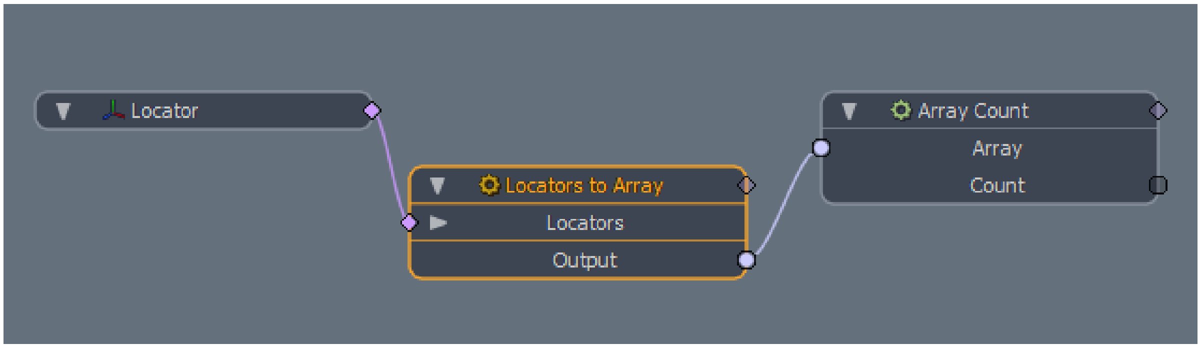
The Array Blend modifier has the following channels and properties:

Channels on the Node by Default	
Input A/B	The values to blend.
Array	The output array.

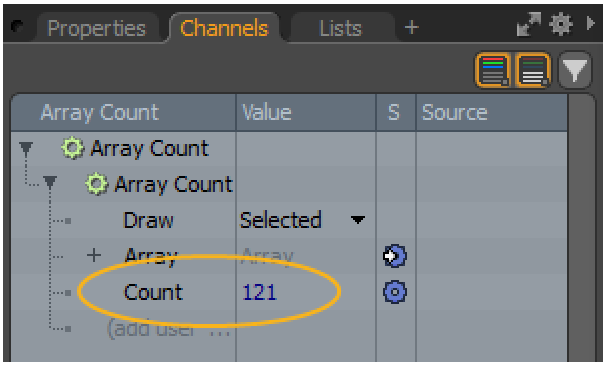
Properties Panel	
Count	Specifies the number of values to generate in the output array. For values that support the value math interface, these steps are interpolated to blend between the two inputs.
Interpolation	Allows you to remap the interpolation using a gradient editor. A line gradient between 0.0 and 1.0 interpolates at fixed samples between the first and last value. Changing the shape allows you to control the distribution of the samples.

Array Count

The Array Count modifier returns the number of elements in an array. In the example below, Array Count outputs the number of elements in the array produced by the Array Modifiers modifier.



The count is displayed in the **Channels** viewport on the right, when **Locators to Array** is selected in the Schematic. Array Count may be useful when you need to know the number of elements for some other aspect of the rig. For instance, if you want to use a Radial Array, you can connect the Helix Generator node's **Count** value to the **Count** channel.



Array Count	Value	S	Source
Array Count			
Array Count			
Draw	Selected		
+ Array	Array		
Count	121		
(add user ...			

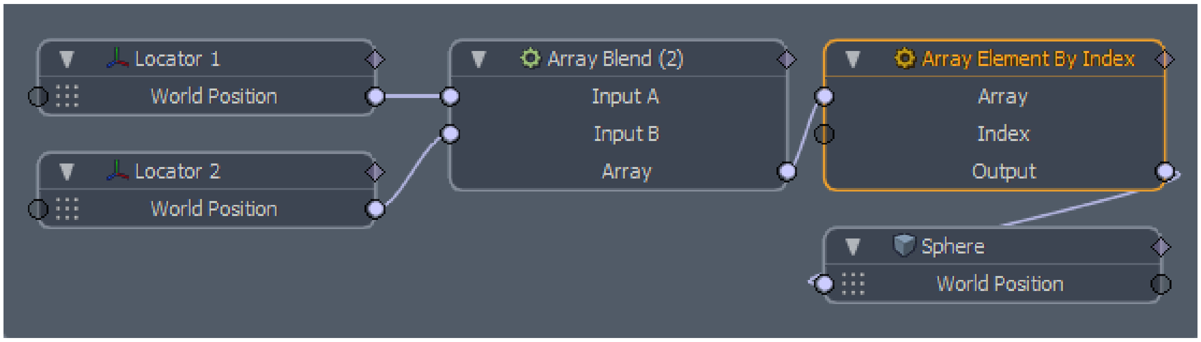
The Array Count modifier has the following channels:

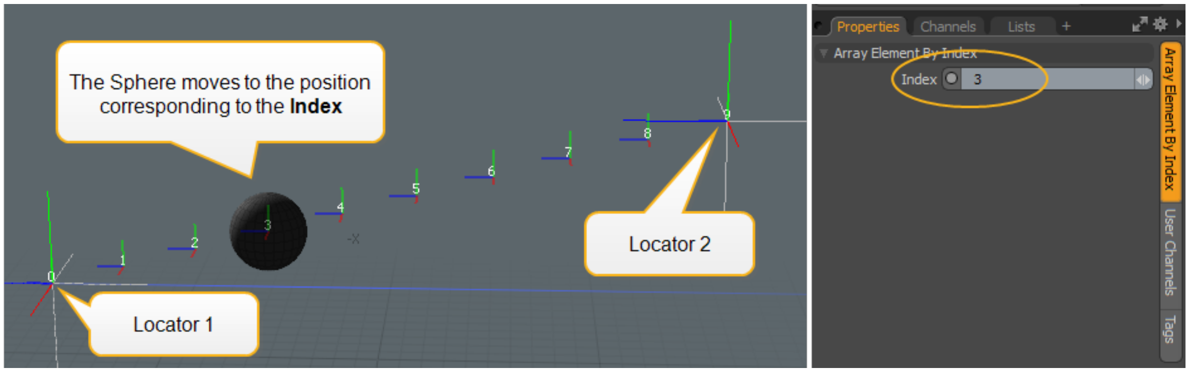
Channels on the Node by Default	
Array	The Array channel to count the number of elements in.
Count	The number of elements in the array.

Array Element By Index

Array Element By Index allows you to access elements within an array using an index. For example, if the array contains the values "I", "Love", and "Modo", accessing index 0 returns "I", element at index 1 returns "Love" and index 2 returns "Modo". If the specified index is out of range, it returns the last element stored in the array. You can then use that element to drive something else in your scene.

The example below shows an array produced from two locator positions using Array Modifiers. Array Element By Index is then used to drive the position of a sphere. When you change the **Index** value of the **Array Element by Index** node in the **Properties** panel on the right, the sphere moves to the corresponding position in the 3D viewport.


Locator 1
World Position
Locator 2
World Position
Array Blend (2)
Input A
Input B
Array
Array Element By Index
Array
Index
Output
Sphere
World Position


The Sphere moves to the position corresponding to the **Index**
Locator 1
Locator 2
0
1
2
3
4
5
6
7
8
9
Properties
Channels
Lists
Array Element By Index
Index
3
Array Element By Index
User Channels
Tags

As array channels can store any value type, the output channel attempts to automatically convert to the channel type it is connected to.

Array Element By Index has the following channels:

Channels on the Node by Default	
Array	The input array channel that contains the elements that are accessed by index.
Index	The index of the element to read from the array.
Output	The element in the array, accessed by index.

Array Element Type

The Array Element Type modifier returns the type of data stored within an array. The output value is the internal name for the value type, for example "matrix4" for Matrix channels. This can be useful for debugging purposes.

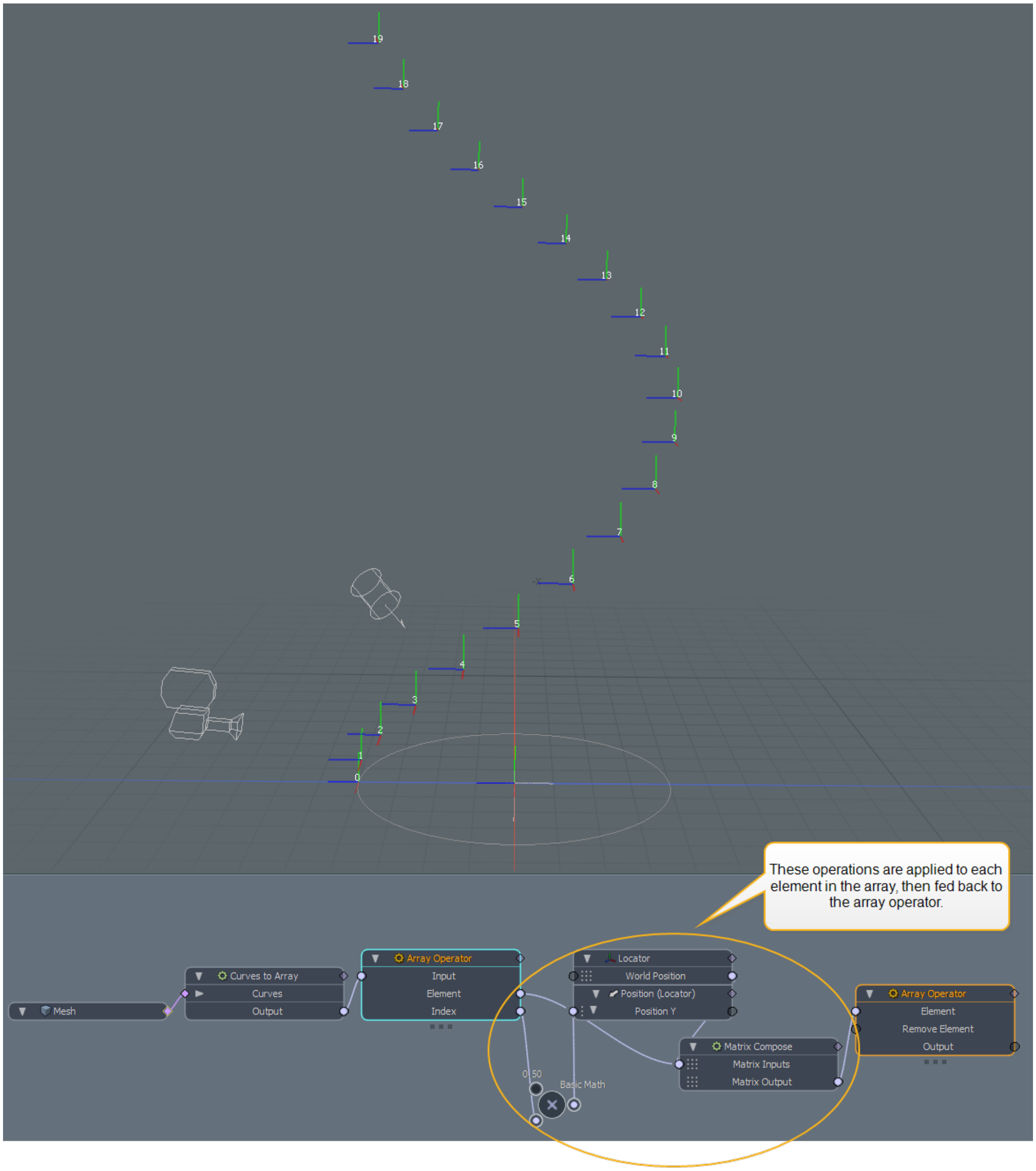
Array Element Type has the following channels:

Channels on the Node by Default	
Array	The input array to evaluate for the type of values it contains.
Index	The name of the value type in the array. If the array is empty, the output value is empty, otherwise it contains the internal name for the value type.

Array Operator

The Array Operator modifier allows you to access individual elements in an array and modify them using a rig. For each element within the input array, the rig is evaluated, calculating an output value that is then written back into the array.

In the example below, an array generated from a curve is the **Input** for the Array Operator. Each element in the array is then multiplied by 0.5, and fed into a Matrix Compose node. The **Output** from the Matrix Compose is then connected to the **Element** channel of the Array Operator. New locators are created in a spiral shape in the 3D viewport, which you can then use for other operations, such as Create Polygons.



These operations are applied to each element in the array, then fed back to the array operator.

19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Mesh
Curves to Array
Curves
Output
Array Operator
Input
Element
Index
Locator
World Position
Position (Locator)
Position Y
0.50
Basic Math
Matrix Compose
Matrix Inputs
Matrix Output
Array Operator
Element
Remove Element
Output

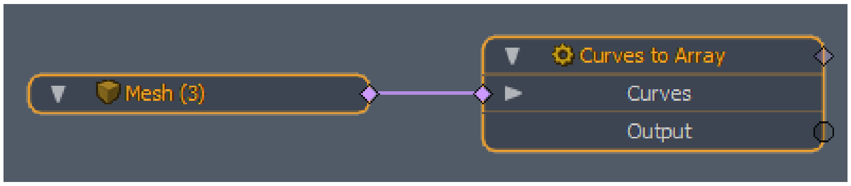
Array Operator has the following channels:

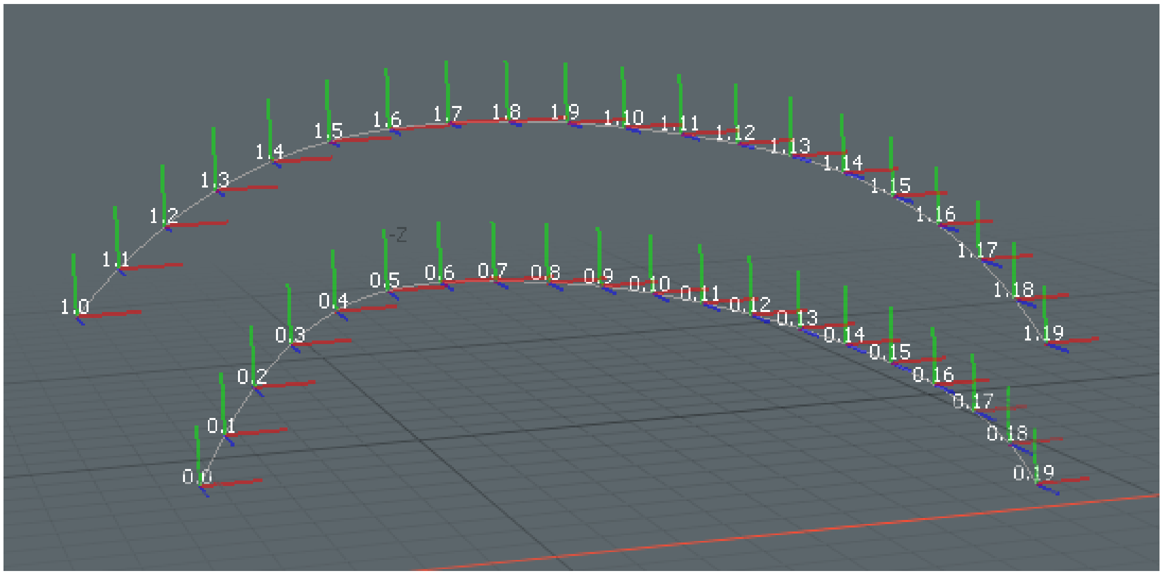
Channels on the Node by Default	
Input	The input array to modify.
Output	The modified output array.

Additional Channels	
Element	The current output element in the array that the rig is being evaluated for. This can read to get the current array element value.
Index	The index of the current element within the array that the rig is being evaluated for.
Element	The current input element in the array that the rig is being evaluated for. This can be driven to set the current array element value.
Remove Element	Specifies whether the element in the array should be removed. This can be driven to remove the array element.

Curves to Array

The Curves to Array modifier samples a curve for different properties along its length and generates an array from the data.


Mesh (3)
Curves to Array
Curves
Output


1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
1.10
1.11
1.12
1.13
1.14
1.15
1.16
1.17
1.18
1.19
-Z
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.10
0.11
0.12
0.13
0.14
0.15
0.16
0.17
0.18
0.19

Any item type that provides a Curve Group channel can be connected to the Curves to Array modifier, but the most common connection is a mesh item containing a curve. If multiple curves are connected, the array will contain a sub-array for each curve, with the curve samples stored in the sub-array.

The modifier can sample four different curve property types:

- **Percentage** - Stores the percentage along the curve for each sample. For example, if the curve is sampled three times between 0.0 and 1.0, the values stored in **Percentage** are 0.0, 0.5, and 1.0.
- **Position** - Stores the world position for each sample along the curve.
- **Tangent** - Stores a normalized vector representing the direction of the curve at the position being sampled. The vector is relative to the center of the world, and stores the direction of the curve.
- **Local Tangent** - Stores a normalized vector representing the direction of the curve at the position being sampled, combined with the position being sampled.

The generated array can then be used by other operations, such as Create Polygons.

Curves to Array has the following channels and properties:

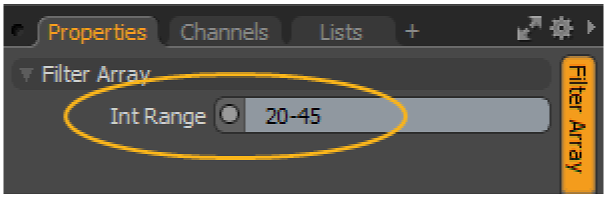
Channels on the Node by Default	
Curves	The input curves used to generate the array.
Output	An array of the sampled property. If multiple curves are sampled, the array will contain sub-arrays for each sampled curve.
Properties Panel	
Index	An integer range that allows you to filter the input curves to be sampled. This can be a single integer to specify a single input curve, a range of numbers, a comma-separated list, or a combination. If the value is empty, or contains "...", all curves are sampled.
Property Type	Specifies the property to sample. The following options are available: • **Percentage** - Stores the percentage along the curve for each sample. For example, if the curve is sampled three times between 0.0 and 1.0, the values stored in **Percentage** are 0.0, 0.5, and 1.0. • **Position** (default) - Stores the world position for each sample along the curve. • **Tangent** - Stores a normalized vector representing the direction of the curve at the position being sampled. The vector is relative to the center of the world, and stores the direction of the curve. • **Local Tangent** - Stores a normalized vector representing the

	direction of the curve at the position being sampled, combined with the position being sampled.
Start/End	By default, the curve is evaluated from start to end. These fields specify the start and end points of the curve.
Steps	Specifies the number of steps along the sampled curve.
Sampling Method	Specifies the method to walk the curve for sampling. • **Fraction** - Divides the curve into equal lengths. • **Parameter** - Uses the divide based on the segments of the curve, adding more samples where the direction of the curve changes more quickly.
Sample Distribution Rate	A gradient that controls the distribution of samples along the length of the curve. By default, the linear gradient places samples at equal distances along the length of the curve. A gradient that eases in places most samples near the start of the curve, and a gradient that eases out places most samples near the end of the curve. The mini Gradient Editor in the properties can be used to define a custom gradient.

Filter Array

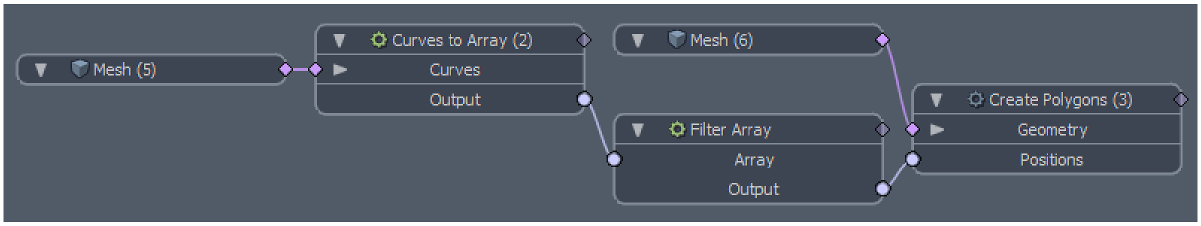
The Filter Array modifier filters an input array, only writing elements from the input array to the output array if they match the filter. The filter is specified by an integer range, but it can also be driven by a pattern.

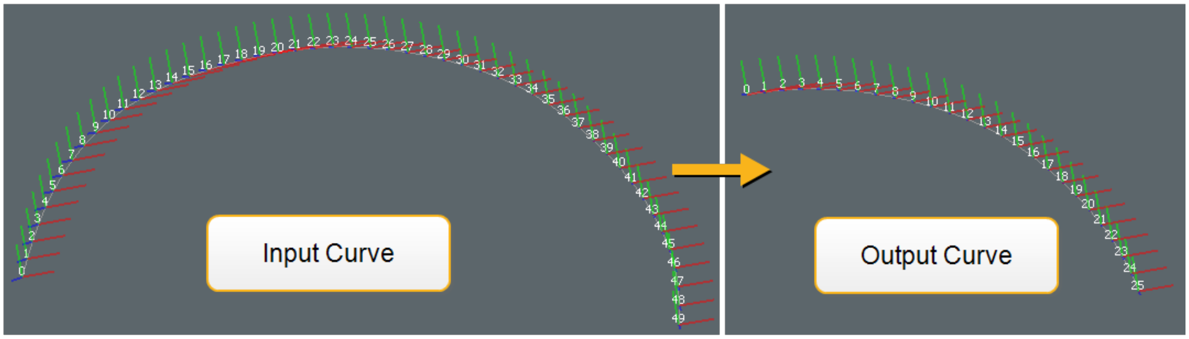
To specify a filter, select the **Filter Array** node and in the **Properties** on the right, enter a value in the **Int Range** field.



Int Range 20-45

For example, the following rig generates an array from a curve. The array is then filtered, and only the positions 20 to 45 are used by the Create Polygons mesh operation to create another curve.


Mesh (5)
Curves to Array (2)
Curves
Output
Mesh (6)
Filter Array
Array
Output
Create Polygons (3)
Geometry
Positions


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
Input Curve
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Output Curve

Filter Array has the following channels and properties:

Channels on the Node by Default	
Array	The input array to be filtered.
Output	The filtered output array.
Properties Panel	
Int Range	The integer range used for filtering. Only elements within the array that match indices within the integer range are written to the output array.

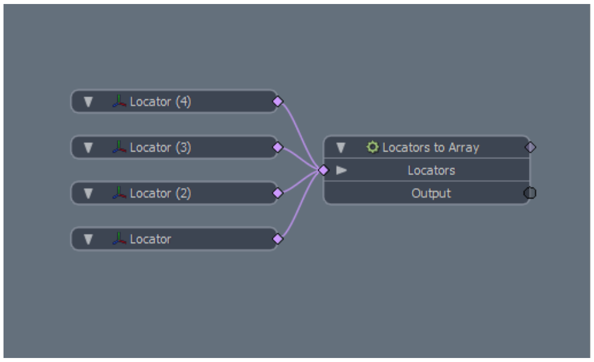
Locators to Array

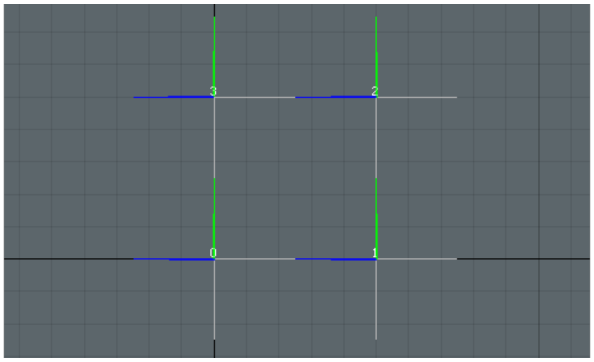
The Locators to Array modifier reads Locator type item transforms into an array, creating an array of Matrix values. The modifier has two modes:

- **Single** - Reads data from a single locator over time.
- **Multiple** - Reads data from multiple locators at the current evaluation time.

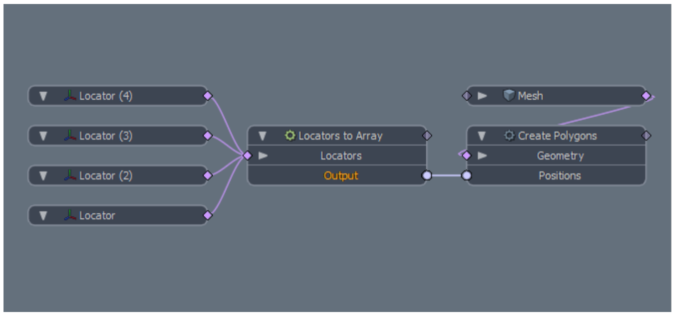
A graph input on the modifier allows multiple locator type items to be connected. This connection allows any item that has a 3D transform to be connected, such as cameras, lights, or meshes.

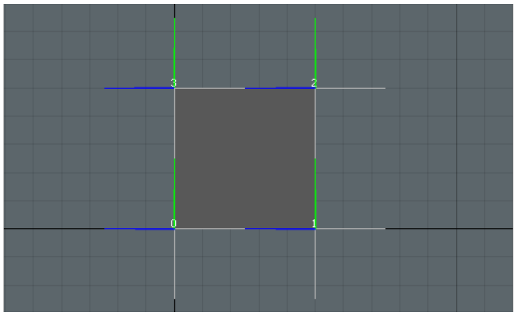
In the example below, four locators are connected to the Array Modifiers node.


Locator (4)
Locator (3)
Locator (2)
Locator
Locators to Array
Locators
Output



The Locators to Array node is then connected to a Create Polygons node, which uses the locator positions to create a polygon.


Locator (4)
Locator (3)
Locator (2)
Locator
Locators to Array
Locators
Output
Mesh
Create Polygons
Geometry
Positions



Locators to Array has the following channels and properties:

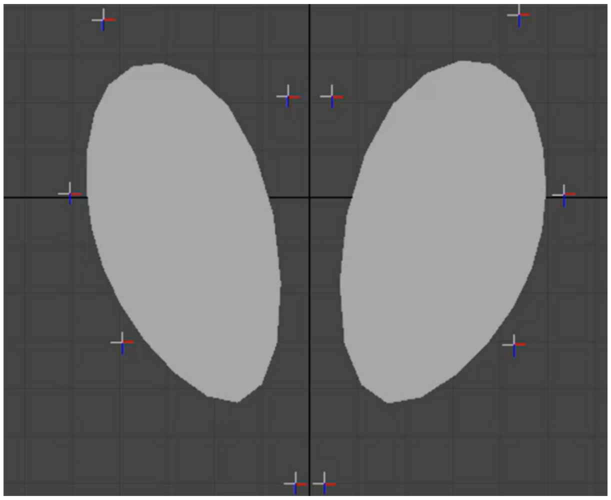
Channels on the Node by Default	
Locators	The locators used to generate the array.
Output	The generated array of Matrix value types.
Properties Panel	
Mode	The Locators to Array modifiers has two modes: • **Single** - Reads a single locator over time • **Multiple** - Reads multiple locators at the evaluation time.
Index	When **Mode** is set to **Single** and multiple locators are connected to the modifier, this allows the locator to evaluate to be specified. If the index is out of range, the array is empty.
Start Time	When **Mode** is set to **Single**, the input locator transform is read over time, with an entry into the array for every frame. This channel specifies the first frame to begin sampling the transform.

End Time	When **Mode** is set to **Single**, this channel specifies the last frame to sample the transform.
Output	The generated array of Matrix value types.

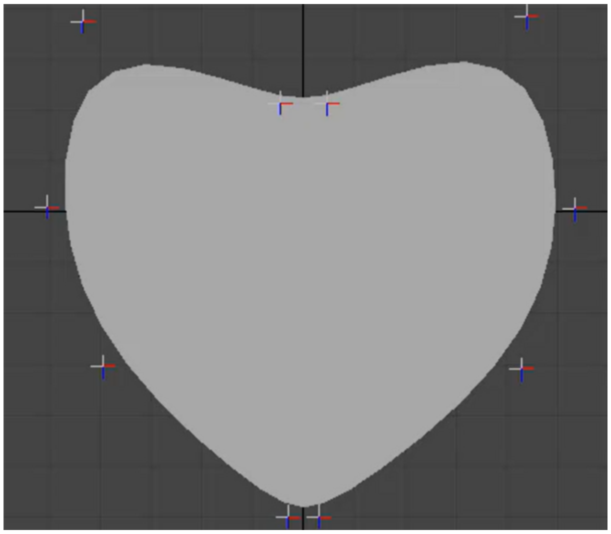
Merge Arrays

The Merge Arrays modifier combines two arrays into a single array. It can operate in two ways:

- **Group** - Groups two input arrays as individual sub-elements in another array.



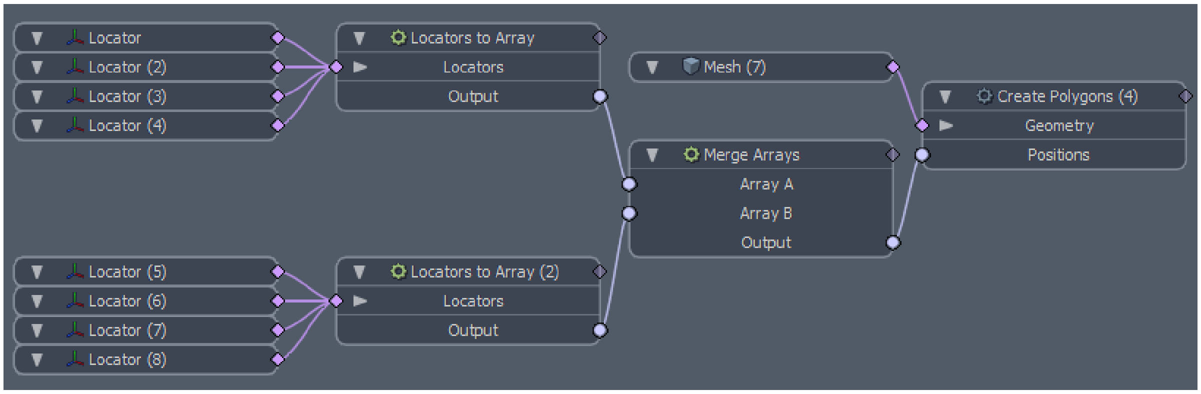
- **Join** - Adds the elements from the second array at the end of the first array.



You can set the **Mode** for the merge operation in the **Properties** panel on the right.

Note: The type of data stored within the two arrays must be identical or compatible with each other. If the two arrays contain different types, one of the types is converted so that the value types match. If neither of the value types can be converted, the merge operation is not completed.

In this example, two arrays are merged to create a single array for the Create Polygons operation.


Locator
Locator (2)
Locator (3)
Locator (4)
Locators to Array
Locators
Output
Mesh (7)
Create Polygons (4)
Geometry
Positions
Merge Arrays
Array A
Array B
Output
Locator (5)
Locator (6)
Locator (7)
Locator (8)
Locators to Array (2)
Locators
Output

Locators to Array has the following channels and properties:

Channels on the Node by Default	
Array A/B	The input arrays to be merged.
Output	The merged output array.
Properties Panel	
Mode	Controls how the two arrays are merged. • Group - Adds the two input arrays as individual sub-elements in another array. • Join - Adds the elements from the second array to the end of the first array.

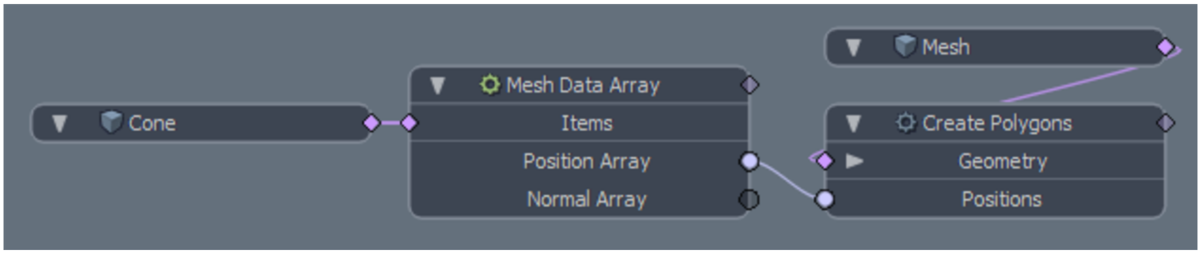
Mesh Data Array

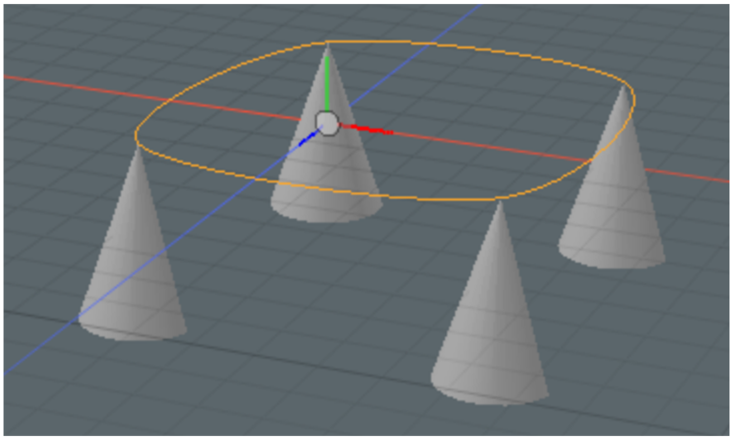
The Mesh Data array modifier creates a normal and position array from connected meshes and mesh instances. It has four different modes for building arrays:

- **Selection Set** - Using the set tag name, a position and normal is generated for each contiguous group of elements. Enable **Per Element** to create a normal and position for each individual element rather than per contiguous element group.

- **Items** - A position is created at each item's position and a normal is created using the X,Y, or Z direction of the item.
- **Materials** - A position and normal is created for the polygons in each material in each mesh.
- **Mesh Part** - Each connected chunk of geometry in each mesh creates a position and normal in the array.

In the example below, the Mesh Data Array takes a mesh with four cones in it. The top vertex of each cone is part of a selection set called **a**. In the Mesh Data Array's properties, **Mode** is set to **Selection Set**, and a is specified as the **Selection Set**. An array is created from the positions of the elements in the selection set. This array is used by the Create Polygons mesh operation to generate a curve.





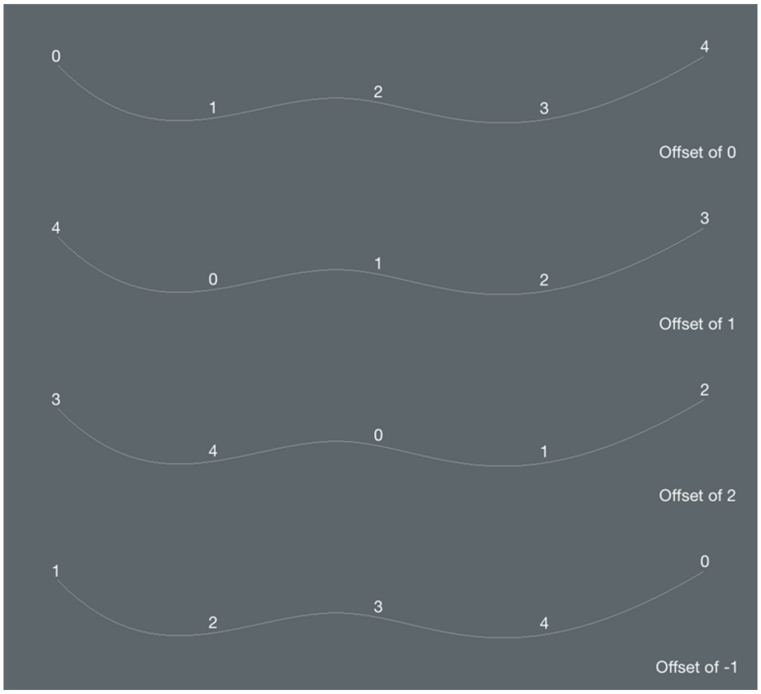
Mesh Data Array has the following channels and properties:

Channels on the Node by Default	
Items	The meshes used to create the array.
Position Array	The array generated from the positions.
Normal Array	The array generated from the normals.
Properties Panel	
Mode	The way the modifier generates the array. The following options are available:

	• **Selection Set** - Using the set tag name, a position and normal is generated for each contiguous group of elements. Enable **Per Element** to create a normal and position for each individual element rather than per contiguous element group. • **Items** - A position is created at each item's position and a normal is created using the X,Y, or Z direction of the item. • **Materials** - A position and normal is created for the polygons in each material in each mesh. • **Mesh Part** - Each connected chunk of geometry in each mesh creates a position and normal in the array.
Per Element	When enabled, array values are created separately for each element. When disabled, array values are created per each contiguous geometry.
Selection Set	When **Mode** is set to **Selection Set**, this specifies the selection set to be used.
Item Axis	When Mode is set to Items, this specifies the item axis to be used.

Offset Array

The Offset Array modifier moves elements in an array forwards or backwards by the specified offset amount. Elements that are pushed off the end of the array are inserted back at the beginning.



Offset Array has the following channels and properties:

Channels on the Node by Default	
Array	The input array to offset.
Output	The offset output array.
Properties Panel	
Offset	The number of elements to offset by. A positive value moves elements forwards and a negative value moves elements backwards.

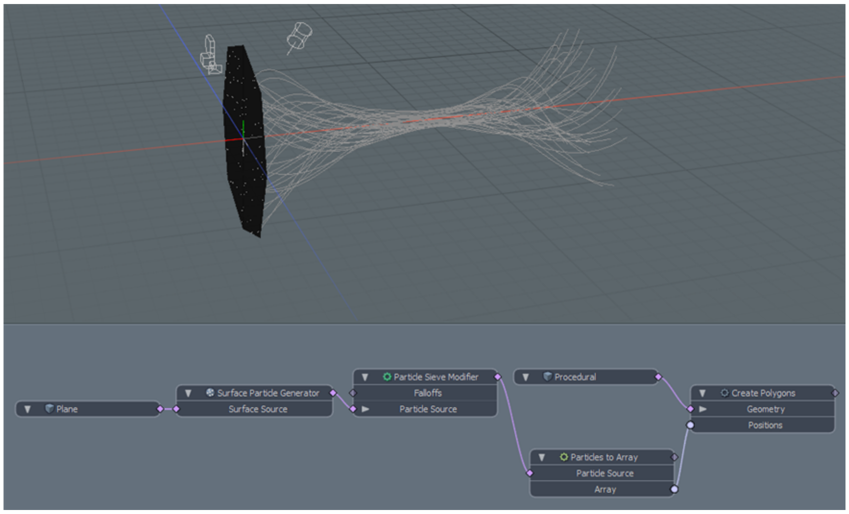
Particles to Array

The Particles to Array modifier samples a particle simulation or static particle source and reads a specified particle feature into the array. You can read multiple particles at the current time, or multiple particles over a specified time period.

When reading particles at a single time frame, the output array will contain the sampled particle features. When reading particles over time, the output array will contain sub-arrays, each containing the particles sampled at the time step.

Note: Reading many particles over time requires a large amount of memory and may cause performance issues.

The example below shows a plane with a Surface Generator creating particles across the plane. The plane transform is animated through space. The Particles to Array modifier samples the particles at every time step and a Create Polygons mesh operation generates curves that trace the particle trajectory.


Plane
Surface Particle Generator
Surface Source
Particle Sieve Modifier
Falloffs
Particle Source
Procedural
Create Polygons
Geometry
Positions
Particles to Array
Particle Source
Array

Particles to Array has the following channels and properties:

Channels on the Node by Default	
Particle Source	The particle source used to generate the array.
Array	An array containing the sampled particle features.
Properties Panel	
Start Time/End Time	When sampling particles over multiple time steps, this specifies the times to start and end sampling.
Feature	Specifies the particle feature to sample into the array. The sub-type stored in the array will change as the feature changes. The supported features on the Particles to Array modifier are: • **ID** - Float • **Position** - Distance3 (Vector) • **Velocity** - Float3 (Vector)

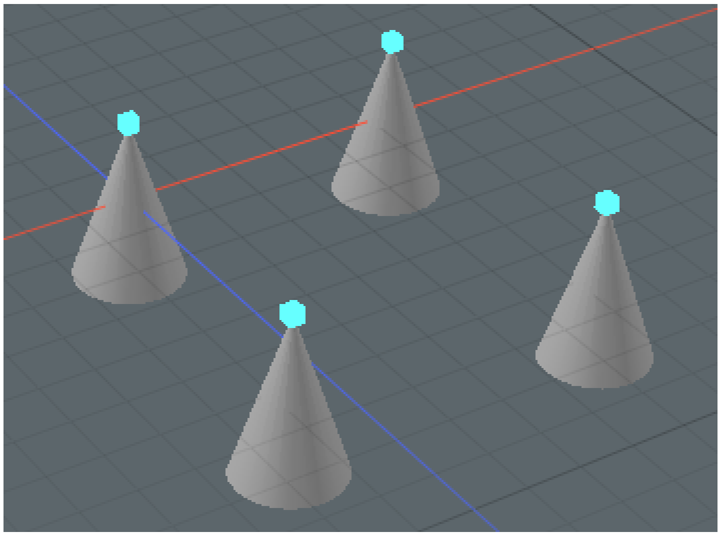
	• **Force** - Float3 (Vector) • **Mass** - Mass • **Age** - Float • **Angular Velocity** - Float3 (Vector) • **Torque** - Float3 (Vector) • **Size** - Float • **Transform** - Matrix3
Mode	Specifies whether to sample the particles at a single time step, or over multiple frames.

Sort Array

The Sort Array modifier creates a new array using the elements of the input array ordered according to the **Sorting Order** channel. The length of the order array does not need to match the input array, and elements can be included more than once. For example, if the input array contains letters, such as "a, b, c, d", "1" in the order array would output an "a", "2" would yield a "b", "3" would output "c", and so on. Setting the order array to "4, 3, 1" would produce the array "d, c, a".

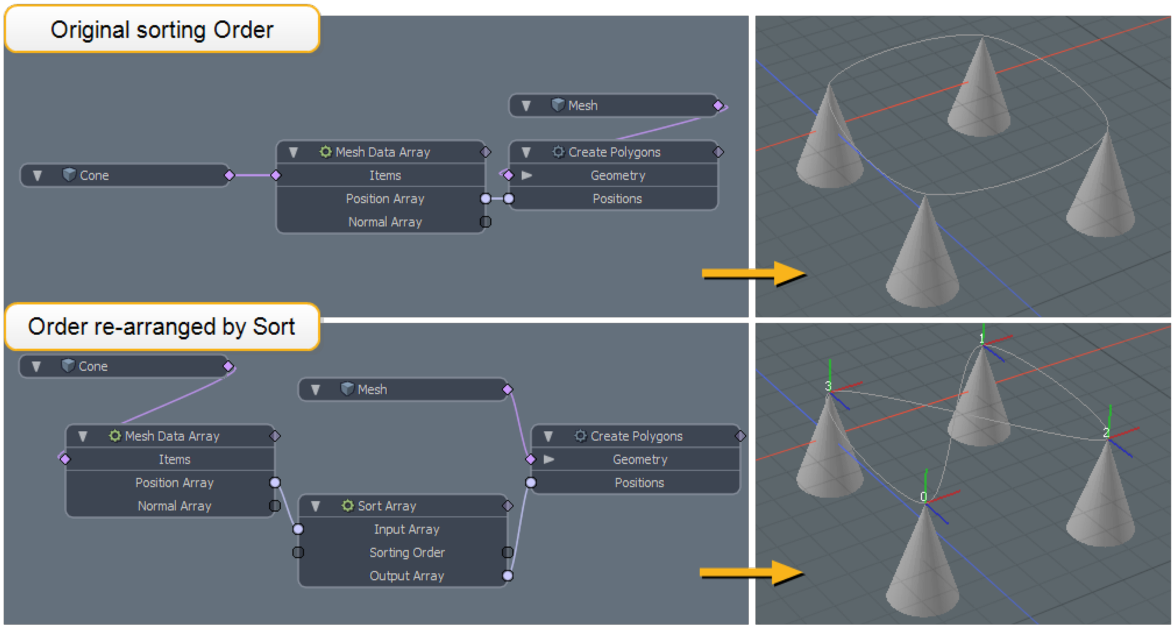
You can set the order using the **Sorting Tool**:

1. Select the **Sort Array** node, then in the **Properties** on the right, click **Sorting Tool**.
2. This creates points in the 3D view, which you can click to set the output array order.



3. Click the points in the order you want to sort them.

New indices are assigned to the points corresponding the order you clicked them. Any operation you connect to the Sort Array's **Output Array** channel will use the newly sorted array.


Original sorting Order
Mesh
Mesh Data Array
Items
Position Array
Normal Array
Create Polygons
Geometry
Positions
Cone
Order re-arranged by Sort
Cone
Mesh
Mesh Data Array
Items
Position Array
Normal Array
Create Polygons
Geometry
Positions
Sort Array
Input Array
Sorting Order
Output Array
0
1
2
3

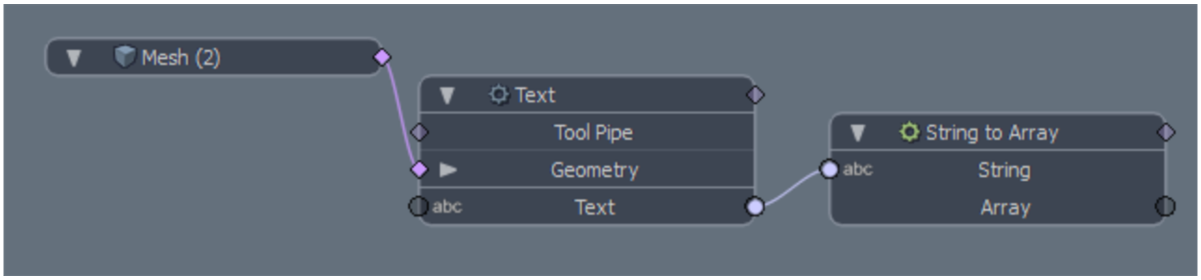
Note: You can also use a String to Array modifier to drive the **Sorting Order** channel.

Sort Array has the following channels:

Channels on the Node by Default	
Input Array	The input array to sort.
Sorting Order	The order you want the elements to be in within the array. Use the **Sorting Tool** in the modifier's **Properties** panel to specify the order.
Output Array	The sorted output array.
Additional Channels	
Layered Array	A True/False value to tell you if the input array contains any child arrays.
Data Type	Displays the type of data stored in the input array.

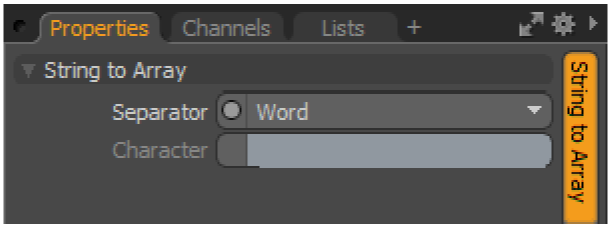
String to Array

The String to Array modifier Converts a string channel into an array, with options to split the string by word, letter, or a custom separator character.



To set how the modifier splits the array. To do this:

1. Select the String to Array Node to open its properties on the right panel.



Separator: Word

Character:

2. Set the **Separator** option. You can choose from the following:
   - **None** - The string is not split up. Instead, an array is created that contains the entire string as the first value.
   - **Character** - The string is split using a character, and any time Modo encounters the character in the string, a new entry is added into the array. For example, if the input string is "/some/file path/file name.ext", and the character specified is "/", the output array will be ("some", "file path", "file name.ext").

Note: The divider character can only be a single character, and not a string of multiple characters.

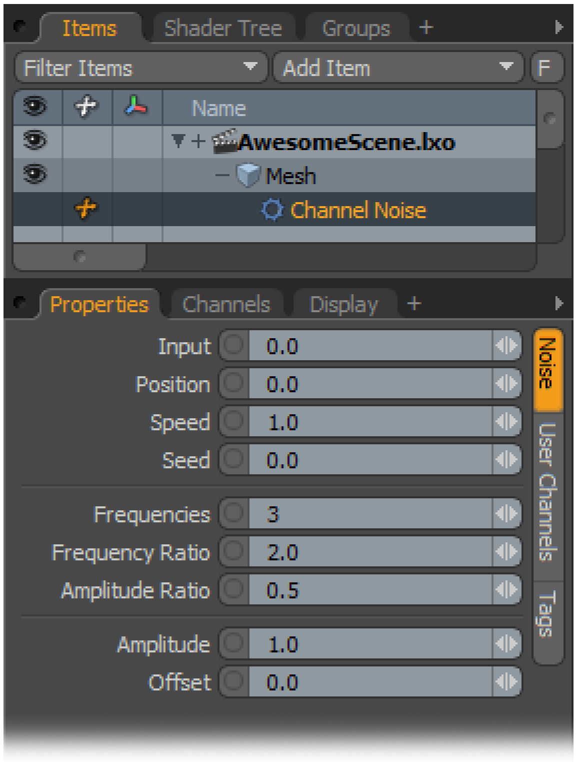
   - **Letter** - The string is split after each letter. For example, if the string is "file name.ext", the output array will be ("f", "i", "l", "e", " ", "n", "a", "m", "e", ".", "e", "x", "t").
   - **Word** - The string is split after every word, using the space character as a splitting character. For example, if the input string is "/some/file path/file name.ext", the output array will be ("/some/file", "path/file", "name.ext").

String to Array has the following channels and properties:

Channels on the Node by Default	
String	The string input used to generate the array.
Array	The array generated from the string.
Properties Panel	
Separator	Controls what element is used to split up the string. The following options are available: • **None** - The string is not split up. Instead, an array is created that contains the entire string as the first value. • **Character** - The string is split using a character, and any time Modo encounters the character in the string, a new entry is added into the array. For example, if the input string is "/some/file path/file name.ext", and the character specified is "/", the output array will be ("some", "file path", "file name.ext"). • **Letter** - The string is split after each letter. For example, if the string is "file name.ext", the output array will be ("f", "i", "l", "e", " ", "n", "a", "m", "e", ".", "e", "x", "t"). • **Word** - The string is split after every word, using the space character as a splitting character. For example, if the input string is "/some/file path/file name.ext", the output array will be ("/some/file", "path/file", "name.ext").
Character	When **Separator** is set to **Character**, specifies the character to be used.

Channel Modifiers

Channel Noise

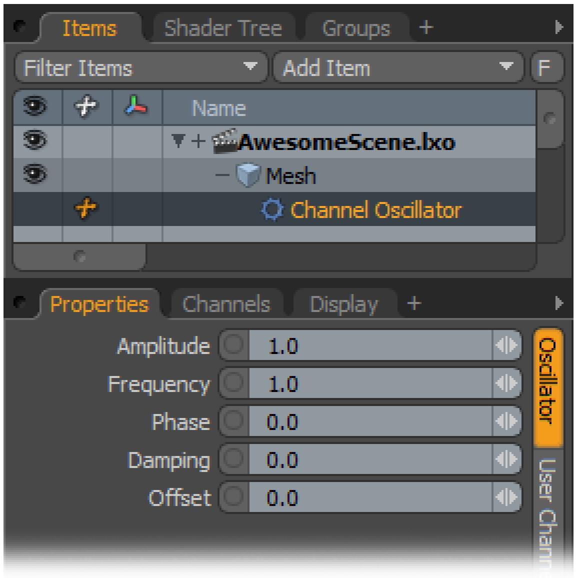

Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Channel Noise
Properties
Channels
Display
Noise
User Channels
Tags

Input	0.0
Position	0.0
Speed	1.0
Seed	0.0
Frequencies	3
Frequency Ratio	2.0
Amplitude Ratio	0.5
Amplitude	1.0
Offset	0.0

This motion generator creates a smoothly changing random value by taking a one-dimensional slice through a 3-dimensional fractal noise field.

- **Input**: This is the x-coordinate for the slice.
- **Position**: This is the y-coordinate for the slice.
- **Speed**: This indicates a scale time for the z-coordinate for the slice.
- **Seed**: This provides a random start time to move the z-coordinate.
- **Frequencies**: Indicates the number of octaves of noise to add together.
- **Frequency Ratio**: Determines the ratio of frequencies of successive octaves.
- **Amplitude Ratio**: Determines the ratio of amplitudes of successive octaves.
- **Amplitude**: This is a scaling factor for Modo to apply to the output.
- **Offset**: This specifies a value for Modo to add to the output.

Channel Oscillator



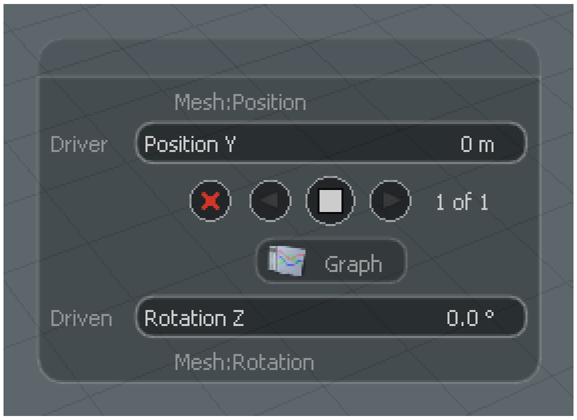
Items
Shader Tree
Groups
Filter Items
Add Item
Name
AwesomeScene.lxo
Mesh
Channel Oscillator
Properties
Channels
Display
Oscillator
User Chann

Amplitude 1.0
Frequency 1.0
Phase 0.0
Damping 0.0
Offset 0.0

This modifier can be considered more of a motion generator because it generates a standard sine wave. When applied to a channel, this modifier creates smoothly undulating values. For a more random effect, use a **Channel Noise** generator.

- **Amplitude**: This indicates the size of base waveform without damping. The channel ranges from -**Amplitude** value to +**Amplitude** value.
- **Frequency**: Determines the frequency of sine wave in cycles per second (Hz).
- **Phase**: Specifies a phase shift for the sine wave as a percent of a wavelength/cycle (for example, 90 deg shift = 90/360 = 25%).
- **Damping**: Indicates a reduction of the sine wave expressed as fraction of amplitude lost per cycle.
- **Offset**: This specifies a DC value for Modo to add to the sine wave. The output value then ranges from **Offset - Amplitude** to **Offset + Amplitude**.

Channel Relationship

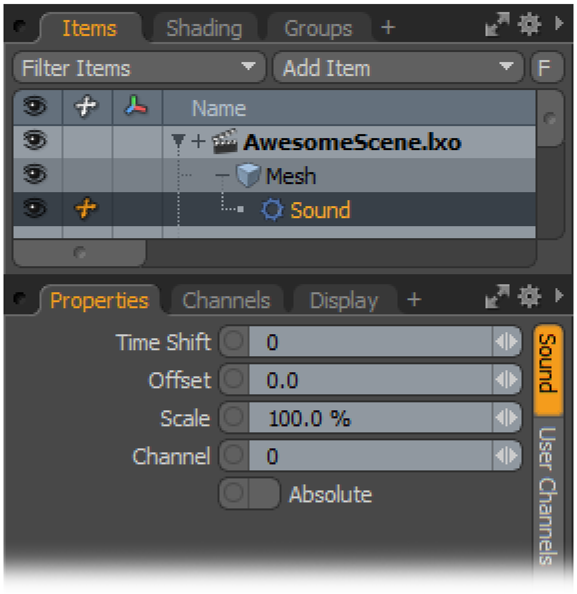


This modifier defines a non-linear relationship between two values that can be modified visually with a Graph curve. Using this modifier is unlike a standard direct link where, once applied, modifying the incoming channel instantly affects the outgoing channel. With this modifier, once you link the items, you must define the relationship itself numerically in the **Relationship** interface window in the 3D viewport. This window only appears when you select the **Channel Relationship** item, itself.

You define the relationships with the buttons in the window. Modo gives the two channels as **Driver** and **Driven**, which means that the values of one output (**Driver**) control the input of another (**Driven**). The square white button sets a relationship key based on the current **Driver** and **Driven** values. Click the arrows to jump between keys. The red **x** deletes the current key. Click the **Graph** button to open the Graph Editor refine the curve.

Channel Sound

Sound



Time Shift 0
Offset 0.0
Scale 100.0 %
Channel 0
☐ Absolute

The **Sound** Channel Modifier uses the amplitude of a source waveform as a channel input value and acts much like the Oscillator Channel Modifier. The output range is -1.0 to 1.0. This range can be multiplied by the **Scale** value and compensated up or down with the **Offset** value. For example, an **Offset** of 1.0 and a **Scale** of 0.5 normalizes all the values to be greater than 0. (This may helpful if the output does produce negative values.)

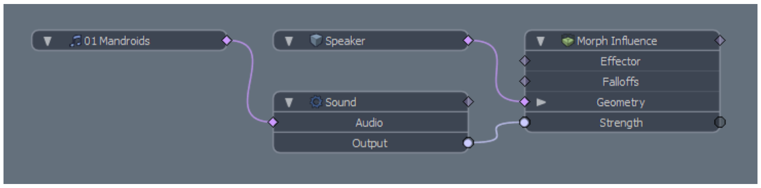
Time Shift: Shifts the time for the Audio forwards or backwards to offset it in frames. With this, the animation can happen before or after the audio event itself.

Offset: This value determines an amount of compensation (either positive or negative) away from the 0 center value.

Scale: This value acts as a multiplier to the sound output, which is limited to floating-point values between -1.0 and 1.0.

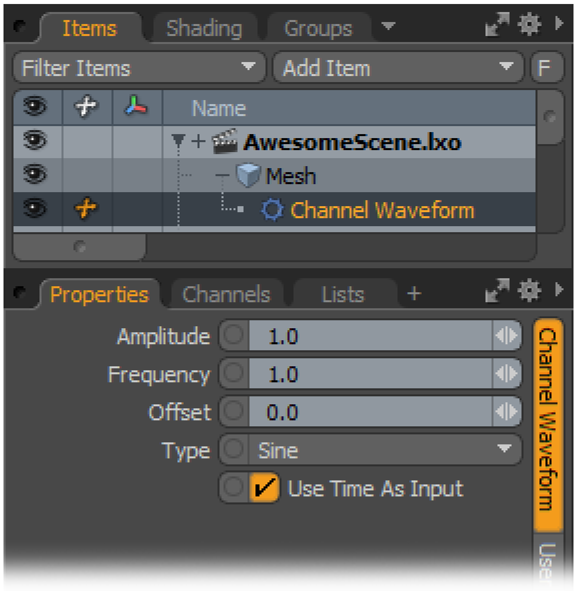
Channel: This specifies the channel of the incoming audio when more than a single channel is present (such as left and right with a stereo file). **0** represents the left channel and **1** represents the right channel. For a combination of both, convert the file to a mono file in an audio-editing application.

Absolute: Select this to convert all the positive and negative output values to positive values.


01 Mandroids
Speaker
Morph Influence
Effector
Falloffs
Geometry
Strength
Sound
Audio
Output

This example shows using a **Sound** channel to drive a Morph animation's **Strength** amount.

Channel Waveform


Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Channel Waveform
Properties
Channels
Lists
+
Channel Waveform
Use

Amplitude 1.0
Frequency 1.0
Offset 0.0
Type Sine
☑ Use Time As Input

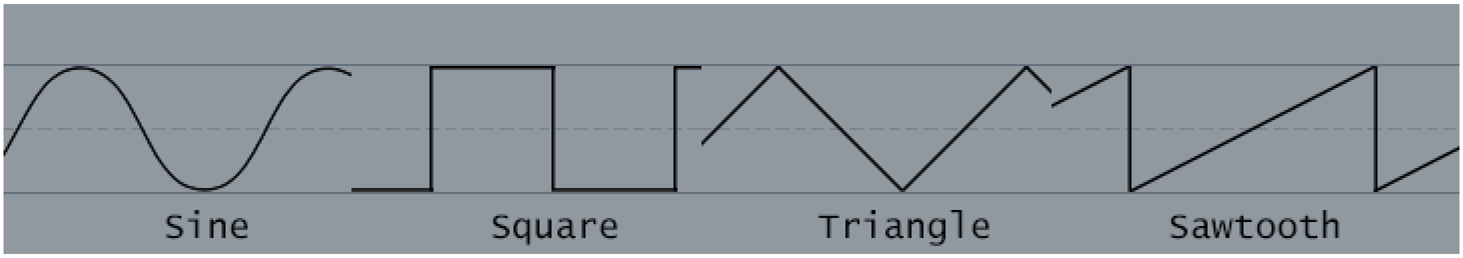
The **Channel Waveform** modifier outputs an oscillating value based on a specified waveform type. Waveforms are graphs that show changes to amplitude over time. With Modo, you can consider these motion generators for creating regular, repeating motions.

Amplitude: This controls the maximum output value between a high and low value. A value of **1.0** oscillates between 1.0 and -1.0.

Frequency: This determines how regular the oscillating motion occurs per second. A value of **1** produces 1 up and down cycle per second; a value of **10** produces 10 cycles per second. Higher values produce a faster oscillation.

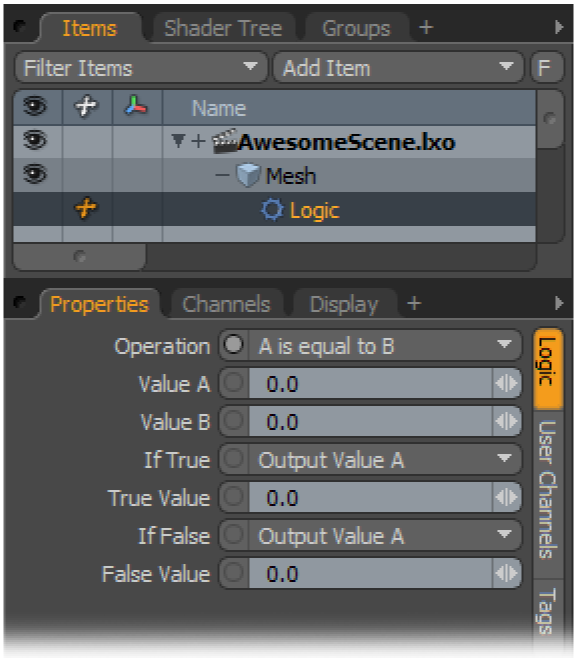
Offset: This value determines an amount of compensation, either positive or negative, away from the **0** center value.

Type: This defines the type of waveform that Modo generates.


Sine
Square
Triangle
Sawtooth

Use Time as Input: To produce a result, the waveform needs timing information to determine the **Frequency** value. This is typically a **Time** channel item linked to the **Waveform** input in the **Schematic** viewport. By selecting **Use Time as Input** you can avoid this extra step by having Modo generate the time input automatically.

Conditional Modifiers


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Logic
Properties
Channels
Display
Logic
User Channels
Tags

Operation: A is equal to B
Value A: 0.0
Value B: 0.0
If True: Output Value A
True Value: 0.0
If False: Output Value A
False Value: 0.0

With the Logic modifier, you apply logic to channel operations.

Operation: You can select from the various logic operation options available in this list:

- **A is equal to B**
- **A is greater then B**
- **A is greater than or equal to B**
- **A is less than B**

- **A is less than or equal to B**
- **A is not equal to B**
- **And** (The result is **True** if both A and B are not zero.)
- **Or** (The result is **True** if A or B are not zero.)

Value A/Value B: These indicate the values of A or B for the operation. Typically you connect the output of another item's channel to these channels for the logic operation.

If True: Specify what Modo outputs when the logic modifier is **True**:

- **Output Value A**
- **Output Value B**
- **Output True Value** as specified below

True Value: Specify the value to use if you select **Output True Value**.

If False: Specify what Modo outputs when the logic modifier is **False**.

- **Output Value A**
- **Output Value B**
- **Output False Value** as specified below

False Value: Specify the value to use if you select **Output False Value**.

Mesh Info Modifier

The Mesh Info modifier allows you to query information about a connected mesh. It outputs the bounding box, surface area, number of elements, and so on. This information can be queried in world space or local space, and has options to query the base mesh or the deformed mesh.

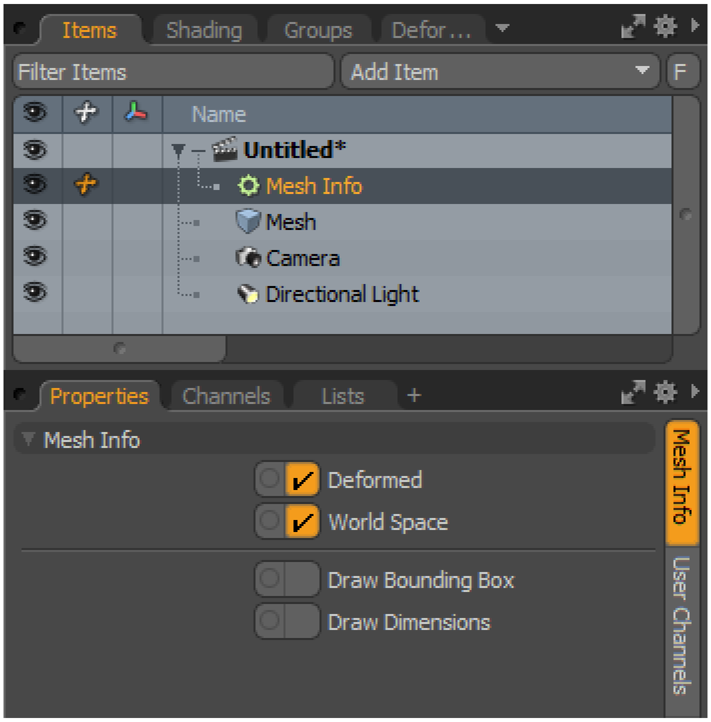
Drawing options are provided, allowing the axis-aligned bounding box of the mesh to be drawn, along with dimensions along the three main axes.

To insert a Mesh Info modifier, follow the steps below:

1. In the **Setup** layout, click the **Modifiers** sub-tab on the left and then select **Add Channel Modifier...**.

   The **Create Modifier** window opens.
2. For **Type**, select **Geometry** > **Mesh Info**.

   The **Mesh Info** channel modifier is added to your scene.


Items
Shading
Groups
Defor ...
Filter Items
Add Item
F
Name
Untitled*
Mesh Info
Mesh
Camera
Directional Light
Properties
Channels
Lists
+
Mesh Info
Mesh Info
User Channels

☑ Deformed

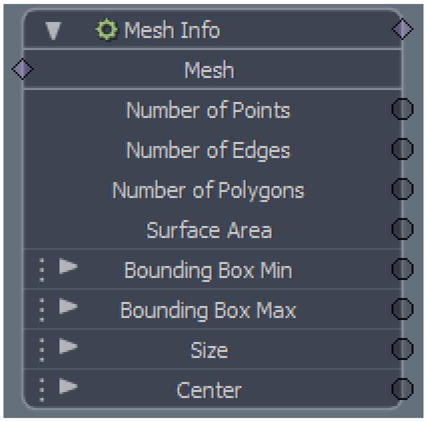
☑ World Space

☐ Draw Bounding Box

☐ Draw Dimensions

You can also insert the Mesh Info modifier in the **Schematic** viewport, by clicking **Add...** on top of the viewport, and selecting **Geometry** > **Mesh Info**.

Along with the following channels, a relationship graph input is provided on the node, allowing any standard Mesh Item to be connected.


Mesh Info
Mesh
Number of Points
Number of Edges
Number of Polygons
Surface Area
Bounding Box Min
Bounding Box Max
Size
Center

Name	Description
Deformed Mesh	Specifies whether the deformed mesh should be queried for its properties. If disabled, the properties of the base mesh are returned.
World Space	When enabled, the bounding box is evaluated with any item transforms on

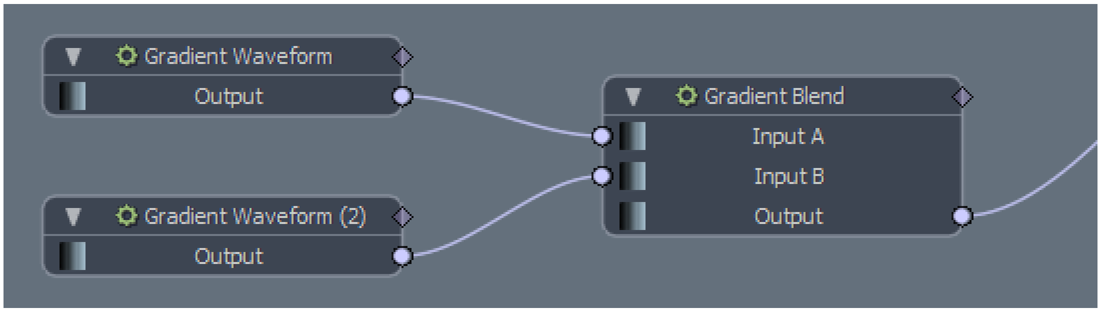
Name	Description
	the mesh taken into account. When disabled, the bounding box is returned as if the mesh is at the center of the world.
Number of Points	Outputs the number of points on the connected mesh.
Number of Edges	Outputs the number of edges on the connected mesh.
Number of Polygons	Outputs the number of polygons on the connected mesh.
Surface Area	Outputs the surface area of the connected mesh.
Bounding Box Min	Outputs the minimum vector of the axis-aligned bounding box.
Bounding Box Max	Outputs the maximum vector of the axis-aligned bounding box.
Size	Outputs the size of the bounding box. This is the distance from the **Bounding Box Min** to the **Bounding Box Max**.
Center	Outputs the center of the bounding box.
Draw Bounding Box	When this option is enabled, the calculated bounding box is drawn in the GL viewport. **Note:** You can override this drawing using the standard **Draw** channel, available on all channel modifiers.
Draw Dimensions	When this option is enabled, the dimensions of the calculated bounding box are drawn in the GL viewport, along the three main axes of the bounding box. **Note:** You can override this drawing using the standard **Draw** channel, available on all channel modifiers.

Gradient Modifiers

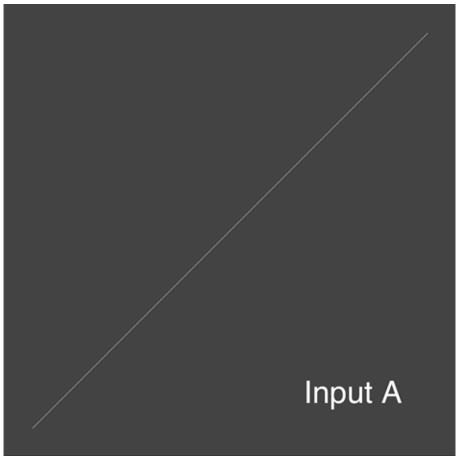
Gradient Modifiers read an incoming gradient, and output a modified gradient curve. For example, if the incoming gradient is linear from 0.0 to 1.0, and the modifier is a Gradient Offset node with a value of 1.0 in the Y axis, the resulting gradient is a linear gradient from 1.0 to 2.0. Gradient Modifiers can be created like any standard channel modifier in the Schematic viewport, and their inputs and outputs can be connected to Gradient channels on any item, including Gradient user channels.

Gradient Blend

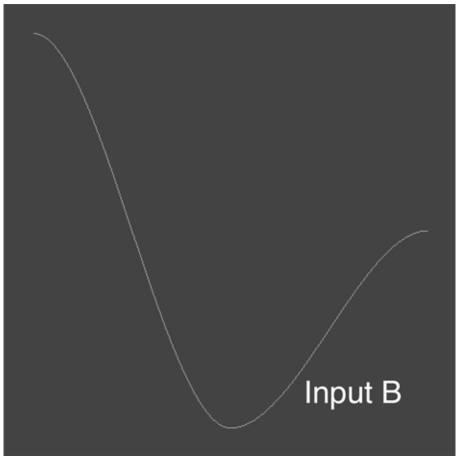
The Gradient Blend modifier performs a linear interpolation between two gradients.



Example

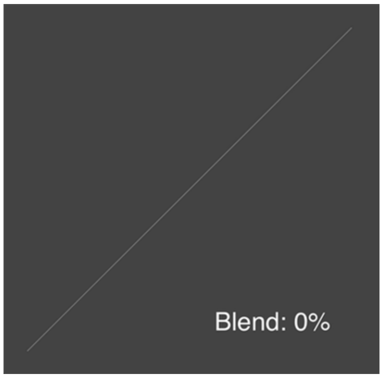


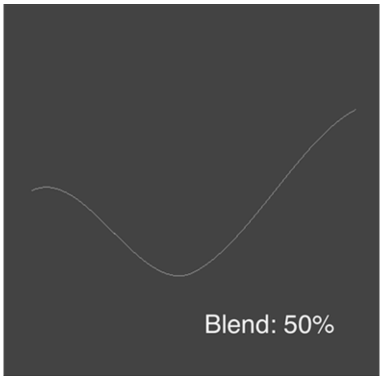
Input A is a linear gradient.

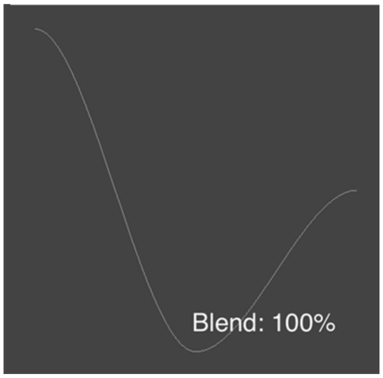


Input B is a curved gradient.

With a blend amount of 0%, the linear gradient is the output. With a blend amount of 100%, the curved gradient is the output. If the blend amount is 50%, the output is a mix of the linear and the curved gradient.





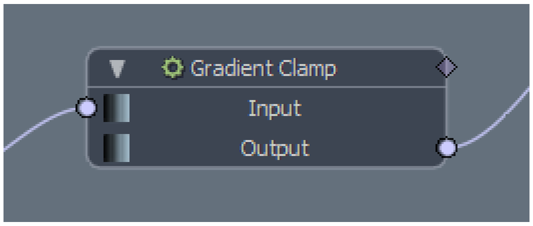


Gradient Blend Properties

Channels on the Node by Default	
Input A	The gradient to blend from.
Input B	The gradient to blend to.
Output	The blended gradient.
Properties Panel	
Mode	The type of value used for modulating the interpolation. The following options are available: • **Scalar** - The default value. The blend is uniform along the length of the gradient, controlled by the **Blend Amount** channel. • **Gradient** - The blend varies along the length of the gradient, controlled by the **Blend Gradient** channel.
Blend	When **Mode** is set to **Scalar**, specifies the amount to blend between Input A and Input B.

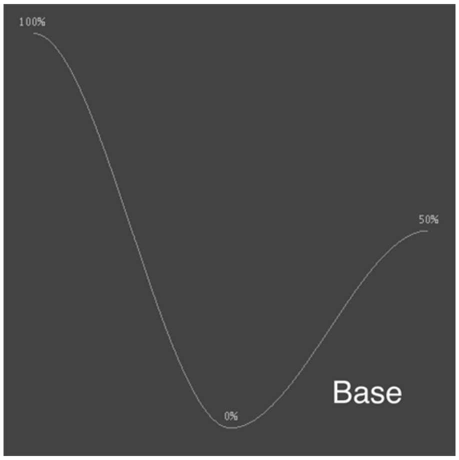
Gradient Clamp

The Gradient Clamp modifier clamps the gradient output value, ensuring that the value never exceeds the specified minimum and maximum range.

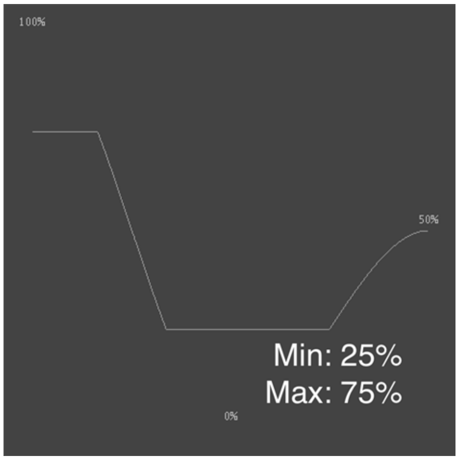


Example

The following images show a gradient clamped at a **Minimum Value** of 0.25 and a **Maximum value** of 0.75.



The input gradient.



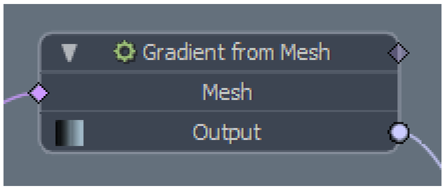
The clamped gradient.

Gradient Clamp Properties

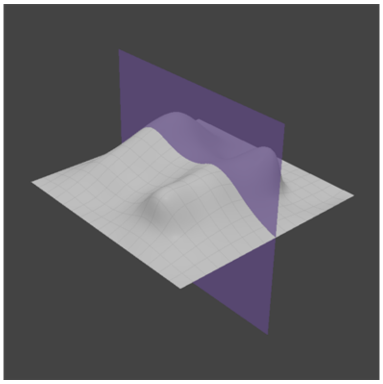
Channels on the Node by Default	
Input	The gradient to clamp.
Output	The clamped gradient.
Properties Panel	
Minimum Value	The lower end of the clamped range.
Maximum Value	The upper end of the clamped range.

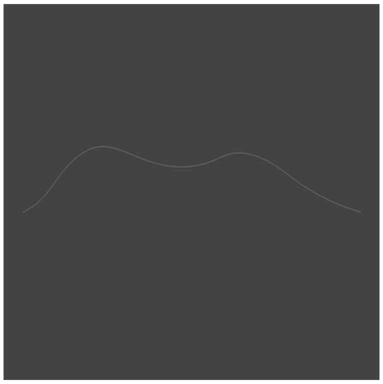
Gradient from Mesh

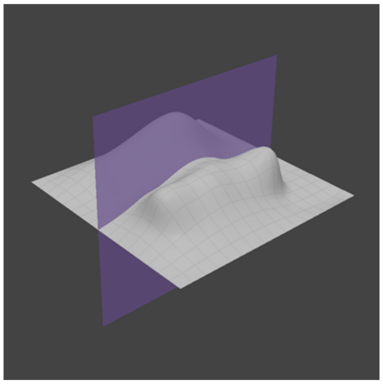
The Gradient from Mesh modifier generates a gradient from a mesh layer. It maps the height of the mesh in either the X, Y, or Z axis to the value of the gradient.

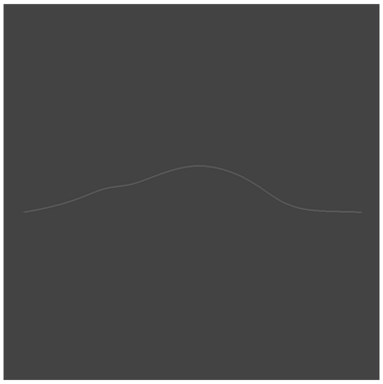


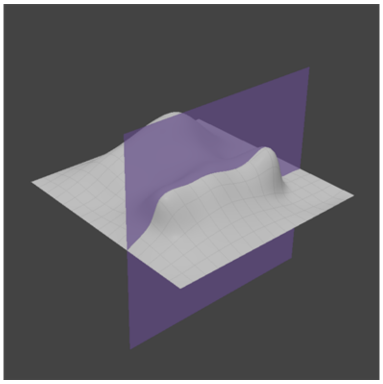
The modifier takes a slice through the input mesh, and raycasts into the slice to extract the height of the surface. You can specify the axis and offset for the slice, as well as animate it, allowing the slope of the gradient to be modified over time.







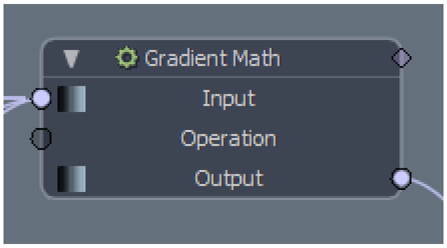




Channels on the Node by Default	
Mesh	The mesh used for producing the gradient.
Output	The computed gradient.
Properties Panel	
Axis	The axis along which the raycast is projected. The height of the mesh in the axis determines the value of the gradient.
Slice Axis	The axis along which the mesh is sliced. U and V maps to the two axes not specified as **Axis** for raycasting. For example, if the raycasting axis is Y, U and V in the **Slice Axis** represent X and Z respectively.
Slice Offset	The offset along the slice axis to slice the mesh. Animating this value moves the slicing plane forwards and backwards, animating the value of the gradient.
Range Minimum/Range Maximum	The bounding box of the mesh is read as the extremes of the gradient, but the way they map to gradient values is determined by the **Range Minimum** and **Range Maximum** options. Changing these values control how the smallest and largest extremes of the bounding box map to the smallest and largest time values in the gradient.

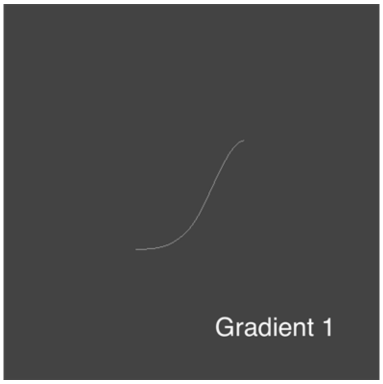
Gradient Math

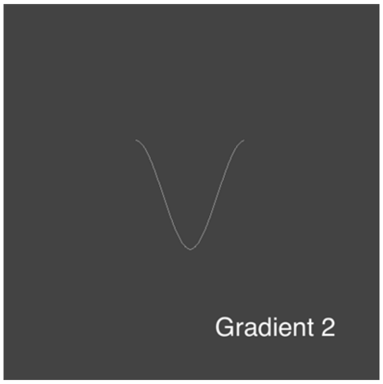
The Gradient Math modifier performs basic math operations on any number of gradient inputs.

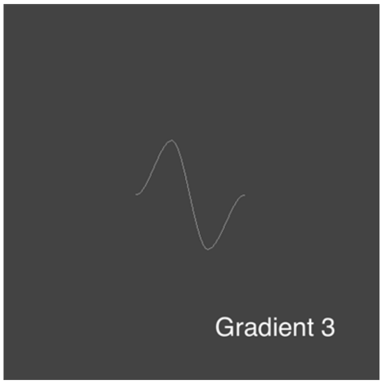

Gradient Math
Input
Operation
Output

Example

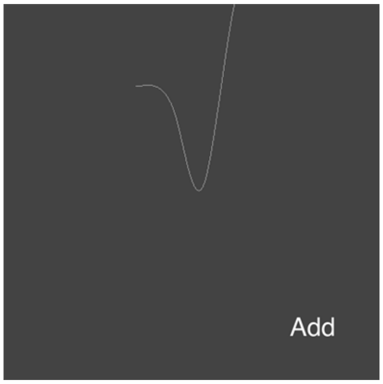
Let's take these three input gradients:

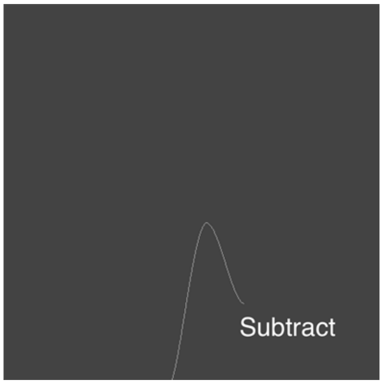


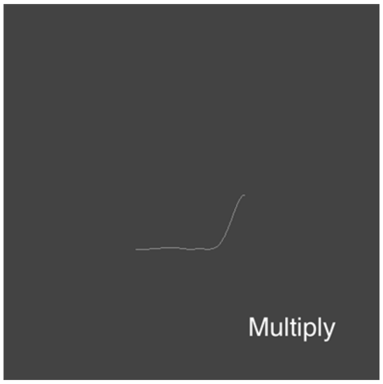


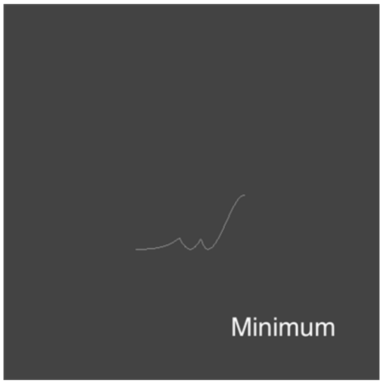


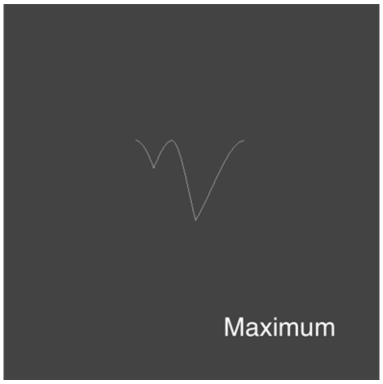
The following results are generated using each math operation:

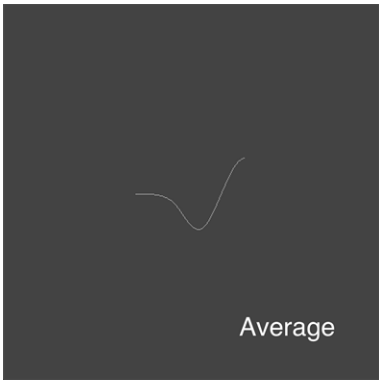








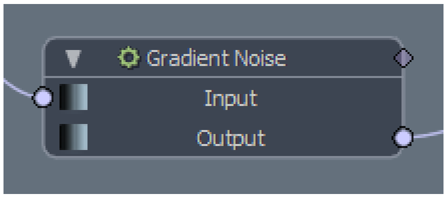


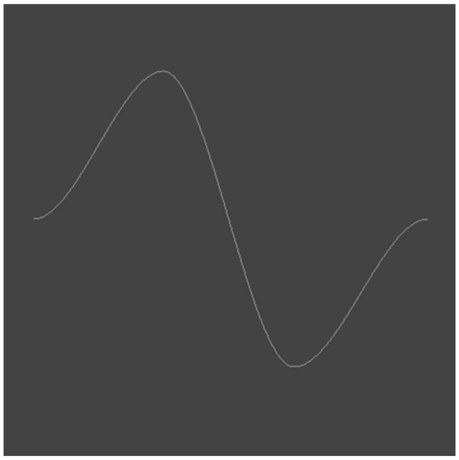


Channels on the Node by Default	
Input	The gradients to perform math operations on. This input allows multiple connections.
Operation	The mathematical operation to perform. The available options are: • **Add** - Adds the gradients to one another and outputs the computed result. • **Subtract** - Subtracts each gradient in turn, from a flat gradient at zero. • **Multiply** - Multiplies the gradients by one another and outputs the computed result. • **Minimum** - Returns a gradient containing the smallest values picked from each of the input gradients. • **Maximum** - Returns a gradient containing the largest values picked from each of the input gradients. • **Average** - Returns the average of all input gradients.
Output	The computed gradient.

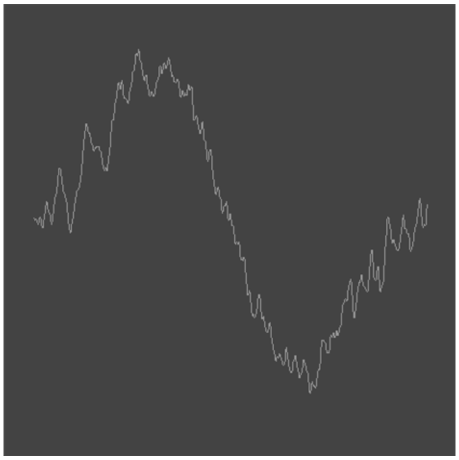
Gradient Noise

The Gradient Noise modifier applies Perlin Noise to an existing gradient. The modifier provides control over both noise amplitude and frequency.





The input gradient.



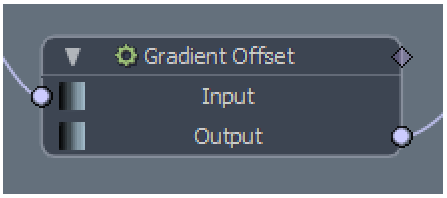
The gradient with noise applied to it.

Gradient Noise Properties

Channels on the Node by Default	
Input	The source gradient.
Output	The computed gradient with noise applied.
Properties Panel	
Frequency	Determines how regularly the noise occurs per second.
Amplitude	Controls how strong the noise offsets the original base gradient.
Seed	Seeds the randomness of the noise generator. Animating this value animates the noise.

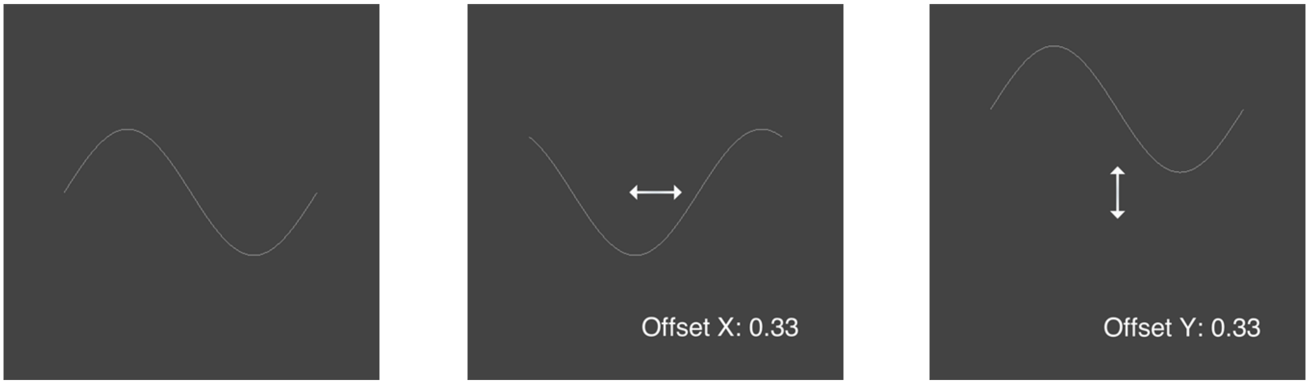
Gradient Offset

The Gradient Offset node offsets all the values in a gradient equally, either by offsetting the value in Y, and/or the time the value is sampled in X.



Example

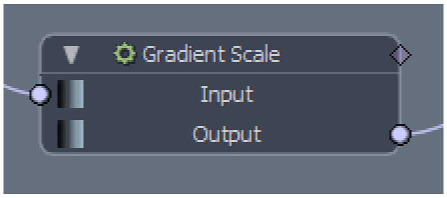
This example shows the source gradient offset by 0.33 on X and by 0.33 on Y.



Channels on the Node by Default	
Input	The gradient to offset.
Output	The offset gradient.
Properties Panel	
Offset X	Offsets the gradient value in X. This offsets the sample time, so reading the gradient at a specific time sample remaps the time sample to read the input gradient at a different time.
Offset Y	Remaps the gradient value, adding the Offset Y value to the input gradient sampled value.

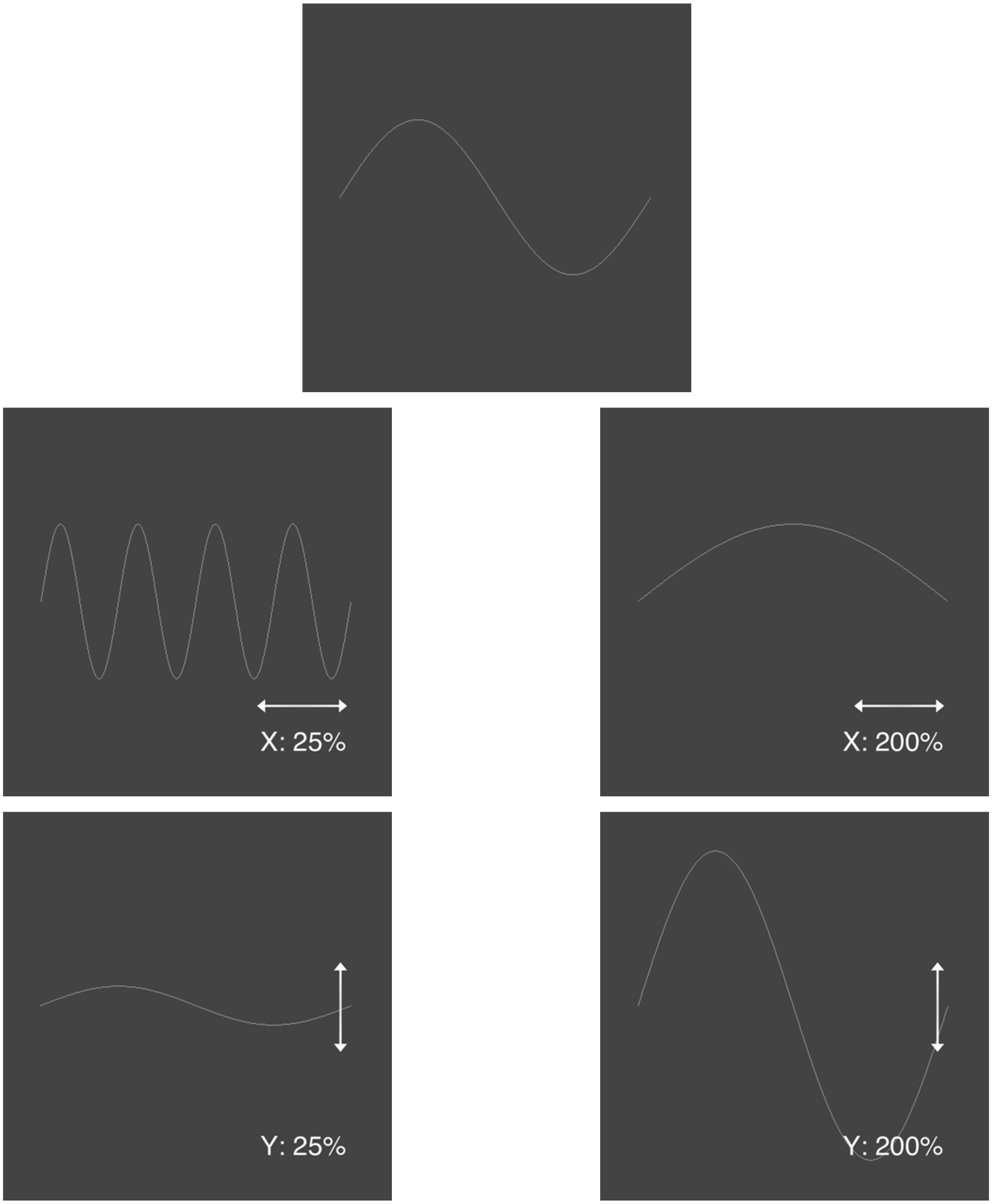
Gradient Scale

The Gradient Scale modifier scales a gradient either in the Y axis, modifying the gradient's value, or the X axis, modifying the time of each value. You can also control the pivot point for each axis the scaling is performed in.



Example

This example shows the input gradient scaled on the X and then the Y axis by 0.25 and 2.0.

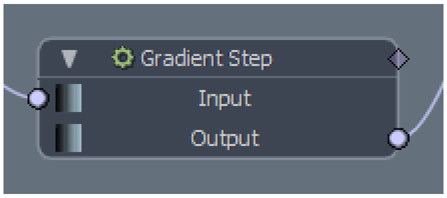


Gradient Scale Properties

Channels on the Node by Default	
Input	The gradient to scale.
Output	The scaled gradient.
Properties Panel	
Amount X/Y	The amount the input gradient is scaled in the X/Y axis. Changing X modifies the sample time of each value in the gradient. Changing Y increases or decreases the gradient values.
Center X/Y	The pivot point to perform the scale in the X/Y axis. The gradient is scaled from this position.

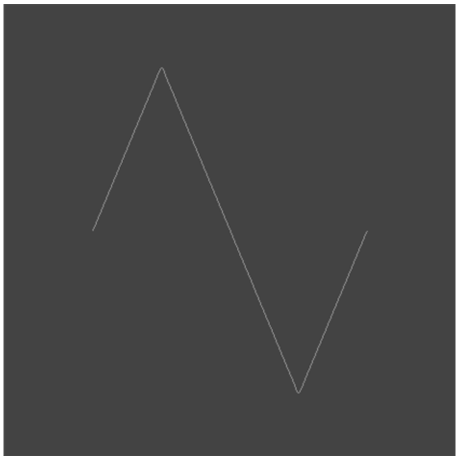
Gradient Step

The Gradient Step modifier rounds values in a floating point gradient to their nearest whole number. For example, 0.25 becomes 0.0, and 0.6 becomes 1.0. This can introduce a step pattern to gradients that cover a large range of floating point values.

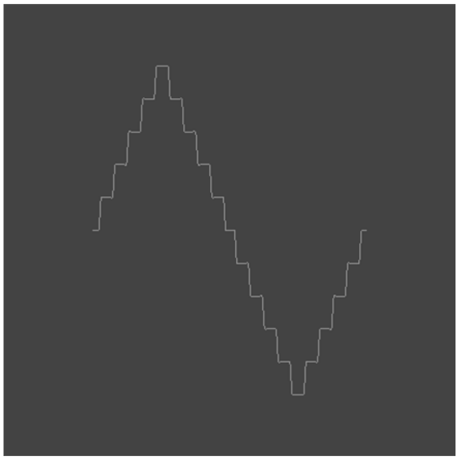


Example

In this example, the base gradient has a **Maximum Value of** 5.0 and a **Minimum Value** of -5.0. This introduces 11 steps across the range, as every value between -5.0 and +5.0 (including 0.0) is rounded to their nearest whole number.



The source gradient.



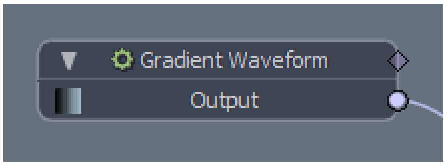
The output gradient.

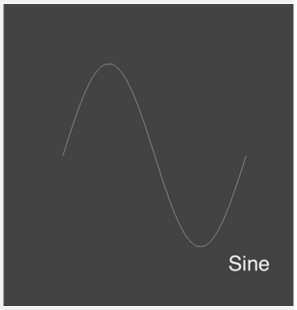
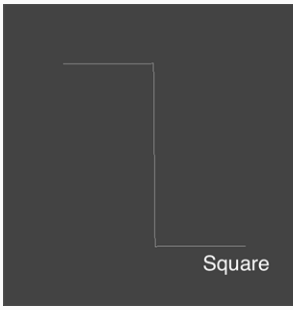
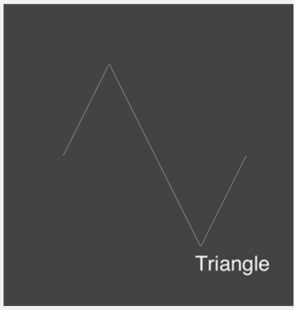
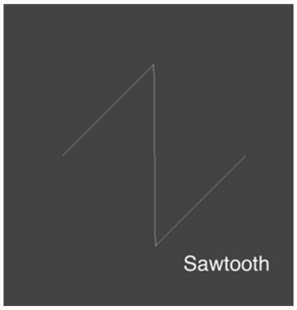
Gradient Step Properties

Channels on the Node by Default	
Input	The input gradient.
Output	The stepped gradient.

Gradient Waveform

The Gradient Waveform modifier generates a gradient from four standard waveform presets: Sine, Square, Triangle, and Sawtooth. You can use this to create regular repeated gradients.



	A sinusoidal waveform that describes a smooth periodic oscillation from one extreme of the amplitude to the other.
	A square wavefrom that rounds to the nearest amplitude value.
	A triangle waveform that blends in linearly from one extreme of the amplitude to the other.
	A sawtooth waveform that blends linearly to the positive extreme of the amplitude, and then snaps back to the negative extreme of the amplitude.

Gradient Waveform Properties

Channels on the Node by Default	
Output	The gradient containing the waveform.

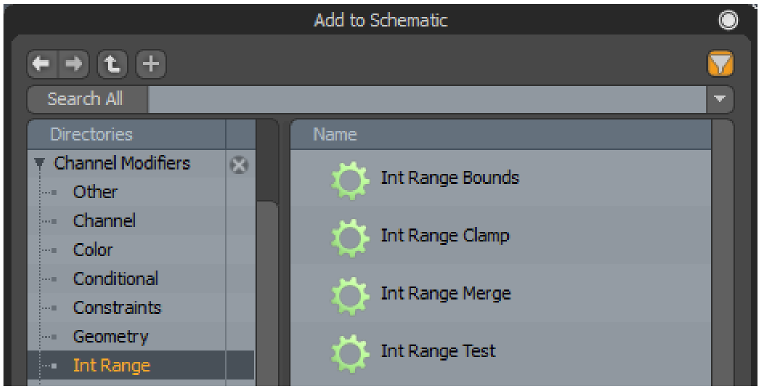
Properties Panel	
Type	The type of waveform to generate: • Sine • Square • Triangle • Sawtooth
Amplitude	The highest and lowest peak of the waveform. A value of 1.0 produces a low of -1.0, and a high of 1.0.
Frequency	Determines how regularly the oscillating motion occurs per second. A value of 1.0 produces one up and down cycle per 1.0 time step.
Offset	Offsets the gradient output value in the Y axis.

Integer Range Modifiers

The following modifiers allow you to manipulate integer ranges. This can be useful in various rigging and procedural modeling situations, such as handling procedural selection values.

To add an Int Range modifier to your scene, at the top of the Schematic viewport, click **Add...** then under **Int Range**, double-click the modifier.

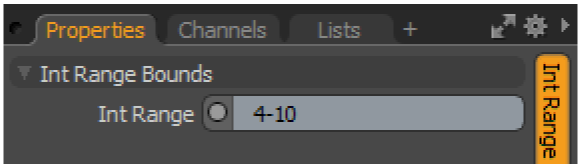
Note: The Schematic viewport is available in the **Setup** layout, or can be opened from the menu bar by clicking **Layout** > **Palettes** > **Schematic**. For more information, see Schematic Viewport.


Add to Schematic
Search All
Directories
Channel Modifiers
Other
Channel
Color
Conditional
Constraints
Geometry
Int Range
Name
Int Range Bounds
Int Range Clamp
Int Range Merge
Int Range Test

Int Range Bounds

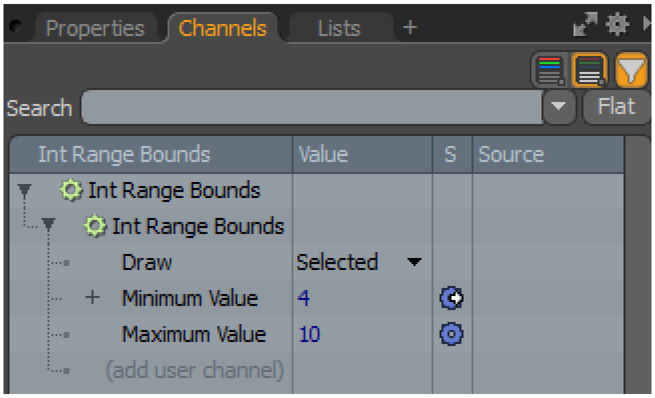
The Int **Range Bounds** modifier takes an integer range and gives you the minimum and the maximum values in that range.

In the Int Range Bound **Properties** on the right panel, enter an integer range to the **Int Range** field.


Properties
Channels
Lists
+
Int Range Bounds
Int Range

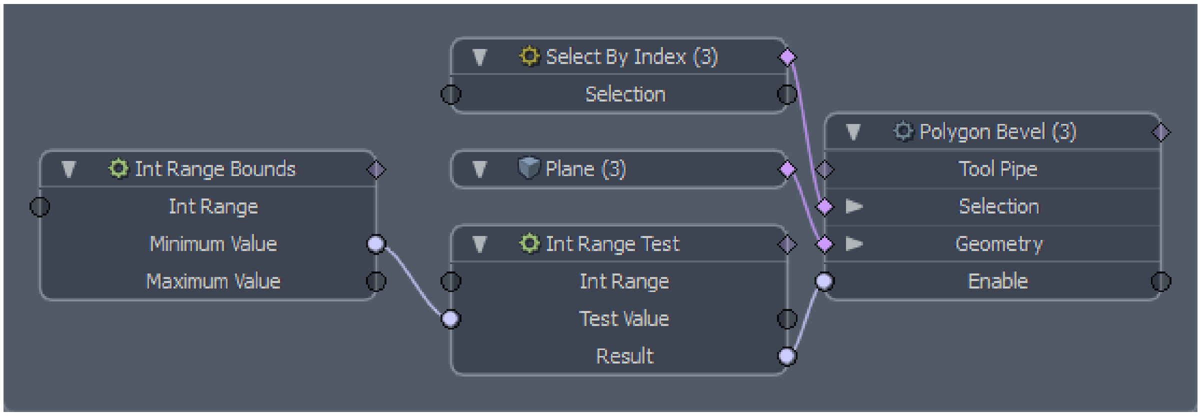
Int Range 4-10

The modifier's **Minimum Value** and **Maximum Value** channels are populated according to the range you specified.


Properties
Channels
Lists
+
Search
Flat

Int Range Bounds	Value	S	Source
Int Range Bounds			
Int Range Bounds			
Draw	Selected		
Minimum Value	4		
Maximum Value	10		
(add user channel)			

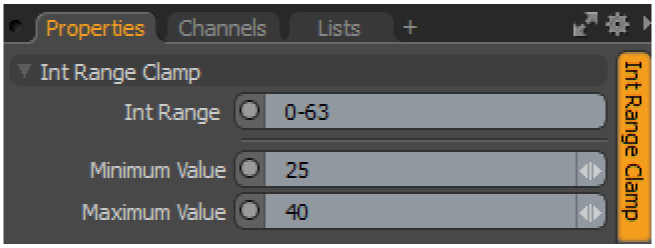
In the example below, an **Int Range Bounds** modifier's **Minimum Value** serves as a **Test Value** for the Int Range Test modifier, which in turn drives the **Polygon Bevel** operation:


Select By Index (3)
Selection
Polygon Bevel (3)
Tool Pipe
Selection
Geometry
Enable
Int Range Bounds
Int Range
Minimum Value
Maximum Value
Plane (3)
Int Range Test
Int Range
Test Value
Result

Int Range Clamp

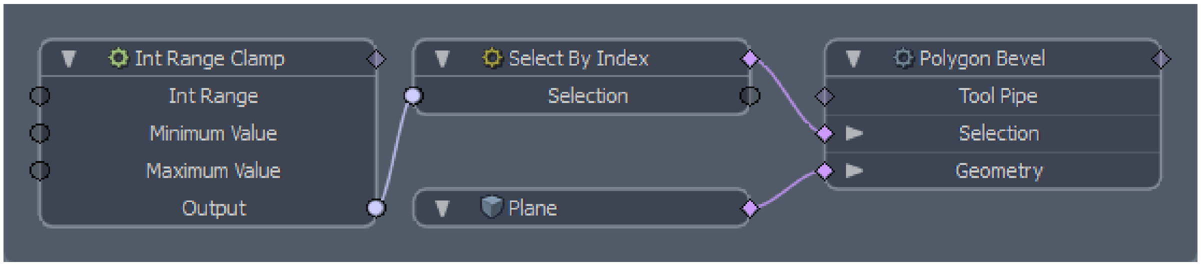
The **Int Range Clamp** modifier limits an output value to the range defined by the **Minimum** and **Maximum** values. Applying this modifier ensures that the output channel never has a value lower than the **Minimum Value** or higher than the **Maximum Value**.

In the Int Range Clamp **Properties** on the right panel, enter an integer range to the **Int Range** field, and specify the **Minimum Value** and **Maximum Value**.


Properties
Channels
Lists
+
Int Range Clamp

Int Range Clamp
Int Range: 0-63
Minimum Value: 25
Maximum Value: 40

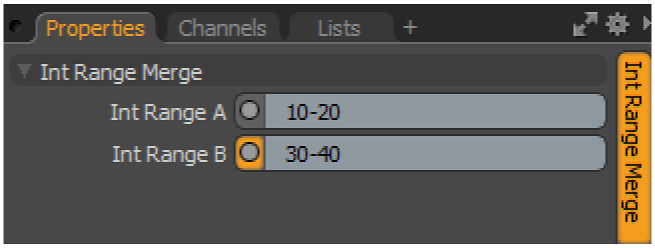
In the above example, the integer range of **0-63** is limited to **25-40**. On the image below, the **Output** value **Int Range Clamp** modifier defines the selection for the **Polygon Bevel** operation applied to the **Plane** mesh.


Int Range Clamp
Int Range
Minimum Value
Maximum Value
Output
Select By Index
Selection
Plane
Polygon Bevel
Tool Pipe
Selection
Geometry

Int Range Merge

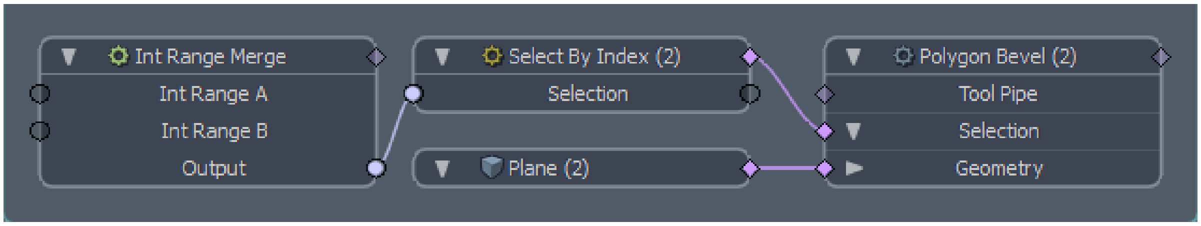
The **Int Range Merge** modifier allows you to specify two integer ranges and merge them into a single one.

In the Int Range Merge **Properties** on the right panel, enter the two ranges to merge in the **Int Range A** and **Int Range B** fields.


Properties
Channels
Lists
+
Int Range Merge

Int Range Merge
Int Range A: 10-20
Int Range B: 30-40

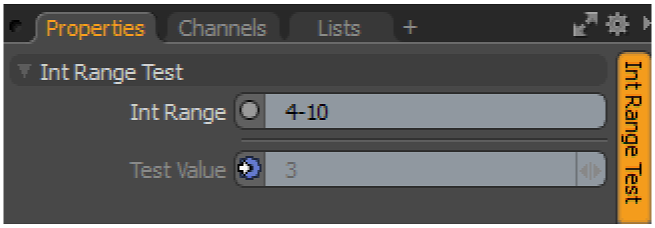
In the example below, the **Output** of the **Int Range Merge** modifier defines the selection for the **Polygon Bevel** operation applied to the **Plane (2)** mesh.


Int Range Merge
Int Range A
Int Range B
Output
Select By Index (2)
Selection
Plane (2)
Polygon Bevel (2)
Tool Pipe
Selection
Geometry

Int Range Test

The **Int Range Test** modifier allows you to specify an integer range and a **Test Value**, and outputs either a true or a false based on whether the value is part of the specified range.

In the Int Range Test **Properties** on the right panel, enter an integer range in the **Int Range** field. You can either enter a **Test Value**, or drive this channel using another modifier. In the example below, the channel is driven by the **Minimum Value** of the **Int Range Bounds** modifier.

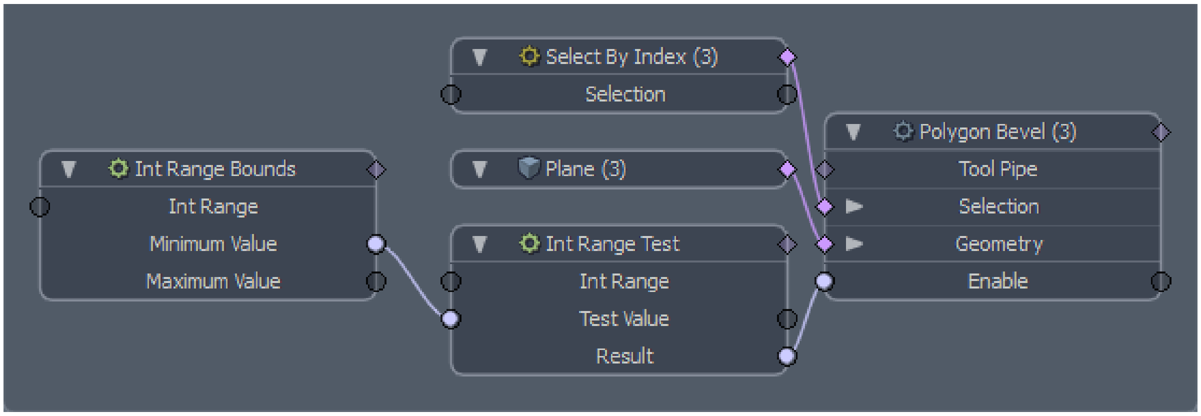

Properties
Channels
Lists
+
Int Range Test

Int Range Test

Int Range: 4-10

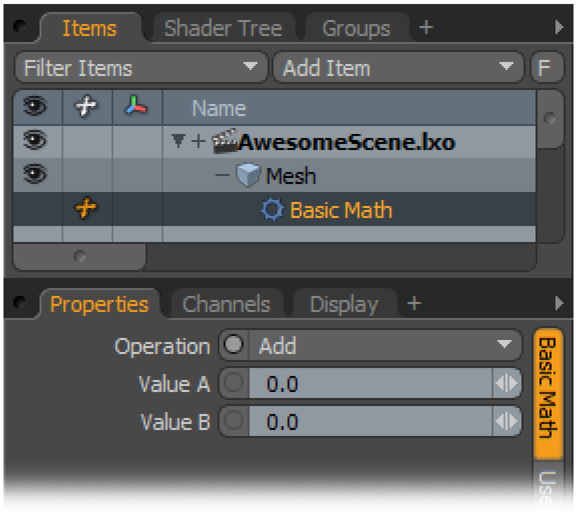
Test Value: 3

In the example below, the **Int Range Test Modifier**'s **Test Value** is determined by the **Minimum Value** of an Int Range Bounds modifier. The **Int Range Test** modifier's **Result** output in turn drives the **Polygon Bevel** operation. If the value is in range, the operation is applied to the mesh. If not, the operation is disabled.


Select By Index (3)
Selection
Polygon Bevel (3)
Tool Pipe
Selection
Geometry
Enable
Int Range Bounds
Int Range
Minimum Value
Maximum Value
Plane (3)
Int Range Test
Int Range
Test Value
Result

Math Modifiers

Basic Math

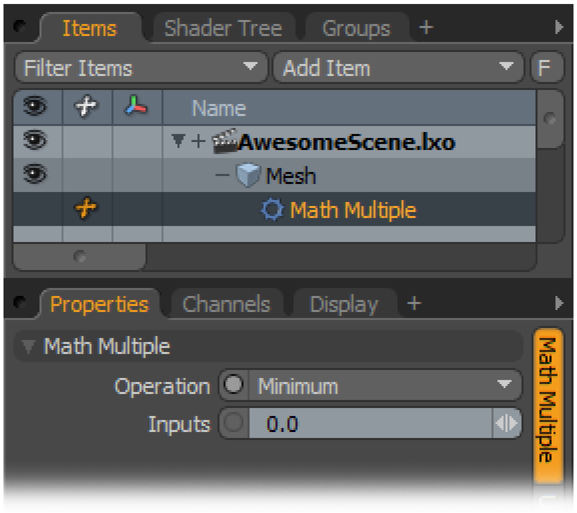

Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Basic Math
Properties
Channels
Display
Operation
Add
Value A
0.0
Value B
0.0
Basic Math

With this Channel Modifier Modo performs basic math operations and passes the result to a channel that can then be read by other Channel Modifiers or linked to a compatible numeric channel on any item. The inputs can be constant or animated values on the **Value A** or **Value B** channels of the modifier itself, or you can link values from other channels. The channel values for the inputs are **A** and **B**. The output value is **C**.

Operations: You can select from these basic math operations:
Add- Value A + Value B = Output C
Subtract- Value A - Value B = Output C
Multiply- Value A * Value B = Output C
Divide- Value A / Value B = Output C
Modulo- The remainder of Value A / Value B = Output C
Average- The average of Value A and Value B = Output C
Minimum- The lesser of Value A and Value B = Output C
Maximum- The greater of Value A and Value B = Output C

Value A/Value B: You can link the input of other channels into this modifier to calculate the output value and link the output channel to the target channel.

Math Multiple


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Math Multiple
Properties
Channels
Display
Math Multiple
Math Multiple

Operation: Minimum

Inputs: 0.0

You can define multiple inputs for the math operation and then pass the result to the result (**C**) output channel.

Operations: You can select from these basic math operations:

- **Add**- Value A + Value B = Output C
- **Subtract**- Value A - Value B = Output C
- **Multiply**- Value A * Value B = Output C
- **Divide**- Value A / Value B = Output C
- **Average**- The average of Value A and Value B = Output C
- **Minimum**- The lesser of Value A and Value B = Output C
- **Maximum**- The greater of Value A and Value B = Output C

Inputs: This is the input value for the numeric calculation. Typically, you link this to another channel.

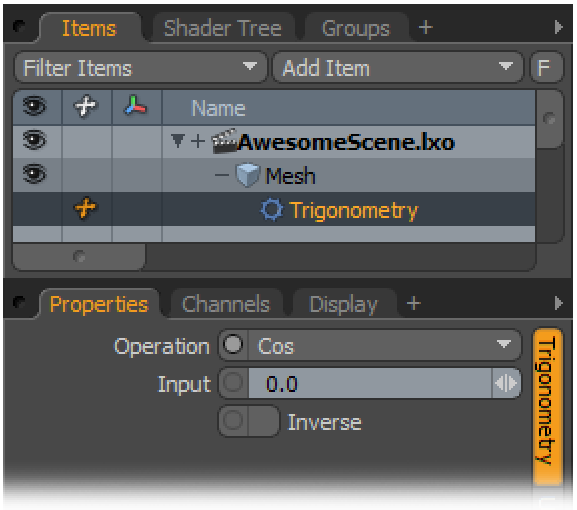
Math

Modo has additional math operators available for calculating other values. As with the other math modifiers, you can link channels to the **Input** channel. Modo feeds the resulting values through the **Output** channel.

- **Operations**: You can select from these additional math operations:
- **Absolute**- Calculates the absolute value for a number regardless of its sign (for example, 3=3 and -3=3).
- **Ceiling**- Calculates a maximum integer (whole number) value.
- **Constant**- Generates common, constant numerical values such as Pi, Eulers number, and the Golden Ratio.

- **Exp**- Calculates the exponent of the input value based on the indicated **Exponent** value (which can also be used as an input).
- **Floor**- Calculates a minimum integer (whole number) value.
- **Log**- Calculates the natural logarithmic value.
- **Log10**- Calculates the common logarithmic value.
- **Power**- Calculates the exponential value for a number based on the input **Power**.
- **Square Root**- Calculates the square root of the input value.

Trigonometry


Items
Shader Tree
Groups
+
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Trigonometry
Properties
Channels
Display
+
Trigonometry

Operation Cos

Input 0.0

○ Inverse

You can apply trigonometric operations to a channel. Trigonometric operations relate the angles of a triangle to the lengths of the sides of the triangle.

Operation:

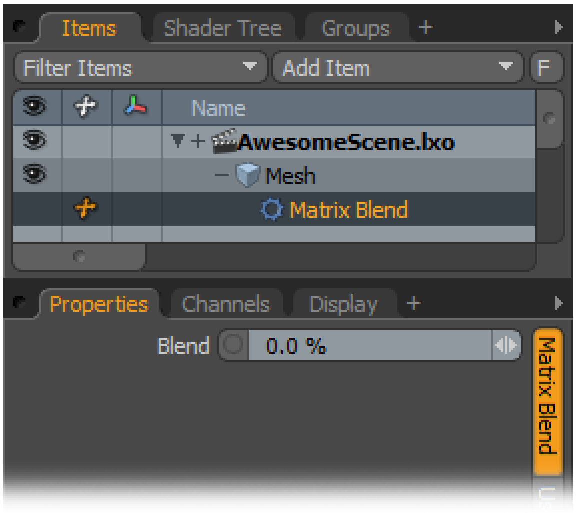
- **Cosine** - Indicates the ratio of the adjacent side to the hypotenuse of a right-angled triangle.
- **Sine** - Indicates the ratio of the length of the side opposite the given angle to the length of the hypotenuse of a right-angled triangle.
- **Tangent** - Indicates the ratio of the side opposite the angle to the adjacent side of a right-angled triangle.

Input: This is the input value for the numeric calculation. Typically, you link another channel to provide this value.

Inverse: If you select this, Modo inverts the output value.

Matrix Modifiers

Matrix Blend


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Matrix Blend
Properties
Channels
Display
Blend
0.0 %
Matrix Blend

You can use this Channel Modifier to blend between a pair of Rotation or Translation input matrices.

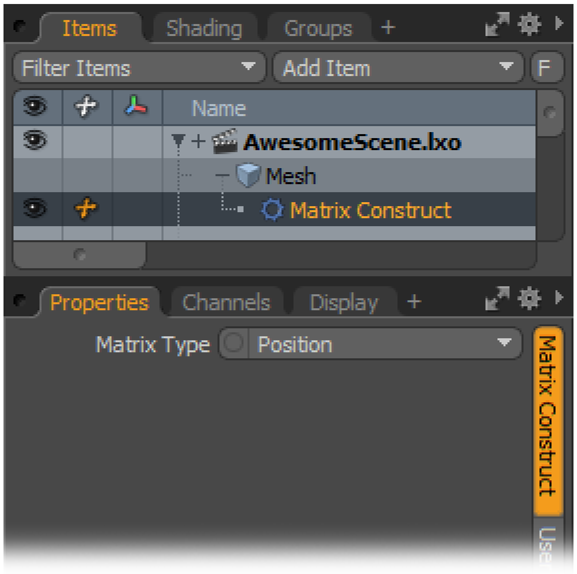
Matrix A/Matrix B	Specify the two matrices to be blended into these channels. These can be World or Local transform matrices from 3D items or the matrix output from other Channel Modifier items. For the best results, link channels with matrices of the same type.
Blend	Modo uses this blend amount to apply to the input matrices. With a value of **0%**, Modo outputs the Matrix A matrix unaltered; with **100%** Modo outputs the Matrix B matrix unaltered. For values between 0% and 100%, Modo interpolates the matrices between the two inputs. For Rotation matrices, Modo uses a quaternion interpolation, and for Translation matrices, Modo uses a linear interpolation.
Matrix Output	Modo passes the result of the interpolation to this channel. You can link this to a compatible matrix channel on a 3D item or another Channel Modifier.

Matrix Compose

The Matrix Compose Channel Modifier multiplies two or more input matrices together and passes the resulting transform matrix.

Matrix Inputs	This channel accepts multiple input links from the matrix channels of 3D items or other Channel Modifier outputs. Modo multiples the matrices together in the order that you used when you linked them this channel.
Matrix Output	Modo outputs the composed matrix to this channel. You can link this into a compatible matrix channel on a 3D item or another Channel Modifier's input.

Matrix Construct

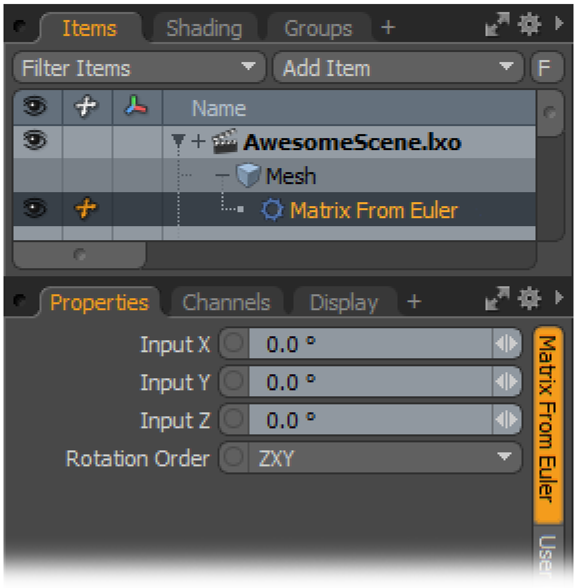

Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Matrix Construct
Properties
Channels
Display
Matrix Construct
Use

Matrix Type: Position

The Matrix Construct modifier creates a 4x4 Transformation matrix from a vector.

Input X/Input Y/Input Z	Describes either the position or the scale of the resulting matrix.
Matrix Type	Specifies whether the output is a Position matrix or a Scale matrix.
Output	Modo passes the composed matrix to this channel.

Matrix From Euler

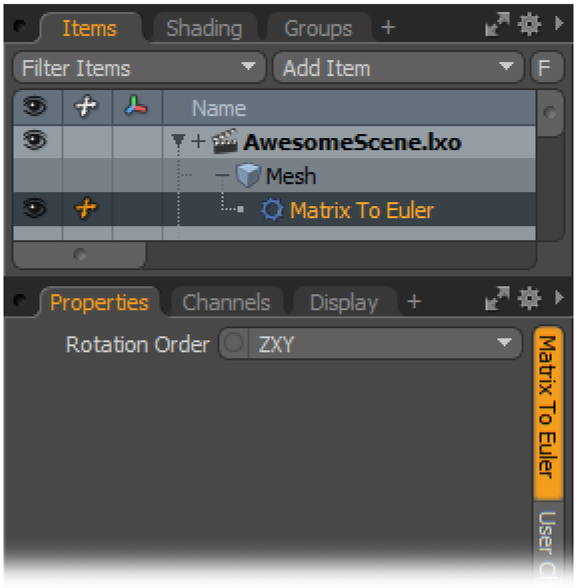

Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Matrix From Euler
Properties
Channels
Display
Matrix From Euler
User

Input X 0.0 °
Input Y 0.0 °
Input Z 0.0 °
Rotation Order ZXY

The **Matrix From Euler** modifier extracts a vector that represents rotation angles in the X, Y and Z axes from a 4x4 Rotation matrix.

Input X/Input Y/Input Z	Describes rotation angles around the X, Y, and Z axes.
Rotation Order	Describes the rotation order of the three axes. With the default **ZXY** order, Modo calculates the final matrix by creating a matrix to represent the Z axis, then creating a matrix to represent the X axis, and finally creating a matrix to represent the Y axis. Modo then multiples the three matrices together to construct the final matrix.
Output	Outputs the resulting rotation matrix.

Matrix To Euler


Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Matrix To Euler
Properties
Channels
Display
Matrix To Euler
User Ch

Rotation Order ZXY

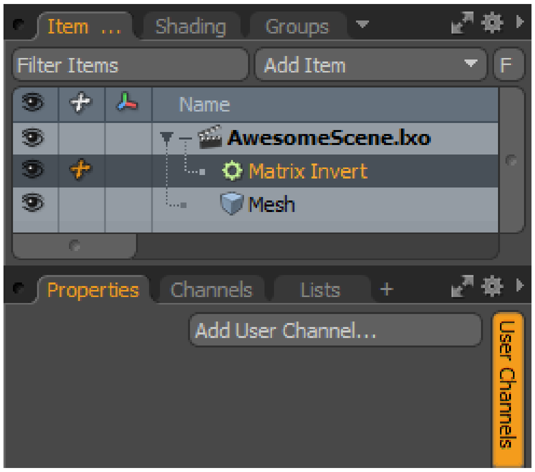
The **Matrix To Euler** modifier extracts a vector that represents rotation angles in the X, Y, and Z axes from a 4x4 Rotation matrix.

Input X/Input Y/Input Z	Indicates the input rotation matrix from which to extract rotation angles.
Rotation Order	Describes the rotation order of the three axes. With the default **ZXY** order, Modo calculates the final matrix by creating a matrix to represent the Z axis, then creating a matrix to represent the X axis, and finally creating a matrix to represent the Y axis. Modo then uses the three matrices to construct the final matrix.
Output X/Output Y/Output Z	Describes rotation angles around the X, Y, and Z axes.

Matrix Invert

The **Matrix Invert** modifier inverts the transforms stored in the input matrix. For example, if the input matrix stores rotations of 90, 45, and 180 degrees, the output matrix stores the opposite: -90, -45, and -180 degrees. It also outputs the inverse of position and scale, or a combination of all three.

This modifier can be extremely useful, as often it's desirable to perform complex 3D math operations at the center of the world. Multiplying a transform matrix by it's inverse transform matrix, results in an identity matrix. That is to say, a matrix that stores no position, rotation, and 100% scale. This allows you to easily transform an item back to its center position, perform a maths operation easily, and then transform it back by multiplying by the original transform.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Matrix Invert
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

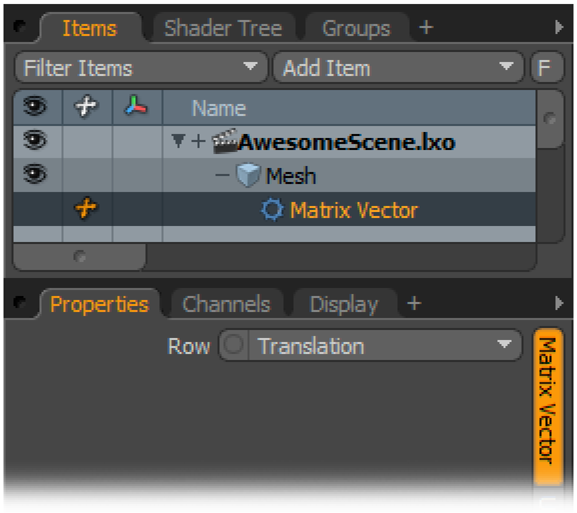
Input	The matrix to invert.
Output	The resulting inverse matrix.

Matrix Transpose

The Matrix Transpose modifier inverts a 4x4 matrix across the main diagonal. Modo inverts any rotation stored in the matrix; therefore, for example, a rotation of 90 degrees in X becomes a rotation of -90 degrees.

Input	Indicates the matrix with the values to transpose.
Output	Indicates where Modo should pass the matrix with the transposed values.

Matrix Vector


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Matrix Vector
Properties
Channels
Display
Row
Translation
Matrix Vector

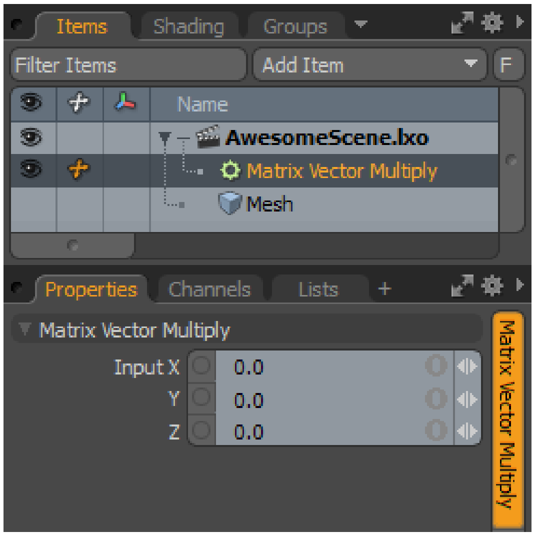
You use this Channel Modifier to extract a normalized direction vector from a Rotation matrix or the translation row from a Translation matrix. Direction vectors in a Rotation matrix are the axes formed after the individual X, Y, and Z rotations have been performed. For example, given the World Rotation matrix of a 3D item, the Z Direction vector points in the direction the item is facing, and the Y axis points towards the top of the item.

Matrix Input	Links the matrix channel from which the vector is to be extracted into this channel. This can be a matrix channel from a 3D item or the output matrix from another Channel Modifier.
Row	This specifies whether to extract the X, Y, Z direction vector from a Rotation matrix or the translation from a Translation matrix.
Output X/Output Y/Output Z	Passes the vector extracted from the matrix to these three channels.

Matrix Vector Multiply

The **Matrix Vector Multiply** modifier multiplies a transform matrix by a vector, and outputs the resulting vector. This essentially applies any transforms that are stored in the matrix to the vector. For example, if the vector stored a value of [1.0, 0.0, 0.0] and the matrix stored a position of 2m in X, then the resulting vector would be [3.0, 0.0, 0.0].

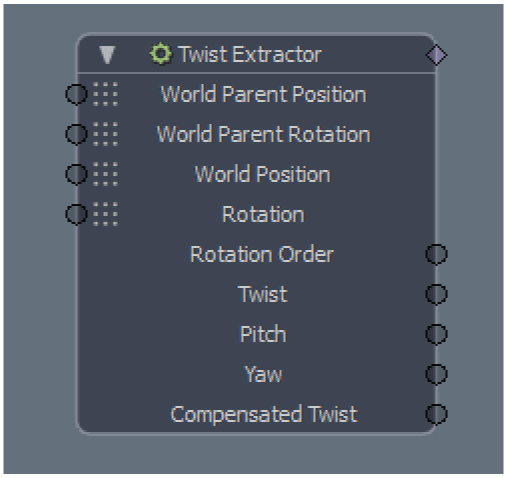
This modifier is really useful when used with rotation matrices. For example, if you have a ray that is fired from a point in space, by multiplying the ray vector by the rotation matrix, you can easily change the direction of the ray, rotating it by whatever value is stored in the matrix.


Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Matrix Vector Multiply
Mesh
Properties
Channels
Lists
+
Matrix Vector Multiply
Input X
Y
Z
0.0
0.0
0.0
Matrix Vector Multiply

Matrix	The matrix to multiply the vector by.
Input	The vector to multiply by the matrix.
Output	The resulting vector.

Twist Extractor

This channel modifier allows you to extract twist (roll), yaw, and pitch from a local matrix. The twist output can then be used to smooth deformations, for example, on the lower arm as the wrist is rotated or the lower leg as the foot is rotated.



To add a Twist Extractor node, in the Schematic, click **Add...** then under **Channel Modifiers** > **Matrix**, click **Twist Extractor**.

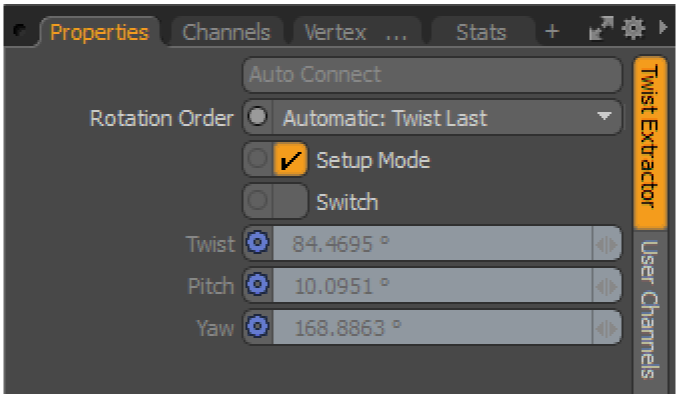
The node compares the twist of a joint to the Setup state, or another joint in the hierarchy.

Note: For more information on Setup mode, see Binding and Setup Mode.

Twist Extractor Channels on the Node

World Parent Position	The world position of the selected joint's parent joint.
World Parent Rotation	The world rotation of the selected joint's parent joint.
World Position	The selected joint's world position.
Rotation	The selected joint's rotation.
Rotation Order	The order of rotation for the matrix extraction.
Twist	The rotation of the selected joint around the longitudinal axis. Also called roll.
Pitch	The rotation of the selected joint around the transverse axis.
Yaw	The rotation of the selected joint around the vertical axis.
Compensated Twist	When a joint is rotated, the path it takes influences the orientation at the destination. For instance, the orientation differs when an arm is held forward compared to when it's held up. With a rotation order of Twist, Yaw, Pitch, Modo computes a compensated twist value to ensure that the orientation is the same at the destination.

Twist Extractor Properties



Rotation Order: Automatic: Twist Last

☑ Setup Mode

☐ Switch

Twist: 84.4695 °

Pitch: 10.0951 °

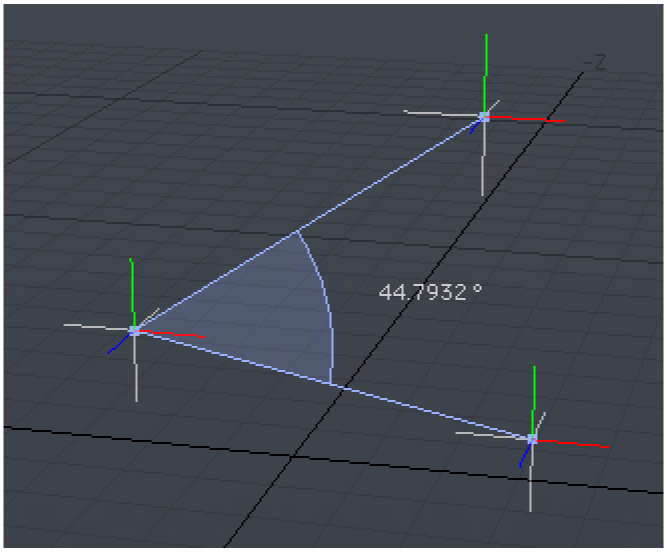
Yaw: 168.8863 °

Auto Connect	When **Setup Mode** is disabled, you can use this to connect selected locators to the Twist Extractor node. With a locator and Twist Extractor selected, click this button to automatically set up the connections between the locator and the Twist Extractor node. **Note:** When **Setup Mode** is enabled, **Auto Connect** is not available.
Rotation Order	The order the rotations are extracted from the matrix. In addition to a specific order of axes, you have the following options: • **Automatic: Twist First** - The default option. It extracts rotation starting with the twist axis and the others follow in direct rotation order. Z is usually the joint rotation axis, so the order is ZXY. • **Automatic: Twist Last** - It uses the twist axis as the last one and the others follow in direct rotation order. Usually, Z is the joint rotation axis, so the order is XYZ.
Setup Mode	When enabled (the default state), the Twist Extractor extracts the twist data from the difference between the current rotation compared to the Setup one. When disabled, the you can use Twist Extractor to get the twist values from a joint compared to its parent or another locator in the hierarchy.
Switch	The Twist Extractor calculates the twist in two different ways. In cases where the default calculation doesn't produce the expected result, enable **Switch** to calculate the twist using another method.

Twist	The rotation of the selected joint around the longitudinal axis. Also called roll.
Pitch	The rotation of the selected joint around the transverse axis.
Yaw	The rotation of the selected joint around the vertical axis.

Measurement Modifiers

Measure Angle



You can use this type of Channel Modifier to measure the angle formed between any three 3D items in the scene.

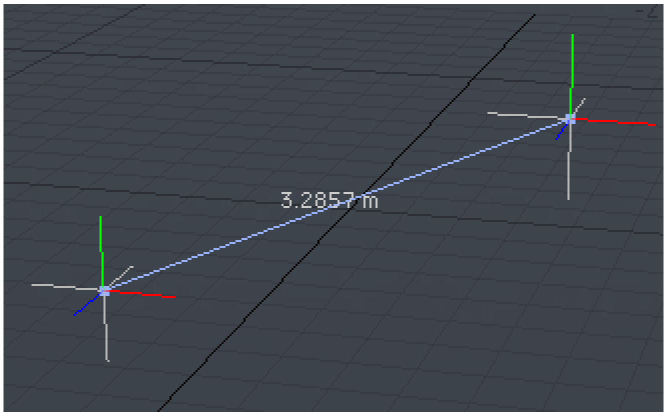
Position From: Use to link the **World Position** transform channel of a 3D item or the **Position** transform output of another Channel Modifier item into this channel.

Position To A: Use to link the **World Position** transform channel of a second 3D item or the **Position** transform output of another Channel Modifier item into this channel.

Position To B: Use to link the **World Position** transform channel of a third 3D item or the **Position** transform output of another Channel Modifier item into this channel.

Angle Output: Modo passes the floating-point value of the angle between the three items to this channel.

Measure Distance



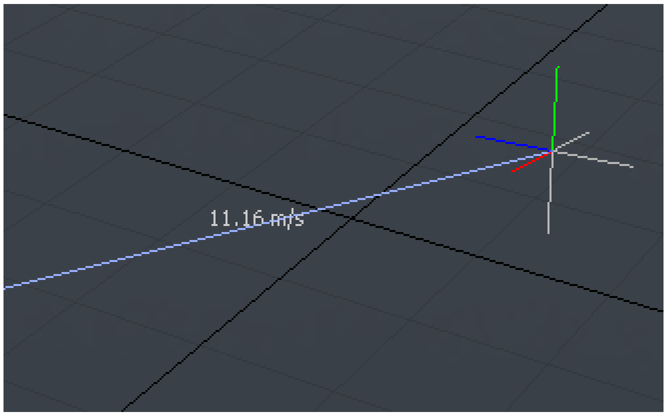
You can use the **Measure Distance** Channel Modifier to measure the distance between any two items in the scene.

Position From: Use to link the **World Position** transform channel of a 3D item or the **Position** transform output of another Channel Modifier item into this channel.

Position To: Use to link the **World Position** transform channel of a second 3D item or the **Position** transform output of another Channel Modifier item into this channel.

> **Note:** If you leave either of these input channels unlinked Modo outputs the distance from the World origin.

Measure Speed



The **Measure Speed** Channel Modifier takes a transformation matrix (such as a **World Position**) and passes a single value, which represents the overall speed of the input item measured in meters per second. Modo draws a blue line in the viewport with the numerical readout to provide a visual representation of the speed.

World Position: Use to link the **World Position** matrix of the moving item into this channel.

Speed: Shows the speed per frame of the input item in meters per second.

Measure Velocity

The **Measure Velocity** Channel Modifier takes a transformation matrix (such as a **World Position**) and passes a vector that represents the velocity of the input item. Each output of the vector represents the speed of the item for an axis and is given in meters per second. Modo draws a blue line in the viewport to provide a visual representation of the velocity.

World Position: Use to link the **World Position** matrix of the moving item into the **World Position** input of the Measure modifier.

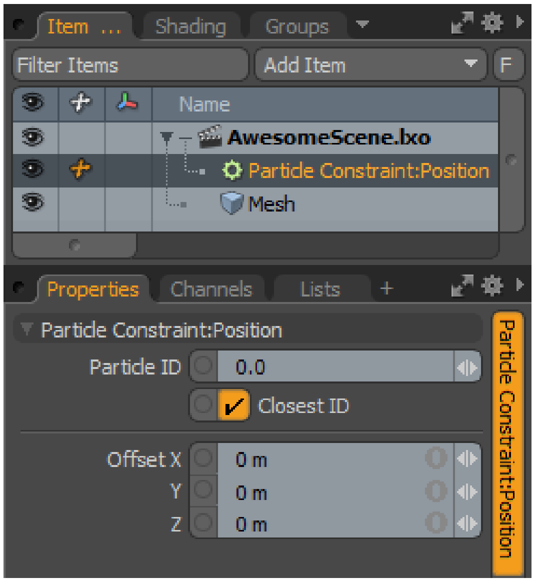
Velocity X/Velocity Y/Velocity Z: Modo calculates the velocity per frame in meters per second from the input item and passes a vector for each axis.

Particle Modifiers

Particle Constraint:Position

The **Particle Constraint:Position** modifier outputs a position matrix for a particle with a specified index. The modifier can either be created manually or by using commands in the **Modifier** menu, which creates the item and connects the required graph links.

As the particle ID is defined as a floating point number between 0.0 and 1.0, the modifier allows an item to be constrained to the closest ID. Essentially, this means that if the ID value is 0.0, and there is no particle with an ID of 0.0, but there is a particle with an ID of 0.1, then the output matrix matches the transform of the particle with the ID of 0.1. This is optional, and can be disabled to match identical IDs only.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Particle Constraint:Position
Mesh
Properties
Channels
Lists
+
Particle Constraint:Position
Particle ID
0.0
☑ Closest ID
Offset X
0 m
Y
0 m
Z
0 m
Particle Constraint:Position

Particle ID: The particle ID to constrain to.

Closest ID: When enabled, the closest ID is treated as a matching ID. When disabled, the ID must match.

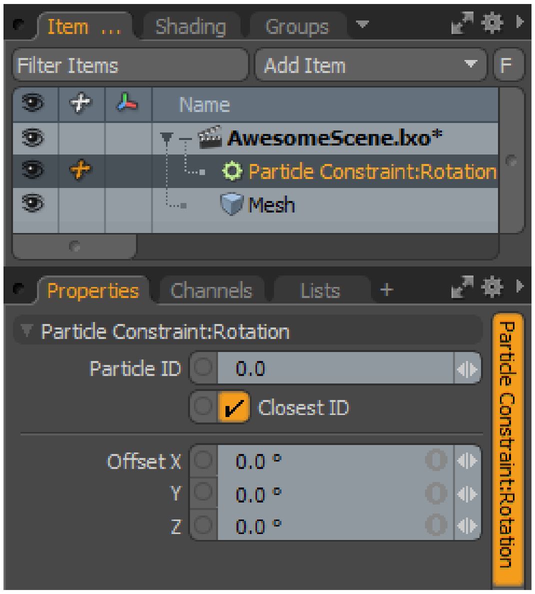
Offset: A position offset to apply to the constrained position.

Output: Outputs the position matrix, matching the particle transform.

Particle Constraint:Rotation

The **Particle Constraint:Rotation** modifier outputs a rotation matrix for a particle with a specified index. The modifier can either be created manually, or by using commands in the Modifier menu which creates the item and connects the required graph links.

As the particle ID is defined as a floating point number between 0.0 and 1.0, the modifier allows an item to be constrained to the closest ID. Essentially, this means that if the ID value is 0.0, and there is no particle with an ID of 0.0, but there is a particle with an ID of 0.1, then the output matrix matches the transform of the particle with the ID of 0.1. This is optional, and can be disabled to match identical IDs only.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo*
Particle Constraint:Rotation
Mesh
Properties
Channels
Lists
+
Particle Constraint:Rotation
Particle ID
0.0
Closest ID
Offset X
Y
Z
0.0 °
0.0 °
0.0 °
Particle Constraint:Rotation

Particle ID: The particle ID to constrain to.

Closest ID: When enabled, the closest ID is treated as a matching ID. When disabled, the ID must match.

Offset: A position offset to apply to the constrained position.

Output: Outputs the position matrix, matching the particle transform.

Probe Modifiers

Channel Probe

The **Channel Probe** is primarily used to simplify assemblies. For example, suppose you have an assembly that requires multiple channels from a camera in order to work. In order to use the assembly, you would normally have to link up each required channel from the camera you want to use. Using a **Channel Probe** in the assembly saves time, requiring just one item link for the camera to allow the assembly to read as many channels as it needs.

The **Channel Probe** works with the internal name of custom channels, not the user name. If the probe has a custom channel with the same name as a channel on the source item, the value of the custom channel is driven by the channel value of the source. The custom channel's user name is set to match the user name of the source channel, so to see the internal name you may need to check command history.

To add a **Channel Probe**, do the following:

1. Navigate to **Animate** > **Add Channel Modifier** > **Probe** and select **Channel Probe**.

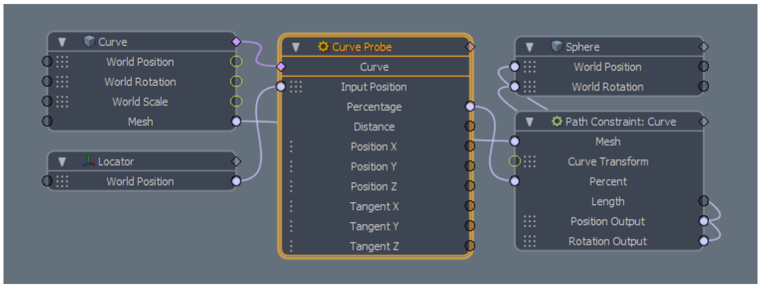
2. Select channels on the source item, either from the **Properties** or the **Channels** list.
3. Add the selected channels to the probe using **Add Selected Channels** button in the probe **Properties**.

   The selected channels are automatically created on the probe, and the probe is linked to the source item.
4. Once the probe and source item are linked, you can:
   - **Add channels** - you can add more channels to an existing probe by selecting the channels on the source item and using the **Add Selected Channels** button in the **Properties**.
   - **Remove channels** - you can remove channels by deleting the custom channels from the channel probe.
   - **Change the source item** - the channel probe item has a schematic connection point for a source item. This is automatically set when channels are added, but can be altered to point to a different source item.

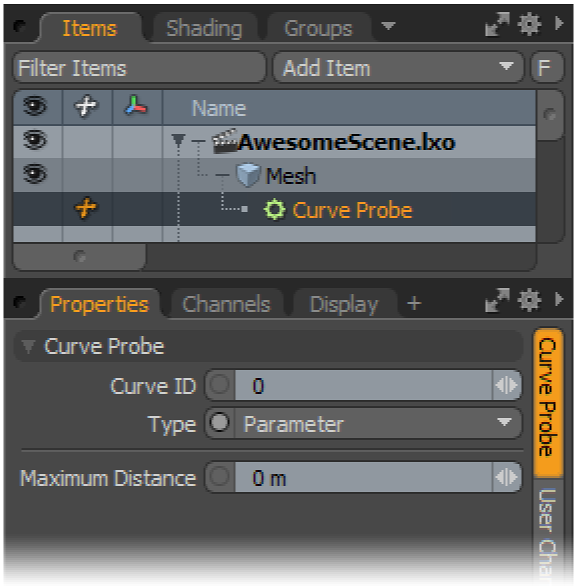
Curve Probe

With the **Curve Probe** Channel Modifier you can sample various attributes from a target curve and apply the sampled value to another channel's input. This could be useful, for instance, to constrain an object along a path without actually using a Constraint modifier or, for Shading, you could use this to create adjustable seams when used with displacements.

When used in a rigging/animation context, you use a Locator item as the **Input Position** (sampling) and connect its **World Position** matrix to the probe's **Input Position** matrix. When used this way, the value generated by the probe represents the closest position along the nearest curve to the Locator. Modo constantly updates the output value as the Locator item moves. When used in a shading context, use a Matrix Construct modifier to convert the **Texture Coordinates** channel from a Texture Locator into the Probe's **Input Position** for it to be read on a per-sample basis.


Curve
World Position
World Rotation
World Scale
Mesh
Locator
World Position
Curve Probe
Curve
Input Position
Percentage
Distance
Position X
Position Y
Position Z
Tangent X
Tangent Y
Tangent Z
Sphere
World Position
World Rotation
Path Constraint: Curve
Mesh
Curve Transform
Percent
Length
Position Output
Rotation Output

Curve Probe



Curve ID: 0
Type: Parameter
Maximum Distance: 0 m

Curve ID: Modo uses this value to designate a specific curve, based on its ID, within an item layer that has multiple curves. When you set the value to -1, Modo probes all curves. Modo tags curves generally in the order they are created, starting with the first curve at id **0**.

Type: This controls how to evaluate the curve. If you select **Fraction**, Modo evaluates the curved based on its overall length to produce a value as a fractional amount of that total. By selecting **Parameter**, Modo evaluates the curve based on the locations of the vertices that define it. For example, with a curve of three points, the 50% position would be the middle vertex — even if it was near the first vertex.

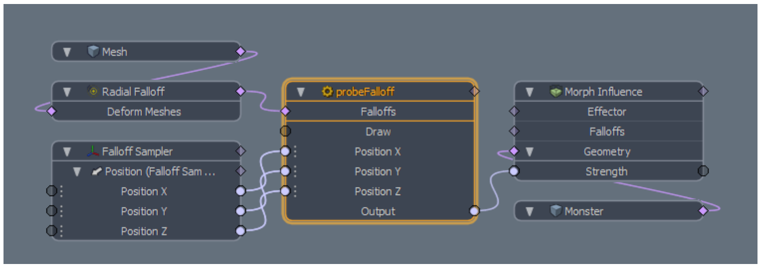
Maximum Distance: This defines an area around the curve for evaluation. If the distance from the point being evaluated is greater than the distance to the bounding box containing the curve plus the maximum distance, then Modo skips the evaluation for that curve.

Falloff Probe

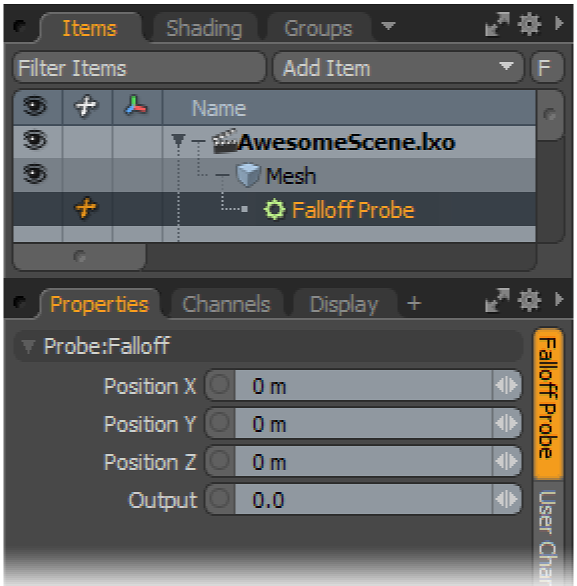
With the **Falloff Probe** Channel Modifier, Modo samples the value returned from a falloff based on the sample position. When used with a falloff, connect the probe to the falloff's graph connection on the node and the XYZ position to the appropriate Locator item channels. The **Output** channel returns a 0 to 1 value that is the falloff multiplied at that location.

Position X/Y/Z: Links the **World Position** transform channel of a 3D item or the **Position** transform output of another Channel Modifier item into this channel.

Output: This represents the strength value of the falloff at the sampled location and can be connected to any single numeric value input channel.


Mesh
Radial Falloff
Deform Meshes
Falloff Sampler
Position (Falloff Sam ...
Position X
Position Y
Position Z
probeFalloff
Falloffs
Draw
Position X
Position Y
Position Z
Output
Morph Influence
Effector
Falloffs
Geometry
Strength
Monster

Falloff Probe


Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Falloff Probe
Properties
Channels
Display
+
Falloff Probe
User Chan

Probe:Falloff

Position X 0 m

Position Y 0 m

Position Z 0 m

Output 0.0

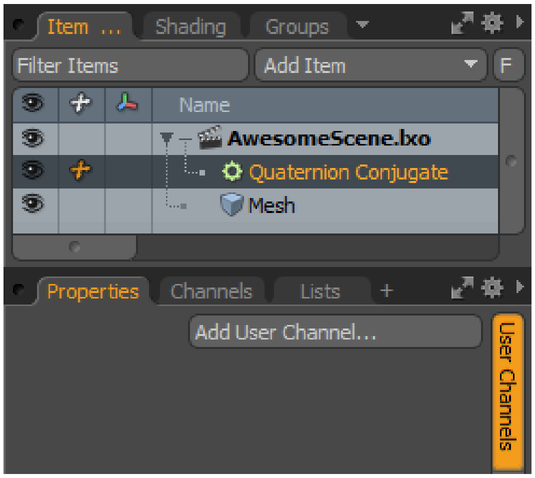
Position X/Position Y/Position Z: These values define the sampling point of the falloff. You can manually enter values to define the location in 3D space or connect the Position inputs of the node to the Position outputs of a Locator.

Output: This value represents the sampled value from the falloff. It is usually connected to the input of another node as a driver of the channel.

Quaternion Modifiers

Quaternion Conjugate

The **Quaternion Conjugate** modifier calculates the inverse of a quaternion. For example, if the input quaternion contains a rotation of 45 degrees in Y, the output quaternion contains a rotation of -45 degrees in Y.

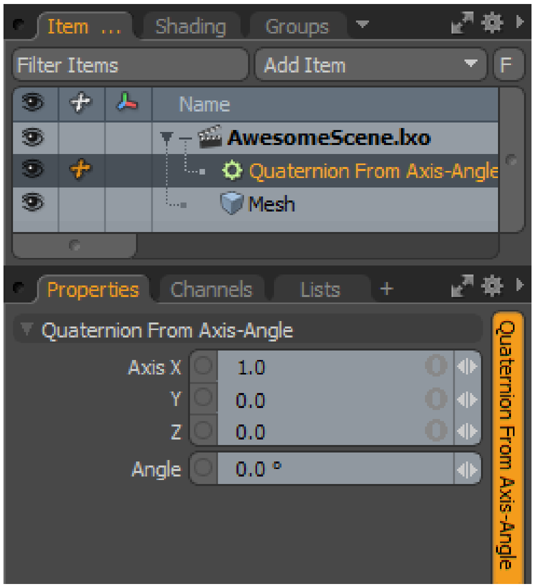

Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion Conjugate
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

Input: The quaternion to conjugate.

Output: Outputs the conjugated quaternion.

Quaternion From Axis-Angle

The **Quaternion From Axis-Angle** modifier converts a rotation around an arbitrary axis into a quaternion.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion From Axis-Angle
Mesh
Properties
Channels
Lists
+
Quaternion From Axis-Angle
Quaternion From Axis-Angle

Axis X 1.0
Y 0.0
Z 0.0
Angle 0.0 °

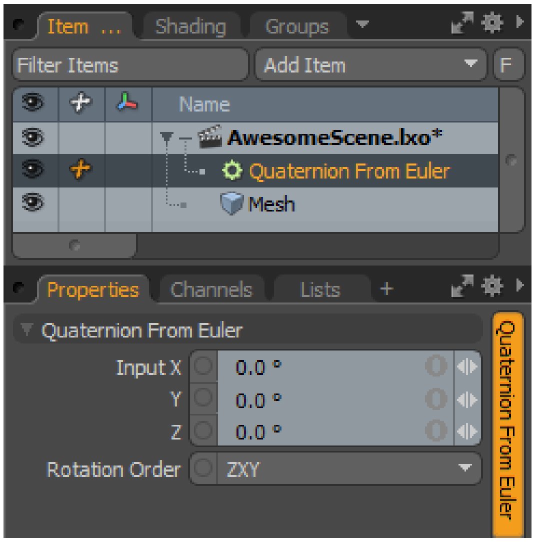
Axis: The axis to perform the rotation around.

Angle: The rotation angle to rotate around the axis.

Quaternion: Outputs the result.

Quaternion From Euler

The **Quaternion From Euler** modifier converts XYZ rotation angles into a quaternion.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo*
Quaternion From Euler
Mesh
Properties
Channels
Lists
+
Quaternion From Euler

Input X 0.0 °
Y 0.0 °
Z 0.0 °
Rotation Order ZXY

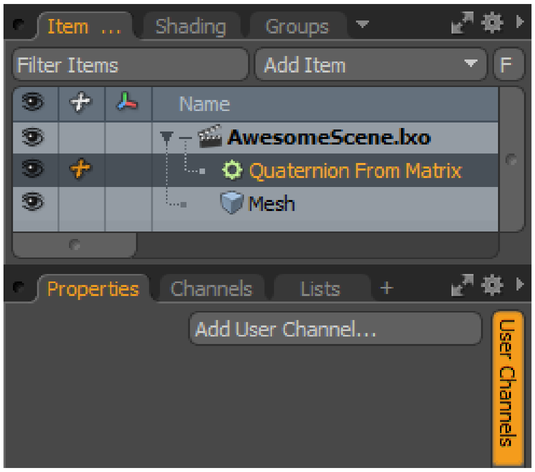
Input: The XYZ rotation angles to convert.

Rotation Order: The order used for combining the three axes. As euler rotations use three gimbals, the order of the rotation matters. For example, if the **Rotation Order** channel is set to **ZXY** (default), the **Z** rotation is performed first, without affecting the other axes. Next, the **X** axis rotates, changing the plane used to define the **Z** axis. Finally, the **Y** axis is calculated, which changes the plane used to define both the **X** and **Z**. In the majority of cases, this value should match the value in the Modo transform properties.

Quaternion: Outputs the result.

Quaternion From Matrix

The **Quaternion From Matrix** modifier converts a rotation matrix into a quaternion.

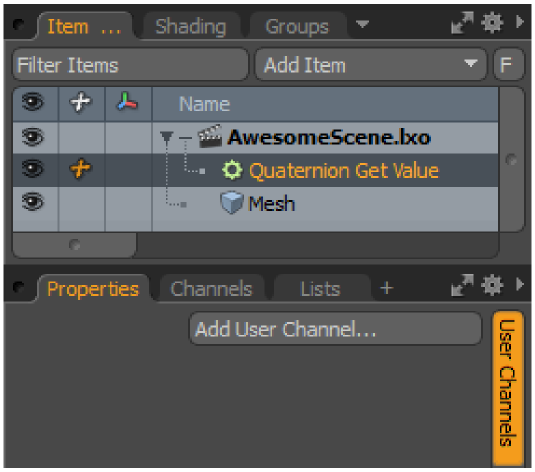

Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion From Matrix
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

Matrix: The rotation Matrix to convert.

Quaternion: Outputs the result.

Quaternion Get Value

The **Quaternion Get Value** modifier returns the four components representing the value of the quaternion.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion Get Value
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

Input: The input quaternion to read for its values.

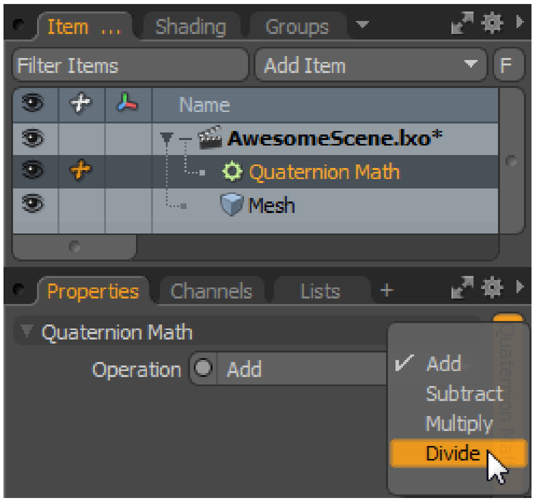
Value: The first three components stored in the quaternion.

Value W: The fourth component stored in the quaternion.

Quaternion Math

The **Quaternion Math** modifier performs simply math operations on two quaternions: **Add, Subtract, Multiply, or Divide**. In some operations, the quaternion inputs are non-commutative, so swapping

Quaternion A and **Quaternion B** around, changes the result. These modifiers show up as four separate modifiers in the interface, but they are in fact a single modifier that has multiple math operations.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo*
Quaternion Math
Mesh
Properties
Channels
Lists
+
Quaternion Math
Operation
Add
Add
Subtract
Multiply
Divide

Quaternion A: The first quaternion used for the math operation.

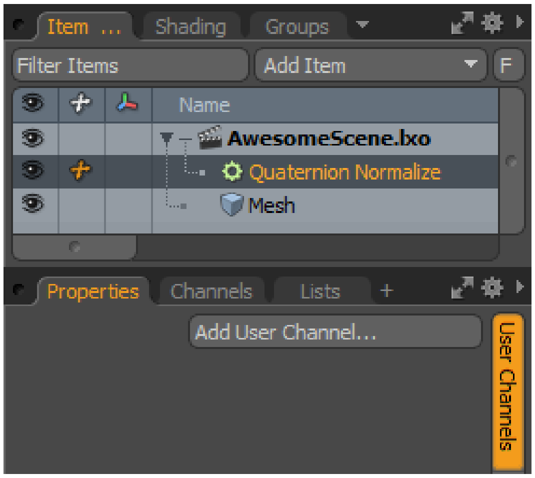
Quaternion B: The second quaternion used for the math operation.

Operation: The math operation to perform: **Add**, **Subtract**, **Multiply**, or **Divide**.

Output: The resulting quaternion, after the math operation has been performed.

Quaternion Normalize

The **Quaternion Normalize** modifier normalizes a quaternion, so that it has a magnitude of 1.0.

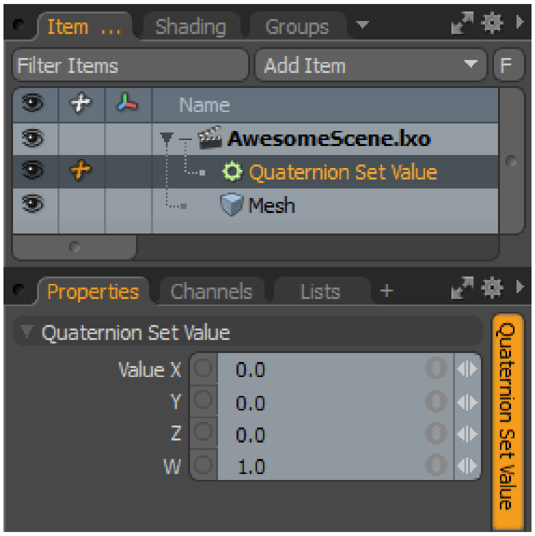

Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion Normalize
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

Input: The quaternion to normalize.

Output: The normalized quaternion.

Quaternion Set Value

The **Quaternion Set Value** modifier sets the four components representing the value of the quaternion.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion Set Value
Mesh
Properties
Channels
Lists
+
Quaternion Set Value
Quaternion Set Value

Value X 0.0
Y 0.0
Z 0.0
W 1.0

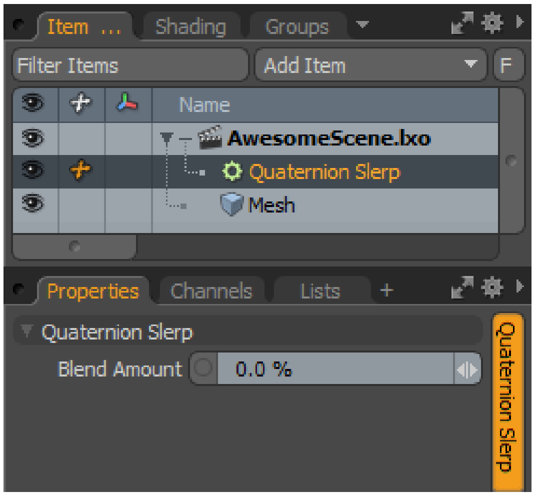
Value: The values used to set the first three components stored in the quaternion.

Value W: The value used to set the fourth component stored in the quaternion.

Output: The resulting quaternion.

Quaternion Slerp

The **Quaternion Slerp** modifier performs a Spherical Linear Interpolation between two quaternions, blending the rotation stored in one quaternion to the rotation stored in another quaternion.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion Slerp
Mesh
Properties
Channels
Lists
+
Quaternion Slerp
Quaternion Slerp

Blend Amount 0.0 %

Quaternion A: The first quaternion value to blend from.

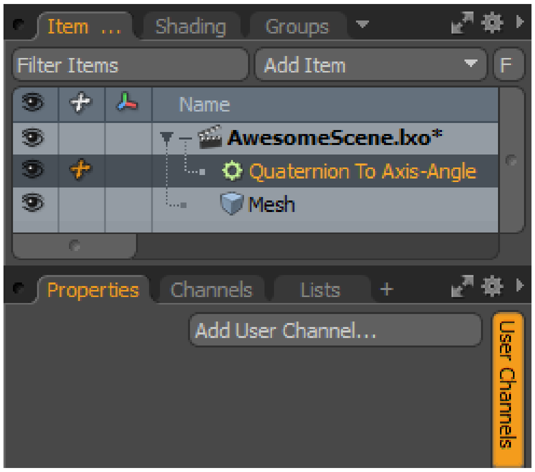
Quaternion B: The second quaternion value to blend to.

Blend Amount: The amount of blend between the two quaternions. A value of 0 returns Quaternion A and a value of 100 returns Quaternion B. A value in between these two values returns an interpolation between the two quaternions.

Output: The resulting quaternion.

Quaternion To Axis-Angle

The **Quaternion To Axis-Angle** modifier converts a quaternion into a rotation around an arbitrary axis.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo*
Quaternion To Axis-Angle
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

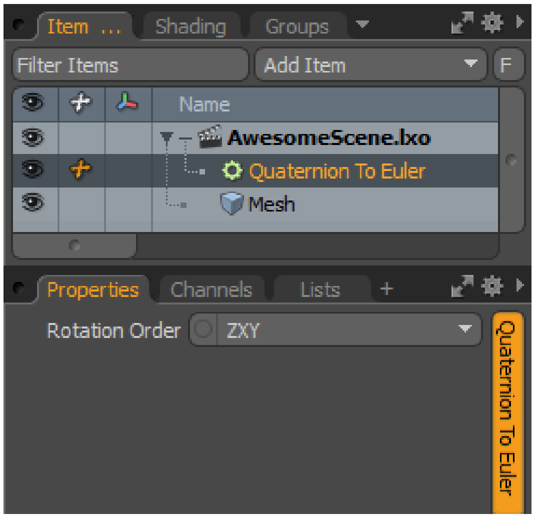
Quaternion: The quaternion to convert to an axis-angle.

Axis: The axis the rotation is performed around.

Angle: The rotation angle performed around the axis.

Quaternion To Euler

The **Quaternion To Euler** modifier converts a quaternion into XYZ rotation angles.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion To Euler
Mesh
Properties
Channels
Lists
+
Quaternion To Euler

Rotation Order ZXY

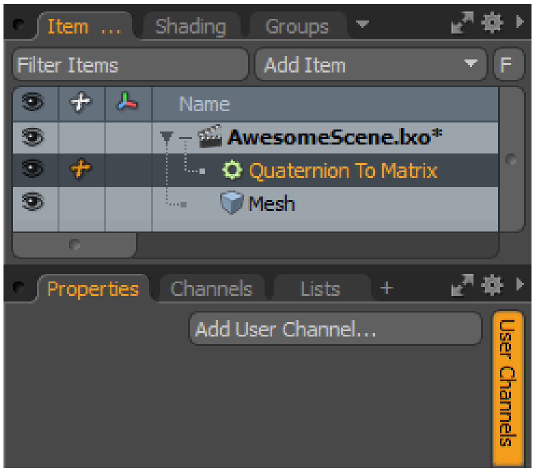
Quaternion: The quaternion to convert into XYZ rotation angles.

Rotation Order: The order used for combining the three axes. As euler rotations use three gimbals, the order of the rotation matters. For example, if the **Rotation Order** channel is set to **ZXY** (default), the **Z** rotation is performed first, without affecting the other axes. Next, the **X** axis rotates, changing the plane used to define the **Z** axis. Finally, the **Y** axis is calculated, which changes the plane used to define both the **X** and **Z**. In the majority of cases, this value should match the value in the Modo transform properties.

Output: The resulting XYZ rotation angles, converted from the quaternion.

Quaternion To Matrix

The **Quaternion To Matrix** modifier a quaternion into a rotation matrix.

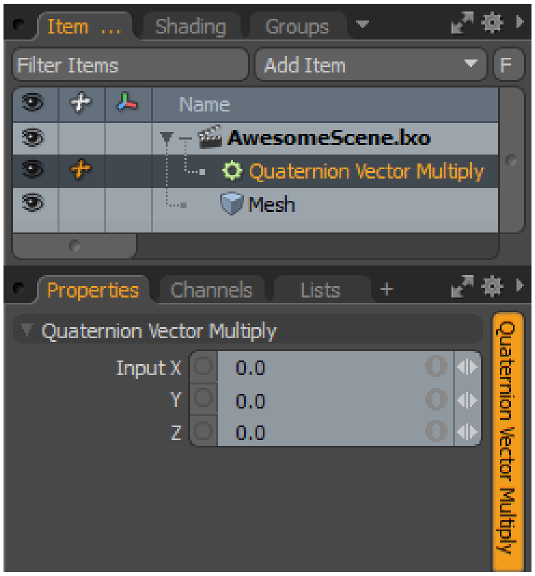

Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo*
Quaternion To Matrix
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

Quaternion: The quaternion to convert into a matrix.

Matrix: Outputs the resulting rotation matrix.

Quaternion Vector Multiply

The **Quaternion Vector Multiply** modifier, multiplies a vector by a quaternion. This essentially rotates the vector around the axis stored in the quaternion, by the angle stored in the quaternion.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Quaternion Vector Multiply
Mesh
Properties
Channels
Lists
+
Quaternion Vector Multiply
Input X
Y
Z
0.0
0.0
0.0
Quaternion Vector Multiply

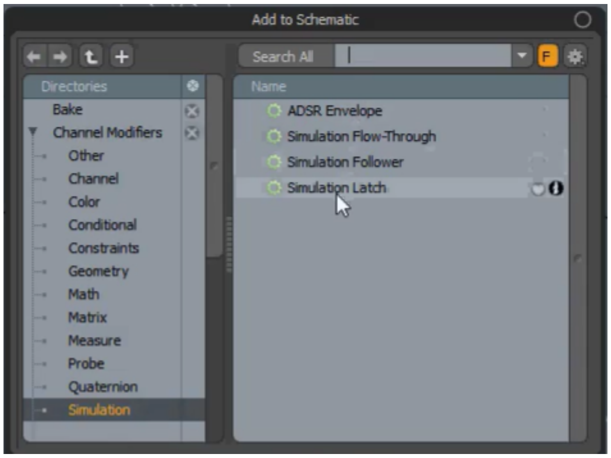
Quaternion: The quaternion to multiply the vector by.

Input: The input vector to multiply by the quaternion.

Output: The resulting vector.

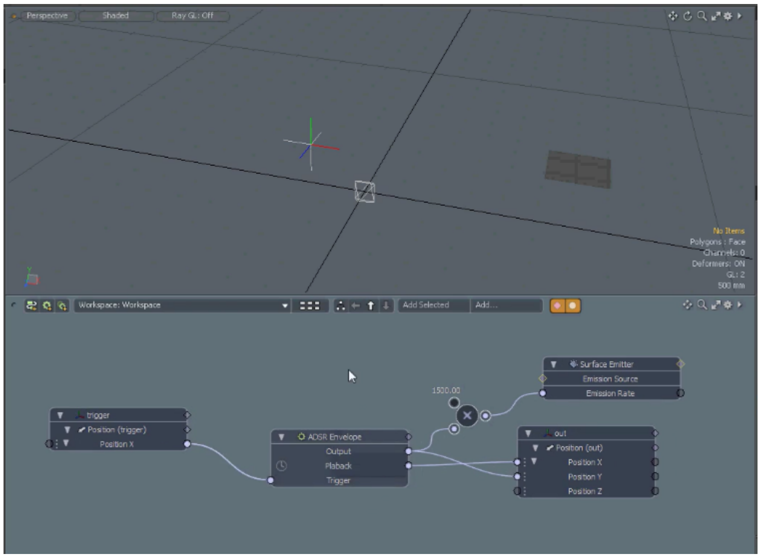
Simulation Modifiers

Simulation Modifiers provide ways to control the animated behaviors based on set triggers. Modo provides a number of simulation modifiers, which allow you to simulate animated workflows similar to those used by third party plug-ins to drive performance in Modo. The following simulation modifiers are available:

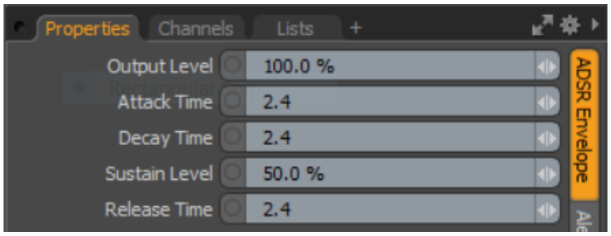

Add to Schematic
Search All
Directories
Bake
Channel Modifiers
Other
Channel
Color
Conditional
Constraints
Geometry
Math
Matrix
Measure
Probe
Quaternion
Simulation
Name
ADSR Envelope
Simulation Flow-Through
Simulation Follower
Simulation Latch

ADSR Envelope

The **ADSR Envelope** modifier allows you to define the following key profiles: **Attack** (to rise), **Decay** (to fall), **Sustain** (if key is held the key sustains its position), and **Release** (to idle). The following is an example displayed in the **Schematic** viewport.


Perspective
Shaded
Ray GL: Off
Workspace: Workspace
Add Selected
Add...
trigger
Position (trigger)
Position X
ADSR Envelope
Output
Playback
Trigger
1500.00
Surface Emitter
Emission Source
Emission Rate
out
Position (out)
Position X
Position Y
Position Z

The Locator is the trigger controlling the out mesh. When the Locator moves to a positive position on the X axis, the ADSR envelope is triggered.


Properties
Channels
Lists
ADSR Envelope

Output Level 100.0 %
Attack Time 2.4
Decay Time 2.4
Sustain Level 50.0 %
Release Time 2.4

ADSR Envelope	
Output Level	Set the percentage calculation level of the results.
Attack Time	Set the time to continue to increase output at the attack rate; if max is reached, the Attack state changes to the Decay state.
Decay Time	Set the time to continue reducing output at the decay rate; if sustain level is reached, the Decay state is changed to the Sustain state.
Sustain Level	Set the percentage level for the output to remain at the sustained level (where nothing happens). If the key is held, it remains at Sustain state.
Release Time	Set the time to continue reducing output at release rate. If zero is reached, the state is changed to idle.

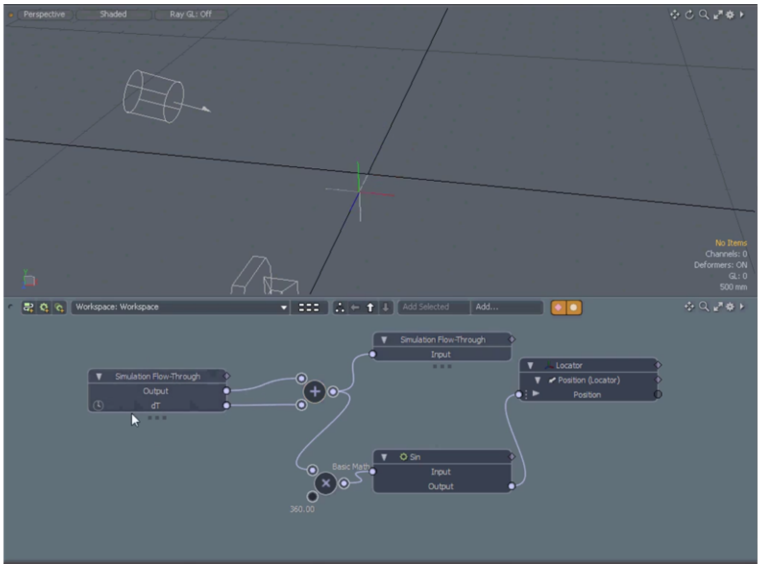
Simulation Flow-Through

During simulation, there is a short interval between two time steps, where the simulation advances and values from the previous time evolve to the next time. The **Simulation Flow-Through** modifier allows you take a value from the current time step and pass it to the next time step. The summation of **dT** is calculated over time.

In the example below, the output of the **Plus** node feeds into the **Input** of the Simulation Flow-Through node, which then becomes the **Output** of the next time step. **dT** is added at each time step, generating an ever-increasing value. This setup is attached to a rig that produces an oscillating motion of the Locator item.

The **Simulation Flow-Through** modifier sees the input to the simulation Flow-Through as a final result of the current stage of the simulation and it sees the output of the Flow-Through as an input to the next stage of the simulation. This is used for dynamic outputs to affect themselves and for things to react.

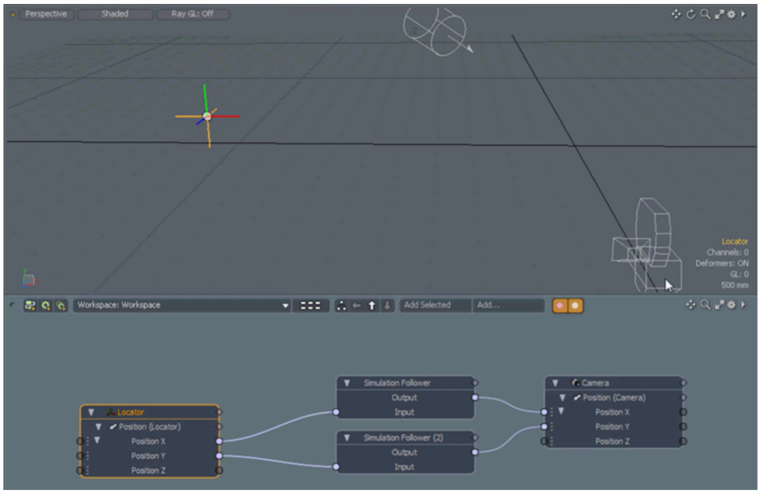
Note: In the example below, the Simulation Flow-Through node's **Input** channel is separated for better visibility.


Perspective
Shaded
Ray GL: Off
No Items
Channels: 0
Deformers: ON
GL: 0
500 mm
Workspace: Workspace
Add Selected
Add...
Simulation Flow-Through
Output
dT
Simulation Flow-Through
Input
Locator
Position (Locator)
Position
Sin
Input
Output
Basic Math
360.00

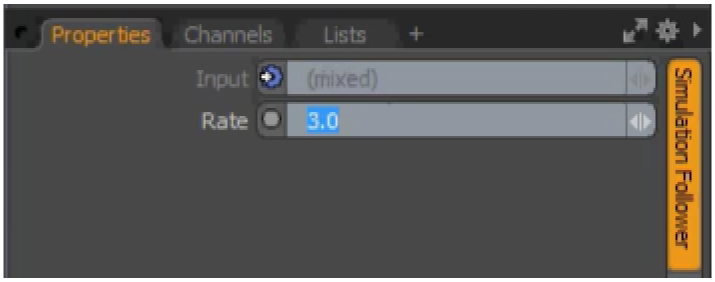
Simulation Follower

The **Simulation Follower** modifier allows you to set target locations for objects to follow in your scene. When the animation is played over time, it displays the object following the Locator item.

The following is a simple example of a Camera following Locators set up on the **X** and **Y** axis. These are passed through two **Followers** which then control the **Camera X** and **Y** values. The image below displays the setup in the **Schematic** viewport.


Perspective
Shaded
Ray GL: Off
Locator
Channels: 0
Deformers: ON
GL: 0
500 mm
Workspace: Workspace
Add Selected
Add...
Locator
Position (Locator)
Position X
Position Y
Position Z
Simulation Follower
Output
Input
Simulation Follower (2)
Output
Input
Camera
Position (Camera)
Position X
Position Y
Position Z

Click **Play Animation** and move the Locator in the scene to view the simulation behavior. The Camera follows the Locator item with a little delay in time. You can control that delay by altering the **Rate** value of the Simulation Follower's **Properties** panel. The lower the rate, the less frames are used to calculate the movement of the camera.


Properties
Channels
Lists
+
Simulation Follower

Input (mixed)

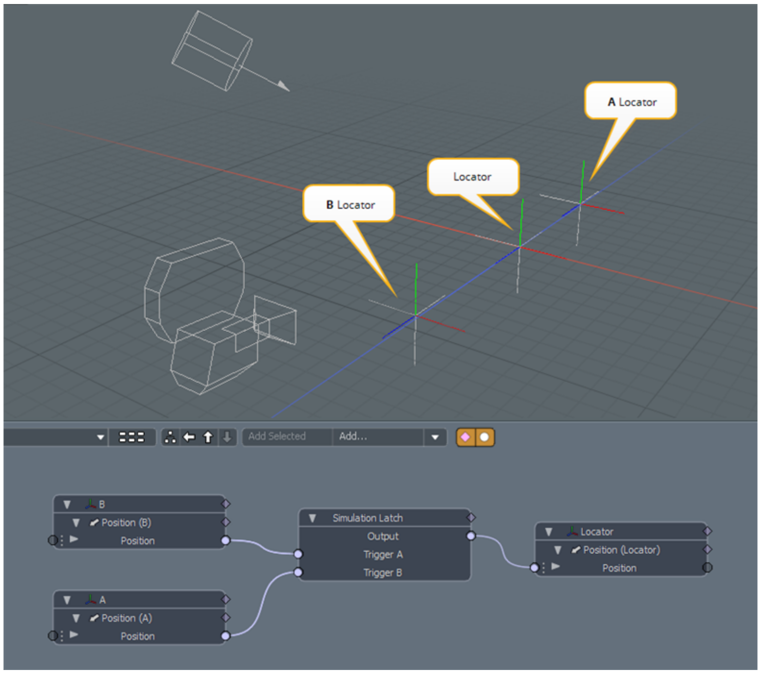
Rate 3.0

Simulation Follower	
Input	The input value for matching.
Rate	Set the lag time speed for the item following the Locator.

Simulation Latch

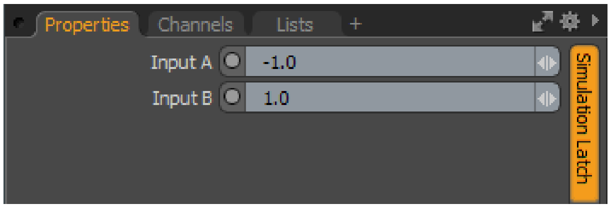
The **Simulation Latch** modifier allows you to latch items to a specified value. When a trigger occurs, meaning when a binary value becomes true, it latches position at a specified value.

Here's a simple example for the Simulation Latch modifier.


A Locator
Locator
B Locator
Add Selected
Add...
B
Position (B)
Position
A
Position (A)
Position
Simulation Latch
Output
Trigger A
Trigger B
Locator
Position (Locator)
Position

The Simulation Modifier's **Output** is driving the locator in the middle of the 3D viewport. There are two triggers, the **A** and **B** locators.

The scene is set up so that when the **A** or **B** locator's **X** value becomes positive, it fires the latch. You can set this in the Simulation Latch item's **Properties** panel:


Properties
Channels
Lists
+
Simulation Latch

Input A -1.0

Input B 1.0

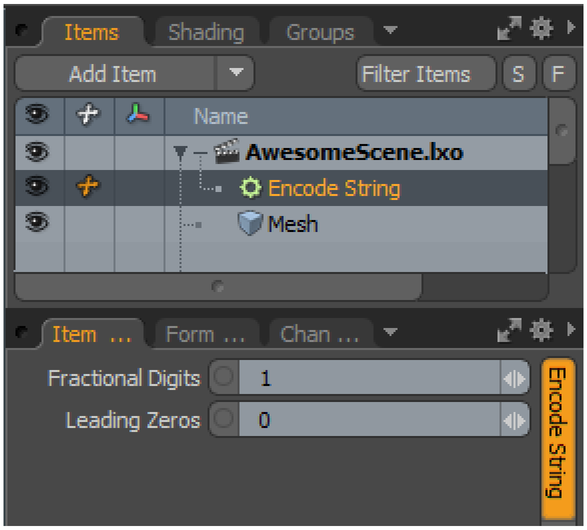
When you click Play Animation, then move the **A** locator to a negative position on X moves the middle Locator to the -1.0 position. Moving B to a negative X position moves the middle Locator to 1.0.

Simulation Latch	
Input A/B	The position an item moves to as a result of trigger A/B

String Modifiers

Encode String

The **Encode String** channel modifier takes a float input and outputs a string encoding the number. There are two other channels controlling the encoding.


Items
Shading
Groups
Add Item
Filter Items
S
F
Name
AwesomeScene.lxo
Encode String
Mesh
Item ...
Form ...
Chan ...
Encode String

Fractional Digits 1

Leading Zeros 0

Fractional Digits: The number of digits after the decimal point.

Zeros: The number of digits in front of the decimal point. If the number has fewer than required, the string is padded with leading zeros. If the number requires more digits it becomes longer.

String Compose

The **String Compose** modifier allows you to compose a string from multiple integer, float, and string inputs. A pattern string channel allows the formatting of the output string to be defined. By constructing a pattern string with certain keywords, you can insert dynamic channel values directly into a string.

The formatting for the keywords, or wildcards, in the pattern follow the convention **$[datatype:index]**. For example, if you want to insert the first integer input value from the **Integers** channel into the output string, enter the following into the **Pattern** field:

```
$[i:1]
```

To write the third float input value from the **Floats** channel into the output string, enter:

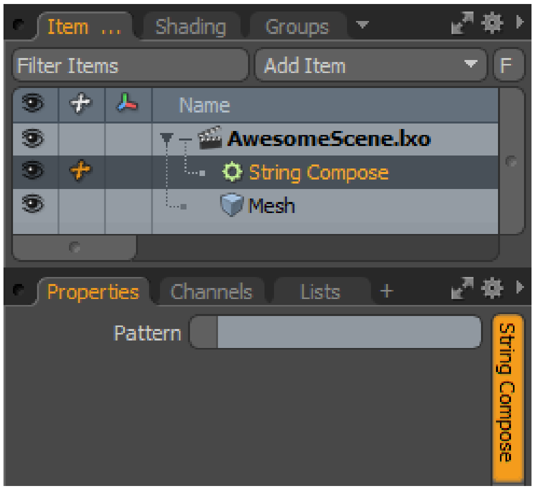
```
$[f:3]
```

Finally, to write the tenth string input value from **Strings** channel, enter:

```
$[s:10]
```

Wildcards can be used in a pattern as well as constant strings. For example, the following pattern is perfectly valid:

```
A string: $[s:1]. An item position: $[f:1]m x $[f:2]m x $[f:3]. A repeated
integer: $[i:1] and $[i:1]
```



Integers: Allows multiple integer channels to be connected, which can be inserted dynamically into the output string.

Floats: Allows multiple float channels to be connected, which can be inserted dynamically into the output string.

Strings: Allows multiple string channels to be connected, which can be inserted dynamically into the output string.

Pattern: The **Pattern** dictates how the input strings are used to compose the output string. As outlined above, the **Pattern** string can contain wildcards using the format **$[datatype:index]**. Whenever the modifier encounters a wildcard in the pattern string, it replaces it with a value matching the data type and index from the three input channels. For example, if it encountered **$[f:1]**, it would replace **$[f:1]** with the first input from the floats channel.

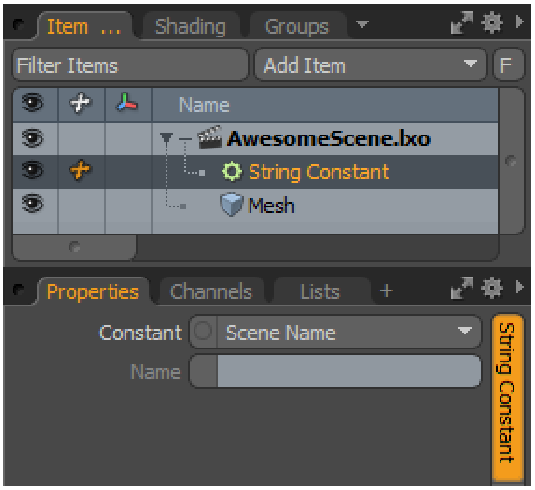
Output: Outputs the composed string.

String Constant

The **String Constant** modifier outputs various pre-defined string values that can be fed into other string modifiers, or into things like string and file path channels.

A **Constant** channel allows the output type to be specified, and in certain cases, a name channel allows the name of the **Constant** to be specified. For example, if the **Constant** channel is set to **Environment Variable**, the name channel specifies which environment variable to output.

An item graph allows items to be connected, such as **item name**, **item type**, and **item tag**, which can be read for their string properties.



Constant: Scene Name

Name:

Constant: Specifies the type of constant to be output. Choose from:

- Scene Name
- Scene FileName
- Scene Filepath

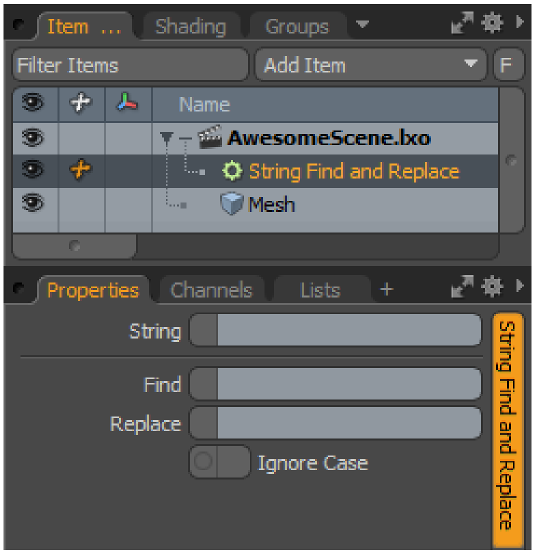
- Scene Title Tag
- Scene Author Tag
- Scene Revision Tag
- Scene Shot Tag
- Input Item Name
- Input Item Type
- Input Item Tag
- Project Path Alias
- Content Path Alias
- Assets Path Alias
- Samples Path Alias
- Scripts Path Alias
- Kits Path Alias
- Environment Variable

Name: When the **Constant** channel is set to either **Environment Variable** or **Input Item Tag**, this field allows the name of the environment variable or item tag to specified.

Output: Outputs the string constant.

String Find and Replace

The **String Find and Replace** modifier allows all occurrences of a search string to be replaced with a target string. By default, the search string and input string are case sensitive, but an **Ignore Case** option allows case sensitivity to be ignored.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
String Find and Replace
Mesh
Properties
Channels
Lists
+
String
Find
Replace
Ignore Case
String Find and Replace

String: The input string to modify.

Find: The sub-string to search for and replace.

Replace: The value to replace all occurrences of the **Find** sub-string in the input string.

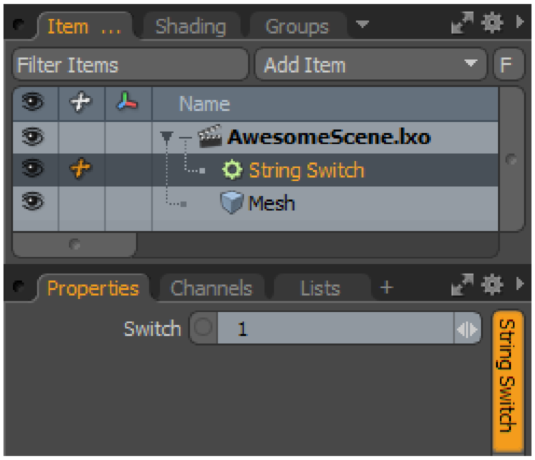
Ignore Case: Specifies whether the operation is case sensitive or not. This value defaults to disabled.

Output: Outputs the modified string.

String Switch

The **String Switch** modifier allows you to dynamically switch between multiple string inputs, in a similar way to the Numeric Switch Modifier modifier, which allows you to switch between multiple scalar inputs. Multiple strings are input and a **Switch** channel allows you to choose which string input to output.

> **Note:** For consistency with the existing **Switch** modifier, the **Switch** channel starts at **1**, rather than **0**.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
String Switch
Mesh
Properties
Channels
Lists
+
String Switch

Switch: 1

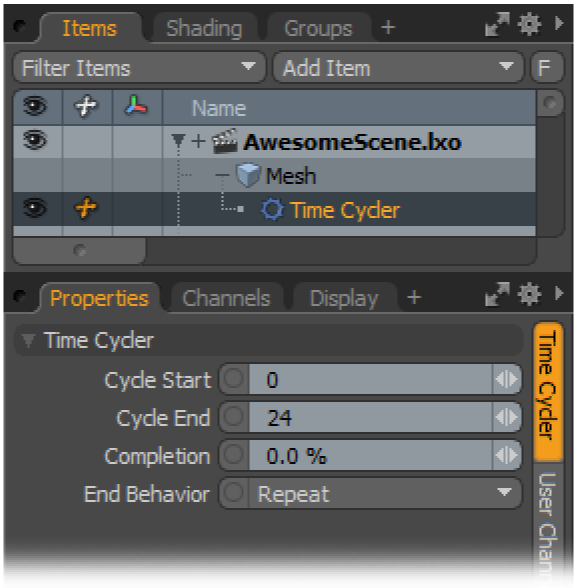
Inputs: Allows multiple string channels to be input.

Switch: Specifies the index of the input string to be output.

Output: Outputs the input string with the index that matches the Switch channel.

Time Modifiers

Time Cycler



Time Cycler

Cycle Start 0

Cycle End 24

Completion 0.0 %

End Behavior Repeat

The **Time Cycler** takes a range of an animation on the Input channel and outputs that animation from the start to an end point based on the **Completion** percent parameter. For example, you could use this for driving a piston animation to follow a wheel's rotation or in cases where you need to re-time a section of an animation. You can use the **End** behavior channel to determine what happens when the parameter is outside the normal 0-100% range. For input and output values, you can use either floating point values or a matrix – or both.

Time Cycler—

Cycle Start/**Cycle End**: Defines the animation frame range as the input item that Time Cycler uses.

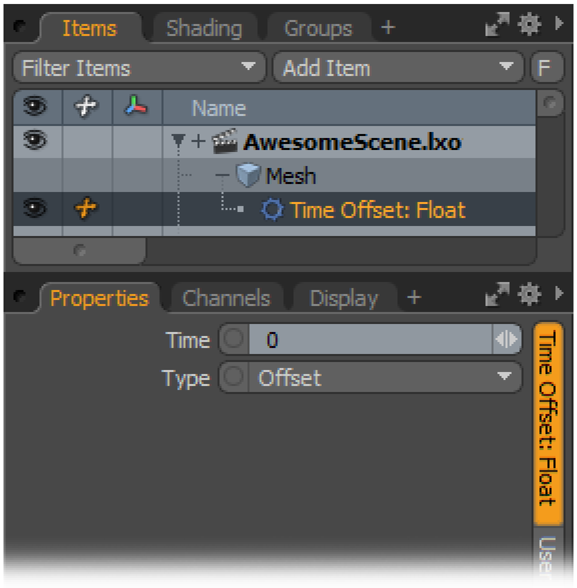
Completion: Specifies the X, Y, and Z coordinates of the input vector.

End Behavior: Defines what action Time Cycler takes for **Completion** values outside the 0-100% range. **Hold** retains the last frame of the animation; **Repeat** reruns the animation cycle; **Ping Pong** plays the animation backwards and forwards in succession; and **Offset Repeat** reruns the animation and offsets it (based on the location of the item at the 100% Completion time) in each subsequent loop.

Time

The **Time** operator provides the current scene's Timeline value. This operator has no user-modifiable controls or channels. You could use the **Time** operator to produce an input for a generation operator (such as a Waveform modifier) that requires a time input. When combined with math operators, you can apply the **Time** operator for interesting effects.

Time Offset



Time 0
Type Offset

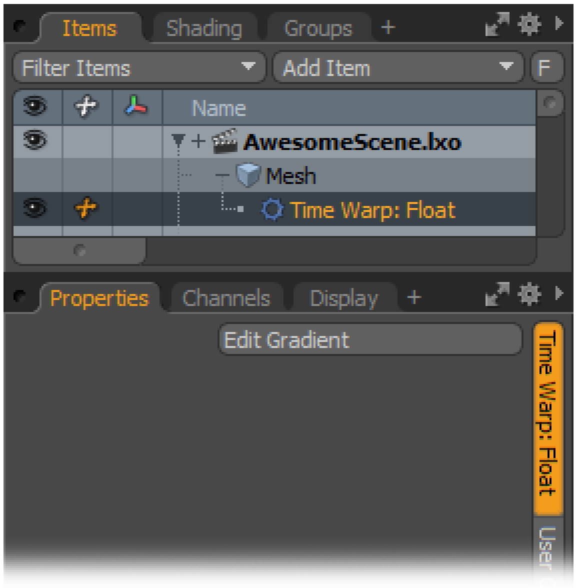
With the **Time Offset** modifier, you can have Modo read a channel value at a different point in time from the current scene time. For example, you can have one item automatically following another. Modo includes two types of time offset modifiers. One takes a floating-point value and the other takes a matrix as input. You can use each to read a respective channel type at different points in time.

Time: Indicates when to read the Input channel, in frames. If you set the **Type** channel to **Offset** and you specify a value of **-1**, Modo reads the channel one frame before the current scene time. With a value of **1**, Modo reads the channel value one frame after the current scene time. If you set the **Type** channel to **Constant**, Modo reads the Input channel at the frame defined in the **Time** channel.

Type: Defines how Modo interprets the **Time** channel. With the default, **Offset**, Modo reads a time for the Input channel by adding the **Time** channel to the current scene time. For example, if the current scene time is 1 second and the **Time** channel is -2, Modo uses -1 as the time to read the Input channel. Select **Constant** to have Modo read the Input channel at specific times in the scene. For example, if the **Time** channel is 2, Modo reads the Input channel at 2 seconds regardless of the current scene time. Use this for animating or rigging the **Time Offset** modifier.

Output: This is the value of the Input channel at the time specified by the **Time** and **Type** channels. Modo passes either a floating-point value or a matrix depending on the operator used.

Time Warp


Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Time Warp: Float
Properties
Channels
Display
Edit Gradient
Time Warp: Float
User

Similar to using the **Time Offset** modifiers, you can use the **Time Warp** modifiers to read a channel value from a different point in time than the current scene time. The key difference between a **Time Warp** modifier and a **Time Offset** modifier is that **Time Warp** modifiers use a gradient to represent the time to read from. You can implement certain effects (such as reversing time and scrubbing time back and forth) depending on the shape of the gradient curve. Modo has two types of **Time Warp** modifiers: one takes a floating-point value as input, the other takes a matrix. You can use each to read a respective channel type at different points in time.

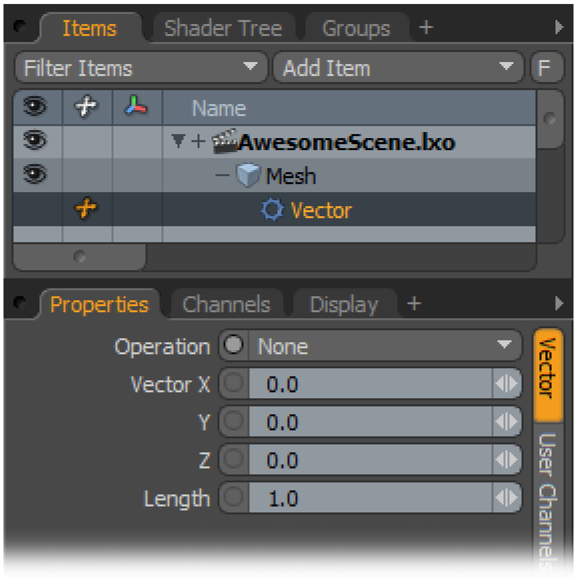
Gradient: Defines the times that Modo reads the Input channel. The value of **X** (along the bottom of the Gradient Editor) is the scene time. The value of **Y** is the time for Modo to read the Input channel. For example, with a key at 1,1 Modo reads the scene time at 1 second and reads the time for the Input channel at 1 second, also. By modifying the shape and slope of the gradient curve, you can effectively slow down time, reverse it, or scrub time back and forth.

Output: This is the value of the Input channel at the time specified by the **Time Warp** channel. Modo passes this as either a floating-point value or a matrix depending on the operator used.

Vector Modifiers

Vector

A vector is a position with both direction and intensity. You can use **Vector** modifiers when creating vectors, setting their length, and normalizing them (setting their length to 1).


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Vector
Properties
Channels
Display
Vector
User Channels

Operation: None
Vector X: 0.0
Y: 0.0
Z: 0.0
Length: 1.0

Operation: The available vector operations are:

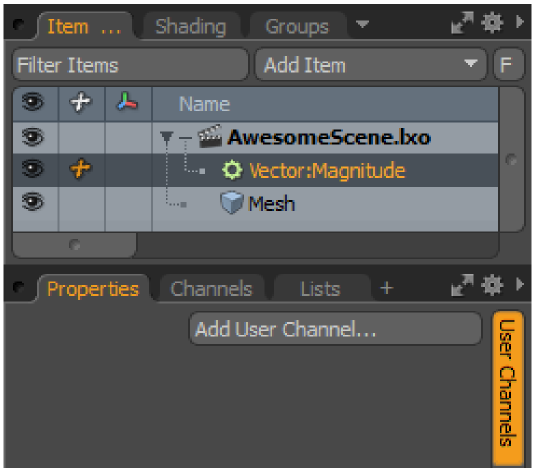
- **None** - Creates a vector to link to other Channel Modifiers or Item channels.
- **Normalize** - Sets the length of the Input vector to 1.0.
- **Set Length** - Sets the length of the Input vector to the value set in the **Length** channel.

Vector X/Y/Z: Specifies the X, Y, and Z coordinates of the Input vector.

Length: Specifies the length of the vector for the **Length** operation.

Vector Magnitude

This modifier measures the length (or magnitude) of a vector. There are no specific options for this modifier.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Vector:Magnitude
Mesh
Properties
Channels
Lists
+
Add User Channel...
User Channels

Start: Specifies the vector to be measured.

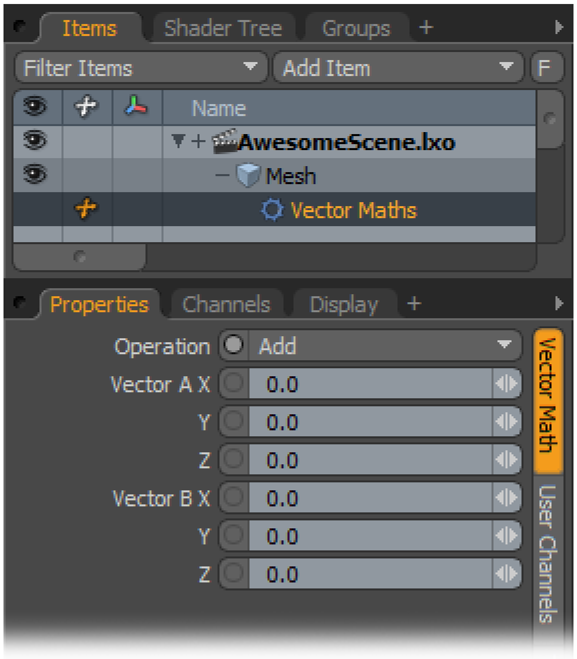
End: Defines the end points of the vector. This is optional, but available in case the vector's origin is not 0,0,0.

Magnitude: Indicates the length of the vector.

Vector Maths

Provides some basic mathematical operations for Modo to perform on 2 Input vectors.

Note: In the case of the dot product, which is a single floating point value, Modo writes the result to the first output channel only (**Output X**).


Items
Shader Tree
Groups
+
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Vector Maths
Properties
Channels
Display
+
Vector Math
User Channels

Operation Add
Vector A X 0.0
Y 0.0
Z 0.0
Vector B X 0.0
Y 0.0
Z 0.0

Operation: Specifies the math operation to apply:

- **Add** - Adds the two Input vectors.
- **Cross Product** - Determines a vector that is at right angles to the two Input vectors on the plane containing them.
- **Dot Product** - Determines the cosine of the angle between the two Input vectors.
- **Subtract** - Subtracts the Input vectors.

Vector A X/Y/Z: Indicates the X, Y, and Z coordinates for vector A.

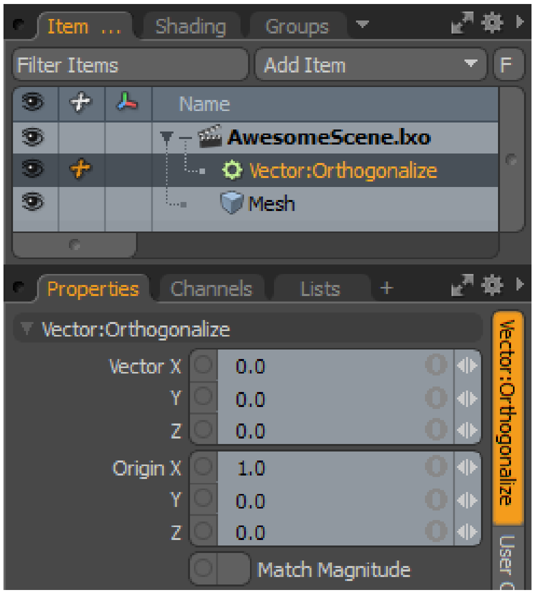
Vector B X/Y/Z: Indicates the X, Y, and Z coordinates for vector B.

Vector Orthogonalize

The **Vector Orthogonalize** modifier orthogonalizes vectors so that no components on the input vector match a second normal vector.

For example, if the input vector is [0.5, 0.5, 0.0] and the normal vector is [1.0, 0.0, 0.0], then the output vector should be [0.0, 0.5, 0.0], as the modifier removes any values aligned to normal - in this case is the entire X component. This also works on non-axis aligned vectors.

The resulting output vector is at a right angle to the normal vector, unless the input vector and normal are aligned, in which case the output vector has a length of zero.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Vector:Orthogonalize
Mesh
Properties
Channels
Lists
+
Vector:Orthogonalize
User C

Vector:Orthogonalize

Vector X 0.0

Y 0.0

Z 0.0

Origin X 1.0

Y 0.0

Z 0.0

Match Magnitude

Vector: Specifies the vector to be orthogonalized.

Normal: Specifies the vector that no components of the input vector match.

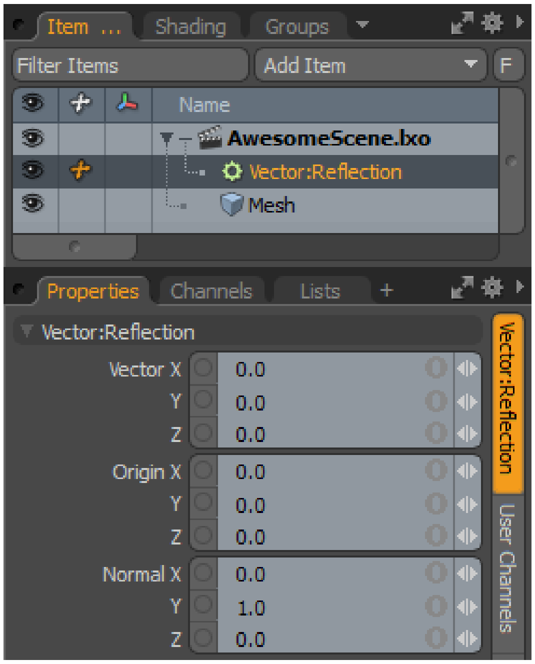
Match Magnitude: When enabled, match the magnitude of the output vector with the input vector. For example, if the input vector is [0.7071, 0.7071, 0.0], it has a length or magnitude of 1.0. If the normal vector is [1.0, 0.0, 0.0], this results in an output vector of [0.0, 0.7071, 0.0], which means that the magnitude or length of the vector has become 0.7071. The **Match Magnitude** channel sets the output vector back to the correct length, resulting in an output vector of [1.0, 0.0, 0.0].

Output: Outputs the orthogonalized vector.

Vector Reflection

The **Vector Reflection** modifier is used for calculating a reflection across a position and normal.

In simple terms, if you have a surface at a position defined by the **Origin** channel, and its normal is defined by the **Normal** channel, then a position defined by the Vector input is reflected to a position that matches the **Output** channel.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Vector:Reflection
Mesh
Properties
Channels
Lists
+
Vector:Reflection
User Channels

Vector:Reflection

Vector X 0.0
Y 0.0
Z 0.0
Origin X 0.0
Y 0.0
Z 0.0
Normal X 0.0
Y 1.0
Z 0.0

Vector: Specifies the input position to be reflected.

Origin: Specifies the origin of the reflection. For the majority of cases, leaving this at [0.0, 0.0, 0.0] should be sufficient.

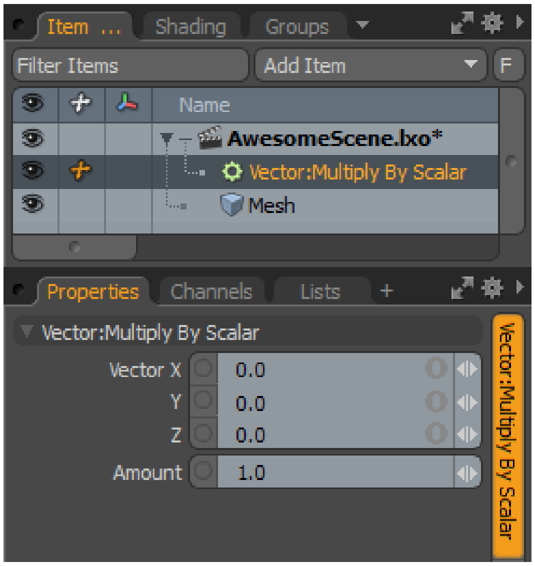
Normal: Specifies the **Normal** of the surface that the input vector is being reflected in. Essentially, this is defining the axis that is used for reflection.

Output: Outputs the reflected position of the input vector.

Vector Multiply by Scalar

The **Vector Multiply by Scalar** modifier takes a vector and multiplies each component by a single scalar input. For example, if the vector is [1.0, 0.5, 0.25] and the scalar is 2.0, then the output vector is [2.0, 1.0, 0.5].

Tip: The **Vector Multiply by Scalar** modifier can be used instead of individual Math nodes to perform the same function.


Item ...
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo*
Vector:Multiply By Scalar
Mesh
Properties
Channels
Lists
+
Vector:Multiply By Scalar
Vector X
Y
Z
Amount
Vector:Multiply By Scalar

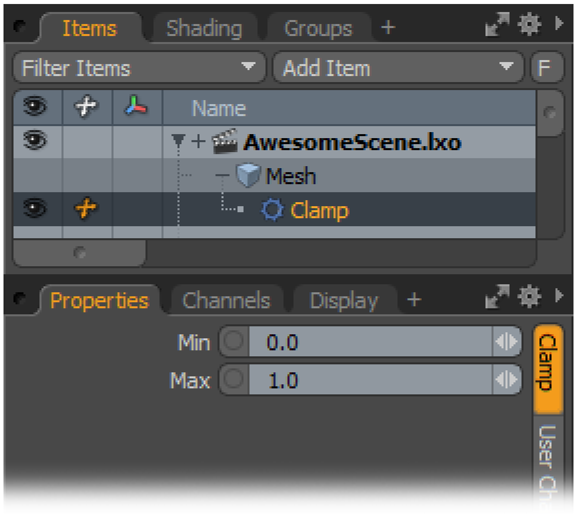
Vector X 0.0
Y 0.0
Z 0.0
Amount 1.0

Vector: Specifies the vector to be multiplied.

Amount: Specifies the scalar amount to multiply the vector by.

Output: Outputs the multiplied vector.

Clamp


Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Clamp
Properties
Channels
Display
Clamp
User Ch

Min 0.0

Max 1.0

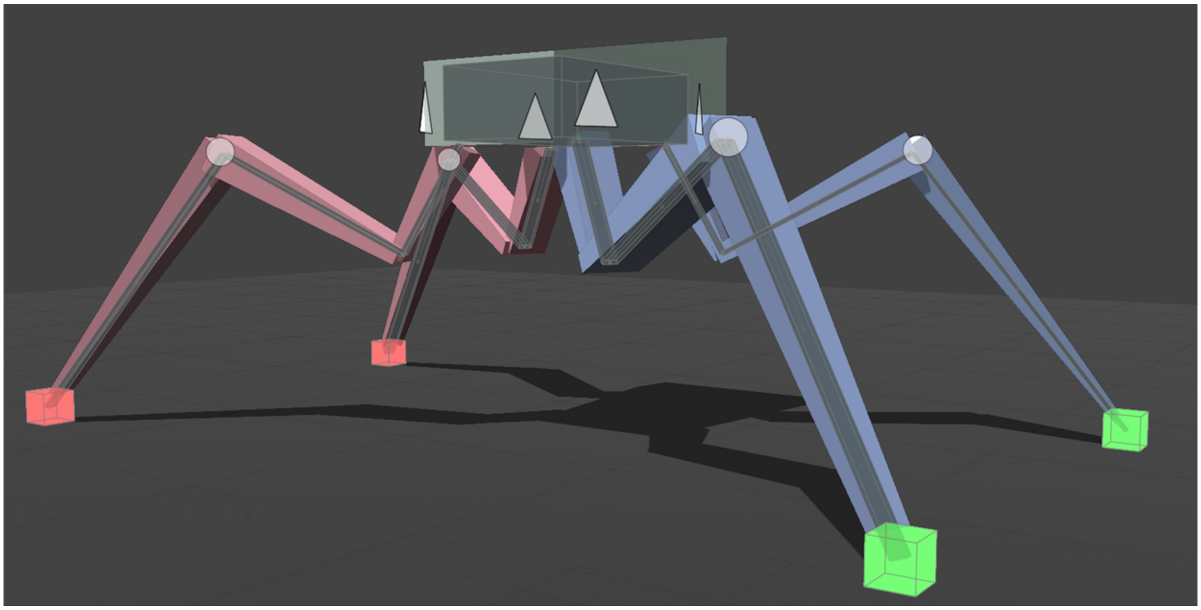
This modifier limits an output value to be within the range defined by the **Min** and **Max** values. Applying this modifier ensures that the output channel never has a value lower than the **Min** value or a value higher than the **Max** value.

Min: Determines the minimum output value of the Clamp operator.

Max: Determines the maximum output value of the Clamp operator.

Output: The **Output** channel passes the value from the input channel, but clamped by the minimum and maximum values.

Planar IK



Inverse Kinematics (IK) is a method of animating items in which software calculates the bending of a joint based on the position of a goal object. In Modo, the goal object is usually a locator. Once you have set up IK for a series of items, the IK hierarchy causes the item to continually reach for the goal when the goal item moves around and modifies the joint rotations as necessary.

With IK, it is much easier to animate the position of one item (the goal) than to animate the rotations of several separate items. For example, to animate a hand on a doorknob with IK, the goal can be positioned at the door knob and the hand is in the proper place in the scene. With a standard hierarchy, both the shoulder and the elbow must be rotated independently to find the correct position for the hand.

Modo's Planar IK system works well for dual joints, such as the arm mentioned in the previous example, but also allows you to animate three locator joints, for instance an animal's leg.

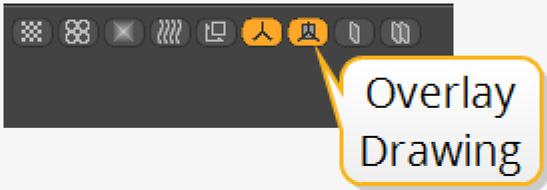
Note: You may also consider the more powerful Full Body IK system, which allows you to rig an entire character.

Applying IK to a Chain of Locators

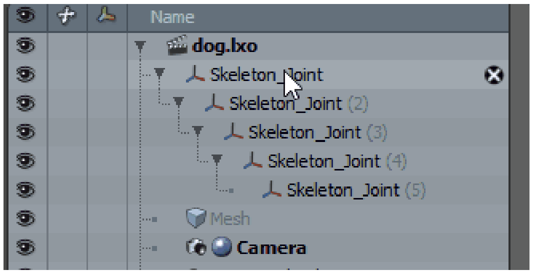
To apply IK to a chain of items, first create the hierarchy:

- Parent the items to each other in a descending chain.
- Make sure the Center positions are in the proper location as well. For Modo to calculate the IK correctly, all Center positions must share at least one axis value to appear in the same plane.
- Make sure the chain has a bend to it so that when Modo applies the IK, it can use a preferred angle for bending.

Tip: Enable Overlay Drawing in the 3D viewport to make the joints visible within the mesh.


Overlay Drawing

1. Once the items are in order, select the items in descending order.

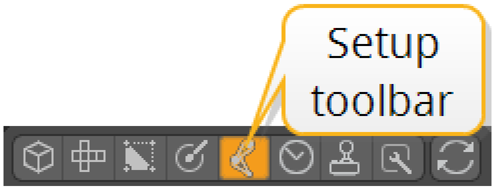

Name
dog.lxo
Skeleton_Joint
Skeleton_Joint (2)
Skeleton_Joint (3)
Skeleton_Joint (4)
Skeleton_Joint (5)
Mesh
Camera

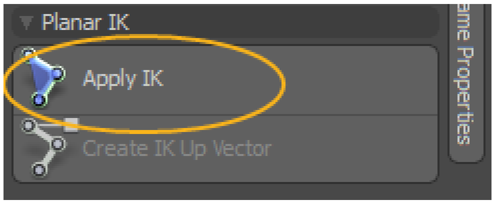
2. Enter Setup mode by clicking the Setup mode button. Items in the Setup action state are understood by Modo to be in their resting or undeformed position.

   A yellow frame appears around the 3D viewport, indicating that you're in Setup mode.

Note: For more information on using Setup mode, see Binding and Setup Mode.

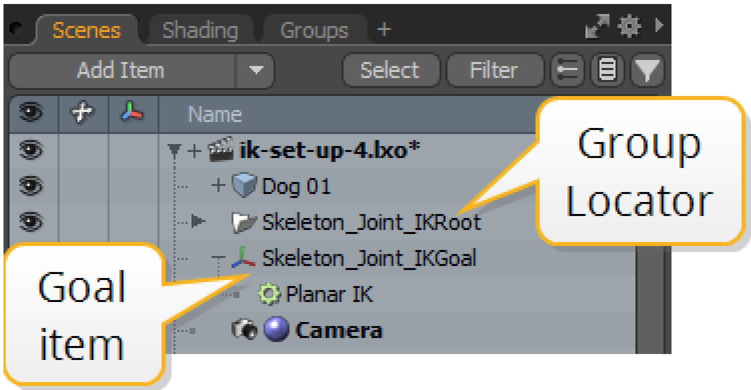
3. In the Setup toolbar, go to the **Inverse Kinematics** tab and under **Planar IK**, click the **Apply IK** button.


Setup toolbar


Planar IK
Apply IK
Create IK Up Vector
ame Properties

Applying the IK modifier to the chain creates a goal item and adds a group locator as the overall parent for the IK chain. The goal item is indicated by a cube that is colored according to the side of the X axis it lies on when the IK is assigned.

Note: A warning is displayed if your joints don't share an axis. Click **OK** to align your items along the X axis.

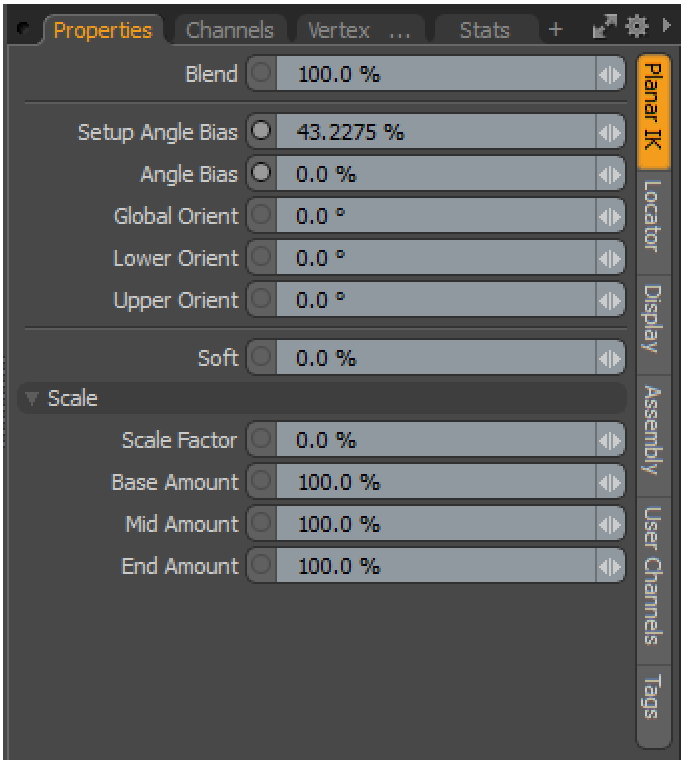

Scenes
Shading
Groups
Add Item
Select
Filter
Name
ik-set-up-4.lxo*
Dog 01
Skeleton_Joint_IKRoot
Skeleton_Joint_IKGoal
Planar IK
Camera
Group Locator
Goal item

4. Exit Setup mode.

   By moving the goal item, Modo modifies the angle of the joints in the IK chain.

Planar IK Properties

Selecting the goal item in the **Item List** displays the properties related to the IK chain.

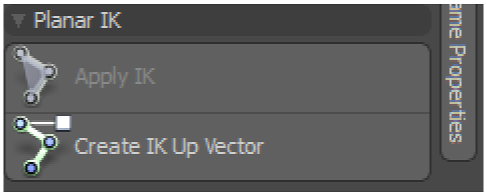


Blend 100.0 %
Setup Angle Bias 43.2275 %
Angle Bias 0.0 %
Global Orient 0.0 °
Lower Orient 0.0 °
Upper Orient 0.0 °
Soft 0.0 %
Scale
Scale Factor 0.0 %
Base Amount 100.0 %
Mid Amount 100.0 %
End Amount 100.0 %

Blend	A setting of 100% fully uses the IK solution. Values below 100% reduce the effect of IK on the chain, and a setting of 0% disables it. **Note:** The orientation lines fade when the Blend value is lowered to indicate the level of the effect.
Setup Angle Bias	Specifies the resting state for the chain. This setting should be adjusted in Setup mode to give a default behavior for the joints in the chain.
Angle Bias	Controls the weighting of the joints towards the start or end of the chain. After leaving Setup mode, you can adjust or keyframe the **Angle Bias** option to alter the behavior of the chain for a specific animation. This amount is added to the **Setup Angle Bias**. So if you have a **Setup Angle Bias** of 50% and an **Angle Bias** of 10%, the overall bias used for the joint angles is 60%. The separate controls allow you to adjust the chain in Setup mode without having to redo any changes made to the bias in animations.

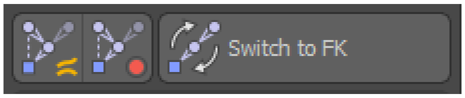
	Note: Angle Bias is available for three-joint chains only.
Global Orient	Specifies the planar orientation angle of the IK chain. For example, for an arm, you can adjust the angle the elbow points toward.
Lower Orient	Specifies the planar orientation angle of the lower two joints of a three-joint chain. **Note: Lower Orient** is available for three-joint chains only.
Upper Orient	Specifies the planar orientation angle of the upper two joints of a three-joint chain. **Note: Upper Orient** is available for three-joint chains only.
Soft	When overextending a joint, it snaps to the goal position by default. Increasing this value removes the snapping and produces a slower transition.
Scale	
Scale Factor	When the goal point is too far from the sum of the joint lengths, the IK solver can scale the joints to match the goal position. 0% means no scaling, while at 100%, the joints are fully scaled. The **Base**, **Mid**, and **End Amount** settings allow you to determine the distribution of the scale amount across the joints.
Base Amount	Scales only the first joint.
Mid Amount	Scales only the middle joint.
End Amount	Scales only the last joint.

Create IK Up Vector



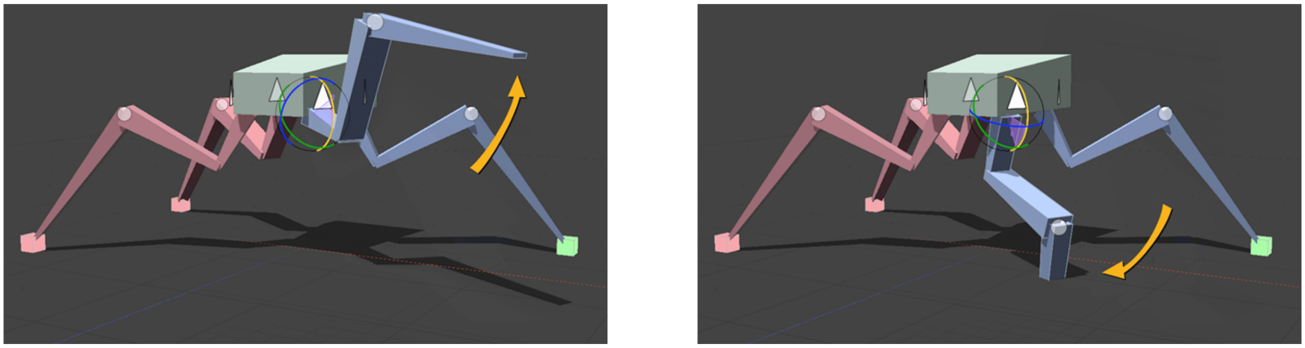
This vector defines the up direction for an IK constraint and can help to keep the chain oriented properly during animation. The **Create IK Up Vector** button is similar to the Assign Up Vector command except you don't need to create and position a locator. With **Create IK Up Vector**, Modo generates the locator automatically based on the preferred bending angle. Once the new Up Vector locator is available, you can animate it to adjust the orientation of the IK rig. This is similar to adjusting the **Global Orient** setting in the IK **Properties** panel. You can also combine IK Up Vector with **Global Orient** to produce any required orientation.

IK/FK Switch

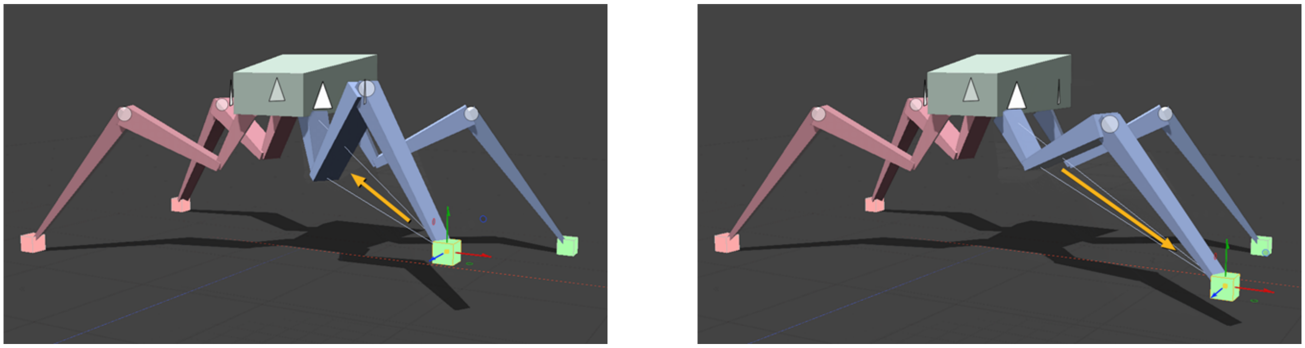


These tools, located in the **Setup** toolbar under **Inverse Kinematics** > **Planar IK**, allow you to switch between IK and FK on Planar IK chains. Artists may need to switch between IK and FK depending on the type of movement they are animating.

Note: The IK/FK Switch tools are only available in Modo versions 14.1v1 and later.



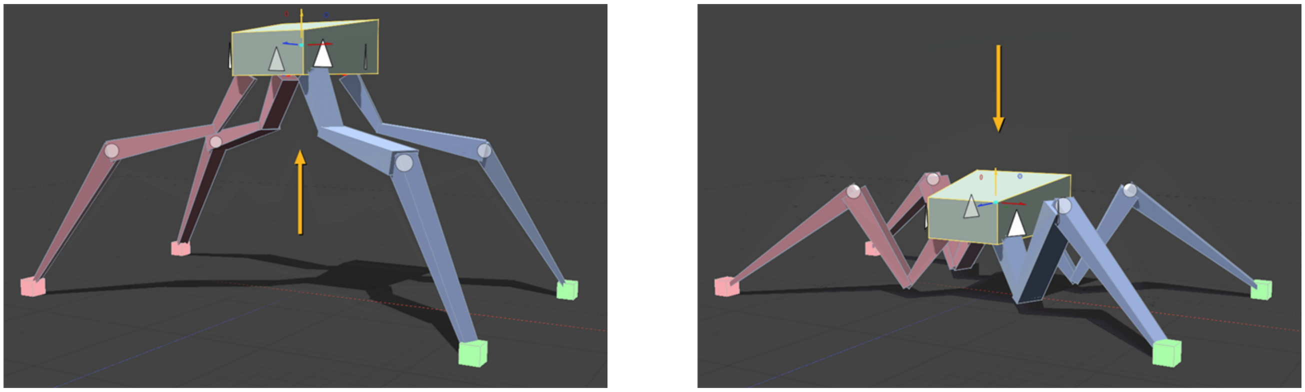
FK example



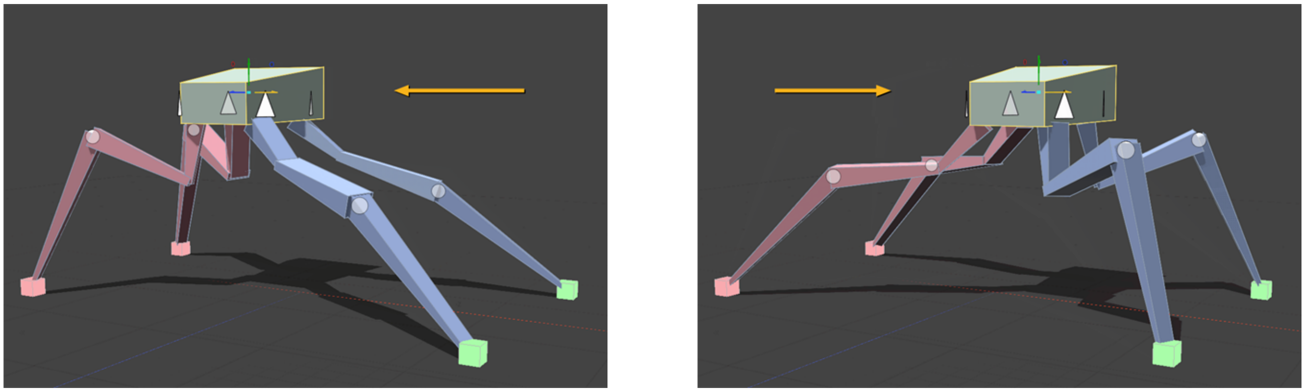
IK example

Using FK means each bone follows the movement of its parent. This method is great for animating movement which requires smooth animation arcs such as a kicking motion. IK works up the skeleton hierarchy from the goal, as opposed to down it. This is useful when a joint needs to remain in place while another joint further up in the hierarchy moves. For example, applying an IK to a character's feet allows them to remain on the ground as the body is moved around. Modo calculates the angles of the joints between the IK and the joint.

In the following examples there is a goal located on each of the character's feet. The feet therefore stay in place while the body is moved. The joints between the goal and the body are calculated accordingly.

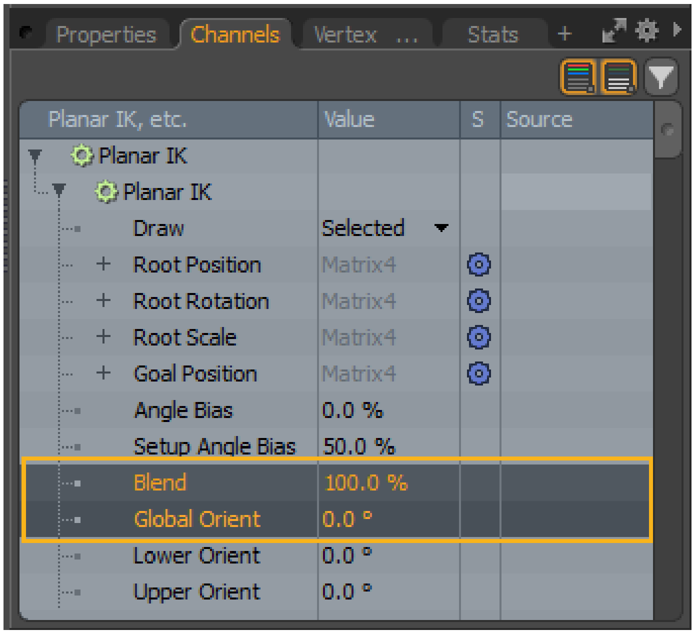


IK Up/Down movement example



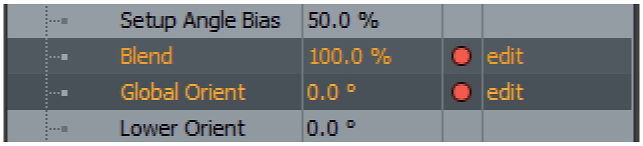
IK Left/Right movement example

The **IK/FK Switch: Select Channels** button selects the channels used by IK/FK Switching on the Solver item(s). This can be used with the goal item or any joint in the chain selected. This option is useful if, for example, you want to add the switch channels to channel only Actors.



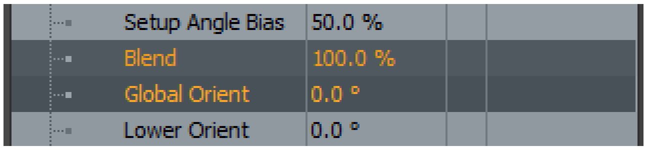
Planar IK, etc.	Value	S	Source
Planar IK			
Planar IK			
Draw	Selected		
+ Root Position	Matrix4	⚙	
+ Root Rotation	Matrix4	⚙	
+ Root Scale	Matrix4	⚙	
+ Goal Position	Matrix4	⚙	
Angle Bias	0.0 %		
Setup Angle Bias	50.0 %		
Blend	100.0 %		
Global Orient	0.0 °		
Lower Orient	0.0 °		
Upper Orient	0.0 °		

The **IK/FK Switch: Set Keys** button sets keys on the IK Solver channels that are used for IK/FK switching. While not required, this can be used at the start of the scene to keyframe the initial values for the channels. Hold **Shift** to change this button from **IK/FK Switch: Set Keys** to **IK/FK Switch: Delete Keys**. When pressed, this button deletes the keys on the Solver IK/FK Switch channels.



Setup Angle Bias	50.0 %		
Blend	100.0 %	●	edit
Global Orient	0.0 °	●	edit
Lower Orient	0.0 °		

IK/FK Switch: Set Keys

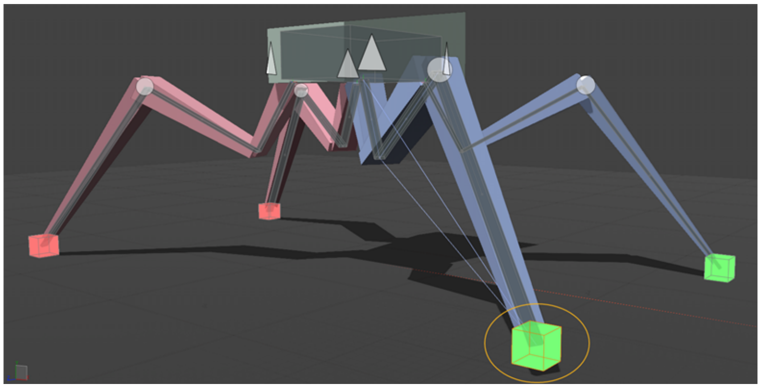


Setup Angle Bias	50.0 %		
Blend	100.0 %		
Global Orient	0.0 °		
Lower Orient	0.0 °		

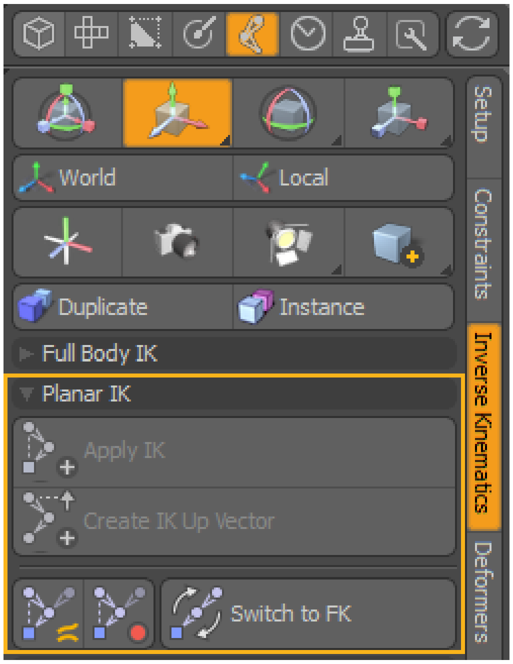
IK/FK Switch: Delete Keys

Switching Between IK and FK

1. Select a joint, goal or solver item belonging to a Planar IK chain.



2. Navigate to the **Inverse Kinematics** tab in the **Setup** toolbar and scroll down to **Planar IK**.


World
Local
Duplicate
Instance
Full Body IK
Planar IK
Apply IK
Create IK Up Vector
Switch to FK
Setup
Constraints
Inverse Kinematics
Deformers

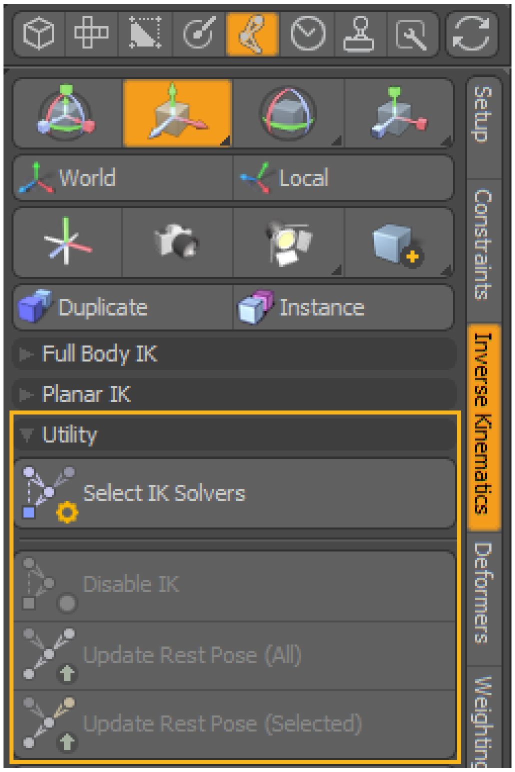
3. Click the **Toggle IK/FK** / **Switch to IK** / **Switch to FK** button.
   The label on the button updates according to the current state of the IK chain. If Inverse Kinematics is active on the chain it reads **Switch to FK**, otherwise it reads **Switch to IK**. If items belonging to multiple chains are selected and those chains are in different states it reads **Toggle IK/FK**.

Note: The Goal item position channels are not automatically keyed by this button so should be keyed manually as normal.

Note: When switching to FK the local rotation and position channels of all the Joints in the chain are keyed. The position channels are keyed to support scaling of the chain.

Utility Options

Utility options have been added to the **Inverse Kinematics** tab under the **Setup** toolbar in Modo 14.1v1. Using these buttons you can now disable both Planar and Full Body Inverse Kinematics in setup mode, as well as update the Rest Pose and quick select IK solvers.


World
Local
Duplicate
Instance
Full Body IK
Planar IK
Utility
Select IK Solvers
Disable IK
Update Rest Pose (All)
Update Rest Pose (Selected)
Setup
Constraints
Inverse Kinematics
Deformers
Weighting

To quick select an IK Solver item, select a Joint or Goal item belonging to the chain from which you want to access the IK Solver item, and click the **Select IK Solvers** button.

To disable an IK from Setup mode, enter Setup mode and click the **Disable IK** button found in the **Setup** toolbar under **Inverse Kinematics** > **Utility**. This toggles the IK off and on. The state is stored as a per scene setting and is saved in the **.lxo** scene file.

Note: When disabling IK, the Joints and Goals are not automatically returned to their Rest Pose.

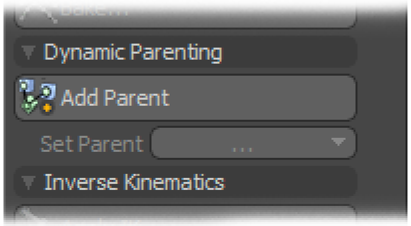
Whilst IK is disabled, the **Update Rest Pose (All)** and **Update Rest Pose (Selected)** buttons are active. These allow the Rest Pose of all IK chains in the scene to be updated, or just selected chains by selecting any Joint or Goal item belonging to the chain you want to update.

Dynamic Parenting

When you parent an item to another, the child inherits that parent's position, rotation, and scale. Any transforms of its own are added to the those of its parent. Normally, this parent/child hierarchy is a fixed relationship. Parenting this way cannot be animated, and the relationship remains in effect for the duration of the scene. (For more information about setting up hierarchies, see the Hierarchy topic.)

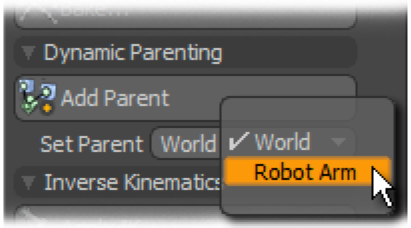
There are many situations where being able to change the parent of an item is very useful. For example, a robotic arm on an assembly line picks up a component, attaches to it, and does something to a product on a conveyor belt. Once the robotic arm completes the task, the product moves off for the next stage of assembly. You could animate this process by having multiple versions of the component: one in the parts bin, one parented to the robot arm, and another parented to the product on the conveyor. By animating the visibility of each of these copies, you could give the impression of the component being picked up and attached to the product being assembled, but this is tedious to implement and difficult to manage.

Ideally, you'd want to specify an item is parented to one item and, then at another point in time, it's parented to another item, and so on. With Dynamic Parenting you can do that.



To define a Dynamic Parent relationship, first select the child item. (In the robotic arm example, this would be the component.) Then select the acting parent item (for example, the robotic arm). Once you have selected both items in order, make sure that **Compensation** is on for the child item to remain in its current relative location. Then, click the **Add Parent** button to define the relationship.

Below the **Add Parent** button, the **Set Parent** list becomes active after you apply Dynamic Parenting. With this list, you specify the current parent for the item. If you open the list, you should see at least two entries: one for the parent that you just assigned and one for the **World**, which effectively means no parent. If you specify the new parent for the child and move the parent, the constrained item follows along just as if it were parented by conventional means. To add more parents, select the constrained item and then the item that is to be the parent. Click the **Add Parent** button again. You can add as many parent items as you wish at any time.

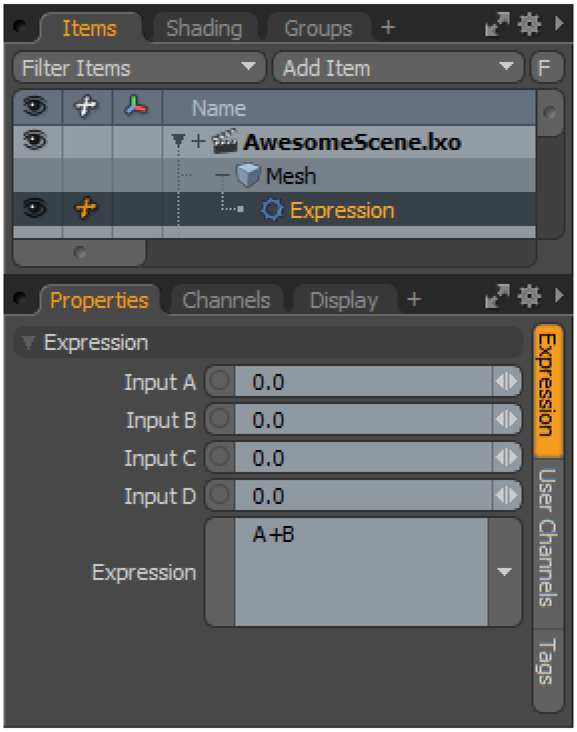


To change the current parent, select the constrained item (the child item) and select the new parent from the **Set Parent** list in the **Properties** viewport for the **Dynamic Parent Channel Modifier** item. (You may need to click the **+** next to the constrained item in the Items (Scenes) List to select the child item.) You can also specify the parent in the tool bar. Once you define all the parent items, you can apply different ones over time with keyframes. If you enable **Auto Key**, then by just moving the Timeline to a new position and specifying a new parent, Modo changes the child item dynamically to the new parent. You can also manually apply keyframes on the **Parent** Channel under the **Dynamic Parent** modifiers channels. You specify the keyframes numerically but represent the items in the list top down. In the robotic arm example, **0** equals **World**, **1** equals **Robot Arm**, **2** equals **Component**, and so on.

Note: You must enable **Compensation** when dynamically parenting items to ensure that items don't jump around when parented.

Expression Modifier

Expression



Input A 0.0
Input B 0.0
Input C 0.0
Input D 0.0
Expression A+B

With the **Expression** Channel Modifier you use various mathematical and logical operators to define an output value based on input values. Expressions are especially useful in creating complex animations that don't require keyframes, and they are easy to adjust and update. The **Expression** modifier uses the muParser expression engine.

Input A/**Input B**/**Input C**/**Input D**: These are the user-defined input variables. Each is a floating-point value and is referred to in an expression by its variable name: **A**, **B**, **C**, or **D**.

Expression: This is the expression that Modo executes based on the defined inputs. For example, **A** passes the value coming in on the **Input A** channel to the output channel, or **C+D** adds the values coming in on the **Input C** and **Input D** channels and sets the sum to the output channel. You can also use built-in functions. For example, **sin(A)** outputs the sine value of **Input A** to the output. You can also use conditional statements. For example, with **A>B?C:D** Modo sends the value in **Input C** to the output if **Input A** is greater than **Input B**; if not, Modo sends the value in **Input D**.

Output: The output value is the value returned by the user-defined expression.

Built-in Functions

By default, Modo with the muParser expression engine includes the following functions. The table lists the function names and a brief description of each with the number of arguments, if applicable.

Name	**Meaning**
sin	sine function
cos	cosine function
tan	tangens function
asin	arcus sine function
acos	arcus cosine function
atan	arcus tangens function
sinh	hyperbolic sine function
cosh	hyperbolic cosine
tanh	hyperbolic tangens function
asinh	hyperbolic arcus sine function
acosh	hyperbolic arcus tangens function
atanh	hyperbolic arcur tangens function
log2	logarithm to the base 2
log10	logarithm to the base 10
log	logarithm to the base 10
ln	logarithm to base e (2.71828...)
exp	e raised to the power of x
sqrt	square root of a value

Name	Meaning
sign	-1 if x is less than 0; 1 if x greater than 0
rint	round to nearest integer
abs	absolute value
min	min of all arguments
max	max of all arguments
sum	sum of all arguments
avg	mean value of all arguments
noise	noise value takes 1-3 arguments
floor(x)	gives the integer value of x (For example, 3.5 becomes 3.)
ceil(x)	rounds up to the next integer value of x (For example, 3.5 becomes 4.)
mod(a,b)	returns the remainder of a divided by b
clamp(a, min, max)	given a value a, if a is less then min, min is returned. If greater then max, max is returned. If in between min and max, a is returned.

Built-in Binary Operators

By default, Modo with the muParser expression engine includes the following binary operators. The table lists the operator, a brief description, and the priority assigned to each operator.

Operator	Meaning	Priority
=	assignment *	-1
&&	logical and	1

Operator	**Meaning**	**Priority**
||	logical or	2
< =	less or equal	4
>=	greater or equal	4
!=	not equal	4
==	equal	4
>	greater than	4
<	less than	4
+	addition	5
-	subtraction	5
*	multiplication	6
/	division	6
^	raise x to the power of y	7

*The assignment operator is special because it changes one of its arguments. You can only apply it to variables.

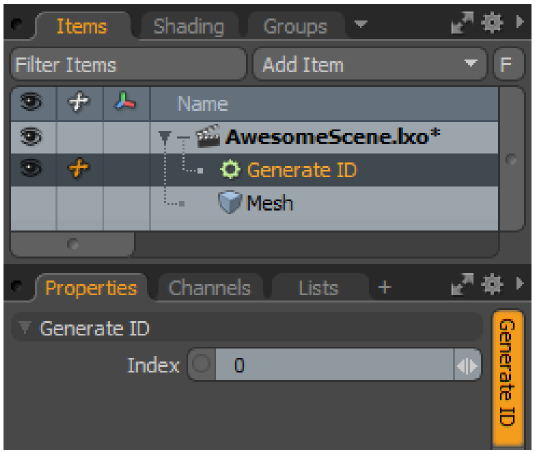
Other Operators

The muParser expression engine has built-in support for the if/then/else operator. It uses a 'lazy' evaluation to make sure only the necessary branch of the expression is evaluated.

Operator	**Meaning**	**Remarks**
?:	if then else operator	C/C++ style syntax

Generate ID

The Particle Constraint modifier uses a floating point value to specify the particle to be constrained to. This matches the rest of the particle system, and allows it to interface with things like the Particle Sample nodal shading node. However, from a user point of view, a floating point index can be quite difficult to use. The **Generate ID** node solves this by providing a conversion from an integer index to a floating point index, used by the particle system.

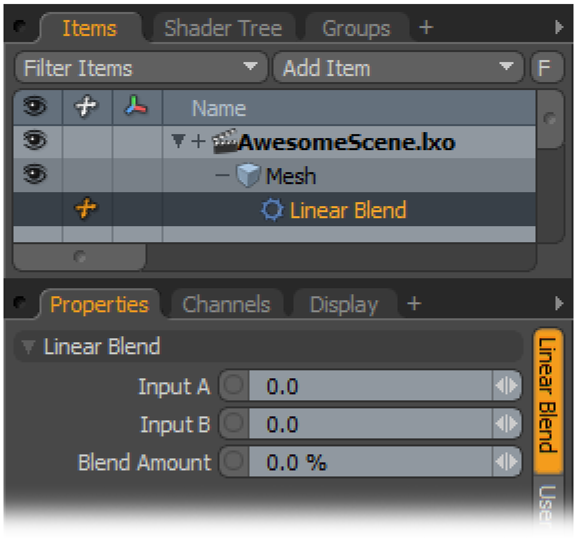

Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo*
Generate ID
Mesh
Properties
Channels
Lists
+
Generate ID
Generate ID

Index 0

Index: The integer index used to generate the floating point ID.

ID: The resulting floating point particle ID.

Linear Blend


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Linear Blend
Properties
Channels
Display
Linear Blend
User

Input A 0.0

Input B 0.0

Blend Amount 0.0 %

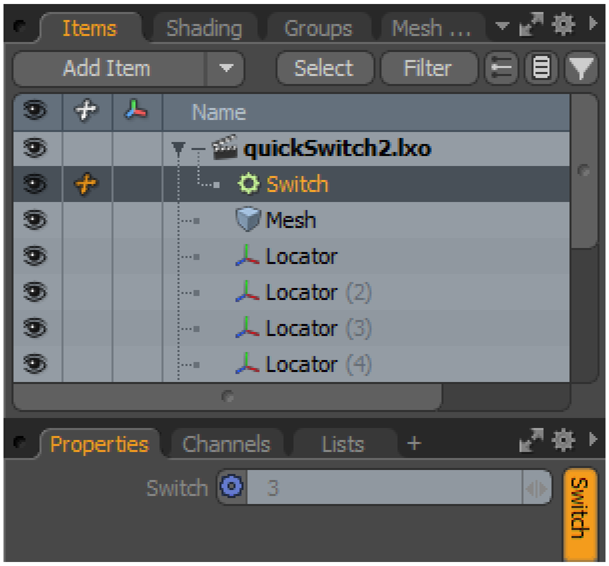
Performs a linear blend between the two input values. The blend value controls the gradation between the two input values. At **0%**, Modo uses only **Input A** and at **100%**, only **Input B**. Modo interpolates any values between 0% and 100% linearly.

Input A: Input value determined from the source input.

Input B: Input value determined from the secondary input.

Blend Amount: Determines the amount of blending between the values for **Input A** and **Input B**.

Numeric Switch Modifier

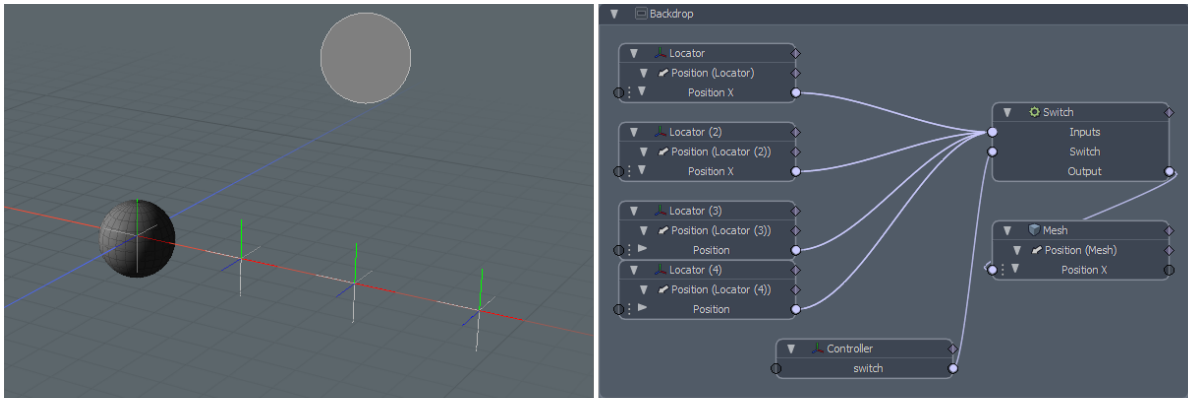

Items
Shading
Groups
Mesh ...
Add Item
Select
Filter
Name
quickSwitch2.lxo
Switch
Mesh
Locator
Locator (2)
Locator (3)
Locator (4)
Properties
Channels
Lists
Switch

Switch 3

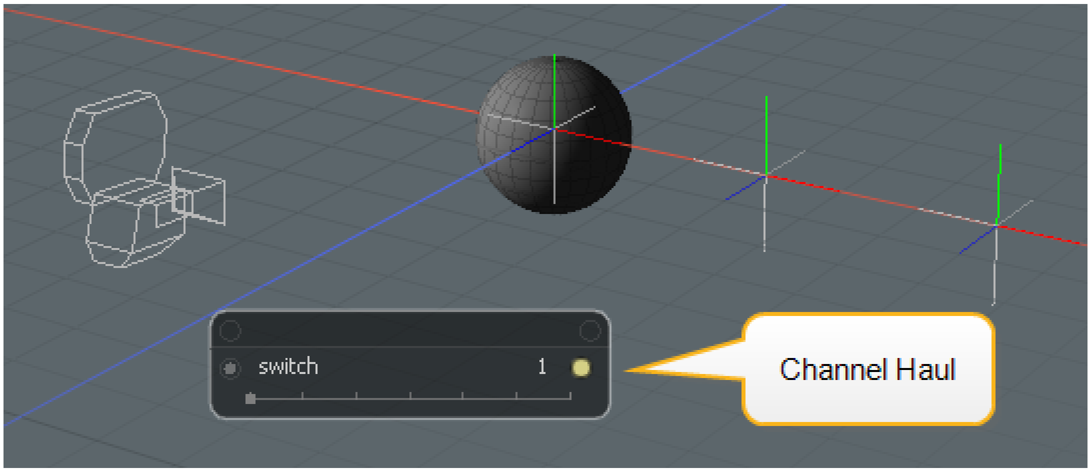
The **Numeric Switch** modifier takes multiple inputs and, based on the options, defines which of them to output. The **Switch** channel indicates the **Input** channel to output. If the **Switch** channel is greater than the number of inputs, the output is 0.

Note: Before Modo 13.0 the Numeric Switch node was called Switch.

The following example demonstrates the use of the Switch modifier. In this case we use it to place a sphere to one of four positions. The four locator items are the four possible positions the sphere can move to. They are all connected to the Switch modifier's **Inputs** channel. The sphere's position is set by the **Controller**, a fifth locator with a **switch** user channel added to it. This **switch** channel drives the Switch modifier's **Switch** channel. The sphere is linked to the modifier's **Output** channel, which moves it to the position specified by the Switch value.


Backdrop
Locator
Position (Locator)
Position X
Locator (2)
Position (Locator (2))
Position X
Locator (3)
Position (Locator (3))
Position
Locator (4)
Position (Locator (4))
Position
Switch
Inputs
Switch
Output
Mesh
Position (Mesh)
Position X
Controller
switch

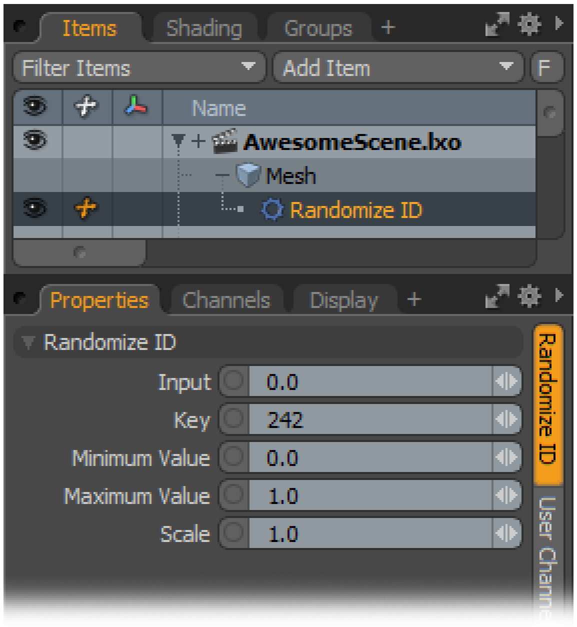
To use the controller, activate Channel Haul by selecting the **Controller** in Item selection mode. Click on a position in the Channel Haul field to move the sphere to the connected locator.


switch
1
Channel Haul

Note: If any inputs get disconnected, Modo may re-order them. This could result in the **Switch** channel no longer providing expected inputs. To avoid this, connect all input channels before animating or rigging the **Switch** modifier.

Randomize ID Modifier

Randomize ID


Items
Shading
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Randomize ID
Properties
Channels
Display
Randomize ID
User Channe

Randomize ID

Input 0.0

Key 242

Minimum Value 0.0

Maximum Value 1.0

Scale 1.0

The **Randomize ID** Channel Modifier generates randomized ID values for creating predictable quasi-random particle effects based on fixed IDs.

Input: Represents the input value that would be randomized.

Key: Modo uses the **Key** channel to initialize the random number generator. A change in the **Key** value changes random values.

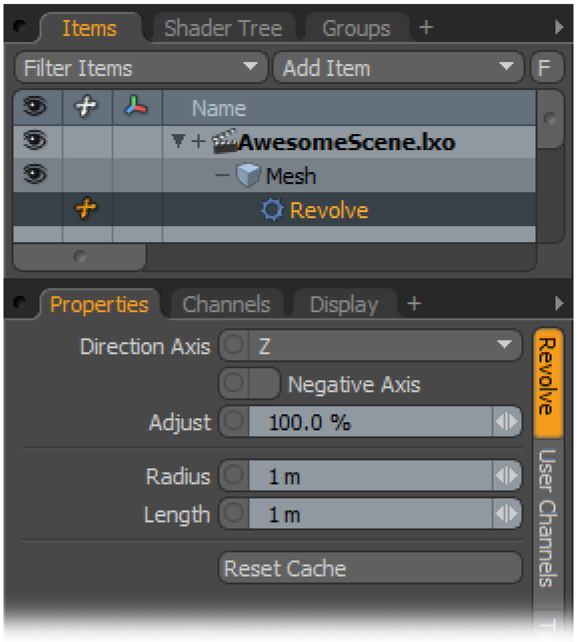
Minimum Value: Defines the minimum value generated by the random number generator.

Maximum Value: Defines the maximum value generated by the random number generator.

Scale: This is a multiplier value that scales the resulting randomly generated value.

Revolve

The primary output of the **Revolve** Channel Modifier is a world space distance traveled along a specified vector, as well as the distance traveled. This modifier also has channels to compute a rotation value from a given radius and a percentage traveled along a specified length. (You can use the rotational value for driving wheels on a vehicle.)


Items
Shader Tree
Groups
+
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Revolve
Properties
Channels
Display
+
Negative Axis
Reset Cache
Revolve
User Channels

Direction Axis: Z
Adjust: 100.0 %
Radius: 1 m
Length: 1 m

Input: Position: Links the World Position transform of the item whose movement is being measured.

Input: Rotation: Links the World Rotation transform of the item whose movement is being measured.

Input: Radius: With a radius set in this channel, the modifier calculates the rotation required to roll a cylinder over the distance traveled without slipping and passes it to the **Angle Output** channel.

Input: Length: Given this length, the modifier passes the percentage of this length traveled to the **Percentage Output** channel.

Input: Adjust: Use this channel to alter the outputs of the modifier. By default, this is set to **100%** and the output is not affected. By setting this to **0%**, the output is static; at **200%** the output is doubled. This value can be used for things like braking, wheel slip, and so forth.

Input: Axis: This specifies which direction vectors (defined by the Position and Rotation transform inputs) Modo uses to work out the distance traveled.

Input: Negative Axis: Selecting this causes the modifier to use the negative axis of the one specified in the **Axis** channel.

Output: Angle Output: This is the angle of rotation calculated from the **Radius** channel. Connect this to one of the rotation channels on the **Rotation** transform of the driven item.

Output: Percentage Output: This is the percentage of the **Length** input channel traveled. For example, if the distance traveled is twice as long as the input length then the output is 200%.

Output: Distance Output: This is the total distance traveled of the driver item along the vector specified by the **Axis** channel.

Notes on using Revolve

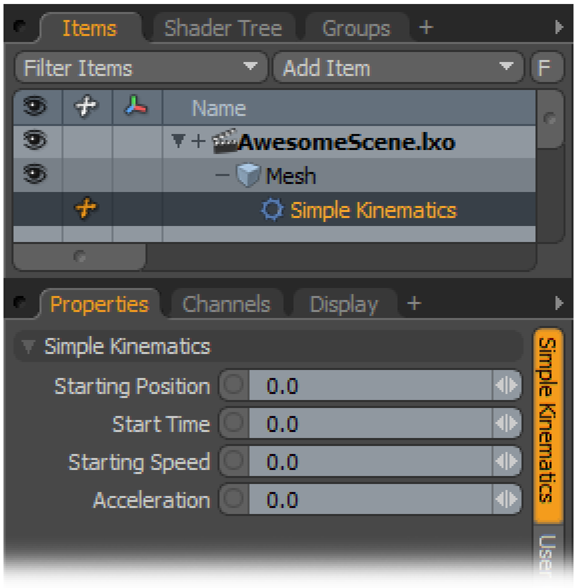
Reset Cache: This modifier uses a cache to store the total distance traveled. You may have to reset this cache periodically when working in the scene — especially after a lot of scrubbing or jumping to different points in time. Click the **Reset Cache** button found in the **Properties** sub-tab for the modifier.

Network Rendering: Scenes using this modifier require that the output be baked and the modifier removed (or its outputs unlinked) to render over a network. This is not necessary when rendering locally if the frames being rendered are in sequence. It is best to reset the cache on each modifier before baking or rendering locally. Modo does not store the caches in the scene file, so they get reset automatically when you reload the scene.

Dependency Loops: To avoid a dependency loop, do not use the World Position and Rotation channels of the item being driven as inputs to this modifier. Create a locator at the same position as the driven item and parent the driven item to it. You can then use the World Position and Rotation channels of the locator as inputs for the modifier.

Simple Kinematics

The **Simple Kinematics** modifier applies a constant acceleration to a channel. Use this to produce a physically based trajectory for something flying or falling through the air and taking gravity into account. (This is much like a **Gravity** modifier but without bouncing.) To create a trajectory for a thrown ball, you apply the modifier to the ball, link the modifier's Output to the ball's Y-position channel, and set the proper options.



Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Simple Kinematics
Properties
Channels
Display
Simple Kinematics
User

Simple Kinematics

Starting Position 0.0

Start Time 0.0

Starting Speed 0.0

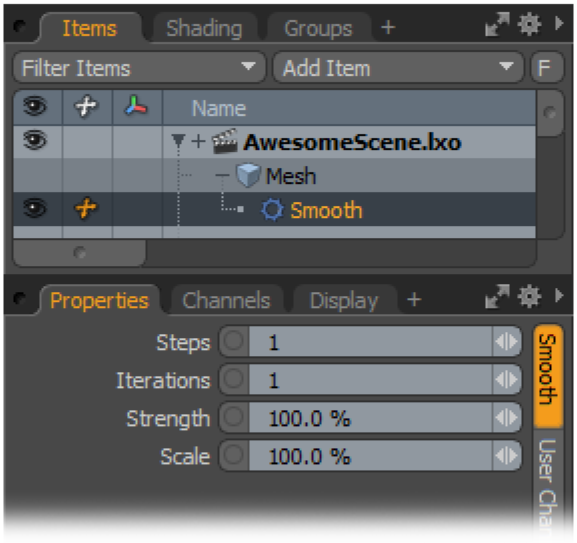
Acceleration 0.0

For example, if you set the **Starting Position** to the Y value that the ball should start moving from, leave the **Starting Time** at **0.0**. For acceleration (meters per second), type **9.8** (for Earth). Set the **Starting Speed** to **22** (meters per second), which would be a 50 mph baseball throw. For something more interesting than a vertical throw, add a constant velocity along the ground by either setting a linear enveloped motion on the X channel or by adding another **Simple Kinematics** modifier with a constant speed and no acceleration on the X channel.

Other uses for this type of constant velocity/constant acceleration include buoyant objects accelerating upwards, possibly cars and vehicles, and any other process where the resulting channel shape should be linear or parabolic.

Smooth Modifier

Smooth



Steps 1
Iterations 1
Strength 100.0 %
Scale 100.0 %

This modifier takes an input value and smooths it by reading the value of the surrounding frames and averaging the values.

Steps: Determines how many frames surrounding the current frame that Modo reads to evaluate the smoothness. For example, a value of **1** means that Modo reads one frame on either side of the current frame. Modo adds the three frame values together and divides that value by 3 to determine the smoothed value for the current frame.

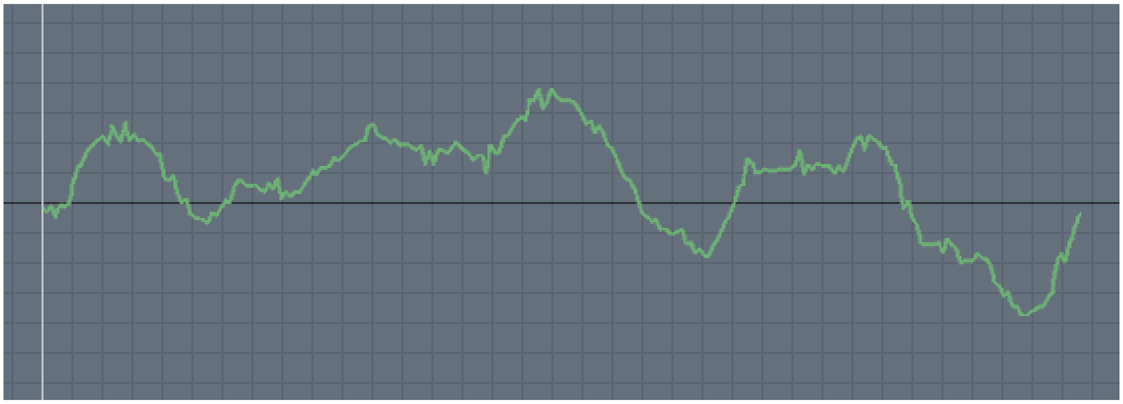
Iterations: Defines how many passes Modo makes for the smoothing function. If you set the **Steps** channel to **1** and the **Iterations** channel to **2**, then Modo evaluates and smooths the surrounding frames of the current frame and then Modo uses the resulting smoothed value to calculate the smoothing for the current frame. Increasing the iterations count, while keeping the number of steps low, results in a smooth curve with minimal loss of data.

Strength: Specifies a blending value for the smoothing function. Use this mainly for animating purposes. Animating based on the integer **Step** and **Iteration** channels only results in a stepped animation.

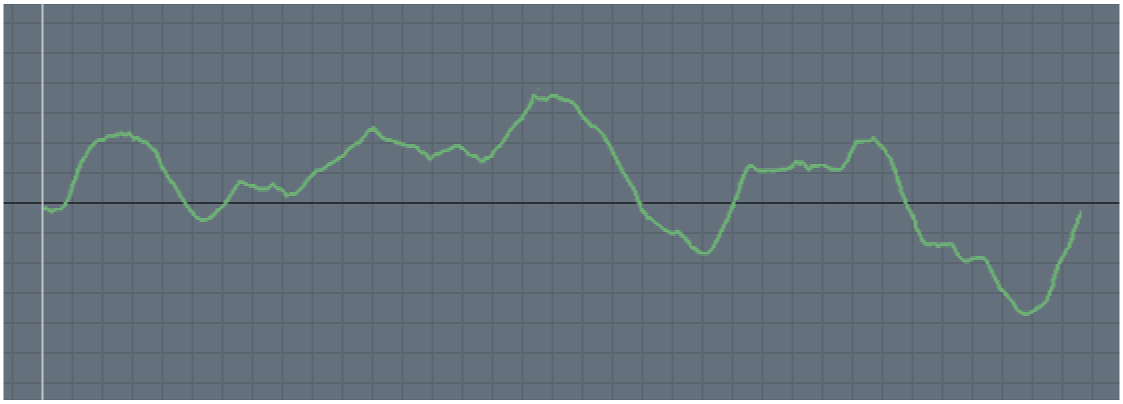
Scale: Multiples the smoothed value. Use this to maintain high peak values if you have Modo overly smooth the Input channel.

Output: Provides the resulting values from the smoothing operation.

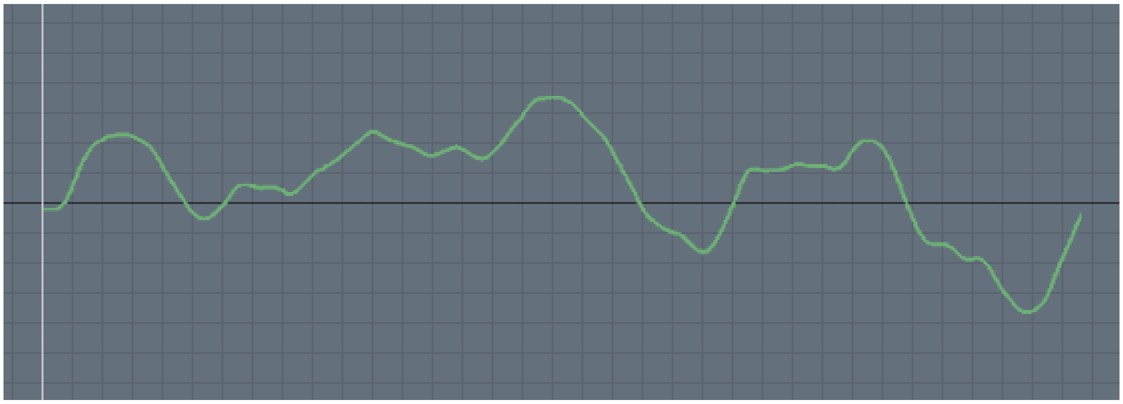
Smooth Channel Modifier Examples



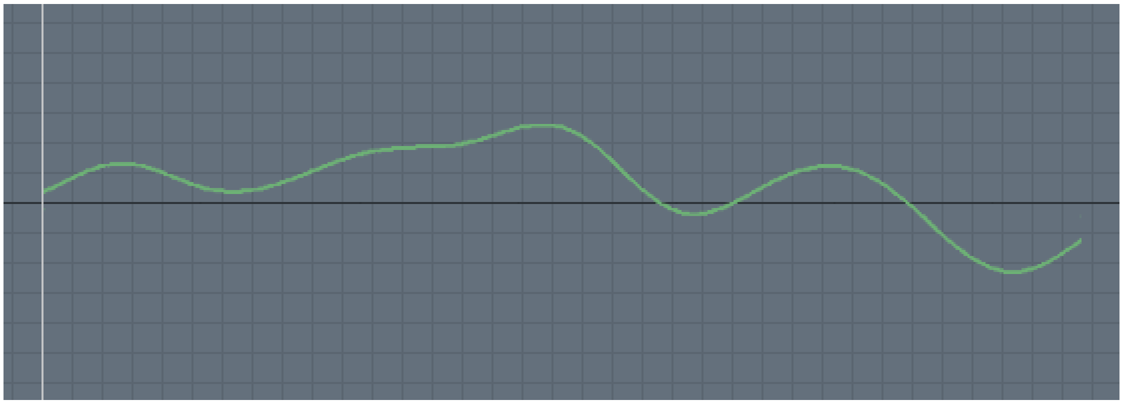
An original, unsmoothed animation curve



The sample curve with the **Smooth** operator using **Steps=1** and **Iterations=1**

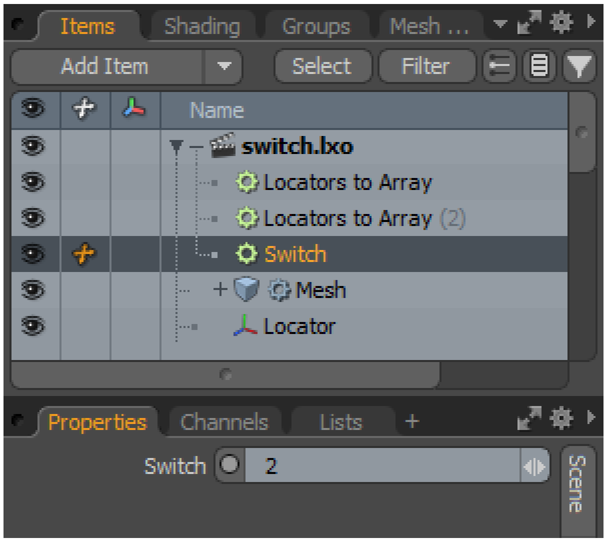


The sample curve with the **Smooth** operator using **Steps=1** and **Iterations=3**



The sample curve with the **Smooth** operator using **Steps=10** and **Iterations=3**

Switch Modifier


Items
Shading
Groups
Mesh ...
Add Item
Select
Filter
Name
switch.lxo
Locators to Array
Locators to Array (2)
Switch
Mesh
Locator
Properties
Channels
Lists
+
Scene

Switch 2

The Switch channel modifier works similarly to the String Switch and Numeric Switch Modifier. You can connect multiple inputs to a single connection, and use an index channel to specify which of the inputs is written to the output. However, unlike the String and Numeric Switch modifiers, Switch can operate on any value type.

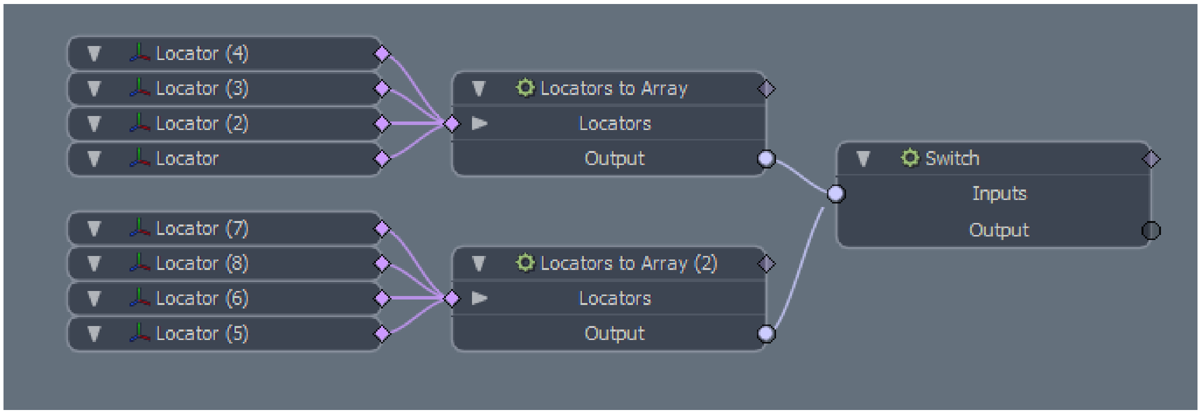
If the types of the objects connected to the Inputs and Output are different, they are automatically converted to the Output type, when possible. For example, if your inputs contain Boolean, Matrix, and Float values, but the output is a string, the input values are all converted to strings.

Using Switch

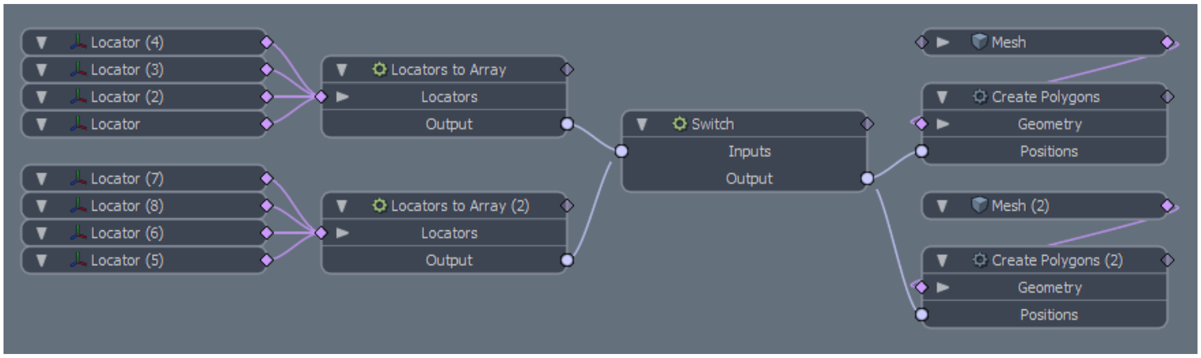
In the example below, the Switch modifier takes two arrays and creates a polygon using one of them, depending on which array's index is selected in the switch.

1. In the Schematic layout, click **Add...**, then navigate to **Channel Modifiers** > **Other**, and double-click **Switch**.

   This adds the Switch node to the Schematic.
2. Connect the inputs to the node's **Inputs** channel. In this case, we use two arrays generated from locators using the Array Modifiers modifier.

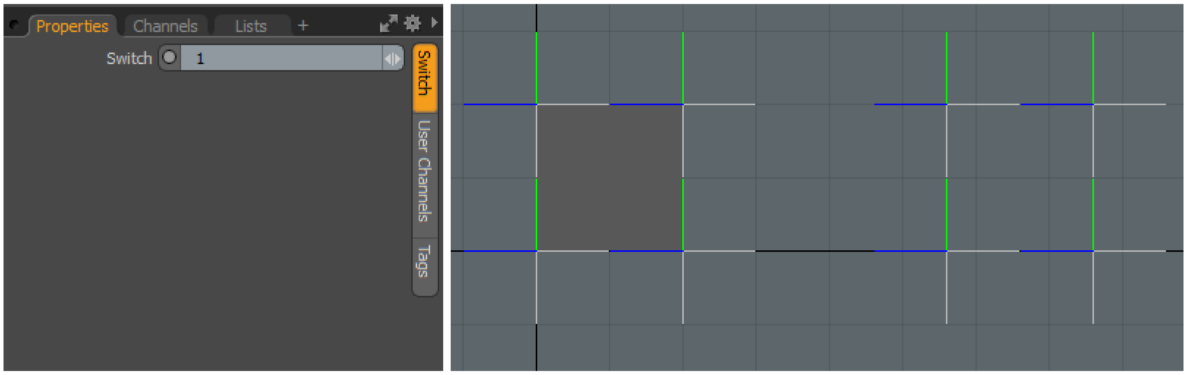

Locator (4)
Locator (3)
Locator (2)
Locator
Locators to Array
Locators
Output
Switch
Inputs
Output
Locator (7)
Locator (8)
Locator (6)
Locator (5)
Locators to Array (2)
Locators
Output

3. Connect the output nodes to the **Switch** node's **Output** channel. Here we have two Create Polygon operations, that generate polygons based on array positions.

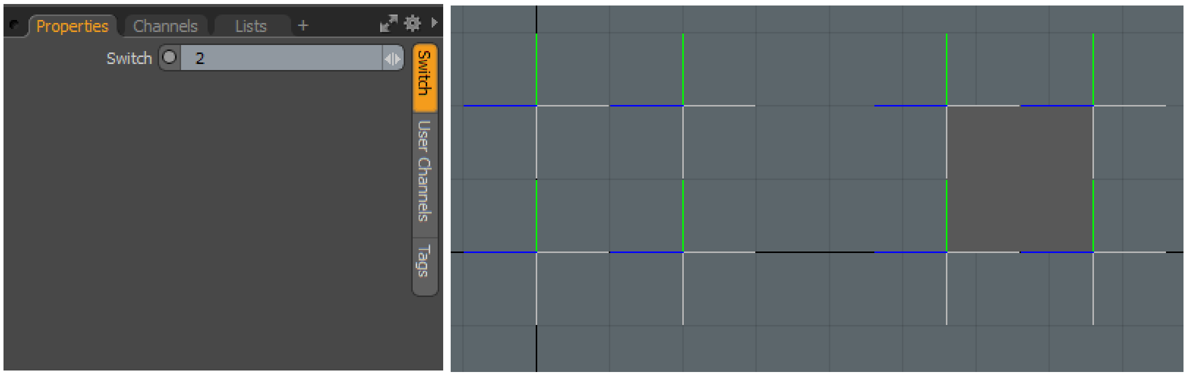

Locator (4)
Locator (3)
Locator (2)
Locator
Locators to Array
Locators
Output
Locator (7)
Locator (8)
Locator (6)
Locator (5)
Locators to Array (2)
Locators
Output
Switch
Inputs
Output
Mesh
Create Polygons
Geometry
Positions
Mesh (2)
Create Polygons (2)
Geometry
Positions

4. Click the **Switch** node in the Schematic, and in the **Properties** panel on the right, set the **Switch** property to the index of the output you want to see. In this case, setting **Switch** to **1** draws a polygon using the first set of locators, while choosing **2** draws a polygon using the second set of locators.

Note: Index assignment depends on the order you connected the nodes to the Switch node's **Inputs** channel.


Properties
Channels
Lists
+
Switch
User Channels
Tags

Switch 1


Properties
Channels
Lists
+
Switch
User Channels
Tags

Switch 2

Using Setup Tools

To help you when setting up elements for rigging and animation, Modo provides a number of tools and functions to do numerous tasks by positioning elements and by defining hierarchies. Click a tool's icon below to see details.

Hierarchy

Parent

Unparent

Insert

Insert at Parent

Extract

Match and Align

Item Position

Center Position

Pivot Position

Item Rotation

Center Rotation

Pivot Rotation

Match Scale

Match

Align

Set

To Workplane

Position

Rotation

Center to Pivot

Mirror Hierarchy

Mirror Position

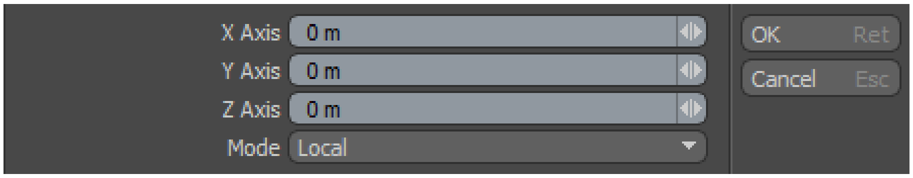
Set

- **To Work Plane Position**: This command is a powerful function for you to position and align Items, Centers, or Pivots relative to the current Work Plane. You can use the Work Plane, as a dynamic function by fixing it to a specific location after selecting a component and then applying the **Align Work Plane** command. Doing this rotates the entire Modo universe to align with the selected element. With the Work Plane's origin now centered on the selection, you can align objects to specific positions.

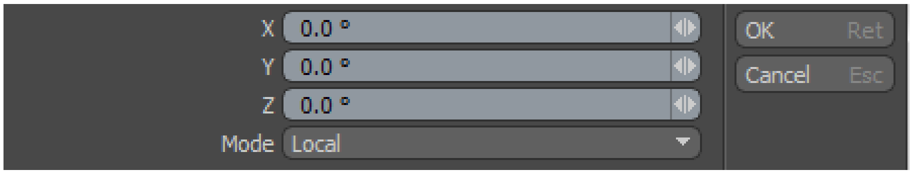
To use this command:

1. Select one or more geometric components to align to, such as vertices, edges, or polygons.
2. Apply the **Align Work Plane to Selection** command found in theUsing the Work Plane menu.
   This fixes the Work Plane in place, centered around the current selection, and aligns it to match the normal direction of the selected elements.
3. Next, select the item to move.
   This can be the item itself or just the item's Center or Pivot in its respective selection mode.
4. Next, choose the **Set to Work Plane** command to move the selected element to the center (0,0,0) position of the current Work Plane.
5. To reset the Work Plane to its normal operation, use the **Reset Work Plane** command or press the **End** key.

- **To Work Plane Rotation**: This command works in the same manner as the **Set to Work Plane Position** command, but this command only affects the rotation of the selected item, its Center, or its Pivot. With this command, you can match the rotation of an element without changing its position.
- **Position**: Use this pop-up requestor to numerically set the position of an item, its Pivot, or its Center in 3D space. Use the **X**, **Y**, and **Z** values to define a specific location. Use the **Mode** option to determine if the transform is a local (offset from the item's current position) or a world (a specific position within Modo's universe) transform.


X Axis 0 m
Y Axis 0 m
Z Axis 0 m
Mode Local

- **Rotation**: Use this pop-up requestor to numerically set the rotation of an item, its Pivot, or its Center in 3D space. Use the **X**, **Y**, and **Z** values to define a specific rotation. Use the **Mode** option to determine if the transform is a local (offset from the item's current position) or a world (a specific position within Modo's universe) transform.

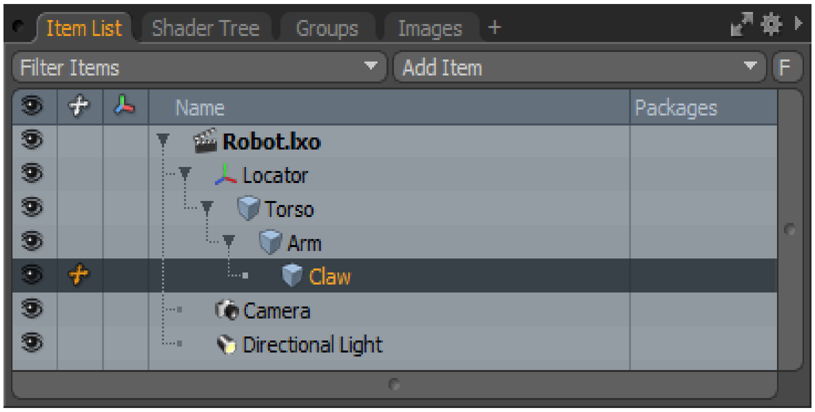

X 0.0 °
Y 0.0 °
Z 0.0 °
Mode Local

- **Center to Pivot**: All of Modo's animation transform values, by default, originate from the Center position. If you set up your model to use pivots instead, use **Center to Pivot** to match the Center position to the Pivot position. Select the item in question (in **Items mode**) and click the **Center to Pivot** button.
- **Mirror Hierarchy**: Use these commands for each axis (**X**, **Y**, or **Z**) to mirror an entire hierarchy of items in a single click. Prior to mirroring, select the target element to mirror. If you select the root element, Modo clones the entire hierarchy and mirrors its position across the desired axis. If you select a target further down the chain, Modo mirrors only the selected element and its children. They remain parented to the same parent. This button is in the **Setup** interface layout in the toolbox under the **Setup** sub-tab. If you click the button, Modo immediately mirrors the elements as directed. The button defaults to the **X** axis, but if you right-click the button, you can specify an alternate axis.
- **Mirror**: This command reflects the selected items over the specified axis (either **X**, **Y**, or **Z**) to produce mirrored items.

Hierarchy Tools

Hierarchy is a series of ordered groupings. In 3D animation, these hierarchies are known as Parenting or Parent/Child relationships. Modo uses hierarchies to define how multi-tiered objects relate to each other. Without these hierarchies, it would too cumbersome to animate anything beyond simple, single-item transforms.

Hierarchies are essential to animation because they define the order of inheritance for transformations in a chain of items. For example, if you move your shoulder, your arm follows along, as does the wrist and hand. In this case, the shoulder bending passes its transforms to the elbow, which can also bend, and both of those transforms pass on to the hand. Modo visualizes these hierarchical chains as cascading items within the **Item List** tab.



The **Item List** tab indents each subsequent item in the list. The top locator item is the parent item, the next its child, which is followed by siblings in lower orders. The ordering of items in a hierarchy defines how the transforms relate to items further down the chain. To learn more about the process for creating hierarchies in the **Item List** tab, see Item List. The **Animation** tab setup provides some additional tools for creating and modifying hierarchies.

Compensation

Compensation: When parenting items together that have prior transforms applied, you might find your items popping out of position. When this happens, the child is inheriting the transforms of the parent and sending it to a position you don't want. Enabling the **Compensate** button, in the top of the **Setup** layout next to the mode buttons, changes the **Parent/Unparent** commands to **Parent in Place/Unparent in Place**. When changed, Modo ignores prior parental transforms, and items retain their initial positions in relation to their parent.

Parent/Unparent: When in item mode, these commands produce the same results as the **Item List** but work with selections in the 3D viewport. When using this function, the selection order sets the relationship properly. The last item selected becomes the parent and everything else becomes the child of the parent. You can't create an entire chain by selecting multiple items. To do that, you need to apply the **Parent** command multiple times.

To use the **Parent** command:

1. First make sure you are in Item mode.
2. Select an item in the 3D viewport. (This is the child item.)
3. Press the **Shift** key and select the second item.
4. With both items selected, click the **Parent** button.

The first item selected has become the child of the second item. The **Unparent** button works the same way. Select any item or items, and then click the **Unparent** button to remove those items from the hierarchy. When setting up hierarchies, any item can be parented to any other item, cameras, lights, and locators.

Insert: This command is useful for adding items into an existing hierarchy without breaking the existing chain. It adds the first selection as a child of the second selection. The new child inherits the new parent's transforms and siblings.

Insert at Parent: This command reverses the **Insert** command and inserts an item as a parent instead of as a child. The item still inherits transforms and siblings. Use this to insert an item into the top of a hierarchy.

Extract: This command removes an item from within a hierarchy without disrupting the rest of the chain. This command automatically sets the children of the extracted item to its original parent, once the item is removed.

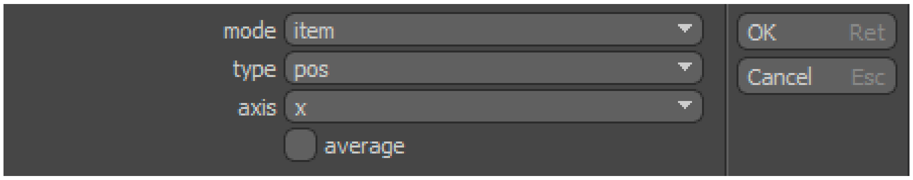
Match and Align

With the Match and Align functions you can position objects in relation to each other in 3D space for animations or for layout purposes. With most of these functions, you make a selection and then click the appropriate button to get the desired result. Also, Modo provides Match and Align dialog panels where you can specify options for more control.

- **Item Position**: This command matches the World Position of one item to that of a second item based on their center points. This button is in **Setup** sub-tab of the default toolbox in the Setup interface layout. Select the desired elements in **Items** mode prior to invoking the command. Because the selection order matters, be careful when you select the items: Modo moves the first item selected to match the position of the second item selected (the target). Modo ignores subsequent selections beyond the first two and also ignores the **Scale** and **Rotation** values of the selected items.
- **Center Position**: This command matches the World Position of one item's Center to that of another item's Center without affecting the geometry location. This button is in the **Setup** sub-tab of the default toolbox in the Setup interface layout. It may be under the **Item Position** button. If so, right-click to open the pop-up menu. Select the desired elements in **Items** mode prior to invoking the command. Because the selection order matters, be careful when you select the items: Modo moves the Center of the first item selected to match the Center of the second item selected (the target). Modo ignores subsequent selections beyond the first two and also ignores the **Scale** and **Rotation** values of the selected items.
- **Pivot Position**: This command matches the World Position of one item's Pivot to that of another item's Pivot without affecting the geometry location. This button is in the **Setup** sub-tab of the default toolbox in the Setup interface layout. (It may be under the **Item Position** button. If so, right-click to open the pop-up menu.) Select the desired elements in **Items** mode prior to invoking the command. Because the selection order matters, be careful when you select the items: Modo moves the first selected item's Pivot position to match the Pivot position of the second item selected (the target). Modo ignores subsequent selections beyond the first two and also ignores the **Scale** and **Rotation** values of the selected items.
- **Item Rotation**: This command matches the world orientation of one item to that of a second item based on their Center points. This button is in the **Setup** sub-tab of the default toolbox in the Setup interface layout. Select the desired elements in **Items** mode prior to invoking the command. Because the selection

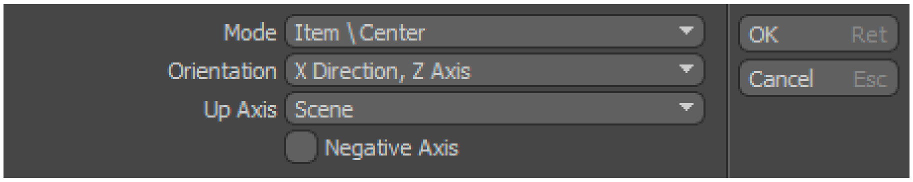
order matters, be careful when you select the items: Modo rotates the item selected first to match the rotation of the second item selected (the target). Modo ignores subsequent selections beyond the first two and also ignores the **Scale** and **Position** values of the selected items.

- **Center Rotation**: This command matches the world orientation of one item's Center to that of another item's Center without affecting the geometry, orientation, or position. This button is in the **Setup** sub-tab of the default toolbox in the Setup interface layout. It may be under the **Item Rotation** button If so, right-click to open the pop-up menu. Select the desired elements in **Items** mode prior to invoking the command. Because the selection order matters, be careful when you select the items: Modo rotates the Center of the item selected first to match the rotation of Center of the second item selected (the target). Modo ignores subsequent selections beyond the first two and also ignores the **Scale** and **Position** values of the selected items.
- **Pivot Rotation**: This command matches the world orientation of one item's Pivot to that of another item's Pivot without affecting the geometrical orientation or position. This button is in the **Setup** sub-tab of the default toolbox in the Setup interface layout. It may be located under the **Item Rotation** button. If so, right-click to open the popp menu. Select the desired elements in **Items** mode prior to invoking the command. Because the selection order matters, be careful when you select the items: Modo rotates the Pivot of the item selected first to match the rotation of Pivot of the second item selected (the target). Modo ignores subsequent selections beyond the first two and also ignores the **Scale** and **Position** values of the selected items.
- **Match Scale**: This command matches the world scale of one item to that of a second item based on their Center points. This button is in the **Setup** sub-tab of the default toolbox in the Setup interface layout. Select the desired elements in **Items** mode prior to invoking the command. Because the selection order matters, be careful when you select the items: Modo scales the item selected first to match the scale of the second item selected (the target). Modo ignores subsequent selections beyond the first two and also ignores the **Position** and **Rotation** values of the selected items.
- **Match** : With this command, you can select precisely how Modo matches two elements. This button is in the **Setup** sub-tab of the default toolbox in the Setup interface layout. Select the desired elements in **Items** mode prior to invoking the command. Because the selection order matters, be careful when you select the items: Modo acts upon the item selected first to match the second item selected (the target). Once you select the items, clicking the **Match** command opens the dialog requester.


mode item
type pos
axis x
☐ average

- **Mode**: This setting determines what is being affected: **Item** modifies the item itself, **Center** modifies only the item's Center point, and **Pivot** modifies only the item's Pivot point.
- **Type**: This setting specifies which value Modo matches from one item to another. You can specify **Position**, **Rotation**, or **Scale**.

- **Axis**: This setting specifies which axis Modo matches from one item to another. You can specify **X**, **Y**, **Z**, or **All**.
- **Average**: If you enable **Average**, Modo averages the values of both items selected when applying final values.
- **Align**: When setting up item hierarchies for Inverse Kinematics, a parent's Center position must point toward its child's Center position for proper results. If the items in the hierarchy were created off-axis (rotated away from any of the plane's axes), the **Align** command helps to orient an item's Center position so that it aligns directly toward its child's Center. You can also use **Align** for other general alignments tasks (such as fixing improperly oriented joints in a skeleton). You can find this in the **Setup** sub-tab of the default toolbox of the Setup interface layout. To use, select the two target item layers in the hierarchy to affect and then choose the command. The target items must have a direct parent-child relationship defined. Selecting the command opens the following dialog box.



Mode: Item \ Center
Orientation: X Direction, Z Axis
Up Axis: Scene
☐ Negative Axis

- **Mode**: Specify to modify either the parent item's **Center** or **Pivot** rotations.
- **Orientation**: Select the orientation of the item when rotated. **Direction** defines the axis that points toward the child, **Axis** defines the actual orientation.
- **Up Axis**: Specify to use the **Scene** or one of the three world axes for rotation.
- **Negative Axis**: When enabled, orients the **Up Axis** in the negative direction.

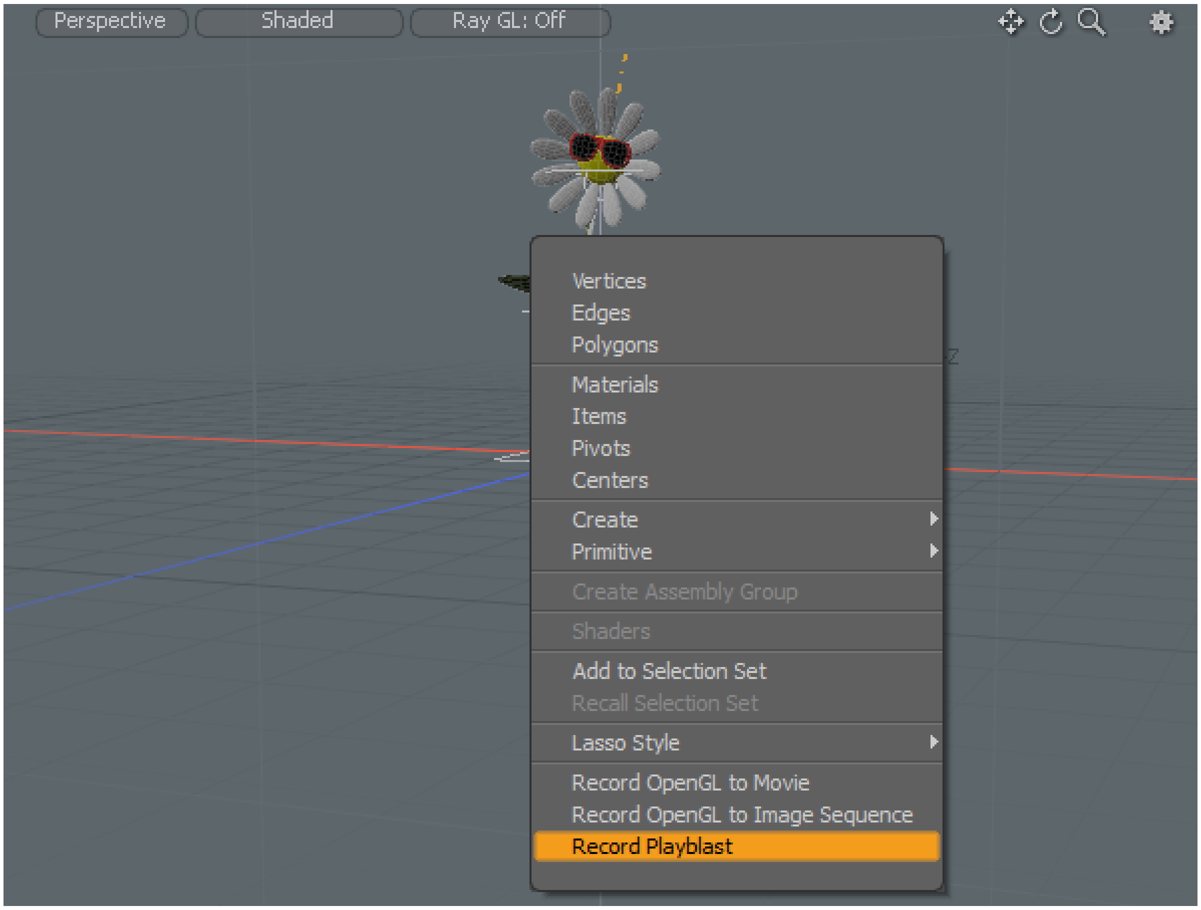
Playblast

The Playblast tool allows you to get a quick preview of your animation, providing you with a realistic idea of your final render result without requiring the time needed for a formal render. Use this tool to quickly check for character and prop positioning and lighting placement. Curved surfaces are rendered as surfaces with a low face count (faceted).

Use this tool to:

- Open a dedicated viewport
- Use the scene frame size as the base resolution
- Set the Resolution Multiplier value to set different viewport sizes for output
- Set video **Video Format**, **Codec**, and **Quality** values
- Select any camera defined in the scene

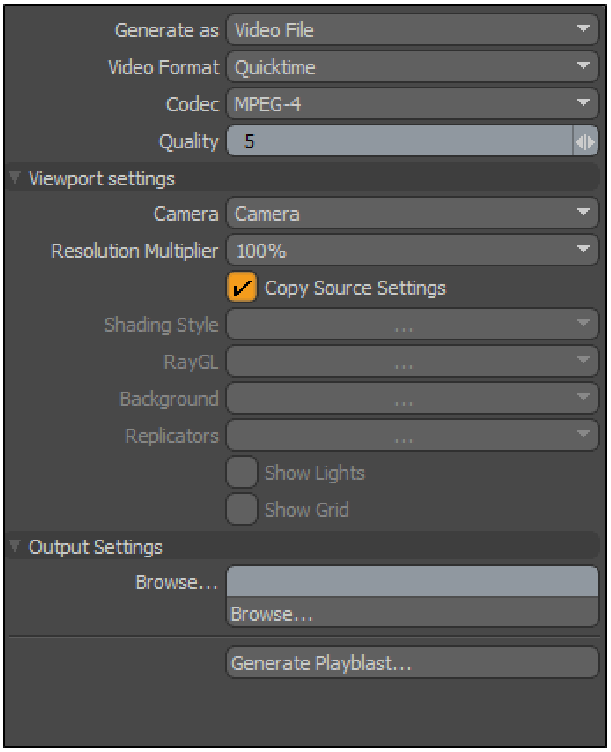
The Playblast tool is available from the 3D Viewport contextual menu. Right-click in the 3D Viewport to view the contextual menu and select **Record Playblast**.


Perspective
Shaded
Ray GL: Off
Vertices
Edges
Polygons
Materials
Items
Pivots
Centers
Create
Primitive
Create Assembly Group
Shaders
Add to Selection Set
Recall Selection Set
Lasso Style
Record OpenGL to Movie
Record OpenGL to Image Sequence
Record Playblast

To create a Playblast, enter the desired Playblast Settings and click **Generate Playblast**. A new viewport displays, using the scene's render size, resolution, and other specified settings.

Playblast Settings

The following Playback **Video File** settings are available:



Generate as Video File
Video Format Quicktime
Codec MPEG-4
Quality 5
Camera Camera
Resolution Multiplier 100%
☑ Copy Source Settings

Viewport settings
Shading Style
RayGL
Background
Replicators
Show Lights
Show Grid
Output Settings
Browse...
Browse...
Generate Playblast...

Option	Description
Generate as	Selects the type of output. The following options are available: • **Video Format** - Stores digital video data on your computer system. • **Image Sequence** - Stores a collection of **.tga** images related by time, such as frames in a movie. For more information about Image Sequence properties, see Image Sequence. **Note: Video Format**, **Codec**, and **Quality** options are not available when **Image Sequence** is selected.
Video Format	Selects the video format. The following options are available: • **QuickTime** • **AVI** • **Mpeg4** • **Flash Video**
Codec	Selects the encoding and decoding digital data stream method. The following options are available:

Option	Description
	• **Mpeg4** • **Microsoft MPEG-4** • **Microsoft video 1** • **Sorensen Video** • **MJPEG** • **Animation** **Note:** Available options vary depending on the **Video Format** option selected.
Quality	Sets the quality value for your output. The default value is 5.
Viewport Settings	
Camera	Sets the camera in your scene that you would like to use in the output.
Resolution Multiplier	Sets the resolution multiplier value for your output. The following options are available: • **25%** • **50%** • **100%**
Copy Source Settings	When enabled, copies the settings from the 3D Viewport. The remaining settings in the **Playblast Settings** dialog override those copied from the source. This setting is enabled by default.
Shading Style	When Copy Source Settings is disabled, the **Shading Style** sets to Shaded by default. The following options are available: • **Wireframe** • **Solid** • **Vertex Map** • **Texture** • **Shaded Texture** • **Default** • **Advanced**

Option	Description
	• **Gooch Tone** • **Cell Shading** • **Reflection** For more information, see 3D Viewport Styles.
RayGL	Enables the 3D viewport rendering Ray-GL, giving you a fast, fully rendered representation of the scene. Choose from the **Fast** or **Full** options to maximize workflow efficiency.
Background	Allows you to set the background to **Environment**. The following options are available: • **Environment** - Displays the settings as determined by the Environment item of the Shader Tree, simulating the rendered output. • **None** - Displays the default 3D Viewport background. This option is set to **None** by default.
Replicators	Enables the replicators in your scene at render time. Replicators control how duplicates of the item are copied onto the target point source. For more information, see Replicators.
Show Lights	When enabled, displays lights in the 3D viewport.
Show Grid	When enabled, displays the grid in the 3D viewport. For more information, see Grid and Workplane Settings.
Output Settings	
Browse	Allows you to specify where to save your Playblast file to.
Generate Playblast	Generates your Playblast output. A new viewport automatically displays the output.

Creating Higher Quality Playblasts with AVP

If your viewport is set to advanced, you can set your playblast to wait for iterations to finish before moving to the next frame.

Enabling Progressive Rendering

To enable progressive rendering in your playblast:

1. Open the **3D Viewport Properties window**

    Alternatively, press O to open the **3D Viewport Properties window**

2. Navigate to the **Advanced Options** tab, and turn **Progressive DOF Mode** on
3. Set **Progressive** to the number of times you want your scene to be redrawn

    The higher the number, the more anti-aliasing that occurs, but this will increase render time

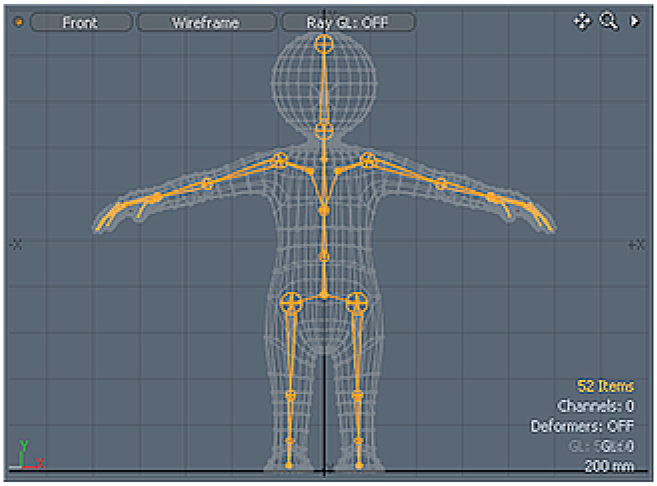
4. Once AVP has been set up in the **3D Viewport Properties Window**, open the **Playblast Settings** window and enable **Use Camera Override**

    Right-click within the 3D viewport and select **Record Playblast** to open the **Playblast Settings** window.

Once set up, you can continue to adjust your playblast settings as you would normally.

Skeleton Tool

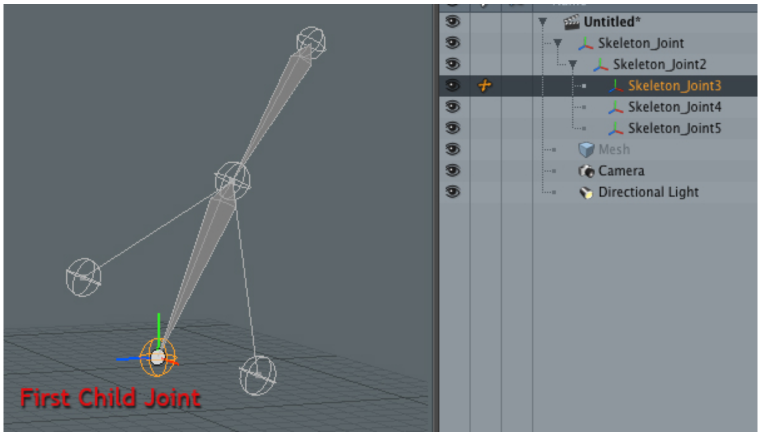
You use the Skeleton tool primarily to create a joint hierarchy for character posing and animations using deformations. Joints are locator items with custom drawing predefined: a wireframe sphere at the joint's location and a geometrical shape drawn to the location of the joint's first child. You can change both shapes in the **Display** viewport if desired. Once you create the skeleton, you must still bind the joint hierarchy to the mesh using the **Bind** command while in the **Setup** action state. For more information, see the Binding and Setup Mode topic.



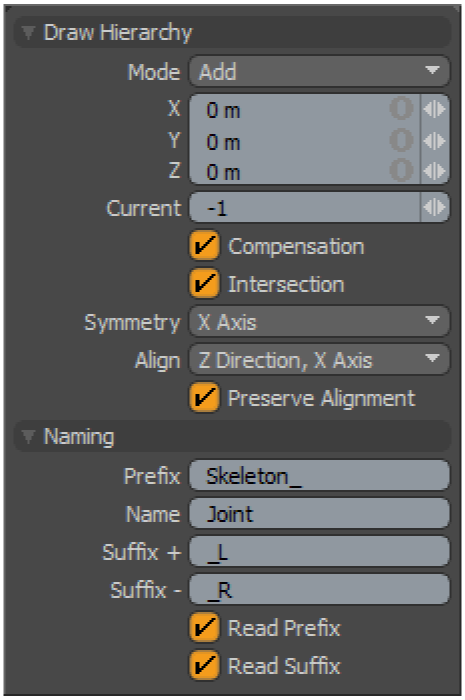
Drawing the Skeleton

Before drawing the skeleton, think about the position of the root element. This is the base joint for the entire chain, and every other joint is a parent/child cascading hierarchy from this joint. For bipedal human characters, the root element is at the hips with the legs and spine branching outward. To set the root, click in the viewport over the target geometry. (You do not need to drag out the joint.)

By default, Modo enables the **Intersection** option, and, therefore, Modo attempts to find the halfway position between the forward- and backward-facing polygons intersecting with the pointer's position, and places the joint segment appropriately. This saves time when placing joints in hands and feet. If there are no back-facing polygons or if **Intersection** is disabled, then Modo creates the joint at the intersection of the pointer when clicked and the geometry's surface. Clicking away from the model creates the joint at the intersection of the pointer and the current Work Plane. After setting the initial point that represents the base of the joint (its rotation's origin), you can click a second time to set the actual joint. By dragging you can interactively scale and position that joint. Subsequent clicks create additional joints in the chain with each being a child of its predecessor. Pressing **Shift** when creating a joint starts a new chain. To add a branch to an existing chain, click a joint to select it and then continue to click to draw the new branch. You can modify the position of joints that have been drawn already by dragging their bases or by entering the Edit action state.



The first child of a joint defines the alignment of its parent joint and is used by other item alignment commands. To make the first joint easier to identify, Modo only draws the geometrical shape to the first child. Links to other child joints are lines. You change which joint is first in the **Item List** by dragging the joint directly under its parent.



Draw Hierarchy

Mode: Add

X: 0 m

Y: 0 m

Z: 0 m

Current: -1

☑ Compensation

☑ Intersection

Symmetry: X Axis

Align: Z Direction, X Axis

☑ Preserve Alignment

Naming

Prefix: Skeleton_

Name: Joint

Suffix +: _L

Suffix -: _R

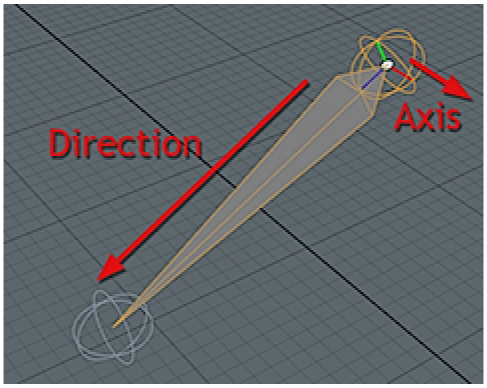
☑ Read Prefix

☑ Read Suffix

The following **Draw Hierarchy** options are available:

- **Mode** - Gives access to various action modes of the Skeleton tool. The specified mode determines what happens when you click a joint.
  - **Add** - Creates an additional joint as the child of the previous joint to produce a cascading hierarchy. This is the default state.
  - **Edit** - Gives the ability to modify the position of individual joints without accidentally creating a new joint.
  - **Insert** - Splits a joint into two. In this mode, clicking the base of any joint splits it into two separate joints.
  - **Delete** - Removes the joint from the hierarchy.
- **X/Y/Z**: Displays the joint's position in 3D World Space.
- **Current**: Displays the joint number (in sequence from the root) to determine the current joint for editing or the most recently created joint.
- **Compensation**: Changes the length of the child joint (when enabled) with the rest of the chain remaining in position. When disabled, the child joints move with their parents.
- **Intersection**: Toggles how Modo draws joints. Disabled by default, when drawing out joints, Modo draws the joint at the intersection of the pointer when clicked and Using the Work Plane. When enabled, Modo attempts to find the mid-point within a volume by determining the forward- and backward-facing polygons under the pointer and centering the joint between them.
- **Symmetry**: Creates mirrored joints across the specified axis automatically. Modo automatically names the duplicated joints according the **Naming** options. To disable symmetry, select **Off** instead of an axis.

- **Align**: Determines how Modo aligns the local axes of a joint toward its first child joint. The **Direction** specifies which axis points towards the child joint; the **Axis** specifies the axis running horizontally through the joint. These can be thought of as roll and pitch, respectively.



Direction: Z axis, Align default: X axis

- **Preserve Alignment**: Maintains (when enabled) the current alignment of joints. When disabled, Modo re-orients the joint to the alignment specified in the **Align** list. Re-orient a number of joints in a hierarchy by specifying the desired alignment and then clicking the target joint to change.

The following **Naming** options are available:

- **Prefix**: Determines the first part of the auto-naming string. Modo adds a number (based on the number of items in the scene with the identical name) at the end of the full name when you create more than one joint with an identical string. You can change the **Prefix** at any time during the joint creation process to denote different areas of the skeleton.
- **Name**: Determines the second part of the auto-naming string. Modo adds a number (based on the number of items in the scene with the identical name) at the end of the full name when you create more than one joint with an identical string. You can change the **Name** at any time during the joint creation process to denote different areas of the skeleton.
- **Suffix +**/**Suffix -**: If you enable the **Symmetry** option, Modo appends a **Suffix** value to the end of the string based on the joint's position on the X plane, relative to the global YZ plane as either + (positive) or - (negative).
- **Read Prefix**/**Read Suffix**: Updates the **Prefix** and **Suffix** boxes (when enabled) with text from the selected joint's current name. When disabled, Modo changes the appropriate part of the joint's name to the text in the **Prefix** and **Suffix** boxes when you click a joint.

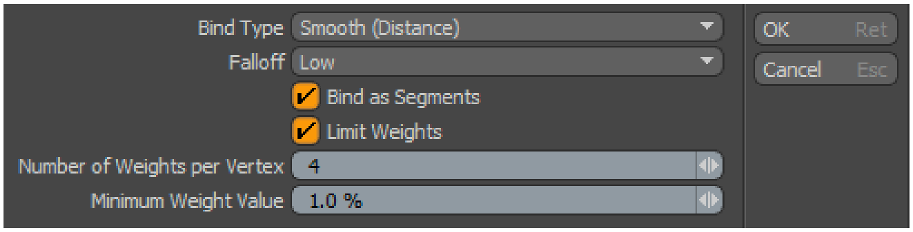
Binding and Setup Mode

Binding

Binding is the process of connecting a skeleton made with the Skeleton tool and attaching it to the target geometry so that it deforms when you manipulate the skeleton. There are a couple requirements that need to be met prior to running the command. Once the geometry is ready to be animated, create a reasonable skeleton with an appropriate hierarchy. Modo creates the hierarchy automatically when you draw the skeleton. See the Skeleton Tool topic for more information.

1. In **Items** mode, select the target Mesh Item(s) and the root of the skeleton's hierarchy.
2. Enter the **Setup** action state by clicking the **Setup** Setup button above the 3D viewport in the **Setup** interface tab.

   Modo indicates the Setup action state by a yellow outline around the viewport when that state is enabled.
3. From within the **Setup** action state, find the **Bind** tool command in the toolbox on the **Deformers** sub-tab. Clicking the button activates the **Bind** process and opens a dialog box with several options for binding the skeleton.



Bind Type: Smooth (Distance)
Falloff: Low
☑ Bind as Segments
☑ Limit Weights
Number of Weights per Vertex: 4
Minimum Weight Value: 1.0 %

Option	Description
Bind Type	By default, the binding process treats the hierarchy as individual segments between the items. Modo assigns weights to vertices along the line between an item and its child (similar to a bone). When an item has multiple children, Modo treats them as a single segment. • **Rigid** - This is the simplest of all binding methods. Modo assigns

Option	Description

	a vertex with a 100% weight to the closest segment in the hierarchy (or to the closest joint when **Bind as Segments** is disabled).
	• **Smooth (Distance)** - Produces a smooth, graduated transition between segments based on its distance from the segment.
	• **Smooth (Visible)** - Produces a smooth, graduated transition similar to the **Distance** option, but Modo only assigns weighting to vertices that are visible to the segment. This reduces or eliminates cross talk between neighboring segments (such as with legs or with fingers). However, it does require that you have the Skeleton contained entirely within the volume of the geometry.
	• **Heat** - This method takes into account the topology of the mesh and the relative positions of the joints. It often produces superior results to the other methods, but the target mesh must conform to a number of requirements:
	  • The mesh must form a volume although Modo does support holes (such as eye sockets).
	  • The target mesh must be polygonal. No single vertices, floating edges, or curves can be present.
	  • No shared vertices, edges, or polygons (non-manifold surfaces) are allowed between multiple components.
	  • All joints must be contained within the volume of the mesh.
	When applying **Heat** weighting with the **Bind** command, if Modo encounters any errors, it reports the error with a message. You can resolve most issues with the geometry by applying the Mesh Cleanup command. Modo indicates the illegal geometry when you exit the command to target the area for you to repair. Modo may also display a Solver Error. This occurs if there are multiple mesh volumes in the same Mesh Item layer, especially if they overlap.
	For example, if you have two primitive spheres in the same layer, they trigger this error. Often you can fix these solver errors by only weighting the main mesh volume (for example, the skin of a character) and placing other elements into another layer (such as

Option	Description
	the eyes, hair, and clothing) before weighting.
Falloff	Determines the attenuation amount between neighboring Weight Maps when using either of the the **Smooth Bind** types. Specifying **Low** produces a softer, more gradual transition between joint weights; specifying **High** produces a more abrupt transition; and specifying **Medium** is a transition between the two extremes.
Bind as Segments	When disabled, Modo calculates the weighting as a radius around each joint. When enabled, Modo calculates the weighting along the segment between a joint and its child.
Limit Weights	When enabled, you can limit the number of joints that influence a weight value during the Bind process. (This is useful for **Unity**, which can handle a maximum of four joints per vertex.)
Number of Weights per Vertex	Determines the maximum number of influencing joints when you enable **Limit Weights**. Modo passes this value on to the Normalizing folder as the **Max Sources** attribute where you can adjust it later.
Minimum Weight Value	Ignores any weight values below this threshold. This is useful when cleaning up **Heat Binding**, which tends to generate many small weight values.

4. After setting up the binding, click **OK**.

   Modo then initiates a series of actions automatically:

   - The first is the weighting of the skeleton to the geometry. Each joint has an associated Weight Map created based on the **Bind Type** and its settings.
   - Next, Modo assigns a series of deform influences to each joint of the skeleton and then generates and assigns an associated Weight Map to each.
   - Finally, Modo links each influence automatically to its originating joint and places the entire hierarchy of deformers into a Normalizing folder, where Modo evaluates all the influences as a single unified deformer.

5. After running the **Bind** command, exit **Setup** mode to begin to deform and animate the character.

   Further refinement is possible. For example, if you wish to fine-tune the automatically created weights, use the Weighting tools in the **Weighting** sub-tab of the relevant toolbox.

Setup

If you are unfamiliar with animation and rigging, you may not know why you need to enter a **Setup** mode to bind a skeleton to the geometry or to set up Inverse Kinematics (IK).

When the binding takes place, items in the **Setup** action state are understood by Modo to be in their resting or undeformed position. By binding in the **Setup** action state, you define the initial position of the individual joints and zero this position so all joints have the same starting value. The same occurs with IK because it defines the base position for segments along the chain. When you add a task or goal, the IK solver knows where to begin its calculations. By having a specific **Setup** mode, you have an easy way to get an animated or deformed model back into its resting position for further refinement without having to remove or disable any animation already applied to the geometry.

Tip: You can also use the **Setup** action state to define resting positions for Using Falloffs when used as deformers. Then Modo knows what areas of the mesh should be under the influence of the falloff.

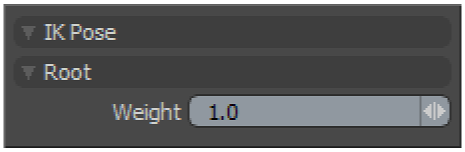
Pose Tool

The Pose tool allows you a simple means to pose and animate item hierarchies, manipulated by an intuitive *pin* and *drag* workflow. While primarily targeted at character skeletons, the Pose tool can be used to modify and animate any hierarchy of locator type items (this includes just about anything that appears in the Items (Scenes) List). The main benefit of the Pose tool outside of its simple to use operation is that it is not necessary to apply any IK to the hierarchy first; the Pose tools takes care of IK automatically, though any rig that does contains IK enhances the use of the tool.

Using the Pose Tool

The Pose tool is found in the 'Setup' and 'Animate' interface tabs in the upper left toolbox. Prior to activating the tool, make sure to select the root item of the target hierarchy. For more information on defining hierarchies, please reference that page of the documentation. Once selected, click the icon button to activate the tool. When active, small white circles draw in the 3D viewport at each items 'Center' point position representing its center of rotation. Hovering the mouse pointer over any of these circles highlights the element yellow. A click and drag moves the highlighted item. If the root element was clicked and dragged, the entire hierarchy moves rigidly, but if subsequent child items are moved, they move as if IK has been enabled, where one element drags along another element earlier in the chain. In order to keep an element from moving it can be pinned, hover the mouse pointer over a joint and right-click; now when any other elements are moved the pinned elements remain in place. Users can pin as many elements as required, and pins can be enabled and disabled freely while positioning the character. The Pose tools always uses the pins to localize the effect of dragging joints, working from the joint being dragging up the hierarchy chain until it finds a pinned joint or gets to the root. The tool also looks down the hierarchy for any pinned joints and tries to find the best fit it can for the skeleton based on the currently pinned joints and their initial orientations; some movement around the pins should be expected when dragging points.

Root—



Weight 1.0

Weight: This option determines how freely the root of the hierarchy moves. A 'Weight' of '0' keeps the root in place.

Posing Controls

- Click drag to move any joint in Screen (viewport) space.
- **Ctrl**+click and drag moves the joint in World space.
- Right-click a joint to pin/unpin, pinned joint displays as red circles.
- **Ctrl**+right-click to fix pin/unpin on subsequent keyframes/poses (requires IK assignment).
- Right-click on background to release all pins.
- **Shift**+click and drag to twist a joint.
- Middle-click drag for traditional FK (forward kinematics - only the dragged joint is affected)
- Click and drag on background to move the root item.

Posing IK Setups

The Pose tool works with any hierarchy of items, they don't need any special alignment or setup, but they can benefit from some hints on which way joints should rotate and on which axes. This is done by assigning full body IK to the hierarchy and setting the joint parameters, for example one can tell the Pose tool to only bend a knee joint on a single axis by setting that up in the IK properties for the knee. The Pose tool reads any IK settings from the joints, taking into account States, Axes, Weights and Limits. For more information on setting up IK, see Setting up a Full Body IK.

Simple Animation

The Pose tool can be used to provide a simple and quick means of animating characters with very little setup using Modo's auto keying function. When set to **Animated**, any modified channels automatically get a new keyframe. Before applying any animation, each item in the hierarchy need to have an initial keyframe. The root item needs a **Position** and **Rotation** keyframe, and all Children only need rotation keyframes. This should all be done on a default, base pose. A quick way to apply keys to many items is to double click the root item (selecting the entire chain) and click the **Key Rotation** button under the Timeline. Don't forget to set the position key for the root as well. Now you can move the Timeline around and adjust poses with the appropriate keyframes being set automatically for joints as the time is changed. Keyframes can also be applied to entire characters at once using the **Actor** and **Pose** options. For more information, see Creating Actors, Actions, and Poses.

Spacing Chart

The Spacing Chart aids animators in a workflow popularized by traditional hand-drawn animation by concentrating on the poses of the character and treating a keyframe position like a single drawing in an animated sequence. With the Spacing Chart, you can define specific key poses for a character before working out the timing. Once the poses are set, then you can modify the timing independently.

The following worflow is based on the approach used by Keith Lango (see References below) where the timing of the animation is done after the creation of all the key poses, demonstrating the use of the **Spacing Chart** interface. This shouldn't be considered the only way of working with this interface though, the animation can be timed out at any point, the use of breakdowns and inbetweens is just as valid whenever the timing is done.

Setup

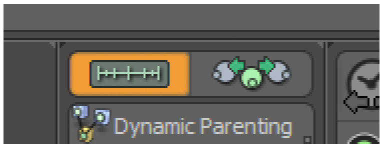
You need to set up a few things before starting:

1. Create an **Actor** for all the items being animated.
2. Create an **Action** for the animation.
3. Set the Action **Interpolation** to **Stepped** or **Linear** in the **Action** pop-up form.
4. Make sure that all the channels you intend to animated have a keyframe at the start frame.

For more information on creating actors and actions, see Creating Actors, Actions, and Poses.

Usage

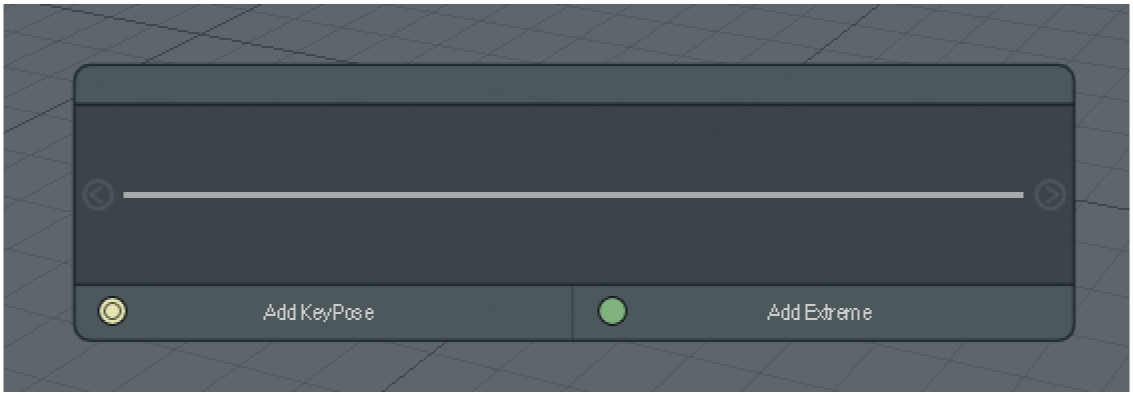
You open the **Spacing Chart** interface (HUD) by clicking the button at the top-left toolbar area in the **Animation** layout.



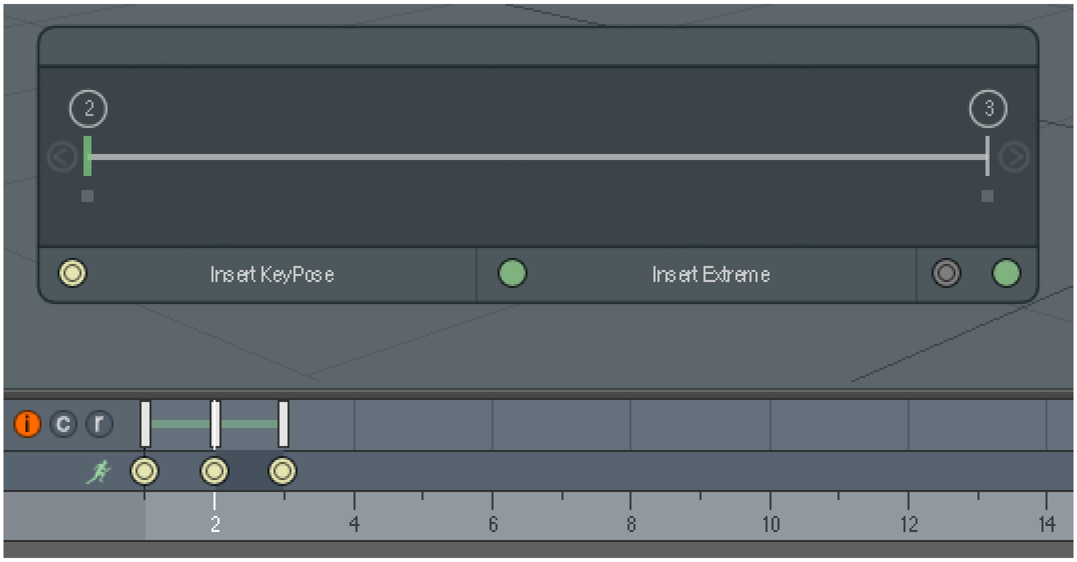
The Spacing Chart UI opens within the 3D viewport where you can reposition it by dragging the header bar area of the UI.

KeyPoses

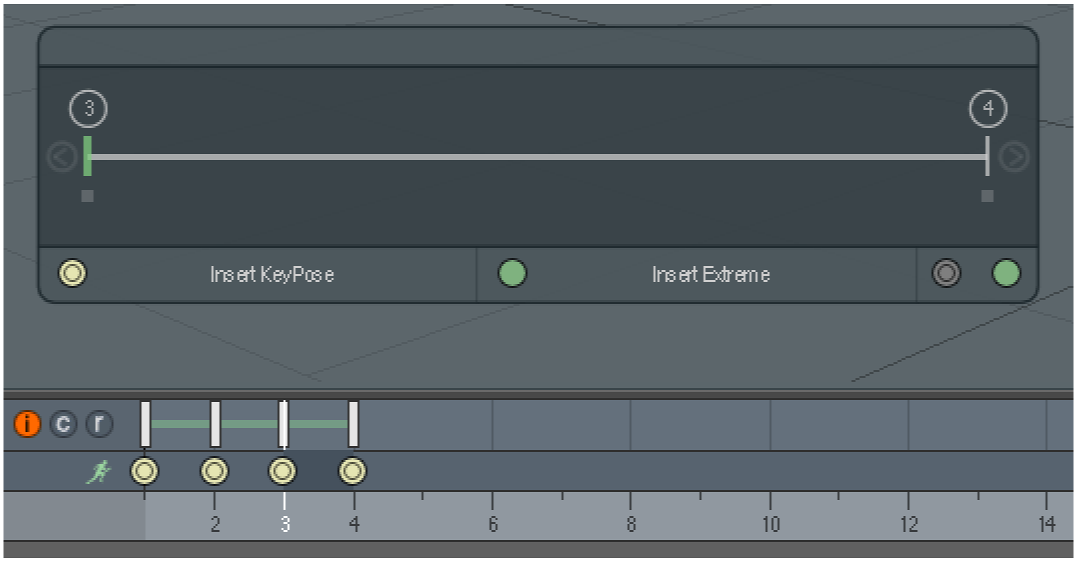
First, create the KeyPoses for the animation. These are also referred to as golden poses, story-telling drawings, or simply keys. These are the poses that tell the story in that they are the specific poses that have to be there to show what's happening in the animation — not necessarily how the character moves or changes unless it's important to the story.



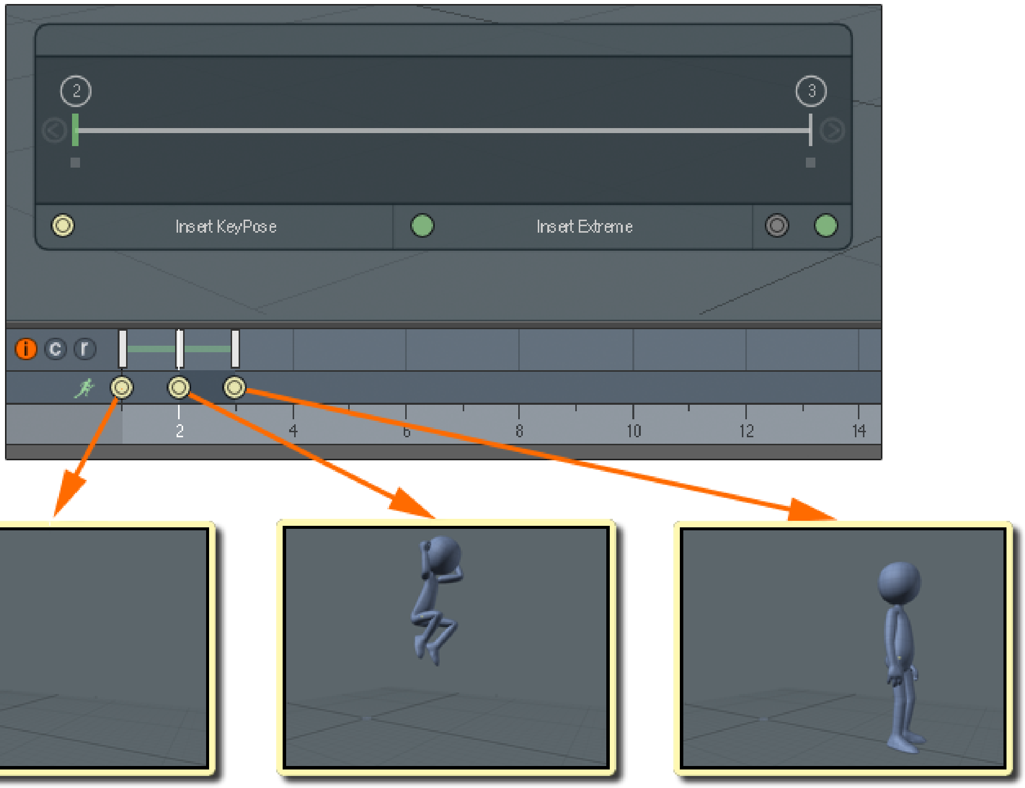
Clicking the **Add KeyPose** button in the HUD adds a KeyPose time marker to the active action and keyframes all the animated channels in the actor. For easy reference, Modo indicates a KeyPose in the Timeline as a yellow double-circle. Next, pose the character in its starting position. Once you create the first KeyPose, move to the next frame of the Timeline. Click **Add KeyPose** again and pose the character as needed for this frame. Repeat this process as many times as necessary until you have added all the story-telling poses. A good way to determine whether you have all the poses you need at this point is to slowly step through each pose. You should be able to tell what's happening in the animation from these poses.


2
3
Insert KeyPose
Insert Extreme
i
c
r
2
4
6
8
10
12
14

If you decide to insert an additional KeyPose to the Timeline, go to the frame (or pose) where you'd like to add it. After you've added a KeyPose, Modo shows the **Insert KeyPose** button. Click this button and Modo inserts a new frame **after** the current frame and moves all subsequent poses along to make room. Modo adds a KeyPose time marker, keyframes the action, and sets the current time to that of the new frame for the new pose.

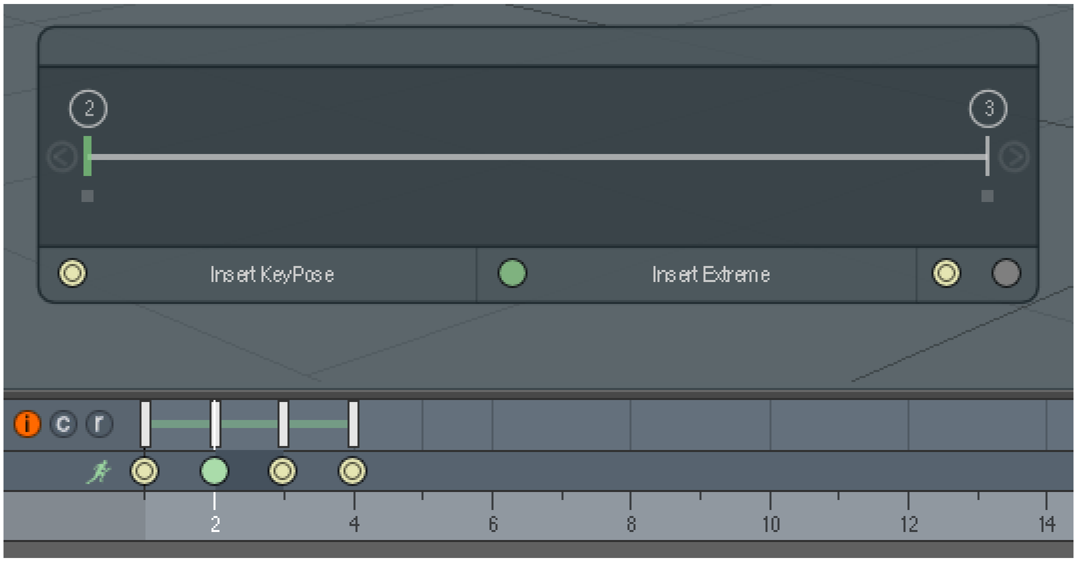

3
4
Insert KeyPose
Insert Extreme
i
c
r
2
3
4
6
8
10
12
14

A complete series of key poses for a jump might look something like this example.

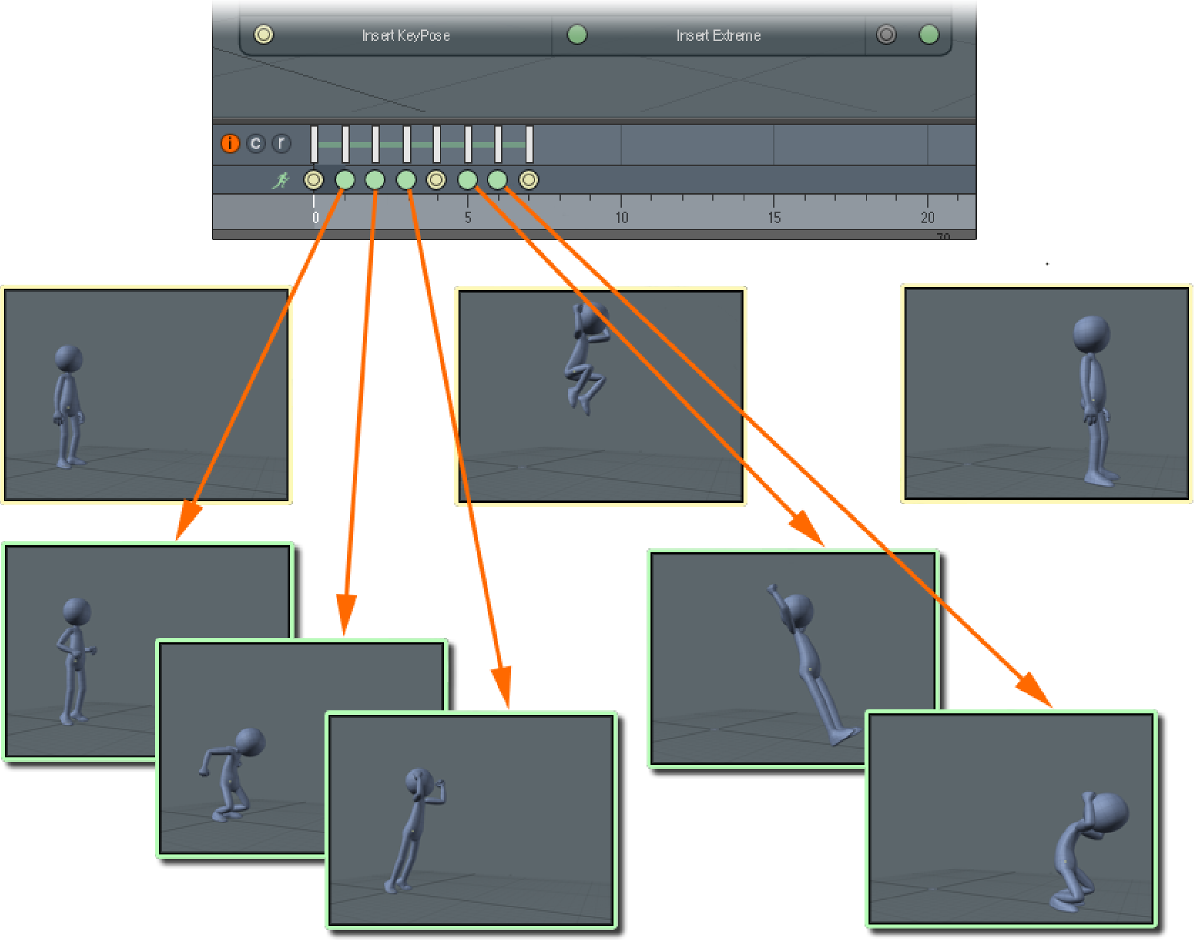

2
3
Insert KeyPose
Insert Extreme
i c r
2
4
6
8
10
12
14

Extremes

After you have added all the story-telling poses, you next add Extremes. These poses show how the character moves between the KeyPoses, and you add them for contact positions (for example, heel contact poses in a walk or when the character changes direction). You insert these between the KeyPoses by setting the current time to the start frame of the pair of KeyPoses between which you add the Extremes by clicking **Insert Extreme**.

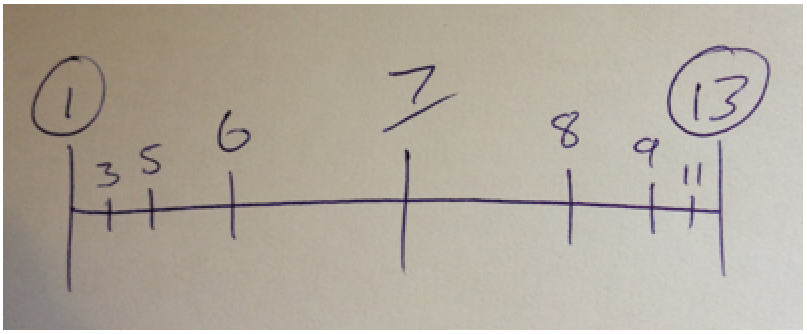

2
3
Insert KeyPose
Insert Extreme
i c r
2
4
6
8
10
12
14

As with inserting a KeyPose, Modo inserts a new frame and keyframes the action. Also, Modo adds an Extreme time marker to the action. The indicators for Extremes are different icons to distinguish them from the KeyPoses. Add as many Extremes as needed to define the animation between the KeyPoses.



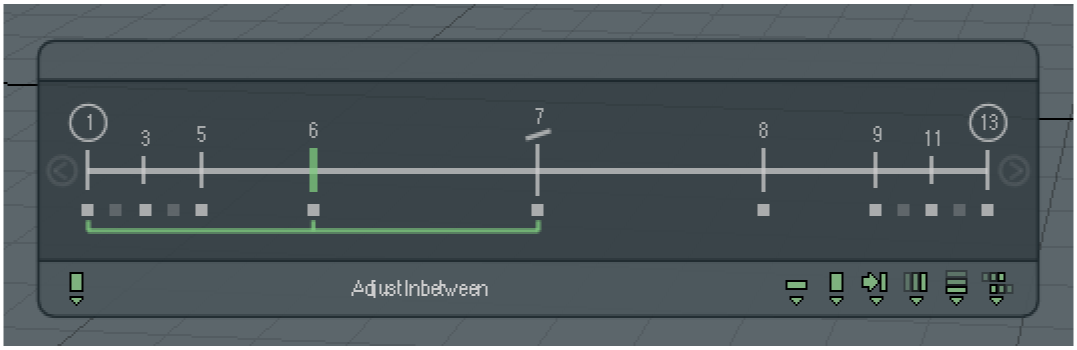
If you add an Extreme and decide it's important enough that it should be converted to a KeyPose or you decide that an existing KeyPose is really an Extreme, you can convert between the two types by using the buttons at the bottom-right area of the toolbar. When showing an Extreme frame, Modo shows the KeyPose button and vice-versa for a KeyPose frame. To convert the current frame, click the available button.

Animation Spacing Charts



Traditional animators who did hand-drawn animations relied on the Animation Spacing Chart. Animators developed this tool to increase the amount of animation they could produce. Instead of creating every drawing in the animation, the animator would only draw the key ones (KeyPoses and Extremes) and one or two breakdowns, which were enough to define the motion. Then, for each pair of key drawings, the animator would create a Spacing Chart that defined where and in what order the in-betweens should be added. An assistant would then take these drawings and, following the Spacing Chart, create the in-betweens: drawings half-way between each of the drawings in the chart and in the order specified.

Modo follows the same principles. It displays a Spacing Chart for each pair of KeyPoses with frame numbers shown circled at either end of the chart. The Primary breakdown shows with its frame number underlined as is traditionally found on these charts. In-betweens appear as lines of decreasing length with their frame number. The length of the lines indicate the order of creation with longer lines first. With the Spacing Chart, you can read how the motion between the key drawings flows. The shorter the gap between the markers, the slower the movement is. Large gaps indicate fast movement.



Clicking an in-between or breakdown marker in the chart sets the current time to that frame, and Modo draws a green chart marker. In the example, the current frame is frame 6. The green brackets shown under the markers indicate which pair of drawings (or poses) were used to create the breakdown or in-between. The frames between the pair of key poses show under the chart markers as small squares: solid for the defined breakdowns and in-betweens and dimmer for those that Modo automatically creates. Use the arrows at either end of the chart to navigate between each chart in your animation.

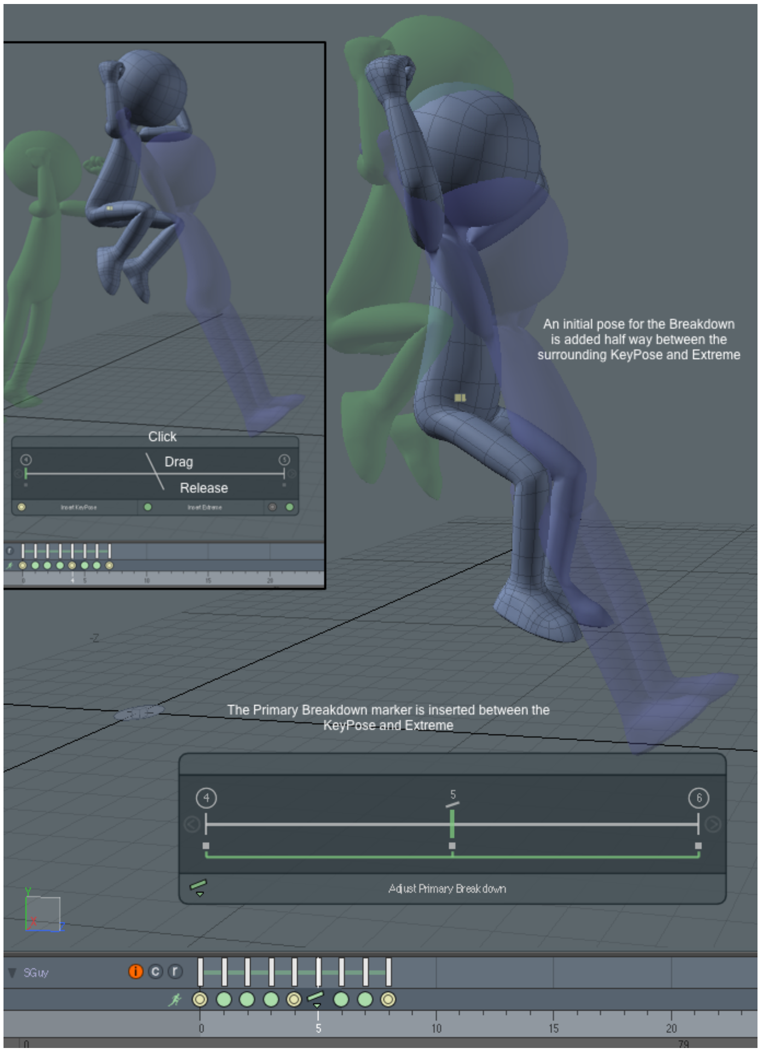
Breakdowns

Breakdowns (or passing positions) are important poses that define how a character moves between key poses. It provides the interest for the movement and is the basis for defining the arcs and delays used throughout.

Creating a Breakdown

You create breakdowns by drawing in the Spacing Chart area of the HUD. Click anywhere above the central line and drag down to release the mouse button below the line. Modo inserts a new frame, keys the actor, adds a Breakdown marker to the Timeline, and sets an initial pose half-way between the surrounding poses.

Note: If you have spaced the KeyPoses and Extremes out already so that there are empty frames between them, then the button becomes the **Add Primary Breakdown** button. Clicking this button creates the pose at the current time without shifting the following poses forward. Similarly if you draw the breakdown on the Spacing Chart, Modo does not shift the following poses but sets a frame half-way between (or as close to half-way as possible) and adds the breakdown pose there.


Click
Drag
Release
An initial pose for the Breakdown is added half way between the surrounding KeyPose and Extreme
The Primary Breakdown marker is inserted between the KeyPose and Extreme
4
5
6
Adjust Primary Breakdown
SGuy
0
5
10
15
20

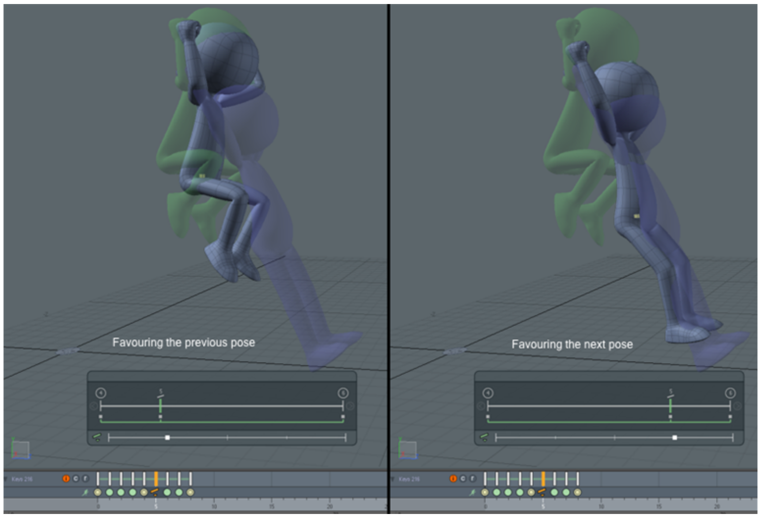
As can be seen in the example, Modo creates a mathematically correct in-between pose (which is good most of the time), but this pose has no personality or feeling because it's just a linear blend between the two poses. For an animator, it is a useful starting point for creating a more interesting pose.

Editing a Breakdown

With the Breakdown & Spacing HUD you can interactively favor a pose for the entire character or just the currently selected items belonging to the actor. You favor a pose by specifying whether it's closer to the previous or next pose. Traditionally, you indicate this on an animation Spacing Chart by drawing the marker closer to the previous or next pose on the chart rather than half-way between.

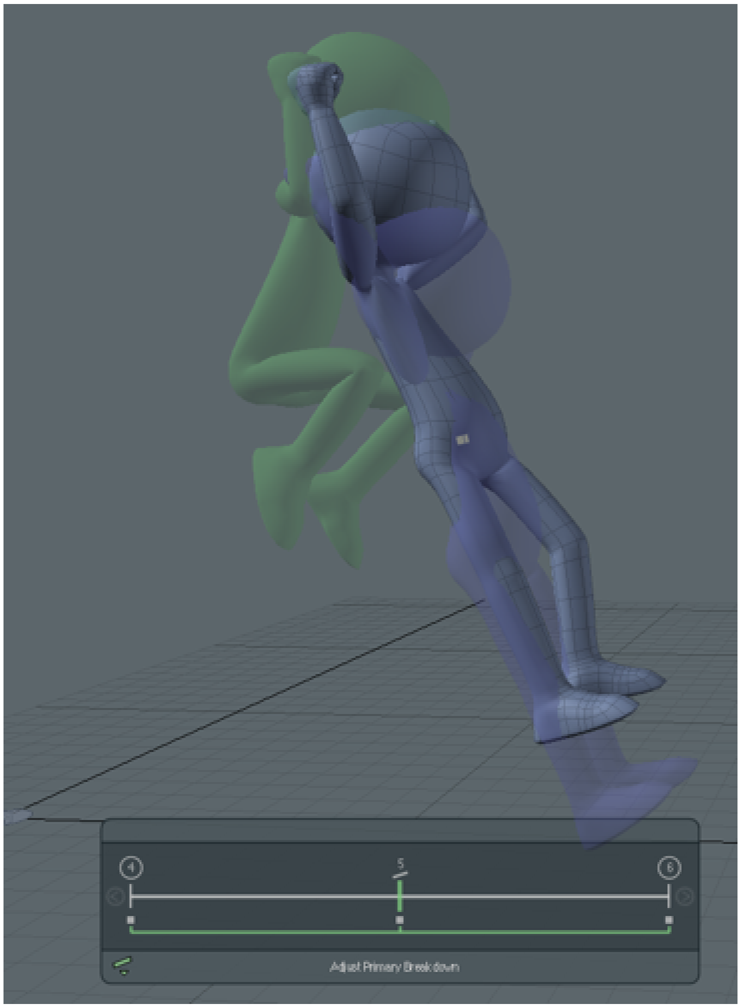
With Modo, you favor a pose by dragging its marker in the Spacing Chart. (You can also click the **Adjust Primary Breakdown** button.) The marker in the Spacing Chart slides to the left or right to indicate how the pose is favored. If you do not select any items, then Modo uses the entire pose. You can limit the in-between process to specific items by selecting them first. If you select items, the position of the marker on the Spacing Chart is not affected.

Of course, you can make adjustments to the pose in any way (for example, by using the Transform tools or the Pose tool). The favoring helps to define the initial pose broadly before fine-tuning it. If you make adjustments to items by favoring selected items or manually by using the Transform tools, you can still use favoring on the entire character, and Modo takes into account any adjustments you have made.

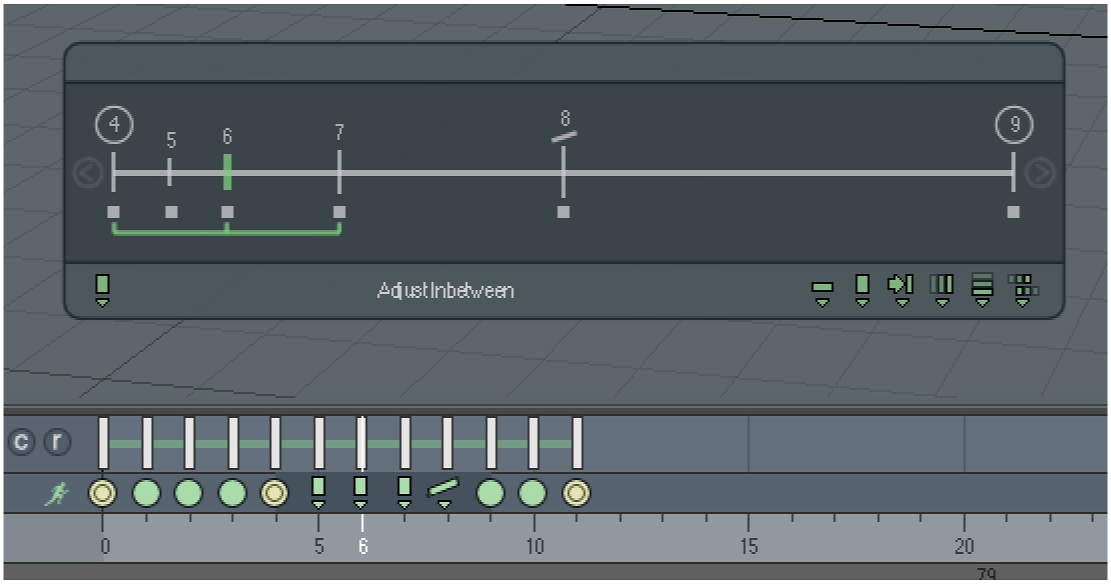


Creating In-betweens

Traditionally, assistant animators draw in-betweens from the Spacing Charts created for each pair of key drawings. The drawings they create are straightforward half-way poses between the animator's key drawings and are drawn in the order indicated on the chart. In this context, Modo is your assistant, and it needs to be told specifically what to do. Modo achieves this in the same way as the assistant animators did — by defining the in-betweens in the required order.



You create in-betweens in the same way as you create breakdowns — by drawing a stroke in the HUD on either side of the Primary breakdown or between existing in-betweens. Modo automatically inserts a new frame, keys the actor, and defines a half-way pose.



Editing an Inbetween

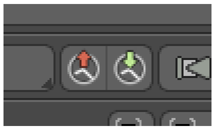
Modo usually does a decent job of creating the half-way poses, but you may need to edit the in-between pose, itself, either for correcting Modo's interpretation (which can happen when interpolating the existing poses) or when the motion needs to be further refined. (For example, you may need to add a secondary breakdown.)

Editing in-betweens is similar to editing breakdowns. The green brackets under the chart markers indicate which frames the in-between was created from. These are the pair of poses that Modo uses to interpolate from as it adjusts for the favoring slider's position or if you dragged the in-between in the chart. This is different from the Inbetween tool that always interpolates between the previous and next keys.

There are several time marker shapes that you can specify for in-betweens that can indicate its purpose. For example, if you've created an in-between and edited it to act as a secondary breakdown or to act as the start of a moving hold, you can change the marker shape by using the icons to the right of the button area. These are purely graphical representations. Changing the time marker's shape has no effect on how the in-between functions. It just serves as a reminder about the purpose of the in-between and as an identifier to make it easier to find a specific in-between.

Timing

The previous information concentrates on collecting the poses (or drawings) for the animation. Depending on how many poses you have, the timing can be done quickly because you've defined the sequence in advance. To start, set the action's **Interpolation** to Linear and generate a GL Preview. Make notes of where time needs to added and use the **Time Edit** buttons on the Animation toolbar to insert and remove frames as needed. Repeat this until you're happy with the timing. When you click the button, Modo inserts or removes one frame at the current time and shifts subsequent keyframes as needed.



If your scene is simple enough to play back the animation in real time, you can make timing adjustments while the scene plays. To do this, click the **Play** button and middle-click to set a keytime along the Timeline. This defines where to insert or remove frames and Modo shows a green line in the Timeline. By using the **Time Edit** buttons, you can then insert or remove frames while you watch the animation play.

To make edits to various parts of the character in the frames that you've added for the timing, use the standard keyframing tools. Because all the main poses have time markers associated with them, you can identify where the adjustment edits occurred. You can adjust the timing earlier than described in this topic. Some animators prefer to roughly time an animation out after the breakdown stage and insert in-betweens afterwards.

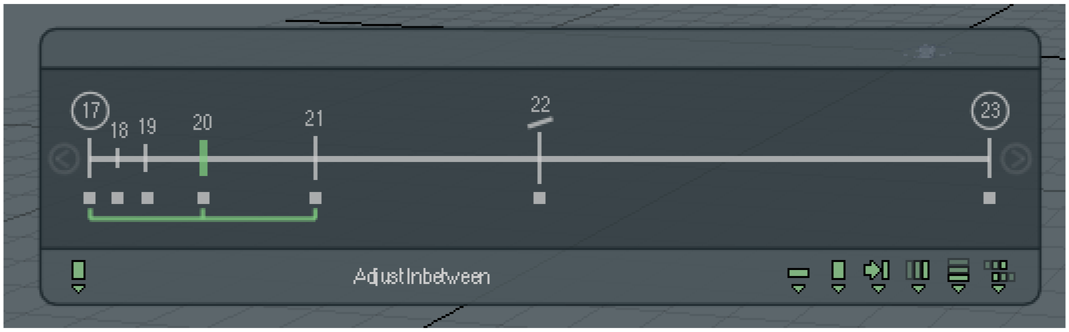
Cleanup

Once you time the animation and adjust the keyframes as needed, you can do the final clean-up process. Now, you set the action's **Interpolation** to Curve and, starting with the root of the character, look at the curves in the Graph Editor and check for any anomalies. Because the data points are defined by the poses, you should only need to make small adjustments.

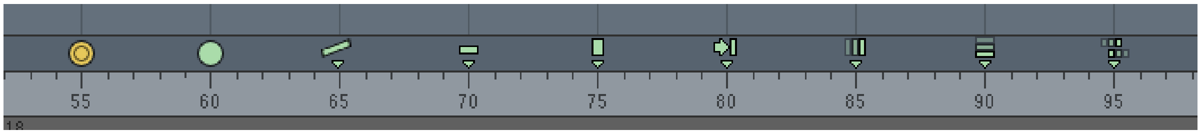
Watch the animation in the 3D view because if you consider making any changes based on the Graph Editor, the anomalies may there for a reason. It doesn't matter how the curves look because no one else sees them. What matters is how the animation plays in the viewport or through the camera. Work outwards from the root along the spine before moving onto the extremities such as the arms and legs.

Deleting Markers/Poses

To preserve the relationship of breakdowns and in-betweens to each other, you must delete them in a specific order: from the lowest level in-between upwards. For example, you cannot delete the Primary breakdown if it has in-betweens derived from it. To determine this relationship, select a marker in the Spacing Chart. If the green brackets enclose other markers (excluding the extents), then you delete those other markers first. You can select range of markers (in any order) and have Modo delete them all in the right order. If it cannot delete any, Modo displays a warning dialog.



Time Markers



The time markers used by the Spacing Chart are different from the normal Time Markers.

- Selecting a marker also selects keyframes at the same time as the marker.
- Deleting a marker also deletes selected keyframes that are at the same time as the marker.
- Dragging a marker also moves its associated keyframes.
- Markers cannot be moved past surrounding markers of this type.
- Only one Primary Breakdown marker can be made between each pair of KeyPoses/Extremes.

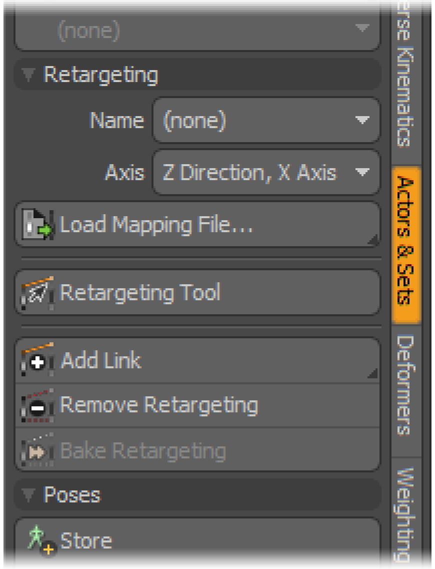
References

In his book *The Animator's Survival Kit* (particularly *Advancing Backwards to 1940*), Richard Williams advocates adjusting the timing of the animation after creating all the key poses. Keith Lango's VTS tutorials explain this in great detail as well and really emphasize the benefits. Eric Goldberg's *Character Animation Crash Course!* has great descriptions and examples of charts, key drawings, breakdowns, and so forth. Also, look at Jason Ryan's tutorials. He does most of his work in 2D before animating in 3D, but the principles are just as valid — more so because this process comes from traditional 2D animation workflows in the first place.

Retargeting

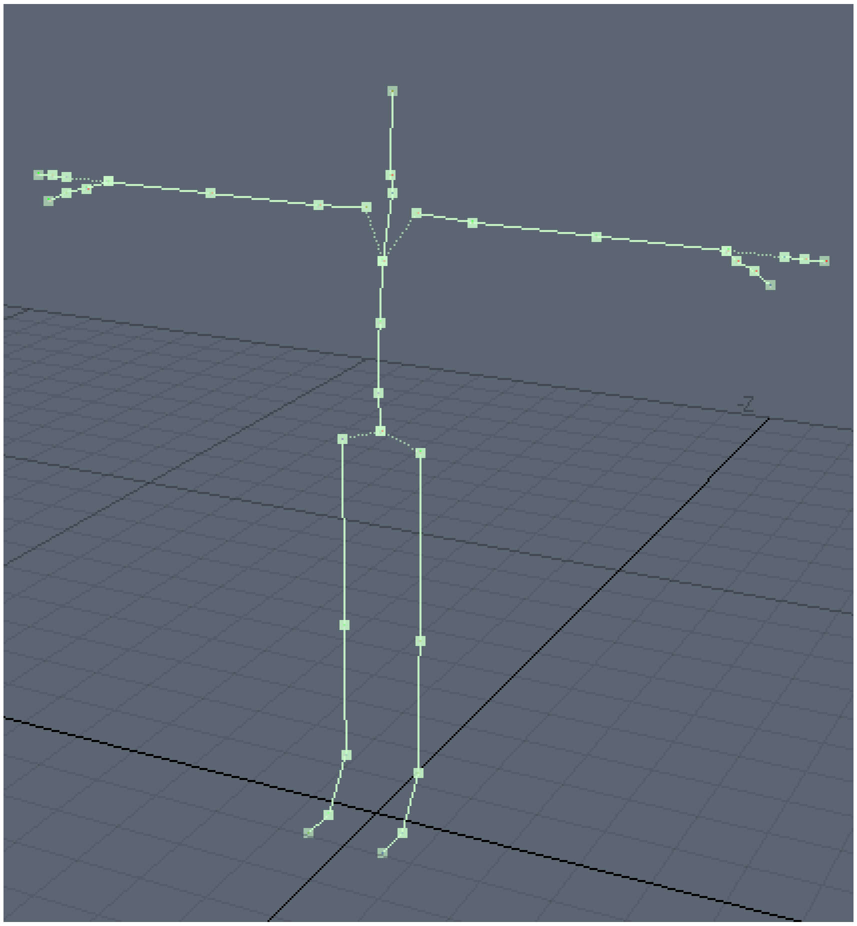
Retargeting is the process of transferring motion from one hierarchy of items to another. Typically this might be a character's walk or run cycle that would be reused on another character, which may have different proportions from the original. The motion can originate from existing keyframe animation in the scene, or can be imported motion, such as with an animated FBX file, or a BVH motion capture format import. The commands and tools for retargeting can be found in the **Retargeting** section of the **Actors & Sets** toolbar found in the toolbox of the **Setup** interface layout.

Preparation for Retargeting


(none)
Retargeting
Name (none)
Axis Z Direction, X Axis
Load Mapping File...
Retargeting Tool
Add Link
Remove Retargeting
Bake Retargeting
Poses
Store
erse Kinematics
Actors & Sets
Deformers
Weighting

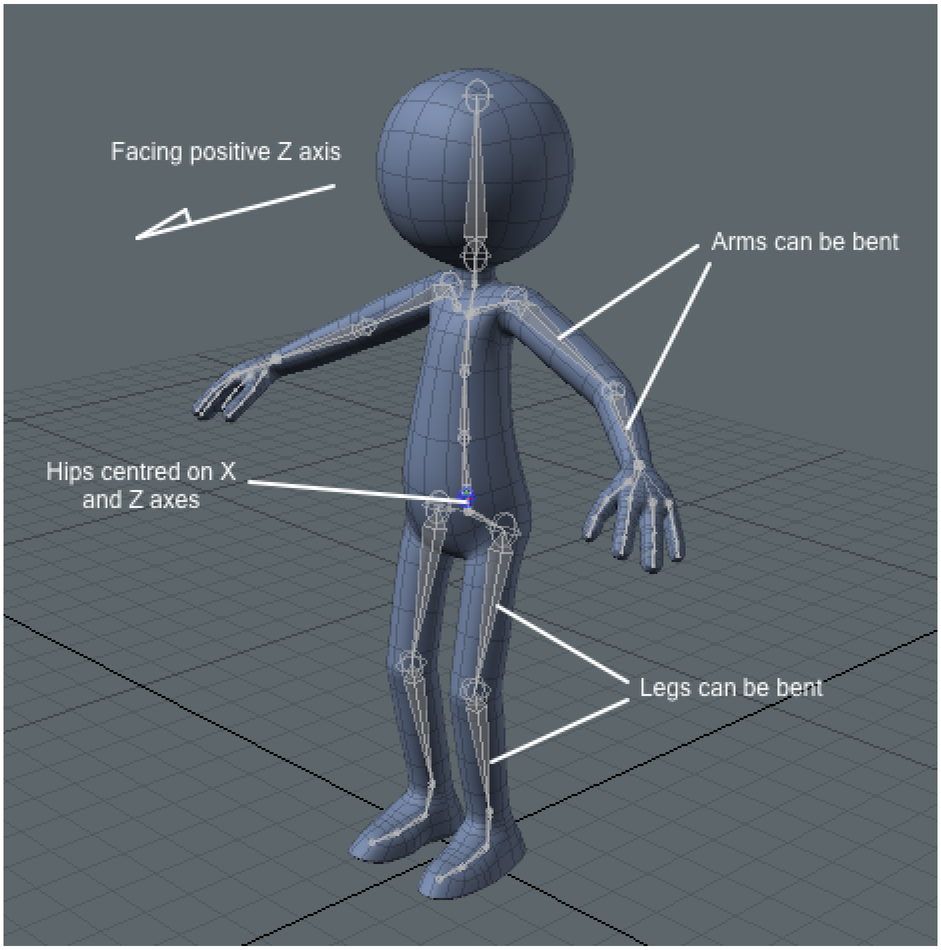
Source Skeleton

The Source skeleton is the motion source for the retargeting (the joints that contain the animation). While in **Setup** mode, this skeleton must be set to a default T-Pose (illustrated below) with all of its rotations set at **0** (zero). The T-pose is commonly used in motion capture data, although some files may have the arms set vertically. See Importing Motions for more information.



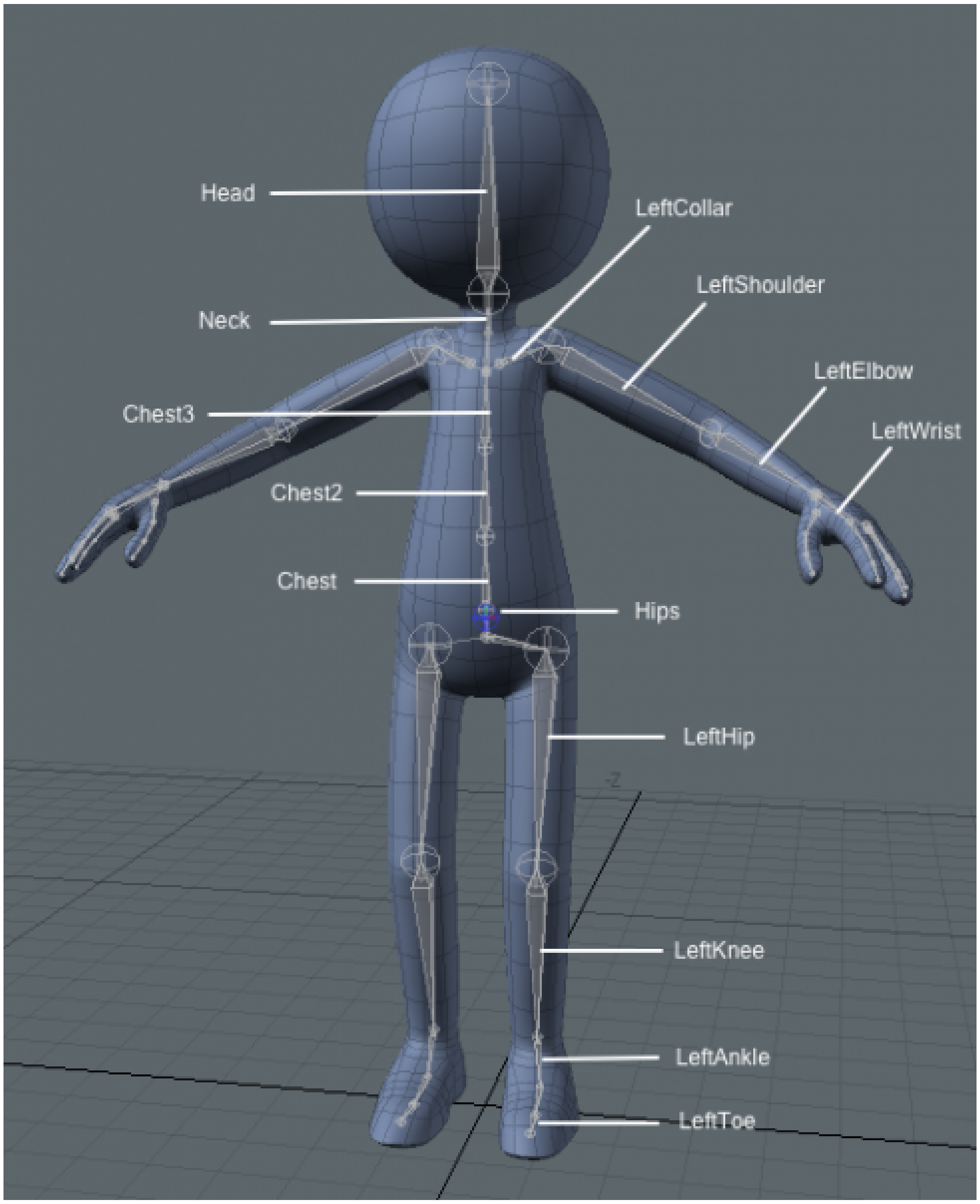
Target Skeleton

The skeleton onto which the motion data is loaded must be a single, continuous hierarchy of items. Motion cannot be retargeted onto individual, unconnected items. This skeleton must also be in a standard T-pose as well. However, since arms and legs are often partially bent to aid deformation and IK solving, Modo compensates for this when applying the retargeted motion. All other joints should be at their rest orientation. The skeleton must also be facing the Z+ axis with its root centered at the origin on the X and Z axes. However, it may be positioned offset on the Y axis.

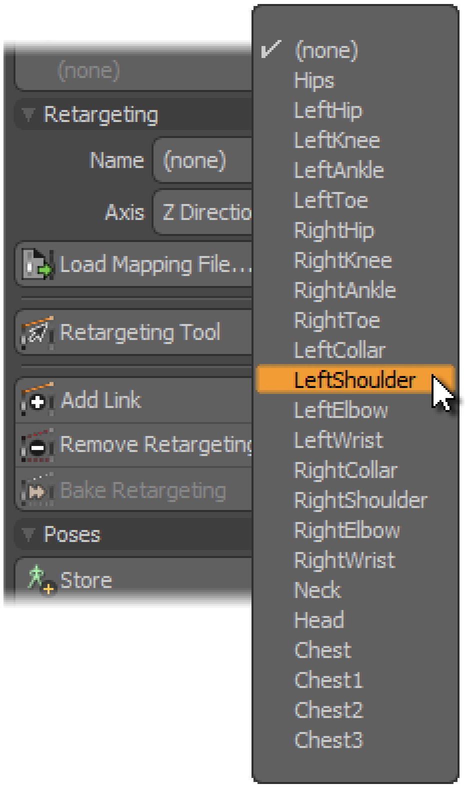

Facing positive Z axis
Arms can be bent
Hips centred on X and Z axes
Legs can be bent

Tagging the Target Skeleton

To help Modo identify particular joints in the target skeleton during the retargeting process, they should have common names assigned that follow a standard scheme (as is often the case for motion capture). Modo uses the *Character Studio Biped* naming convention (illustrated below) as this is found in the greatest majority of freely available motion capture files. **Note**: This step is optional but it greatly reduces the amount of time it takes to retarget the motion onto the skeleton. This naming structure allows Modo to perform various tasks automatically, such as scaling the motion to adapt for different sized skeletons, and so on.

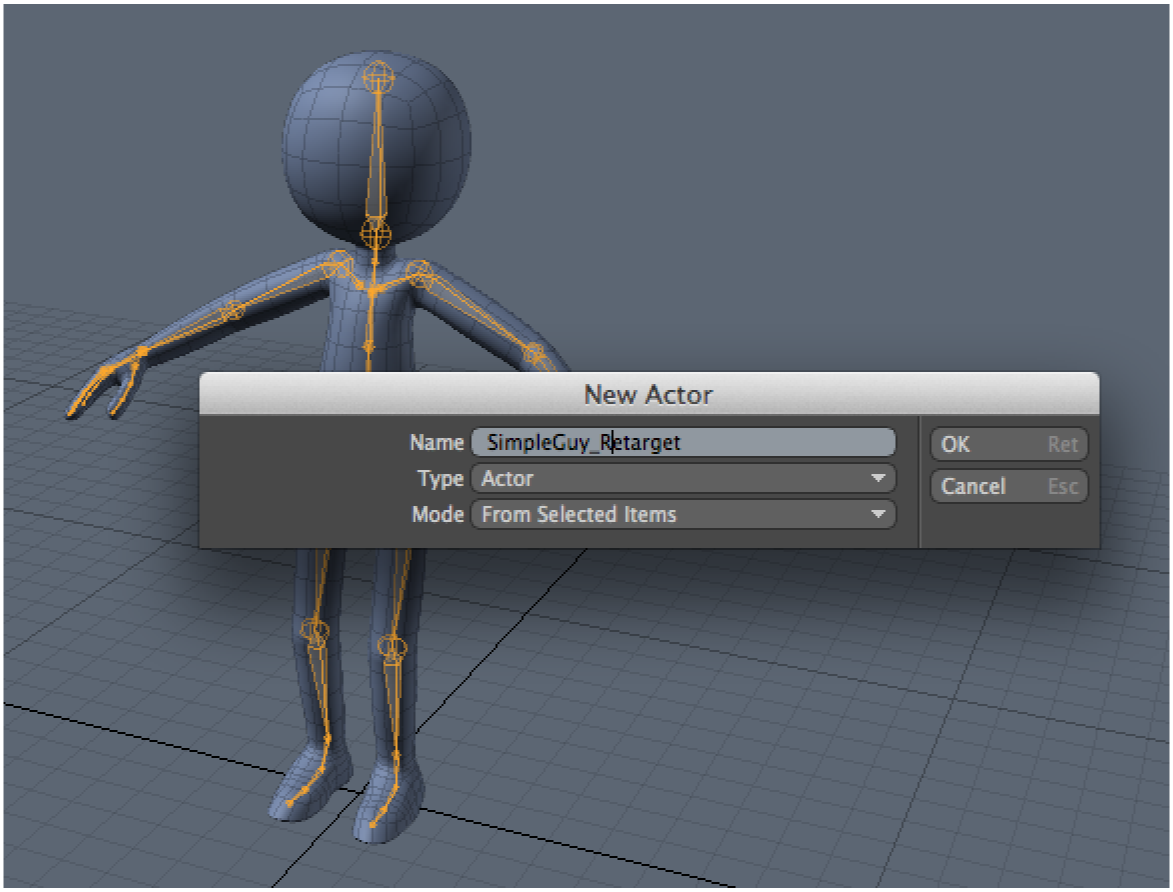

Head
LeftCollar
LeftShoulder
Neck
LeftElbow
Chest3
LeftWrist
Chest2
Chest
Hips
LeftHip
LeftKnee
LeftAnkle
LeftToe

To assign a name to a joint, simply select the target item and choose the appropriate pre-defined name from the pop-up menu in the **Retargeting** section of the **Actors & Sets** toolbar (found in the **Name** menu). When an item is selected, the menu updates to show the currently assigned name. To remove a tag from an item, select the **(none)** option from the top of the menu.


(none)
Retargeting
Name
(none)
Axis
Z Directio
Load Mapping File...
Retargeting Tool
Add Link
Remove Retargeting
Bake Retargeting
Poses
Store
(none)
Hips
LeftHip
LeftKnee
LeftAnkle
LeftToe
RightHip
RightKnee
RightAnkle
RightToe
LeftCollar
LeftShoulder
LeftElbow
LeftWrist
RightCollar
RightShoulder
RightElbow
RightWrist
Neck
Head
Chest
Chest1
Chest2
Chest3

Defining the Target Actor

Retargeting uses the Actor, Action & Pose functionality to help identify the target joints. If you are not familiar with this functionality, see the Creating Actors, Actions, and Poses topic. To assign a hierarchy as an Actor, double-clicking on the root item of the target skeleton (generally this is the hips) selecting the entire hierarchy, next choosing the **Actor** > **New** option from the **Actors & Sets** toolbar opening the create **New Actor** dialog. Make sure the **From Selected Items** option is set to **Mode** and enter a name for the skeleton (for example, 'SimpleGuy_Retarget') and press **OK** to make the Actor group.



New Actor

Name SimpleGuy_Retarget

Type Actor

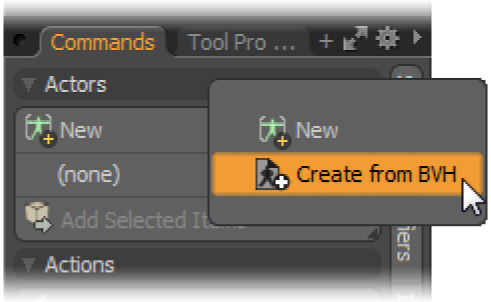
Mode From Selected Items

Importing Motions

BVH Format Motions

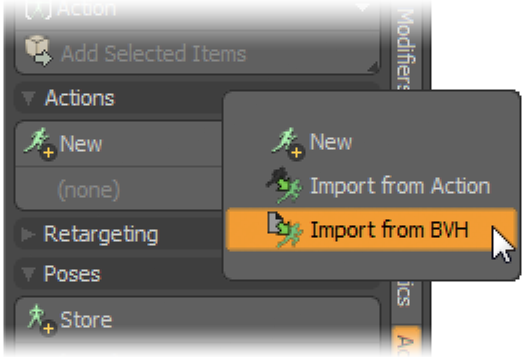
The **.bvh** format was developed by a now defunct motion capture company named Biovision as a means for them to deliver the motions to their customers. The format has since gone on to become an industry standard because of its simplicity and ease in implementation. Each Biovision (*.**bvh**) motion capture file contains a definition for the skeleton along with the actual motion data. The skeleton definition in the file needs to be created in Modo in order to apply the initial motion data to it. However, once that has been done, further **.bvh** files that use the same skeleton hierarchy can be loaded as additional Action states to the same Actor.

To load a **.bvh** file, select the **Create from BVH** option in the **Actors** > **New** tool menu in the **Actors & Sets** toolbar (shown below). This opens an OS-specific file requester, navigate to the desired **.bvh** file and click **Open** to load the file. Once loaded, Modo automatically constructs the skeleton from the definition in the **.bvh** file, create an Actor containing the skeleton items, and create an Action containing the motion capture data. The resulting Actor is named 'ActorBVH' by default and the Action containing the motion is named by the imported file (for example 'Run.bvh').



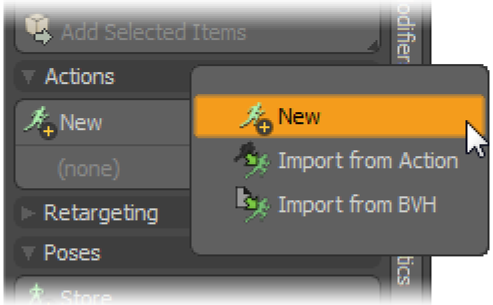
To alleviate various sizing issues, the generated skeleton is automatically scaled to 2m in height (approx. 6 feet) with appropriate adjustments made to the imported motion to match. It also centers the Setup pose at the origin (if necessary) and position the root upwards in Y+ direction, placing the feet on the ground. The new Action containing the imported motion is made active, allowing the animation to be previewed in the 3D viewport.

If there are additional **.bvh** files that use the same skeleton, their motion can be loaded as additional Actions into the same Actor. Choose **Import from BVH** from the **Actions** > **New** tool menu. This opens an OS-specific file requester, navigate to the desired **.bvh** file and click **Open** to load the file. Once loaded, a new Action is created with the name of the **.bvh** file. Any number of additional motions can be loaded this way, as long as they all use the same skeleton. Switching between the various motions is done in the normal way, by choosing the desired 'Action' from the available choices in the pop-up menu.

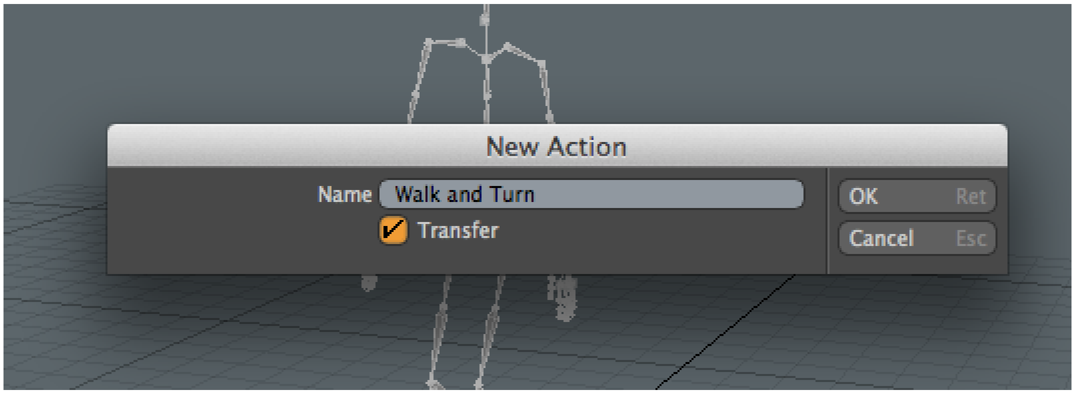


FBX Format Files

The **.fbx** format is a scene data exchange format for transferring geometry, animation and other data between applications. Files that contain skeletons with animation data can be loaded and retargeted onto other characters. The **.fbx** file must first be loaded using the **File** > **Import...** command, loading the files' contents into the current scene. This creates a skeleton with the motion information applied.
For retargeting, this skeleton must be assigned as an Actor for the entire skeleton using the same method described above. The skeleton must also be in the default T-pose while in **Setup** mode. If not, then all joints must be rotated until it is. Once this is done, the motion in the skeleton has to be transferred to an Action. To do this, first make sure to exit **Setup** mode, then create a new Action, using the **New** button.


Add Selected Items
Actions
New
(none)
Retargeting
Poses
Store
New
Import from Action
Import from BVH

This opens the **New Action** dialog box. Give the action an appropriate name and make sure the **Transfer** option is enabled to move the animation from the scene into the Action.

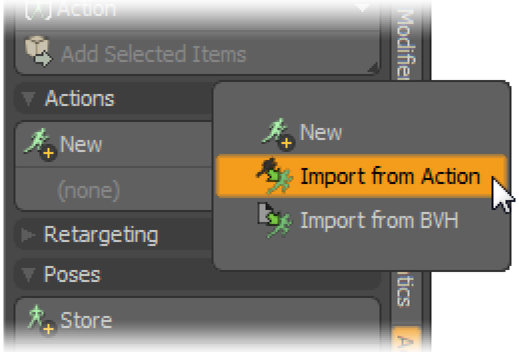

New Action
OK Ret
Cancel Esc

Name Walk and Turn

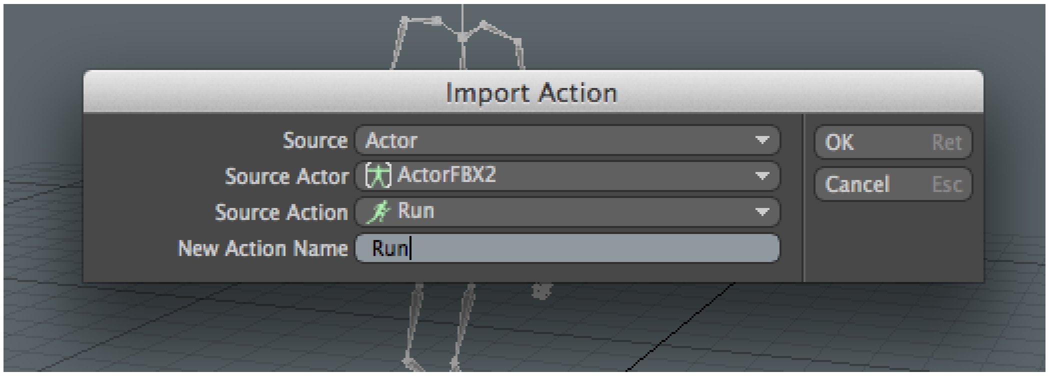
☑ Transfer

Since there is no way to load the FBX motion data without the skeleton, the process of loading multiple motions into the same Actor is different from that used for **.bvh** files. The steps below need to be followed for each **.fbx** file.

1. Import the **.fbx** file. Create a new Actor for the skeleton.
2. Create a new Action and transfer the animation as described above.
3. Once that has been done, the motion can be copied to the target Actor, making the original Actor current by selecting it in the pop-up menu.
4. Next, select the **Import from Action** command from the **New** Action tool menu.


Add Selected Items
Actions
New
(none)
Retargeting
Poses
Store
New
Import from Action
Import from BVH

In the resulting dialog, make sure that **Actor** is set as the **Source**. Then select the desired Actor that contains the motion that was just loaded and the **Source Action** to transfer. Designate a name for the resulting action in the **New Action Name** field (it defaults to the name of the source Action if left blank).



Source: Actor
Source Actor: ActorFBX2
Source Action: Run
New Action Name: Run

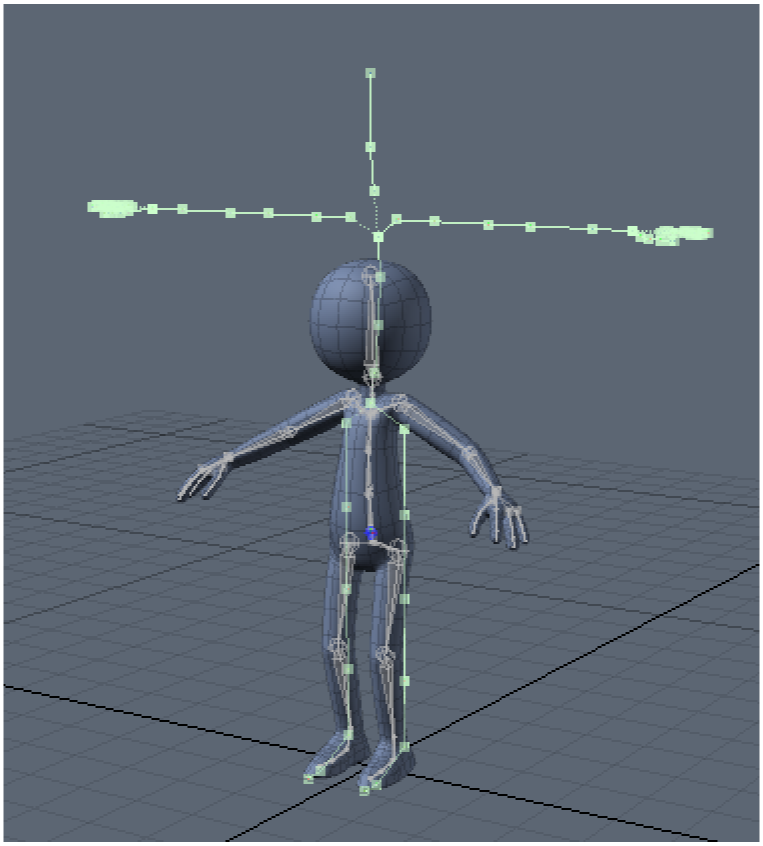
Click **OK** for the Action to be copied to the current Actor. This sequence can be repeated for each FBX motion. Once all the motions are transferred, the (now unnecessary) source skeletons can be deleted.

Retargeting

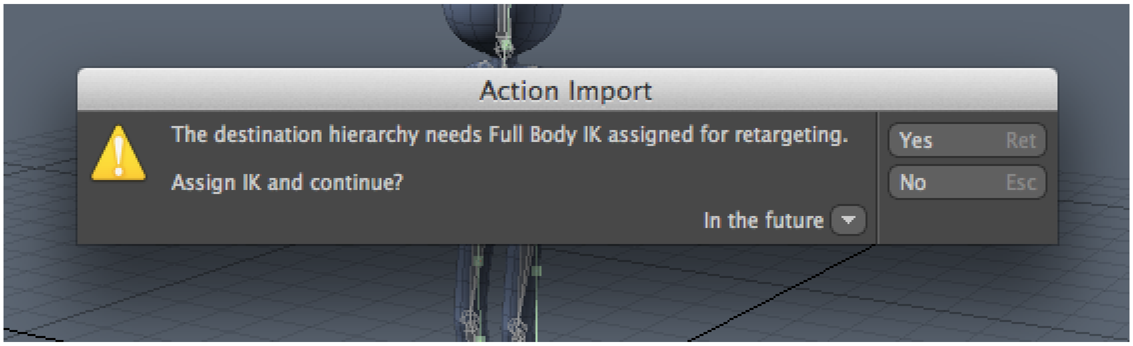
Retargeting Tool

Now that the Source and Target skeletons have been properly set up, its time for the actual retargeting. This is performed in **Setup** mode using an interactive tool.

Note: The skeletons can have different sizes and proportions.



To activate the tool, first select the root joint of the source skeleton, followed by the root joint of the target skeleton (in that particular order). Then click the **Retargeting Tool** button. It can be found in the **Retargeting** section of the **Setup** layout toolbox (under the **Actors and Sets** sub-tab, it may also be minimized by default). Modo uses its Full Body IK to help solve the retargeting, the Retargeting tool first checks if IK has been assigned to the target skeleton and add it if required, you are prompted with the following dialog.



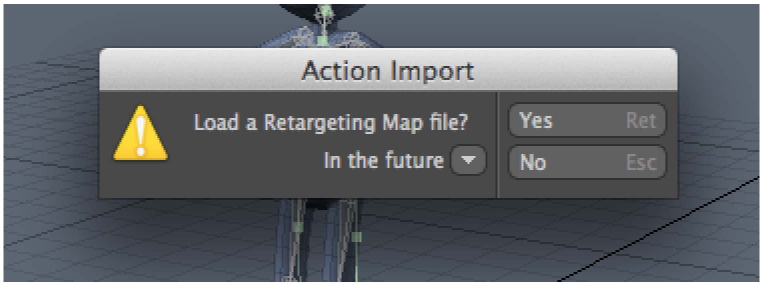
Action Import

The destination hierarchy needs Full Body IK assigned for retargeting.

Assign IK and continue?

The Retargeting tool then asks for a retargeting map file. These files store the retargeting information for a pair of skeletons, allowing the retargeting to be applied automatically. If this is the first time these skeletons have been mapped, choose **No** to continue to the next step. If the skeletons have been mapped before and a mapping file has been saved for them (see Mapping Files) then choose 'Yes' to open a file requester.

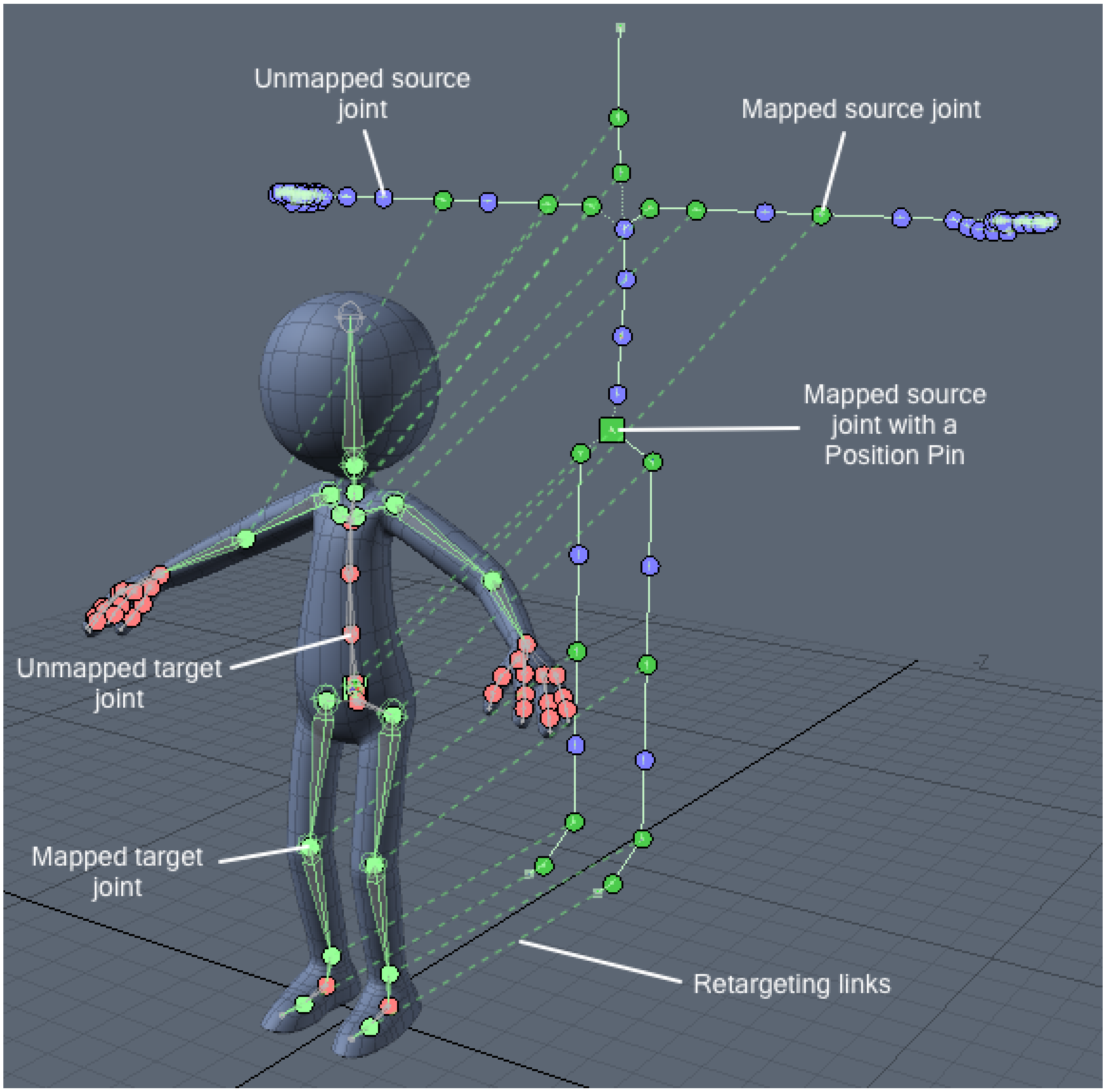
Choose the mapping file and click **OK**. All the retargeting is then applied and you can view the motion on the character immediately in the 3D viewport.



Action Import

Load a Retargeting Map file?

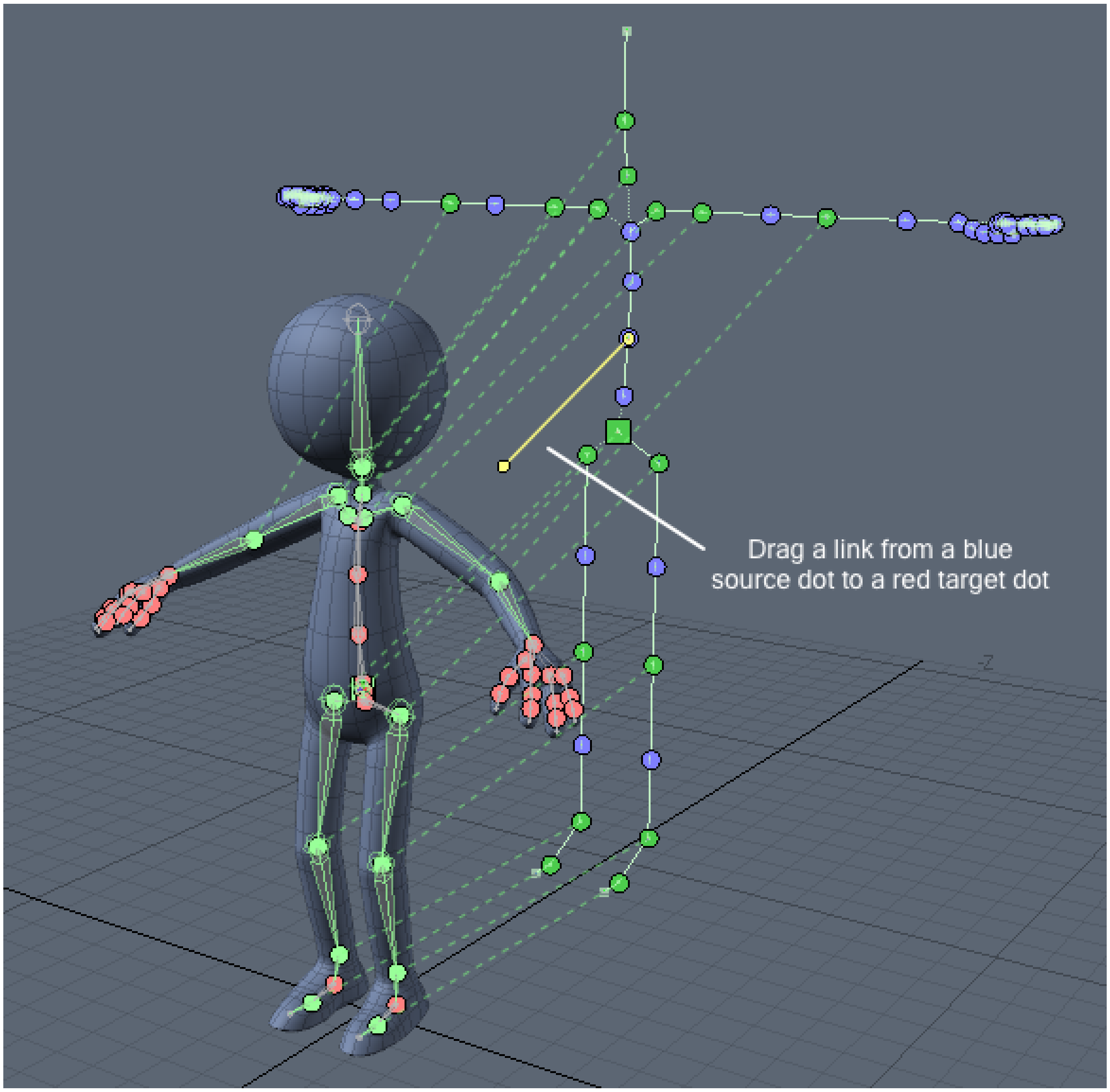
The retargeting tool then draws its handles in the 3D views. The source skeleton is moved back temporarily to help make the connections clearer, it is restored when the tool is disabled. If a mapping file was not chosen, the tool examines the skeletons and make as many connections as it can automatically, using the joint names defined in the setup phase to identify matching joints. Since the root items can be awkward to access in the 3D views, the tool maps these automatically and also add a Position Pin (described below).


Unmapped source joint
Mapped source joint
Mapped source joint with a Position Pin
Unmapped target joint
Mapped target joint
Retargeting links
-Z

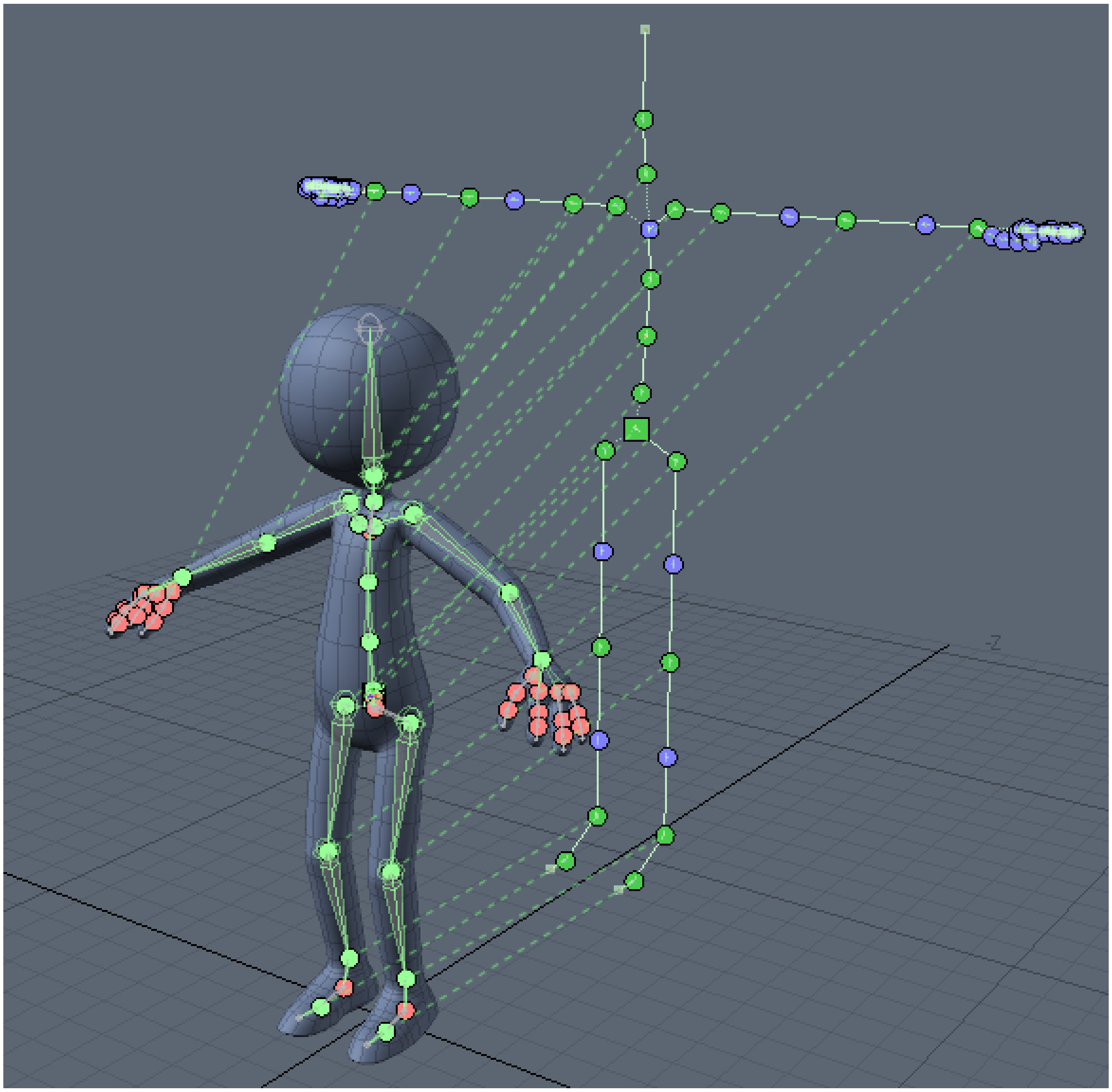
The tool uses color coded connection handles (drawn as solid circles) to allow new links to be made and provide feedback on existing connections.

- **Blue**: An unmapped joint on the source skeleton
- **Red**: An unmapped joint on the target skeleton
- **Green**: A mapped joint

Retargeting links are drawn as dotted lines between mapped joints. The wireframe color of mapped joints on the target skeleton is changed to green to show that the joint is currently driven by retargeting, even when the Retargeting tool is off. Once retargeting is removed from the skeleton, the original color of the joint is restored. New links are added by clicking and dragging from a blue source connection handle to a red target handle. Links can be removed by **Ctrl**+clicking on a source or target connection handle.


Drag a link from a blue
source dot to a red target dot
-Z

Adding a new link.

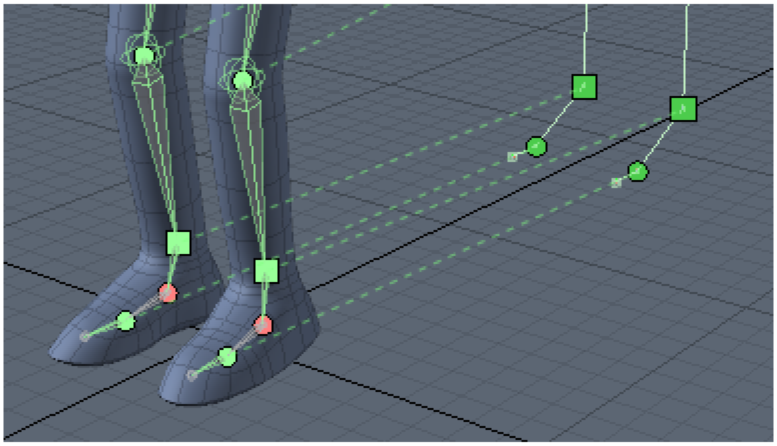


The skeleton fully mapped.

Position Pins

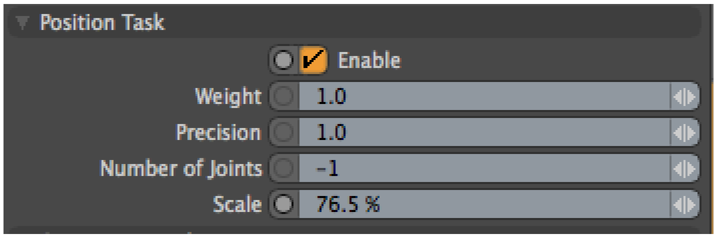
The retargeting links, described above, map the rotation of a source joint to the rotation of a target joint. Translational mapping (the actual movement), is achieved by the use of Position Pins. These take the position of a source joint and apply it, scaled, to that of the target joint. The scaling is based on the difference in proportions and size of the target skeleton, compared to its source. Position Pins almost always need to be applied to the root of the target skeleton, so they are added automatically when the Retargeting tool is activated.

When a Position Pin is assigned to a joint, the green circular handle changes to a green square shape. Position Pins are enabled as a toggle, turning them on and off by right-clicking on a source or target handle. You can remove a Position Pin by **Ctrl**+clicking on it.



The automatic scaling that is applied when linking skeletons of different scales, helps to reduce any foot sliding. Adding a Position Pin to the character's feet can further improve the results. These pins can be applied to other parts of the skeleton as well, when desired. For example, if a motion involves the character's hands resting on the ground, such as with a handstand, Position Pins can be applied to the wrists to help keep them in place.

Note: If retargeting tags have not been assigned to the target skeleton, then automatic scaling cannot be applied and needs to be set and adjusted manually. The Full Body IK settings **Position Task** attribute has a **Scale** channel which can be used for this purpose. This is also useful to make adjustments (if necessary) to the automatic scale that's calculated from the retargeting tags.



Position Task

☑ Enable

Weight 1.0

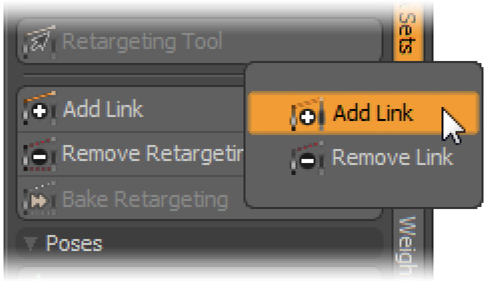
Precision 1.0

Number of Joints -1

Scale 76.5 %

Adding & Removing Links Manually

Some joints can be awkward to access with the Retargeting tool. This might be because they may be very close to other joints or even in the same position. In these cases links can be made manually using the **Add Link** and **Remove Link** buttons found under the **Retargeting Tool**. These can be used with the tool on or off. To add a link, select the source joint followed by the target joint and click the **Add Link** button. To remove a link select the target joint and click the **Remove Link** button.


Retargeting Tool
Add Link
Remove Retargetir
Bake Retargeting
Poses
Add Link
Remove Link
Sets
Weigh

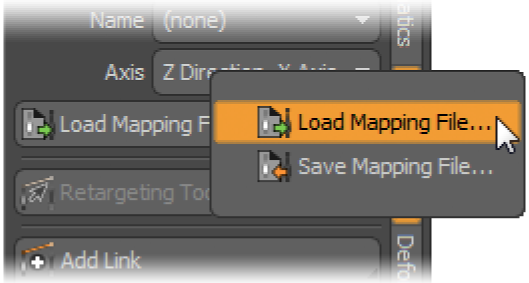
Previewing the Motion

Once all the retargeting links have been assigned, exiting **Setup** mode allows you to preview the retargeted motion now applied to the character. The connector dots are drawn to make the mapping clearer. You can go in and out of **Setup** mode while the Retargeting tool is active as many times as necessary while editing the retargeting.

If the source Actor has multiple Actions, it's possible to switch between them and see each motion applied to the character.

Mapping Files

A mapping file stores the retargeting information for a skeleton pair, allowing retargeting to be reapplied, if required, automatically. Once retargeting has been finalized it is a good idea to save a mapping file for the skeletons, in case further motion needs to be applied in the future, after retargeting has been removed from the target skeleton.

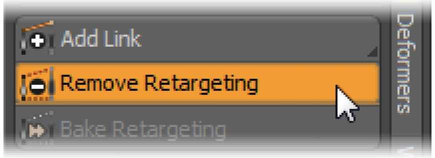

Name (none)
Axis Z Direction X Axis
Load Mapping F
Retargeting Too
Add Link
Load Mapping File…
Save Mapping File…

Mapping files can be saved at any time provided the Actor for the target skeleton is selected in the Actor pop-up menu and retargeting is applied to the skeletons joints. Mapping files are loaded when activating the Retargeting tool, a requester appears asking if one should be loaded.

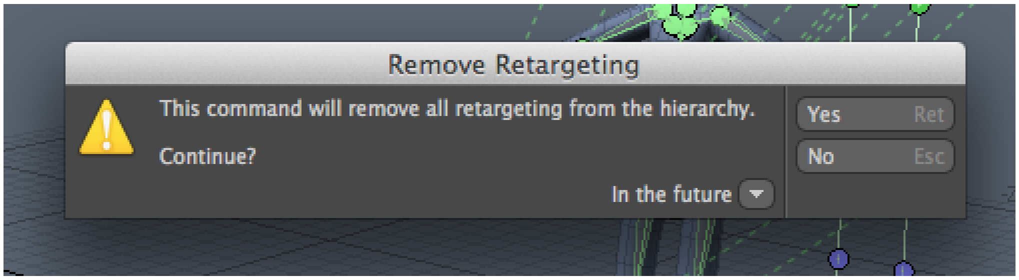
Note: The items in the file are identified by name so it's best to perform retargeting in a scene with a minimum of other items.

Removing Retargeting

Retargeting can be removed from a skeleton by clicking on the **Remove Retargeting** button in the toolbar. This must be performed in **Setup** mode with the target skeleton selected.



Click the button to initiate the removal, opening a warning confirmation dialog box. Clicking **Yes** removes retargeting links, Position Pins, and tasks from the skeleton and the joints are restored to their settings and original colors. Full Body IK is not currently removed.

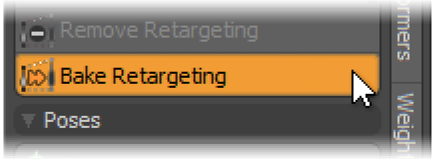

Remove Retargeting

This command will remove all retargeting from the hierarchy.

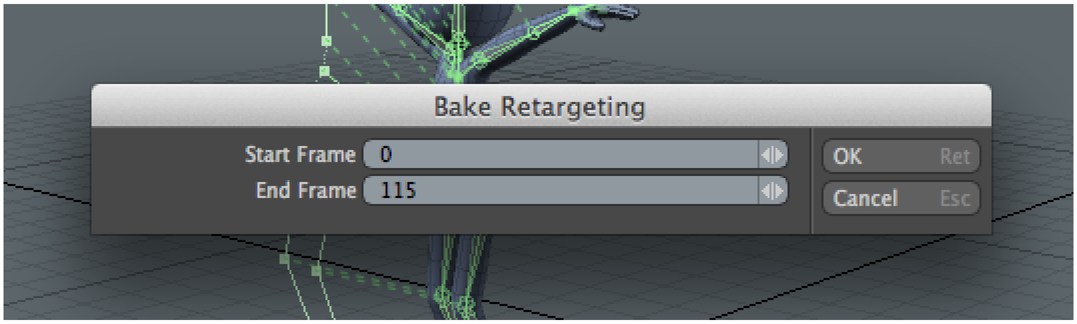
Continue?

Baking the Animations

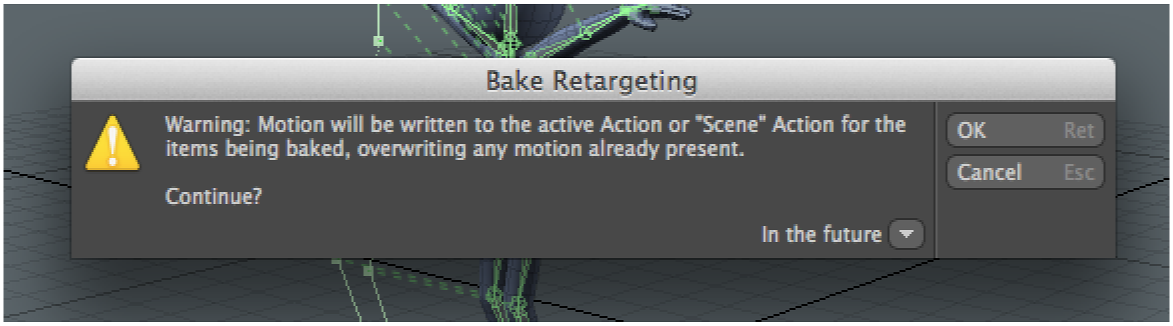
Baking applies the retargeting to the joints of the target skeleton by generating actual keyframes (at every frame) of the motion, for each joint in the skeleton. Once baked, the retargeting can be removed allowing keyframes to be manually edited, for instance to adjust the motion or clean up any jittering found in the motion capture data. Baking is initiated by selecting the root element of the target skeleton and clicking the **Bake Retargeting** button in the toolbox.



This action opens the **Bake Retargeting** dialog panel where the **Start** and **End** frames are defined.



Baking writes keyframes for every frame to the currently active Action for the target skeleton or to the scene Action if one is not active. Clicking **OK** opens a warning dialog before baking commences.



Warning: Motion will be written to the active Action or "Scene" Action for the items being baked, overwriting any motion already present.

Continue?

To avoid overwriting an action the simplest option is to make sure that no Action is currently selected before baking for the target Actor. A new Action can be created afterwards if the baked motion is satisfactory. Clicking **OK** on the warning dialog shows a progress bar as the baking is applied to the retargeting. Once baking has finished a dialog is displayed asking if the retargeting should be removed from the target skeleton. Select **Yes** if a single motion is being baked to the target skeleton or **No** when there are further motions on the source skeleton that need to be baked. Removing retargeting removes all the retargeting links, Position Pins and restores the original settings and colors of the target joints. It also deactivates the Retargeting tool if necessary.

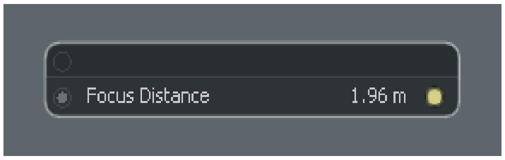
Once baking has been completed, a new Action can be created for the baked motion. Choose the Actor for the target skeleton and then click on **New...** in the **Actions** section of the toolbar. Enter a name for the new Action in the dialog and make sure the **Transfer** option is checked to copy the motion from the scene Action to the new Action.

Baking Multiple Motions

If the source Actor has multiple motions these can all be baked together before the retargeting is removed. Choose the Action to be baked from the source Actor. Select the root of the target skeleton, making sure there are no active Actions for the target Actor. Click on the **Bake Retargeting** button. When prompted in the dialog if retargeting should be removed select **No**. Create a new Action for the target Actor and transfer the baked motion from the scene Action (as outlined earlier). Repeat these steps for each motion belonging to the source Actor. When all the motions have been baked the retargeting can then be removed from the source skeleton and there is a separate Action for each of the motions in the target Actor.

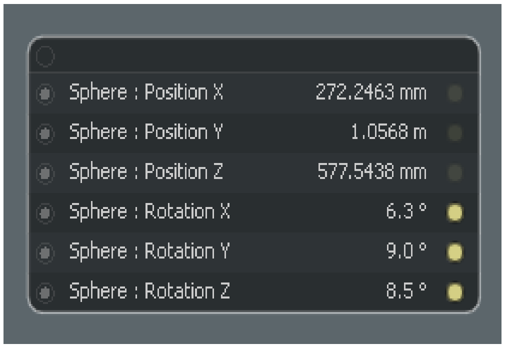
Tools and Commands

Channel Haul: With the Channel Haul tool, you can adjust any selected channel by dragging the pointer in the active viewport.



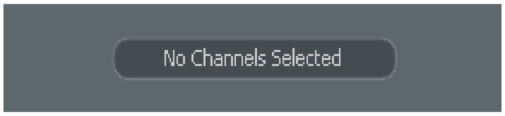
Focus Distance 1.96 m

To activate the Channel Haul tool, select a channel within either the **Channels** or **Properties** viewport and click the **Channel Haul Tool** button or press the **C** key. Modo displays the Channel Haul tool's interface within the 3D viewport. By clicking a channel name and dragging horizontally, you can interactively adjust the value within the viewport. Alternatively, if you click the value itself, you can type a new value. With the Channel Haul tool, you can modify up to 6 selected channels simultaneously. When any value is edited, a yellow dot appears to its right, clicking the yellow dot will discard any changes made to the channel.



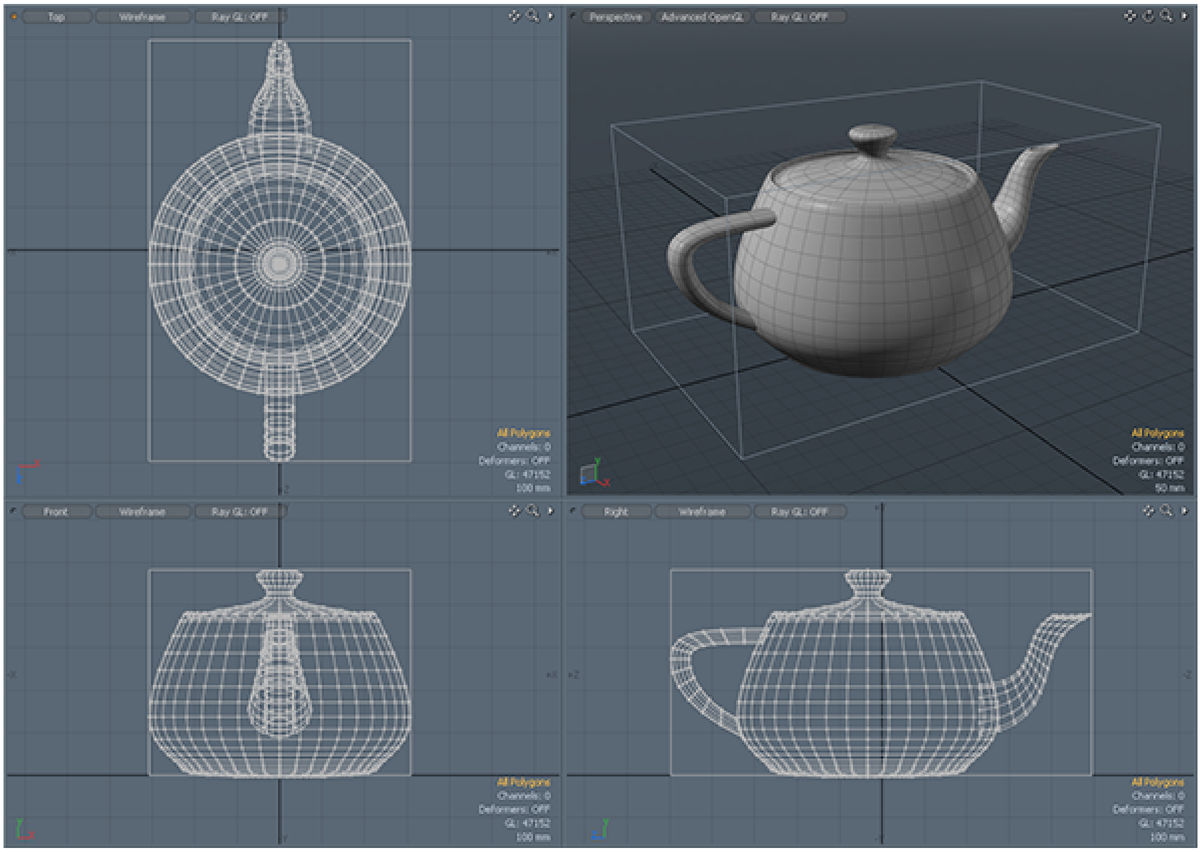
Sphere : Position X	272.2463 mm
Sphere : Position Y	1.0568 m
Sphere : Position Z	577.5438 mm
Sphere : Rotation X	6.3 °
Sphere : Rotation Y	9.0 °
Sphere : Rotation Z	8.5 °

If you do not select a channel, Modo displays this reminder that the tool is still active.



No Channels Selected

Center to Bounding Box: Each Mesh Item in Modo, regardless of shape, has a virtual rectangular bounding box that is defined by an item's overall size (height, width, and depth). With the **Center to Bounding Box** command, you can easily re-position an item's center to a specific position in that bounding box.


Top
Wireframe
Ray GL: OFF
Perspective
Advanced OpenGL
Ray GL: OFF
All Polygons
Channels: 0
Deformers: OFF
GL: 47152
100 mm
50 mm
Front
Wireframe
Ray GL: OFF
Right
Wireframe
Ray GL: OFF

By choosing **Edit** > **Center to Bounding Box** > **position**, you can precisely position the center element in relation to the geometry in that layer.

Center: Positions the item's center element at the very center of the bounding box itself (as opposed to the World Origin).
Top/**Bottom**/**Left**/**Right**/**Front**/**Back**: Positions the item's center element at the center of the plane that defines that specific surface of the bounding box's volume.

Pivot to Bounding Box: By choosing **Edit** > **Pivot to Bounding Box** >**Position**, you can precisely position the pivot element relation to the geometry in that layer. These command works similarly to the **Center to Bounding Box** commands; however, only the pivot moves and the center element remains unchanged. For information about centers and pivots, please see the Animation Overview topic.

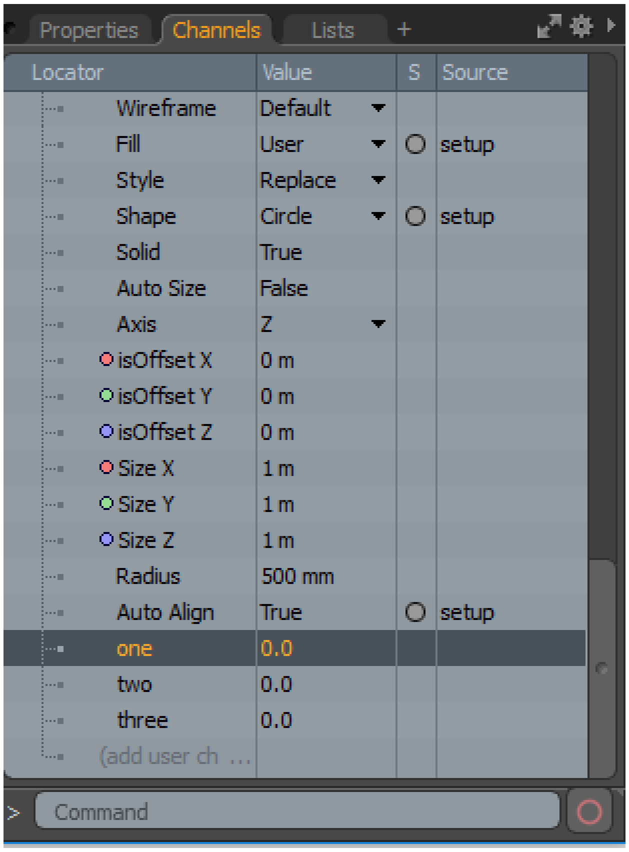
Center: Positions the item's pivot element at the very center of the bounding box itself (as opposed to the World Origin).
Top/**Bottom**/**Left**/**Right**/**Front**/**Back**: Positions the item's pivot element at the center of the plane that defines that specific surface of the bounding box's volume.

Displaying All Available User Channels in the 3D Viewport

Modo allows you to control the state of the User Channels while working in the 3D Viewport. By using the **Channel Haul** tool, you can view all of the available User Channels of selected items in the 3D Viewport. The following procedure explains the workflow. For more information about the Channel Haul tool, see Tools and Commands.

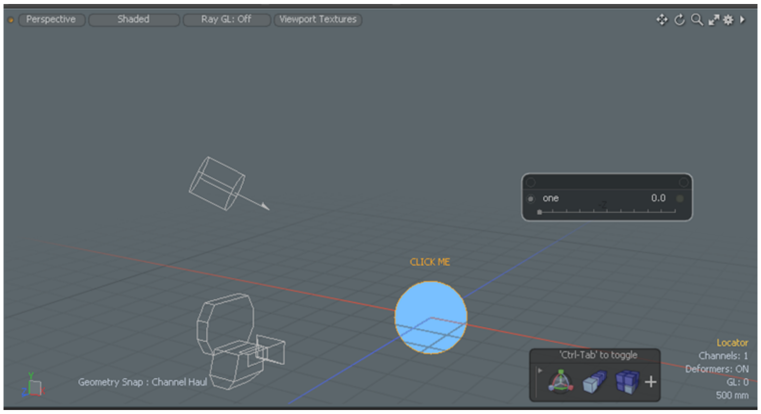
To view all the available channels:

1. Download our example scenes and open **channel_haul.lxo**.
2. On the right panel, open the **Items** tab and select **Locator**.
3. In the right panel, open the **Channels** tab, in the lower portion of the panel, select a channel. In this example, select **one**.



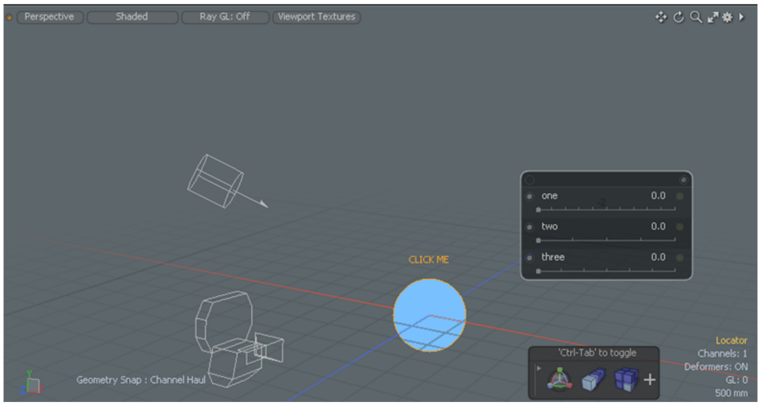
Locator	Value	S	Source
Wireframe	Default		
Fill	User	○	setup
Style	Replace		
Shape	Circle	○	setup
Solid	True		
Auto Size	False		
Axis	Z		
isOffset X	0 m		
isOffset Y	0 m		
isOffset Z	0 m		
Size X	1 m		
Size Y	1 m		
Size Z	1 m		
Radius	500 mm		
Auto Align	True	○	setup
one	0.0		
two	0.0		
three	0.0		
(add user ch ...			

4. Press **C** on your keyboard.


Perspective
Shaded
Ray GL: Off
Viewport Textures
one 0.0
CLICK ME
'Ctrl-Tab' to toggle
Locator
Channels: 1
Deformers: ON
GL: 0
500 mm
Geometry Snap : Channel Haul

The available User Channel displays.

5. On the top right corner of the **Channel Haul** tool, click the **circular radio button** to view all available User Channels.

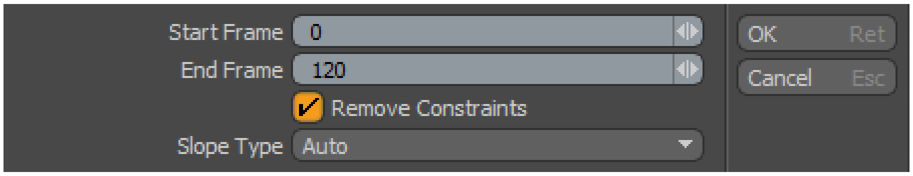

Perspective
Shaded
Ray GL: Off
Viewport Textures
one 0.0
two 0.0
three 0.0
CLICK ME
'Ctrl-Tab' to toggle
Locator
Channels: 1
Deformers: ON
GL: 0
500 mm
Geometry Snap : Channel Haul

6. Press **C** to dismiss the **Channel Haul** tool.

Bake

Baking sets (or freezes) the effects of a constraint on an item over time so that, when the constraint is removed, the item retains the modifications. You may use this feature when exporting an animation out of Modo or if network rendering and using pre-evaluated transforms would reduce the possibility of errors. You can also use baking to convert cached **Dynamics for Rigid Bodies** to individual keyframes.

To use, select one or more items that are affected by a constraint, Channel Modifier, or dynamics, and then choose **Animate > Bake**. This command opens a dialog for you to specify the bake options. After you assign the desired values, click **OK** to actually bake the specified keyframes.



Start Frame 0

End Frame 120

☑ Remove Constraints

Slope Type Auto

Start Frame/End Frame: Defines the frame range to convert the constraint-related values on the keyframes.

Remove Constraints: When enabled, removes the constraints driving the item at the completion of the bake process. The item solely uses the baked keyframes for its motion. When disabled, the constraints remain.

Slope Type: Specifies which type of slope Modo assigns to the resulting keyframes: **Auto**, **Stepped**, or **Flat**. The slope defines the type of interpolation between keyframes. See the Graph Editor topic for more information.

Channel Handles

Channel Handles allow you to drive numeric channel values by dragging directly on controls in the 3D viewport. A single handle can drive any number of channels, and a single channel can be driven by any number of handles. These handles are always active and can be used with the normal transform tools. Modo derives Channel Handles from standard locator items. They have their own transform channels and can be positioned and orientated as needed and parented to other items.

You add Channel Handles to a scene by:

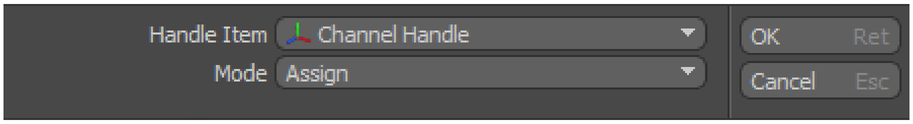
1. In the **Item List**, click **Add Item** and navigate to **Other > Channel Handle**.

   A **Channel Handle** item is created with the default shape at the scene origin (0,0,0).

> **Tip:** You can delete **Channel Handle** items from the scene by pressing the **Delete** key or using the **Delete** command in the right-click menu.

2. Position the **Channel Handle** using the standard **Transform** tools. Keep in mind that, because it is meant as a control (always active), you can't click the item to select it in the 3D viewport. You must use right-click selection lasso to select the item.
3. To assign a channel to the handle, select the target item that to be controlled by the handle.
4. In the **Properties** viewport, right-click to the left of any attribute (channel) and choose **Assign Handle**.

   The **Assign Channel Handle** dialog displays.



Handle Item: Channel Handle

Mode: Assign

5. Specify the required handle from the **Handle Item** dropdown.

> **Tip:** You might find it helpful to rename the handle, by clicking on it in the **Item List**, prior to assigning the channel.

6. Set the **Mode** to **Assign** and click **OK** to create the association.

   You can unassign a handle from a channel with the same method, but set the **Mode** to **Remove**. You may also remove channels from a handle by using the channel control's right-click menu.

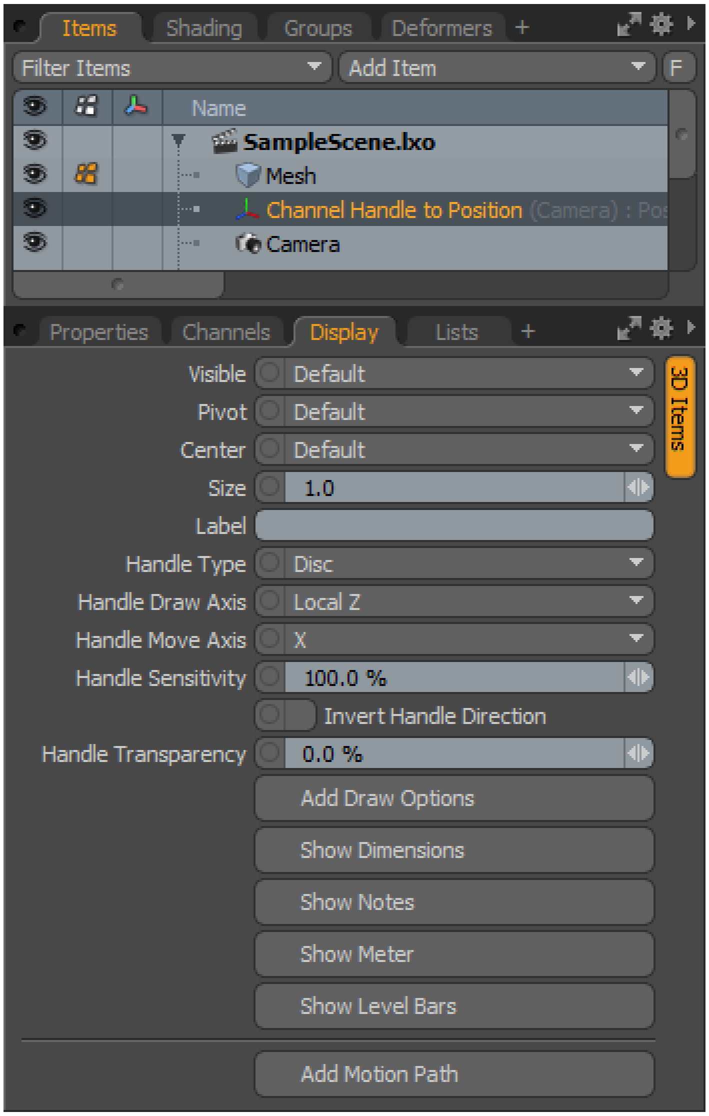
Once you have assigned a channel, you can drag over the handle's representation in the 3D viewport to adjust the numeric value of the associated channel. The handles themselves are always active in the 3D views, the channels they're linked to can be edited by dragging in the direction of their Move axis. (Other tools, such as the Transform tool, may be active at the same time).

Pressing **Ctrl** while dragging allows you to make finer adjustments by halving the amount of value change; pressing **Shift** as you drag doubles the value change. Dragging with the right mouse button shows the name and value of the first linked channel in the 3D views. If the handle is linked to more than one channel then the value that the handle is adding to each channel is shown instead.

Handle Links

When you select a channel handle by lasso selection in the 3D viewport or by clicking it in the **Item List** tab, the channels it is linked to are shown in the **Properties** viewport under the standard transform channel controls. These channels are also accessible in the 3D views by middle-clicking the handle, where Modo displays the handle's name and sensitivity control followed by controls for each of the channel it is linked to. You can easily edit the channel values and their keyframe state.

Handle Display Properties


Items
Shading
Groups
Deformers
Filter Items
Add Item
F
Name
SampleScene.lxo
Mesh
Channel Handle to Position (Camera) : Pos
Camera
Properties
Channels
Display
Lists
3D Items
Add Draw Options
Show Dimensions
Show Notes
Show Meter
Show Level Bars
Add Motion Path

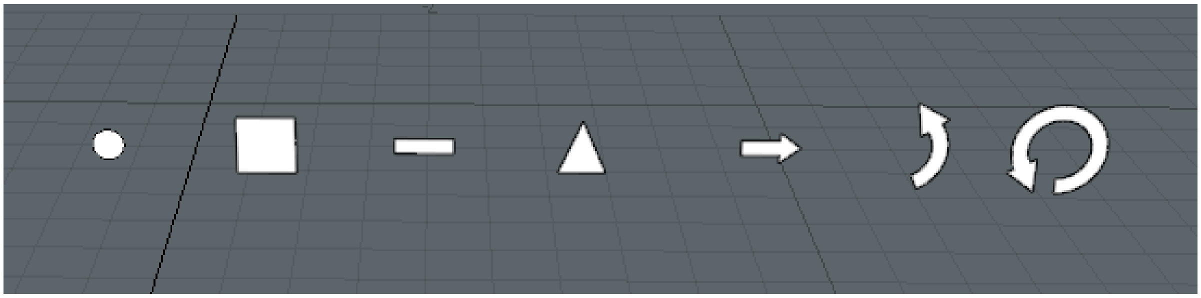
Visible: Default
Pivot: Default
Center: Default
Size: 1.0
Label:
Handle Type: Disc
Handle Draw Axis: Local Z
Handle Move Axis: X
Handle Sensitivity: 100.0 %
☐ Invert Handle Direction
Handle Transparency: 0.0 %

There are a number of controls available to customize the on-screen look of the viewport handle widget. These options are in the **Display** viewport tab when you select a channel handle item.

- **Visible** - Sets visibility for the item itself in the 3D viewport: **Default** respects viewport settings, **Yes** sets the item as visible always, and **No** disables visibility entirely.
- **Pivot** - Sets visibility for the 3D item's pivot: **Default** respects viewport settings, **Yes** sets the pivot as always visible, and **No** disables pivot visibility entirely.
- **Center** - Sets visibility for the 3D item's center: **Default** respects viewport settings, *Yes* sets the center as always visible, and **No** disables center visibility entirely.
- **Size** - Defines a multiplier that adjusts the display size of the item's representations.

- **Label** - Displays the provided text positioned next to the 3D item in the viewport for easy identification. It also adds two additional properties to control the display of the label in the viewport: **Show Label** and **Label Offset**.
- **Show Label** - Temporarily enables or disables the display of the label in the viewport without losing settings. This is only available when you specify a label.
- **Label Offset** - Specifies the distance away from the locator where the label appears. This is only available when you specify a label.
- **Handle Type** - Controls the display in the 3D viewport for the handles. You can select from different icons: **Disc**, **Square**, **Rectangle**, **Triangle**, **Arrow**, **Curved Arrow**, and **Spin**. You can use these as visual indicators of what the handle action does.
- **Handle Draw Axis** - Controls the axis orientation of the 3D viewport widget display. You can select from the three cardinal axes (XYZ) or two screen axis (vertical and horizontal).
- **Handle Move Axis** - Controls the direction the mouse should be dragged to modify the associated value. Directions are relative to the three cardinal axes (XYZ) in the scene. The thin yellow indicator line through the middle of the widget indicates this drag direction. The Z axis cannot be used for view-orientated handles.
- **Handle Sensitivity** - Adjusts how much the mouse movement while dragging on the handle affects the driven channel's values. Increasing the value results in smaller drags to create larger value changes; decreasing it means you have to drag further to make the same change.
- **Invert Handle Direction** - Dragging a handle in the positive direction of the Move axis, by default, increases the driven channel's value and dragging in the negative direction decreases it. You use this control to invert this behavior.
- **Handle Transparency** - Makes Channel Handles transparent; however, you cannot make Channel Handles completely transparent. To hide a channel handle, use the visibility options.
- **Add Draw Options** - Opens the Draw Options dialog for the **Display** viewport. You can use the **Wireframe Color** and **Fill Color** options to control the coloring of the 3D viewport widget.

The rest of the controls are identical to those found with a regular locator item and aren't relevant to the channel handle item. For details, see the Display Viewport topic.



Handle Type Samples

Using Audio

Since the release of *Steamboat Willie*, the first animated film to include synchronized sound and music, audio has been an important part of animation. For a modern CG character animation workflow, including audio is essential. Modo supports a variety of audio formats for playing directly from the Timeline. Because of this, you can synchronize animated actions with audio events easily. You can also use audio as Channel Modifiers to have the audio waveform itself modulate specific item attributes. You can load multiple audio files at a time, but Modo can only play back one file at a time. Once you import audio files, you can access them from the **Audio List** palette.

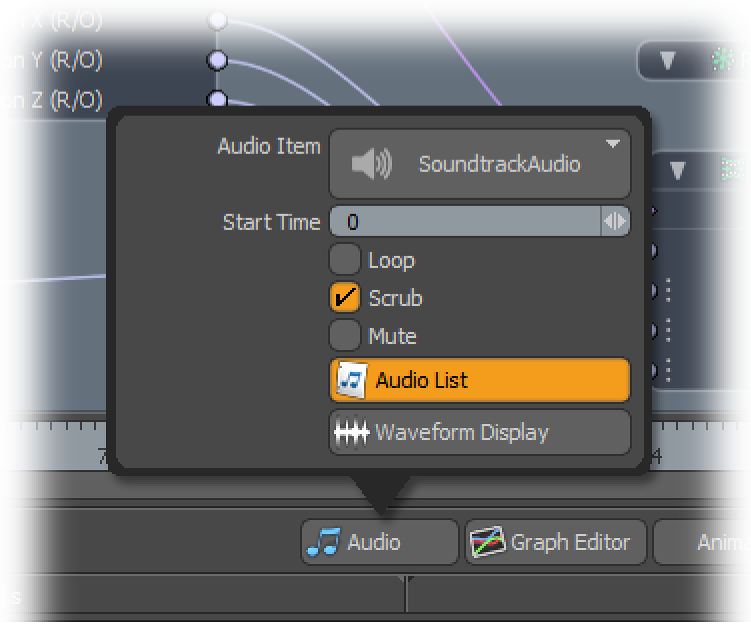
Loading Audio

You can load audio files into a scene for playback in several ways:

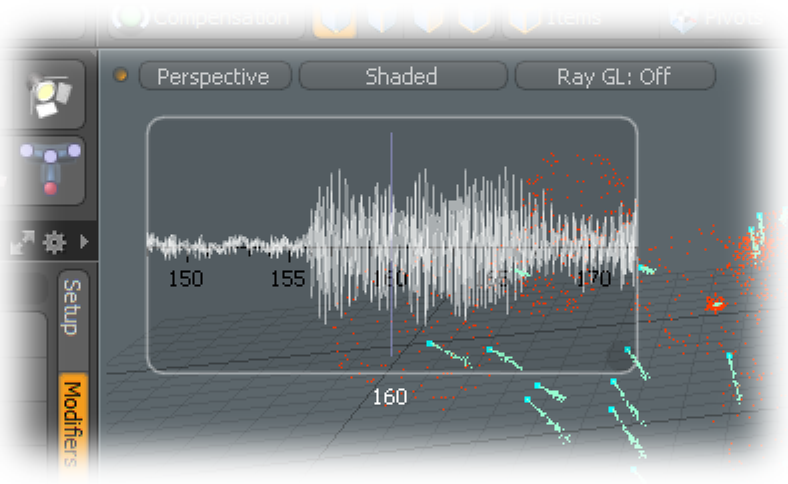
- To load audio files directly, use **File > Load Audio to Timeline**. This command loads the audio file to the **Audio List** palette and places it on the timeline. When the Timeline is played, the audio plays back in sync with the scene.
- You can also load audio into the scene by using **File > Load Audio**. This commands adds the audio to the **Audio** list where you can use it for different purposes.
- Lastly, you can go to any **Layout**, **Setup**, **Animate**, or **Render** interface and use the **Audio** button (with the musical notes icon). If you click this button, you can specify audio options in the pop-up window. (See below for details.) The pop-up window has an **Audio List** button. If you click this button, the **Audio** palette opens. With this palette, you can load audio files by clicking the **Add Audio** button and selecting specific audio items. By using this method, you can adjust the associated attributes of each audio file as it relates to the scene.

Audio Pop-up

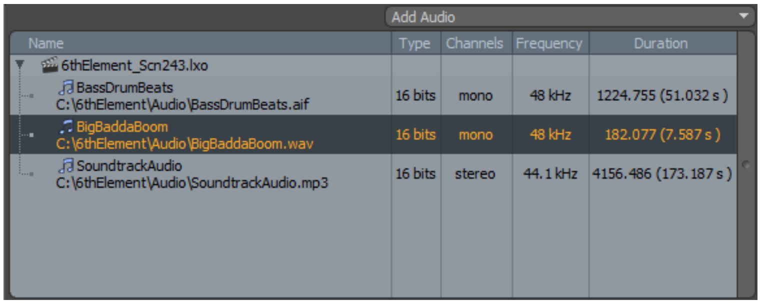
The **Audio** button is available in several default interfaces: **Layout**, **Setup**, **Animate**, and **Render**. When you click this button, the **Audio** dialog opens for you to specify an audio file (in the **Audio Item** list) to use for Timeline playback. You can also specify options for playback.



- **Audio Item** - Clicking this button opens an **Audio List** window for you to load or select a specific audio file to play back from the Timeline.
- **Start Time** - This value determines the frame on the Timeline where the audio file begins to play back.
- **Loop** - If the specified audio file is shorter than the interval determined by the scene's start and end times, then selecting **Loop** plays the audio file repeatedly until the end time.
- **Scrub** - To scrub the Timeline, you drag the **Time** slider backwards and forwards to manually control the rate of playback by the drag speed. If you select **Scrub**, the audio plays back when you scrub the Timeline. When clear, the audio only plays back after clicking the **Play** button.
- **Mute** - If you select this option, Modo suspends the audio playback until you clear the **Mute** checkbox.
- **Audio List** - Clicking this button opens the **Audio List** palette. (See Audio List.)
- **Waveform Display** - This is a toggle button that shows or hides the audio waveform overlay in the 3D GL viewport. The overlay, in a resizable transparent window, displays a 60-frame section of the waveform (30 frames on either side of the current frame) for you to see the amplitude variations in the audio file.


Perspective
Shaded
Ray GL: Off
150
155
160
170
Setup
Modifiers

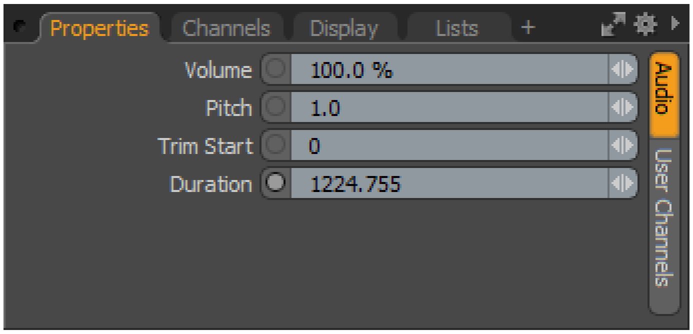
Audio List



Add Audio

Name	Type	Channels	Frequency	Duration
6thElement_Scn243.lxo				
BassDrumBeats C:\6thElement\Audio\BassDrumBeats.aif	16 bits	mono	48 kHz	1224.755 (51.032 s)
BigBaddaBoom C:\6thElement\Audio\BigBaddaBoom.wav	16 bits	mono	48 kHz	182.077 (7.587 s)
SoundtrackAudio C:\6thElement\Audio\SoundtrackAudio.mp3	16 bits	stereo	44.1 kHz	4156.486 (173.187 s)

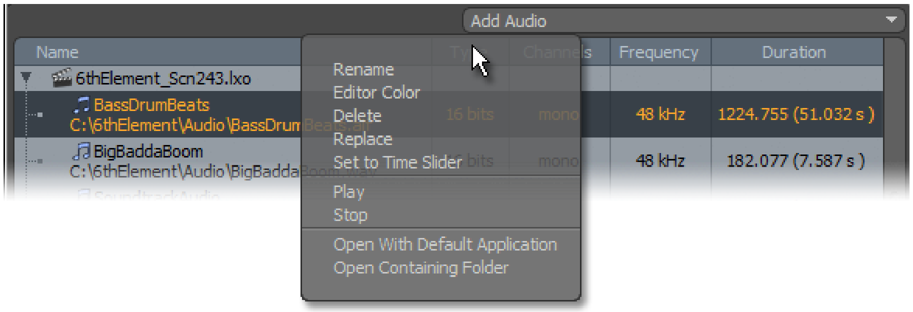
The **Audio** palette lists all audio files in a given scene. You can access any loaded audio file from here. To open the palette, use the **Layout** > **Palettes** > **Audio** command. From this palette, you can drag an audio track directly onto the Timeline. Doing this adds the audio to the Timeline, and when you click **Play**, the audio plays with the animation. If you select an audio item in this palette, the following attributes are available to control the audio playback:


Properties
Channels
Display
Lists
+
Audio
User Channels

Volume 100.0 %
Pitch 1.0
Trim Start 0
Duration 1224.755

- **Volume** - This controls the loudness of the audio playback. The default value, **100**%, plays the audio back at the recorded volume level.

- **Pitch** - Pitch describes how high or low a sound is. In Modo, the **Pitch** value controls the frequency of the audio file, which is its rate of playback. When an audio file plays faster, its pitch gets higher; at slower speeds, its pitch is lower.
- **Trim Start** - By specifying this value, you can crop the start of an audio file. Based on a number of frames, this value acts as the new starting point of the audio file when it's placed on the Timeline.
- **Duration** - This value determines how long the audio file playback is. By default, this value is the actual length of the imported clip (in frames). You can shorten the clip, which crops the end of the audio file.


Add Audio
Name
Frequency
Duration
6thElement_Scn243.lxo
BassDrumBeats
C:\6thElement\Audio\BassDrum
16 bits
mono
48 kHz
1224.755 (51.032 s)
BigBaddaBoom
C:\6thElement\Audio\BigBadda
mono
48 kHz
182.077 (7.587 s)
Rename
Editor Color
Delete
Replace
Set to Time Slider
Play
Stop
Open With Default Application
Open Containing Folder

If you right-click any **Audio** list entry, a menu offers additional audio-related commands.

- **Rename** - Use to change the name that references the file without changing the actual file name, itself.
- **Editor Color** - Use to assign a color to the audio item. This shows as an outline in the **Schematic** viewport.
- **Delete** - Removes the file from the **Audio** list and the scene. Any place where the audio file was used no longer has audio associated with it.
- **Replace** - Use to insert a new file in place of the selected file. Modo replaces all instances where the original file was used with the new file.
- **Set to Time Slider** - Adds the audio file to the Timeline for playback.
- **Play** - Plays the audio file. You can listen to the file without adding it to the Timeline.
- **Stop** - Controls the playback of the audio file.
- **Open with Default Application** - Automatically opens the audio file in your system's default sound editor for you to edit the audio file.
- **Open Containing Folder** - Opens the folder that contains the audio file in a system window.

Tip: Modo does not embed the audio file, itself, in the scene but only links to it. Therefore, the audio file needs to remain in its position relative to the animation file; otherwise, you may need to re-import the audio file.

Supported Audio Formats

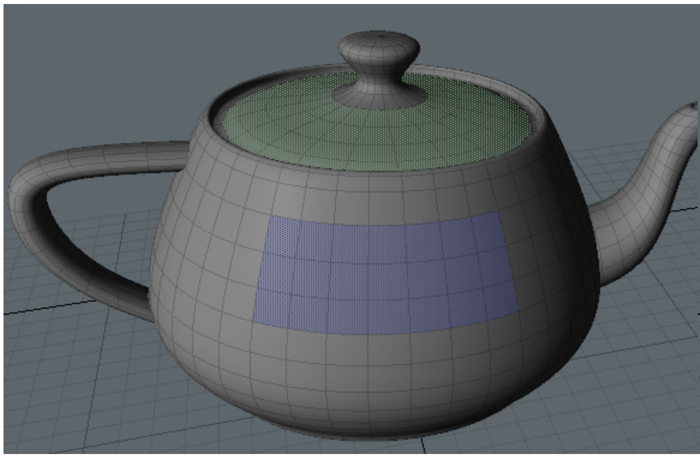
Format Name	**Extension**	**Note**
Waveform Audio File	**.wav**	
Audio Interchange File Format (AIFF)	**.aif**	
MPEG-2 Audio Layer III	**.mp3**	
Core Audio Format	**.caf**	Mac only
MPEG-4	**.mp4**	Mac only
Dolby Digital	**.ac3**	Mac only
Windows Media Audio	**.wma**	Windows only
Audio File Format	**.au**	Windows only
Digital Sound File Format	**.snd**	Windows only

Note: Modo only exports Timeline audio when saving to a QuickTime video file format.

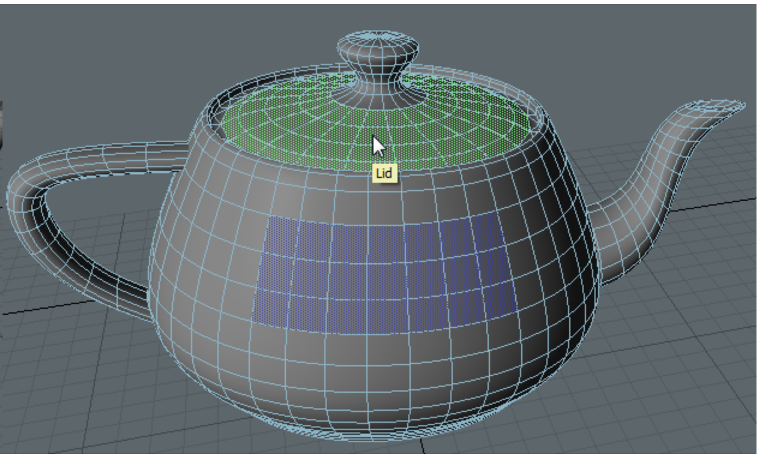
Command Regions

Command regions allow you to execute commands or scripts and to select other items in the scene by clicking on areas of a mesh. Command regions are defined from a selection of polygons on a Mesh item and can be assigned a color to help identify them in the 3D views.

Command regions are always visible in **Polygon Selection** mode, but in **Item Selection** mode, they are only drawn when the pointer moves over the mesh they belong to. Holding the pointer over a region in **Item Selection** mode displays the command region's name in a tooltip.



Command regions in Polygon Selection mode.



Command regions in Item Selection mode.

Assigning Command Regions

The first step is to select the polygons to include in the region in **Polygon Selection** mode:

1. Make sure you're in the **Setup** layout, so that the **Controls** tab is displayed on the left of the viewport.
2. Enable Polygon Selection mode in the viewport by navigating to **Select** > **Polygon** in the menu bar, clicking the Polygon Selection mode icon, or pressing **3** on your keyboard.
3. Select the polygons you intend to assign as a command region.

Tip: Don't worry if you miss a few polygons, you can add them to the command region once it has been created. See Editing Command Regions for more information.

4. In the **Controls** tab, under the **Command Regions** menu, click **Assign Command Region**.

   The **Assign Command Region** dialog displays.
5. Enter the required information into the dialog:

Note: Although **Item to Select** and **Command to Execute** are optional, you must use one or the other to successfully create a Command Region.

- **Name** - Enter a descriptive name for the new command region. The name must be unique on the Mesh Item that the polygons belong to. A dropdown to the right of the Name field lists all the names currently assigned to command regions on the mesh - if a name is chosen from this menu, you must edit it so that it is unique.

- **Item to Select** - An optional item to select can be chosen from the dropdown. This item is then selected before any command or script is executed. There are a few restrictions on which items can be selected by command regions:
    - The item must have a unique name. The automatic names given to items by Modo are not allowed, including numeric suffixes added by Modo.
    - The Mesh Item the polygons belong to cannot be used. This item may be displayed in the dropdown menu as it's impractical to exclude it, but an error is displayed if it is selected.
- **Command to Execute** - Specifies an optional command or script. The dropdown to the right of the field lists a few common commands followed by all the commands currently assigned to command regions on the mesh.

Tip: To run a script, rather than a command, prefix the script name with @.

- **Drawing and Rollover Color** - Select a color to draw the command region and its rollover highlight. Clicking on the color control opens a color picker.

6. Click **OK** to assign the selected polygons to the command region.

   A confirmation dialog displays.
7. Take a moment to confirm your choices and click **Yes** to create the Command Region.

   The Command Region is created.

Assigning Command Regions in the Schematic Viewport

If you prefer to work with the schematic viewport in the **Setup** layout, you can also assign Command Regions with the following method:

1. Make sure you're in the **Setup** layout, so that the **Controls** tab is displayed on the left of the viewport.
2. Enable **Polygon Selection** mode in the viewport by navigating to **Select** > **Polygon** in the menu bar, clicking the **Polygon Selection** mode icon, or pressing **3** on your keyboard.
3. On the schematic viewport, click **Add** and search for **Polygon Command Region**.
4. Double-click **Polygon Command Region** to add it to the schematic viewport.
5. Select the polygons on the mesh you intend to assign as a command region.
6. In the **Controls** tab, click **Add Polygons** under **Nodal Command Regions**.

   The polygons are added to the Command Region.

Note: Double-click the yellow diamond on the **Geometry** input of the **Polygon Command Region** node to expose the input mesh data.

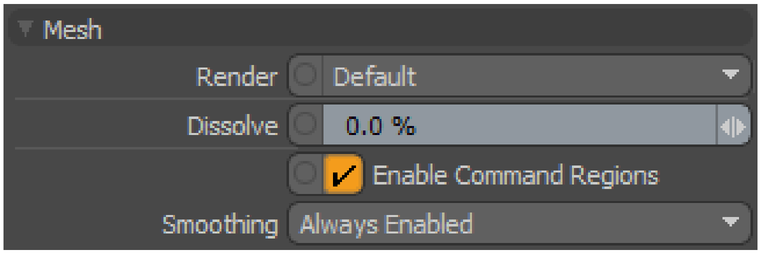
Note: Enable **Command Region** in the **Controls** tab when you want to see Command Regions on your meshes and execute commands with them.

Tip: If you want to change the color of the command region, click the **Polygon Command Region** in the schematic viewport and change the color with the **Color** option in the **Properties**.

Tip: You can also assign edges and vertices to Command Regions.

Disabling Command Regions on a Mesh

Command regions on a mesh are enabled by default, but you can disable the command region by disabling the **Properties** > **Mesh** > **Enable Command Regions** control.



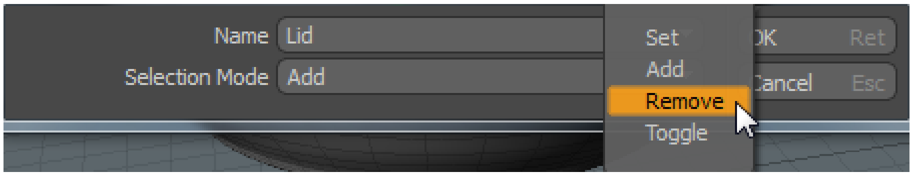
Mesh
Render: Default
Dissolve: 0.0 %
☑ Enable Command Regions
Smoothing: Always Enabled

Editing Command Regions

Once you have assigned one or more command regions, you can edit them using the controls in the **Command Regions** tab to the left of the viewport.

Selecting Regions

You can select regions by clicking the colored region in the viewport, or you can use the **Select by Name** button in the **Controls** tab on the **Setup** layout to automatically select, deselect, or invert a selection within a region.



Name: Lid
Selection Mode: Add

The buttons under **Select by Name** provide the same options:

Icon	Name	Description
	Select Set	Selects all polygons belonging to command regions that have polygons in the current selection. Those that were not assigned to a command region are dropped from the selection.
	Select Add	Selects all polygons belonging to command regions that have polygons in the current selection, keeping any other polygon selections.
	Select Remove	Removes all polygons from the selection that are assigned to command regions.
	Select Toggle/Invert	Selects all polygons belonging to command regions that have polygons in the current selection, dropping the original selection. Selected polygons which do not belong to a command region are not affected.

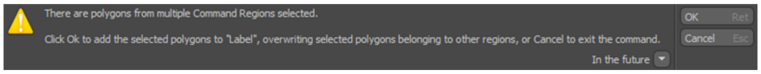
Adding Polygons to Command Regions

Polygons can be added to existing command regions or reassigned to a different region by clicking **Add Polygons**.

1. Select the polygons that you want to add or reassign along with one or more polygons belonging to the target command region.
2. Select the command region in the schematic viewport where you want to add the polygons.
3. Click **Add Polygons** under **Nodal Command Regions** in the **Control** tab.

   The new polygons are added to the target region.

   If there are polygons in the selection belonging to multiple regions, the polygons are added to the first region selected and a dialog displays, asking if the other polygons should be overwritten.



There are polygons from multiple Command Regions selected.

Click Ok to add the selected polygons to "Label", overwriting selected polygons belonging to other regions, or Cancel to exit the command.

4. Click **OK** to proceed or **Cancel** to abort the operation.

Removing Polygons from Command Regions

Polygons can be removed from a region by manual selection or by region name:

- Select the required polygons from all regions and click **Clear from Selection**, or
- Click **Clear by Name** and select the region name to remove.

Note: Clear by Name removes the entire region, rather than a subset of polygons from within the region.

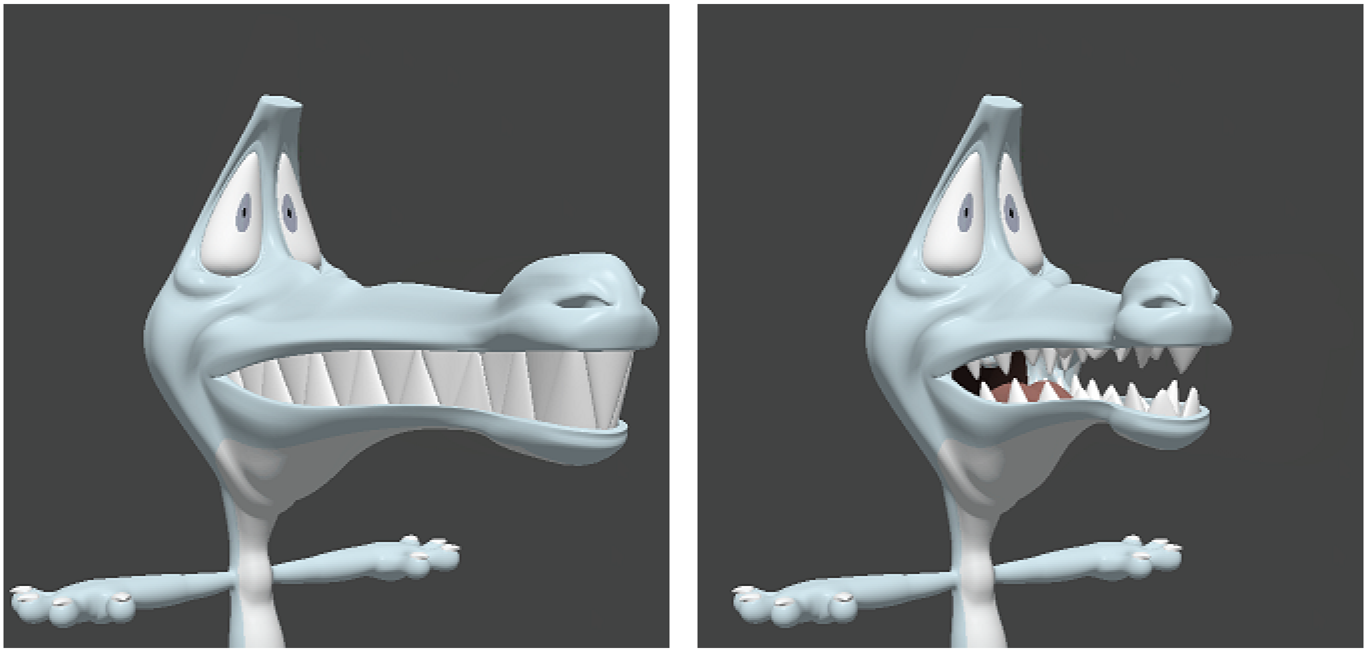
Renaming Command Regions

You can rename a command region by right-clicking the command region in the schematic viewport, and selecting **Rename**. Provide a new name on the command region and press **Return** / **Enter**.

Assigning Mouse Gesture Commands to Command Regions using Rig Clay

Once you've identified the faces on a mesh where you'd like to assign a Command Region, you can assign a command region gesture to the command region. A gesture allows you to define mouse movements so that when you select a command region and perform that gesture, it executes an action on any mesh property, such as the position, rotation, or scale.

For example, you could create a gesture that specifies when you click and drag your mouse in an upward motion on a command region, it moves specific faces on a mesh upward. This allows you to create complex and intuitive rigs for posing and animating characters and faces without the need for channel hauls and other controls taking up real estate in your viewport.



A character posed using mouse gestures and command regions on the head, nose, and teeth.

Model courtesy of William Vaughn

To assign a gesture to a command region:

1. Create a command region on your mesh.

   See Assigning Command Regions to learn how to add a command region to a mesh.
2. In the **Setup** layout, click **Add** on the schematic viewport.
3. Search for **Command Region Gesture**.
4. Double-click **Command Region Gesture** to add it to your schematic viewport.
5. Drag the output of your polygon command region and input it to the **Command Region** input on the command region gesture.

   Modo creates a relationship between the command region and the command region gesture.

Note: If you can't see your command regions when you roll your mouse over, make sure **Show Selection Rollovers** is enabled in the **Visibility** tab of the **3D Viewport Properties**, and that **Command Regions** is active in the **Controls** tab of the **Setup** layout.

Tip: You can add multiple Command Region Gestures to a single Command Region. This allows you to define multiple gestures on one Command Region to execute multiple actions.

Defining Mouse Gestures

Once you have assigned a gesture to a Command Region, you can specify what gesture is performed on the Command Region in order to trigger an event or action.

With the Command Region Gesture selected, the **Properties** allow you to define the behavior of the mouse gesture.

Option	Description
Enable	Enables or disables the gesture.
Tooltip	Allows you to enter a tooltip that appears when the mouse hovers over the Command Region. If you have several Command Regions on a mesh, the tooltip can be used to describe what the Command Region controls.
Button	The mouse button which performs the action when pressed or held, either the **Left**, **Middle**, or **Right**.
Mode	Defines what kind of mouse gesture performs the action when the **Button** is pressed or held down: • **Click** - when the button defined in **Button** is clicked, the action is performed. • **Drag** - when the mouse is dragged with the button defined in **Button** held down, the action is performed. When **Drag** is selected, the **Direction** option becomes available.
Direction	Defines the direction of the mouse drag when **Drag** is selected in **Mode**. The options are: • **Horizontal** - when the mouse is dragged horizontally, the action is performed. • **Vertical** - when the mouse is dragged vertically, the action is performed.
Invert	When enabled, inverts the direction of the mouse drags. For example, by default, moving the mouse from left to right when **Direction** is set to **Horizontal** creates a positive output. When inverted, moving the mouse from right to left creates a positive output.
Sensitivity	This setting is used to control how far the mouse needs to move for an output amount on an action. A higher sensitivity results in higher output amounts.
Control Multiplier	Adjusts the sensitivity of the mouse gesture when the **Ctrl**/**Control** key is held down. For example, a **Control Multiplier** of 50% means when the **Ctrl**/**Control** key is held down, the sensitivity becomes 50% of what is currently defined in **Sensitivity**.

	Note: You must hold the modifier key down before you begin the gesture on the Command Region.
Shift Multiplier	Adjusts the sensitivity of the mouse gesture when the **Shift** key is held down. For example, a **Shift Multiplier** of 200% means that when the **Shift** key is held down, the sensitivity becomes 200% of what is currently defined in **Sensitivity**.

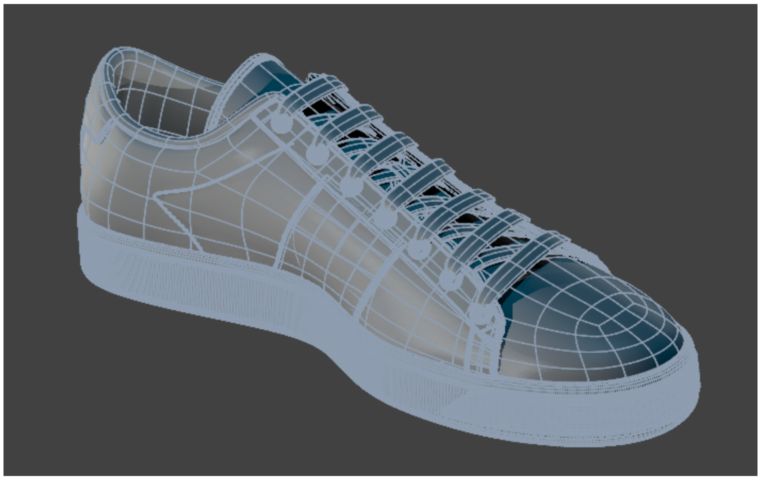
Defining Command Region Actions

You can add an action to a mouse gesture to trigger an event or action on the Command Region. You can add commands, Channels, and Channel Sets as items to be modified by the gesture. See Command History, Channels Viewport and Channel Sets - used for collections of channels that can be viewed and modified directly in the 3D viewport as an overlay. to learn more.

The following example walks you through how to add an action that moves your mesh along the Y-axis when a mouse gesture is performed.

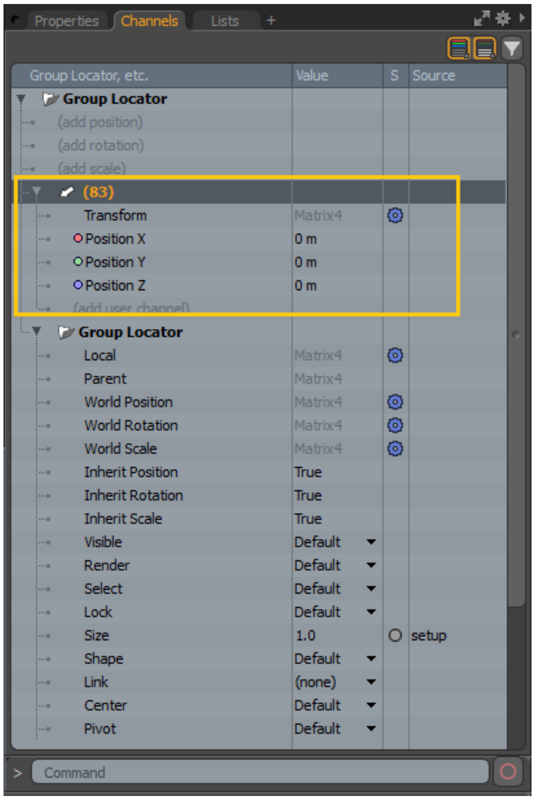
Note: Make sure you have a Command Region and Command Region Gesture set up for your mesh. See Assigning Command Regions and Defining Mouse Gestures to learn how.

1. Select the mesh that you would like to affect in the **Items** list.



Multiple meshes selected in the viewport with a Group Locator.

2. Click the **Channels** tab in the lower-right of the layout.
3. Click **(add position)** under the mesh name to add position channels to the mesh.


Properties
Channels
Lists
+
Command

Group Locator, etc.	Value	S	Source
Group Locator			
(add position)			
(add rotation)			
(add scale)			
(83)			
Transform	Matrix4	⚙	
Position X	0 m		
Position Y	0 m		
Position Z	0 m		
(add user channel)			
Group Locator			
Local	Matrix4	⚙	
Parent	Matrix4		
World Position	Matrix4	⚙	
World Rotation	Matrix4	⚙	
World Scale	Matrix4	⚙	
Inherit Position	True		
Inherit Rotation	True		
Inherit Scale	True		
Visible	Default		
Render	Default		
Select	Default		
Lock	Default		
Size	1.0	○	setup
Shape	Default		
Link	(none)		
Center	Default		
Pivot	Default		

The **Channels** tab for the selected Group Locator showing the **Position X**, **Position Y**, and **Position Z** channels.

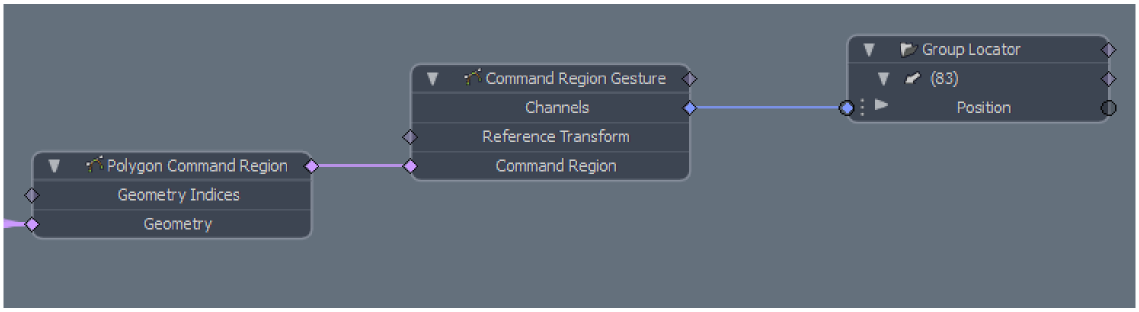
4. Hold **Shift** and select **Position X**, **Position Y**, and **Position Z** and drag the selection onto the schematic viewport.

   The position channels are added to the selected mesh node on the Schematic Viewport.

5. Hover over the **Position Y** property on the mesh item node, right-click and select **Separate Channel**.

   The **Position Y** property is separated from the original mesh item node and is now available to be plugged into other nodes.

6. Drag the **Channels** output on the **Command Region Gesture** to the **Position Y** input.


Group Locator
(83)
Position
Command Region Gesture
Channels
Reference Transform
Command Region
Polygon Command Region
Geometry Indices
Geometry

7. Activate **Command Region** in the **Control** tab on the Setup layout.

Click the yellow command region to move the mesh along the Y-axis.

Hovering over your command region and performing your defined gesture now moves the mesh item up and down along the Y-axis.

Using Mesh Operation Nodes with Command Regions

From **Modo 15.0** you can set command regions on procedural meshes, making it easier to perform gestural actions to make changes quickly. The following example guides you through the minimum required nodes to establish a command region on a procedural mesh.

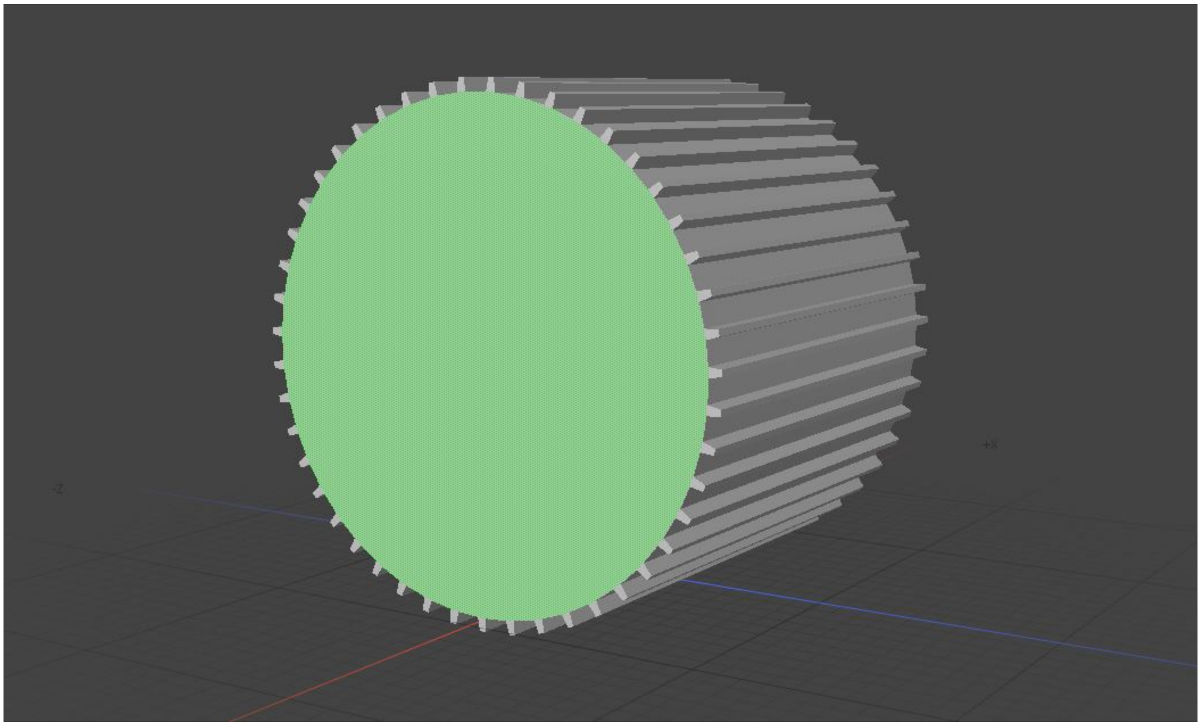
1. Set Modo's layout to **Setup**.
2. In the **Mesh Ops** tab, click **Add Operator** and add a procedural mesh to your item list. For example, a cylinder. For information on adding and working with procedural meshes, go to Procedural Modeling with MeshOps.
3. In the schematic workspace, click **Add...**

   The **Add to Schematic** window opens.
4. Search for a **Command Region** mesh operation, for example **Command Region: Polygon** and double click to add the node to the schematic workspace. This node is used to set the command region to a selected polygon or group of polygons on a procedural mesh. You can also set edges and vertices as command regions to procedural meshes.

5. Click **Add...** and add a **Select by Index** mesh operation node to the schematic workspace.
6. Click on the **Select by Index** node and go to its **Properties** on the right hand side. Go to **Selection** and type the index number of the polygon or range of polygons you want to use as the command region or use Polygon Selection mode to select polygons.

> **Note:** The number you enter in the **Selection** will depend on the number of polygons in your procedural mesh. For example, on a simple cylinder mesh, the front facing polygon might have the index number 25, but you may want the polygons in the range 21-24 as your command region. You can always change this later if you prefer.

7. Click on the **Command Region** node and go to its **Properties** on the right hand side. Adjust a color channel so that your command region is more visible. For example, to set this command region as green, adjust the values of just the green channel.
8. Link the **Select by Index** node to the **Selection** attribute on the **Command Region** node in the schematic.

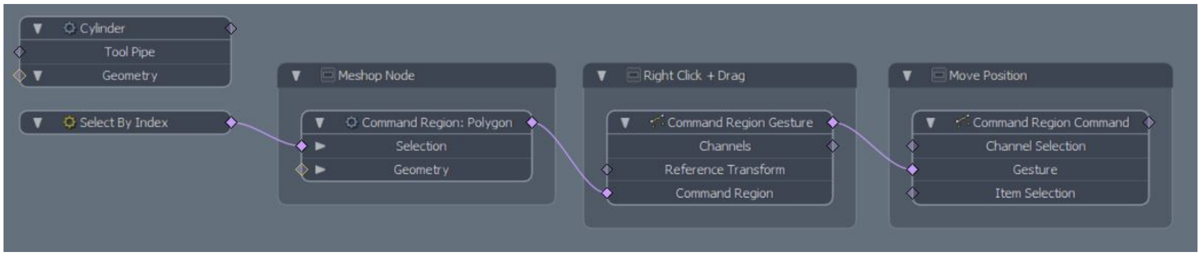


The command region on this procedural mesh is set to a single polygon and the color set to green.

9. In the **Control** tab on the left hand side of the **Setup** layout, make sure the **Command Region** button is active.

   In **Item Selection** mode, each time you hover the cursor over your command region, it highlights the color set in the **Properties** of the **Command Region** node.

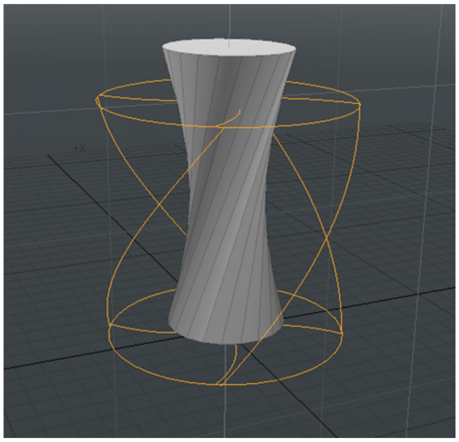
10. Now you can add a **Command Region Gesture** node and link it to the **Command Region** node to assign a mouse gesture to your command region. See Assigning Mouse Gesture Commands to Command Regions using Rig Clay for more information.


Cylinder
Tool Pipe
Geometry
Select By Index
Meshop Node
Command Region: Polygon
Selection
Geometry
Right Click + Drag
Command Region Gesture
Channels
Reference Transform
Command Region
Move Position
Command Region Command
Channel Selection
Gesture
Item Selection

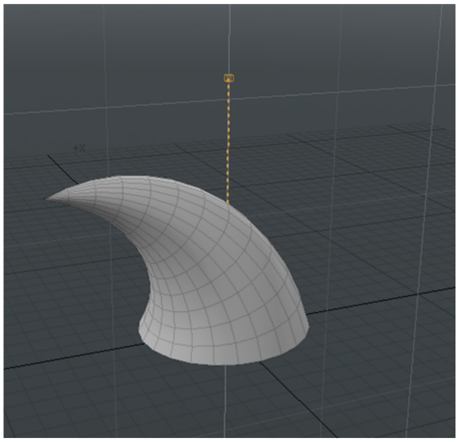
At a minimum, your schematic would look like the above example. Backdrop nodes have been added to label each node in this case.

Applying Deformers

Modo has a powerful set of deformers for you to twist, bend, and otherwise distort your creations in many ways and to provide simplicity and power in an easy-to-use workflow.



A Vortex effector applied to a cylinder.



A Bend effector applied to a cone.

You can fully animate each deformer, so deformations can happen over time. You can combine and stack these flexible deformers in the **Deformers** viewport and control them by using deformer falloffs.

In Modo, deformers have two basic parts: an effector and an influence. These two parts collectively are considered deformers. Effectors provide instructions on how to affect the Target, and the influence determines what gets affected. Having these two parts provides flexibility.

For example, say a cartoon car needs to bend as it goes around a tight corner. The car has multiple parts, which include its four wheels. You can apply a Bend effector to the body to distort it as it goes through the curve, but the wheels, if included in the deformation, produce undesirable results. To keep the wheels connected to the car as it bends, you could add an additional influence and then connect it to the same Bend deformer. In this case, have it change the items' positions and rotation but not actually deform the geometry itself. This keeps the wheels in proper position relative to the bending car and all items use a single simple-to-control effector. You could attach additional deformers to the wheels to further fine-tune their local deformations while under the influence of the prior effector.

As indicated in the example, each deformation is layered upon the previous to produce an easy-to-understand workflow that provides an incredible amount of depth with Modo's ease of use.

Modo offers a variety of options for working with deformers. If you are just starting out with deformers, follow the steps outlined below to learn how to apply them for specific situations. Deformers in Modo have multiple pieces that need to be connected together (usually in the **Schematic** viewport). By selecting a Target mesh and applying the appropriate commands, Modo automatically makes these connections for you. If you are a more advanced user, you can add the effectors and influences separately and then connect them manually in the **Schematic** viewport. By combining deformers with modifiers, Modo provides a powerful rigging toolbox that allows the whole to be much greater than the sum of its parts.

Applying Deformers

Modo has some procedural deformers (such as the Bend, Magnet and Vortex effectors), which you can animate over time and work with in a manner similar to Modo's modeling tools. For example, to animate a car's antenna whipping in the wind, you could use the Bend deformer. (If you combine this with a modifier that modulates the bend amount, you could create a simple animation with very few keyframes.)

For the procedural deformers, the position of the effector defines the origin of its influence over the Target, so animating its position changes how it affects what it's deforming. By selecting the effector, you can edit its attributes in the **Properties** panel. By selecting the influence, you can adjust which parts of the mesh get influenced.

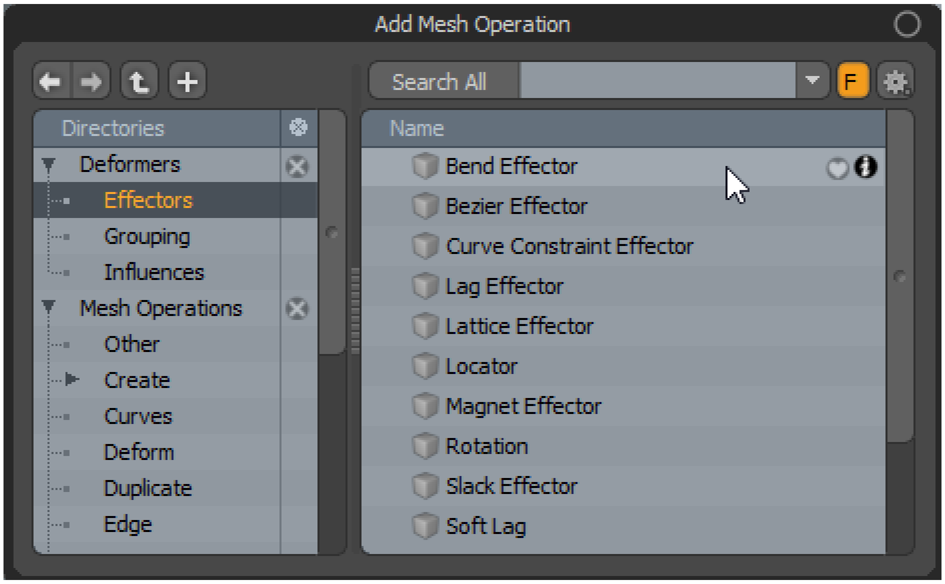
Modo also has reactive deformers (such as the Slack or Lag deformer), which are dependent on the object's keyframed motions. These deformers create deformations based on the positions or motions of an item and can apply secondary motions to an animation. Deformers, such as the Rotation and Transform effectors, provide you with a riggable way to connect items in a controlled manner.

When you assign an effector, you also assign a General Influence. The effector determines how to deform the Target, and the influence specifies what to affect (by default, the entire mesh). Once you have applied the deformer, you can select the effector in the **Items** list and modify its attributes in the **Properties** panel to deform the Target as needed.

Applying Deformers Using the Item List

1. Right-click on the item in the Items (Scenes) List and select **Add Mesh Operation**.

   This opens the Preset Browser.


Add Mesh Operation
Search All
F
Directories
Deformers
Effectors
Grouping
Influences
Mesh Operations
Other
Create
Curves
Deform
Duplicate
Edge
Name
Bend Effector
Bezier Effector
Curve Constraint Effector
Lag Effector
Lattice Effector
Locator
Magnet Effector
Rotation
Slack Effector
Soft Lag

   The different types of deformers can be found under the **Deformers** directory.

2. Double-click on a deformer name to apply it to the item.

Applying the Deformers using the Toolboxes

You can also assign a deformer by selecting the item and then clicking the associated deformer button. Deformers can be found in the following locations:

Note: Certain modes offer a subset of deformers according to the context.

- **Deformer** sub-tab in the **Commands** tab in **Setup** mode.
- **Deform** sub-tab in **Model** mode.

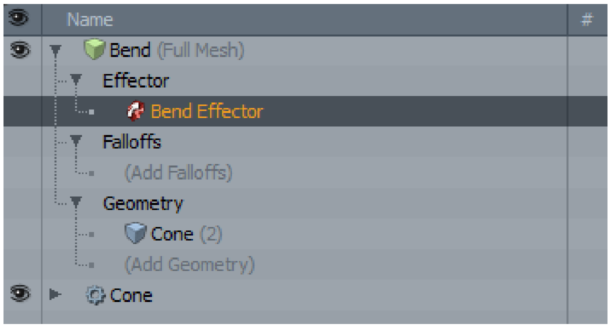
Applying Deformers Using the Mesh Operations List

Deformers are tightly integrated into the procedural modelling workflow. Deformers and mesh operations are managed from the same list, and can be mixed together, to apply deformers before or after mesh operations and vice versa. The following steps assume you have a mesh in the list that you wish to deform.

1. Click **Add Operator** in the Mesh Operations List to open the Preset Browser.

2. Click a category under **Deformers** to view the options.
3. Double-click on a deformer to add it to the list.

The deformer is added as a General Influence with the deformer defined under the **Effector** branch.

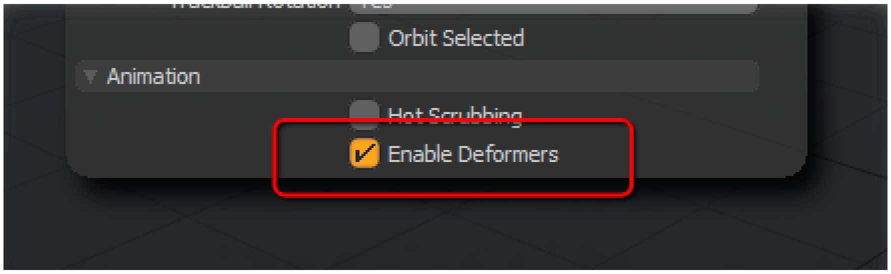


Select the effector to edit the properties in the **Properties** panel. The procedural item has the following sub-tools:

- **Falloffs** - You can add one or more Using Falloffs to the deformer. Click **(Add Falloffs)** to open the Preset Browser and select a falloff.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

Viewing Deformers in 3D Viewports

Because the modeling and animation workflows are significantly different, Modo provides visibility options on the 3D Viewports that are set appropriately for each workflow. When modeling, you most likely want to see the mesh in its relaxed, un-deformed state. Therefore, in the **Model**, **Model Quad**, **Paint**, and **UV** tabs, Modo has the **Enable Deformers** checkbox clear (disabled) by default. If you are working with one of these tabs when you add a deformer, you do not see any effect. The **Animate** and **Render** tabs have this option selected (enabled) by default. To change the state, press the **O** key with pointer over a 3D viewport. Alternatively, at the bottom of the **Viewport Properties** dialog, click the **Enable Deformers** checkbox to enable or disable this option.


☐ Orbit Selected

Animation

☐ Hot Scrubbing

☑ Enable Deformers

Applying Deformers

With Modo, you mainly deform items by applying effectors that provide instructions on how to deform a target and influences that determine what to deform. You apply deformers to items by selecting the target item and then clicking the deformer in the toolbox, which automatically links the elements together.

You may prefer to manually add elements in the **Deformers** viewport and connect them through the **Schematic** view.

Click any link below to view the topic for a particular Deformer.

Deformers		
	Bend Effector	The **Bend Effector**, when assigned to an item, deforms the object in a curving or arcing fashion.
	Bezier Effector	The **Bezier Deformer** is an excellent way to apply deformations in a continuously smooth manner. This can be extremely useful for long tube-like structures (such as tentacles, tails, and whips).
	Curve Constraint Effector	The **Curve Constraint Effector** attracts nearby vertices, constraining them to the nearest position along the defined curve.
	Lag Effector	The **Lag Effector**, when assigned to an item, deforms the object along a directional vector that is determined by the target item's animation. (Essentially, it is deformed along the direction it is traveling.)
	Lattice Effector	The **Lattice Effector**, when assigned, creates a structured array of points across the mesh like a cage. You can move each individual vertex of this cage to deform the underlying mesh. Modo smoothly interpolates the deformation between the vertex locations.
	Magnet Effector	The **Magnet Effector**, when assigned to an item, draws points within its defined range toward the center of the magnet, itself.
	Rotation Effector	The **Rotation Effector**, when assigned to an item, rotates the target item. You can link it to rigging and can control it through the **Schematic** view links when combined with the various modifiers because it is an item with channels.

Deformers		
	Slack Effector	The **Slack Effector**, when assigned to an item, deforms the target as if it were suspended between two points.
	Spline Effector	Use the **Spline Deformer** to apply deformations in a continuously smooth manner. This can be extremely useful for long tube-like structures (such as tentacles, tails, and whips).
	Transform Effector	The **Transform Effector**, when assigned to an item, moves, rotates, or scales the target item. You can link it to rigging and can control it through the **Schematic** view links when combined with the various modifiers because it is an item with channels.
	Vortex Effector	The **Vortex Effector**, when assigned to an item, deforms the object in twisting motions. It has **Decay** options to provide a spinning-tornado type of deformation to the target item.

Influences		
	General Influence	With the **General Influence** you can assign any vertex tagging methods to define a grouping of vertices for the deformer to influence. This limits their application to specific user-defined areas of a mesh (or the entire mesh).
	Item Influence	Use an **Item Influence** when a rigid item needs to stay in the proper position relative to a deforming surface or when multiple items need to transform in a uniform way (such as a tower of spheres that bends around on itself).
MDD	MDD Influence	Daisuke Ino developed MDD files during his work on a dynamics engine built as a plug-in to a host 3D application. These files are a simple point cache that stores a location in space for every vertex in a model for every frame in the animation.
	Morph Influence	With morph maps you can save alternate user-defined deformations for any mesh. To animate any morph map, you must first apply a morph deformer to the model.
	Push Influence	The Push Influence deforms geometry by pushing vertices along their average normals.
	Wrap Influence	With the Wrap Influence, you can create a custom, low-resolution cage as a proxy to deform the associated mesh. Typically, the cage approximates the shape of the target mesh layer to make it easier to manipulate and control.

Bezier Effector

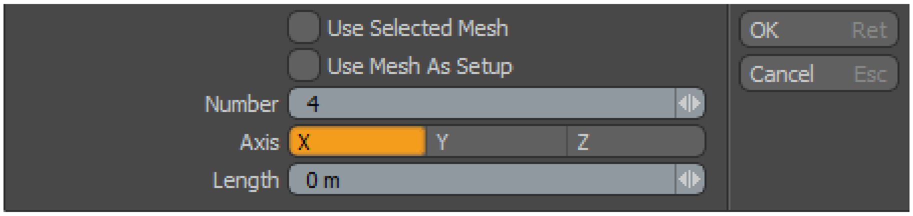
The **Bezier Effector** is an excellent way to apply deformations in a continuously smooth manner. Use **Bezier Effectors** for long tube-like structures (such as tentacles, tails, and whips). The **Bezier Effector** is functionally identical to the **Spline Effector**, but uses Bezier curve handles, which are somewhat easier to manipulate. You use the curve handles to control the line interpolation between anchor positions with additional handles. Doing this can provide for a more intuitive workflow.

The **Bezier Effector** works as a group of Locators that define a Bezier-style curve through their collective locations. Their initial positions (defined as the **Setup Position**) determine the undeformed state of the target mesh, and as Modo moves each Locator, it calculates the difference from the unmodified curve and applies it to the mesh as the deformation. (This is why Curve Deformers always appear to have two curves associated with them.) As with many functions of Modo, there are several ways for you assign and apply a Bezier curve for deformations.

No Initial Curve

You don't need to create a Bezier curve prior to applying a **Bezier Effector**. Modo opens the **Create Spline** dialog box whenever you apply a **Bezier Effector** to a Mesh Item layer. The resulting curve aligns along one of the three axes in a straight line and centers on the specified axis. You either model the target geometry to match this requirement or enter **Setup Mode** after the fact and reposition the generated curve to coincide with appropriate locations relative to the geometry it's meant to deform.

To apply the Deformer itself, you first select the target Mesh Item layer either in the **Items** list or in the 3D viewport. With the target item selected, you then either click the **Bezier** button in the **Deformers** sub-tab in the **Setup** interface layout, or you right-click the target layer in the **Items** list and choose choose **Item Operations** > **Add Mesh Operation**, then under **Effectors**, double-click **Bezier Effector**.


☐ Use Selected Mesh
☐ Use Mesh As Setup
Number 4
Axis X Y Z
Length 0 m

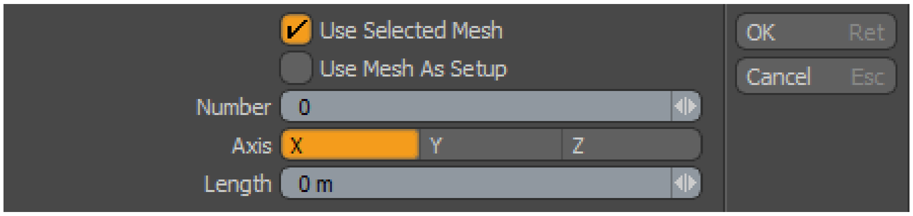
In the **Create Spline** dialog box, you only need to modify three values: **Number**, **Axis**, and **Length**. **Number** indicates the number of **Bezier Node** control points along the length of the curve. More control points provide finer control over the deformation, but they are also more difficult to deal with. Because the deformation smoothly attenuates between each location, you should define the minimum number of points necessary to obtain the desired effect. The **Axis** value indicates to Modo which axis it should use to generate the resulting spline. **Length** determines overall length of the resulting curve when centered at the World origin. Modo places the base of the generated curve on the negative side of the specified axis and draws it towards the positive direction. By clicking **OK** or pressing **Enter**, you save the values, and Modo generates the **Bezier Effector** and adds a number of **Bezier Node** items into the **Item List** as indicate with the **Number** value. These are the items that Modo manipulates in the scene to deform the target geometry.

If you need to make modifications, press the **Setup** button in the **Setup** interface layout to enter the **Setup** action state. In this state, you can modify the curve relative to the target geometry without generating any deformations. (This also temporarily disables any position changes applied to the curve outside of **Setup**.) When you properly position the curve, you can exit the **Setup** action state and change the positions of any of the **Bezier Node** items to deform the target mesh. To do this, select the **Bezier Node** item, itself, and move it with the transform tools. Set keyframes as necessary for a desired animation. To change the incoming and outgoing curve shape more intuitively, left-click the extended control handles on the Bezier node item.

User Created Curve

To create a custom Bezier curve to use as a Deformation controller, generate the curve in the same **Item** layer as the target mesh it's meant to deform. (You can also do this by cutting and pasting prior to invoking the add **Bezier** deformer command.) You can draw the spline with the Bezier tool.

Once you create the Bezier curve, you add it as a deformer to the mesh. To do so, you must be in the **Polygons** component mode with the target curve as the only curve selected. With the target Bezier curve selected, you can then either click the **Bezier** interface button in the **Deformers** sub-tab in the **Setup** interface layout, or you can right-click the target layer in the **Items** list and choose **Add Mesh Operation**, then under **Effectors**, double-click **Bezier Effector**.


☑ Use Selected Mesh
☐ Use Mesh As Setup
Number 0
Axis X Y Z
Length 0 m

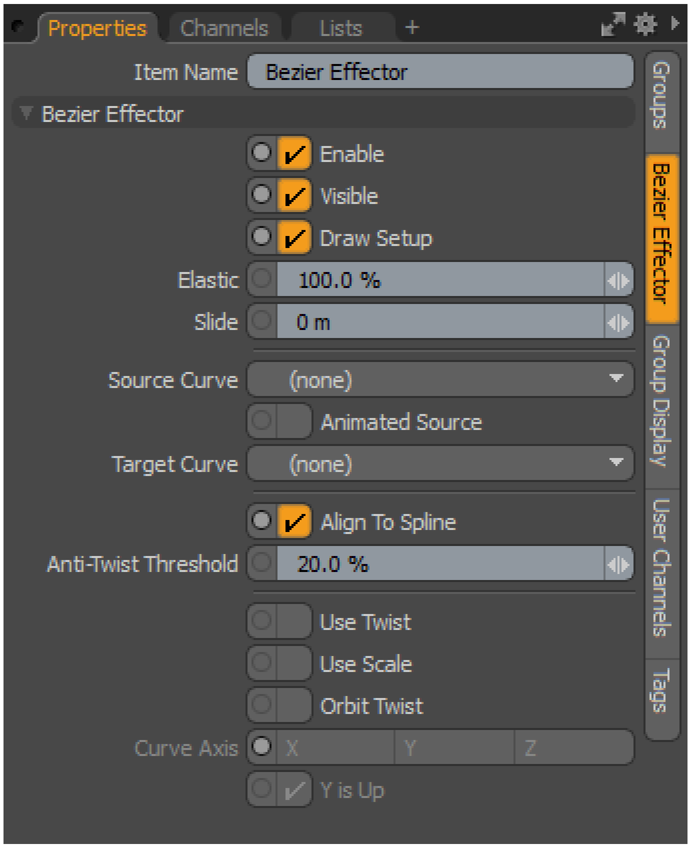
For a user-created curve, you only need to select **Use Selected Mesh** to have the tool use the selected curve. Everything else keys off the curve itself.

If you enable **Use Mesh As Setup**, Modo sets up the Bezier deformer so that it uses a curve polygon in the mesh to define the setup position for the deformer, then creates locators in appropriate places for the animated deformation.

After you click **OK** or press **Enter**, Modo accepts the values and generates the Bezier Deformer. Modo also adds a number of Bezier Node items into the **Item List** (based on the number of vertices in the specified Bezier curve). Modo uses the Bezier Node items to actually deform the target mesh. To have Modo do this, select the Bezier Node item itself, move it with the transform tools, and set keyframes as necessary for a desired animation. To change the incoming and outgoing curve shape more intuitively, left-click the extended control handles on the Bezier node item.

Bezier Effector Properties

To access the options related to the Bezier Effector, select the Bezier Effector item itself within the Groups Viewport as the parent of the Locators or in the Deformers palette and view the associated attributes in the **Properties** panel.

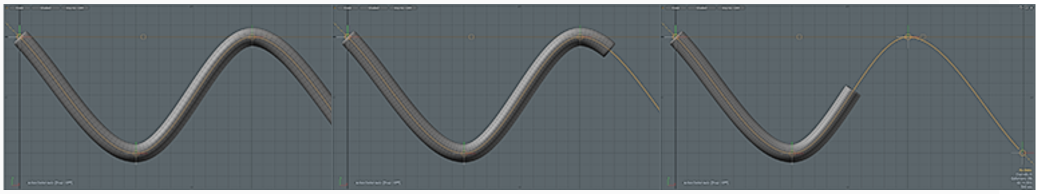
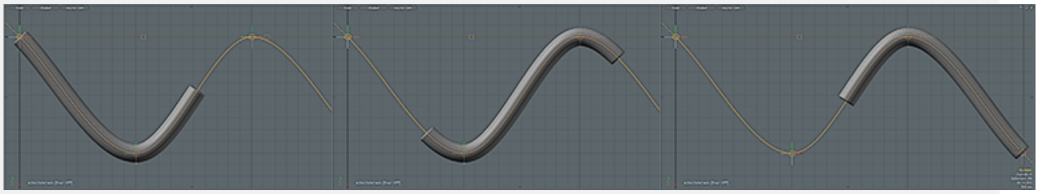


Item Name: Bezier Effector

Bezier Effector

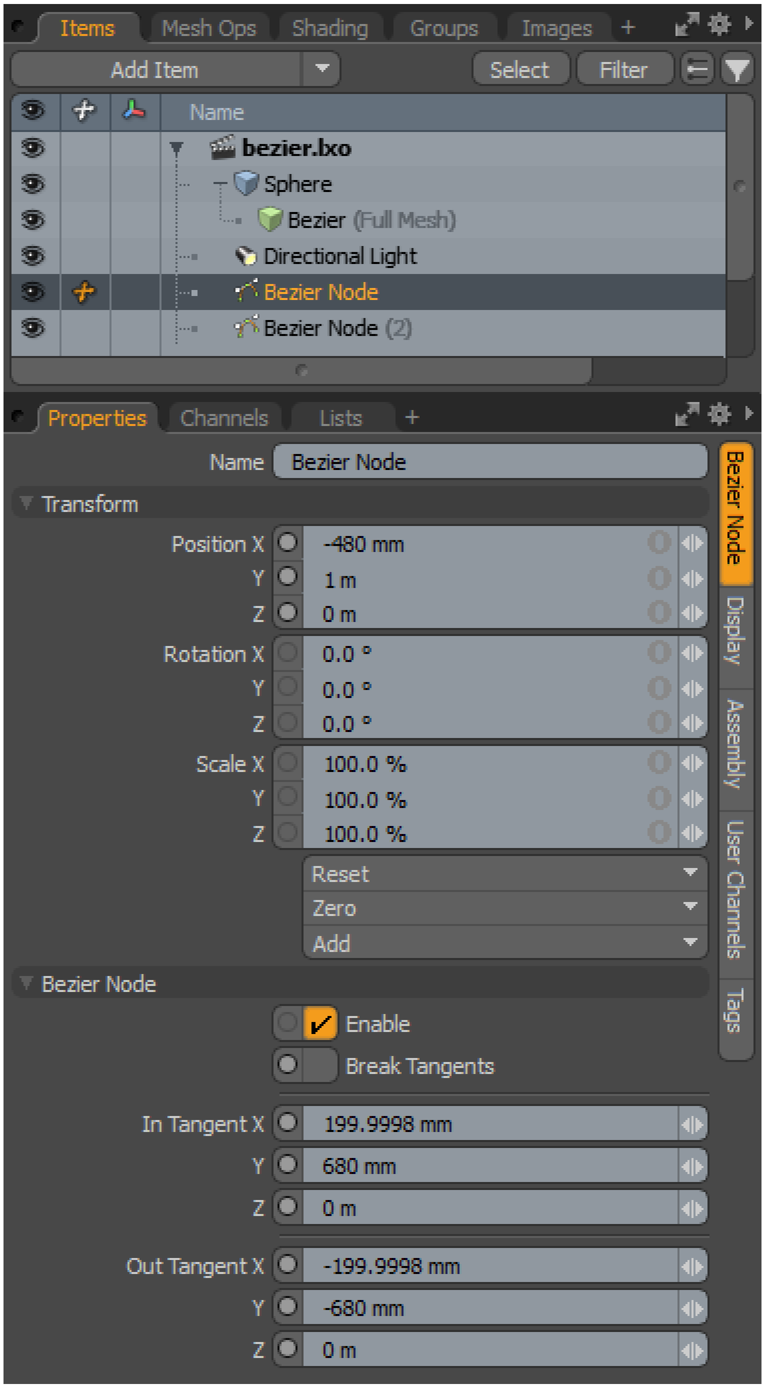
☑ Enable
☑ Visible
☑ Draw Setup
Elastic: 100.0 %
Slide: 0 m
Source Curve: (none)
☐ Animated Source
Target Curve: (none)
☑ Align To Spline
Anti-Twist Threshold: 20.0 %
☐ Use Twist
☐ Use Scale
☐ Orbit Twist
Curve Axis: X Y Z
☑ Y is Up

Bezier Effector	
Enable	Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Bezier Effector	
Visible	Toggles the visibility of the curve representations in the 3D viewport.
Draw Setup	Toggles display of the setup curve in the 3D viewport.
Elastic	Determines to what degree Modo deforms the item as Modo stretches it along the curve to match its overall length. A setting of **100%** (the default) stretches the deformed area to the same length as the target curve. With a setting of **0%**, the item retains its original undeformed length along the curve. Deforming always originates from the start of the curve itself. For items that are shorter than the target deformation curve, use the **Slide** option to position the item along the curve.  Elastic Value 100% Elastic Value 50% Elastic Value 0%
Slide	Moves the target item along the length of the curve. Use this especially when the target item is shorter than the deforming curve. For example, use this to animate a snake along a predetermined path by setting the **Elastic** value to **0%** and then using the **Slide** option to deform the snake along the curve.  Slide Value 0m Slide Value 1m Slide Value 2m
Source Curve/Target Curve	Provides an additional way for you to deform a target mesh using a custom-created curve that is itself deformed by another Deformer. (This bypasses the need to create handles for manipulating the curve.) First, create the curve and then duplicate the curve as a new item to have two identical versions in separate item layers. (Each item layer should have only a single curve.) Next, assign the deformation to the **Target Curve**. (For example, create a Morph for the curve.) Next, add a **Spline Curve Deformer** to an **Item** but do not indicate a specific curve to have Modo create the Effector. You can then

Bezier Effector	
	and the undeformed version as the **Source Curve**.
Animated Source	Indicates when you animate the **Source Curve**. When you select this, Modo constantly evaluates the **Source Curve**. If clear, Modo only evaluates the **Source Curve**'s position on the first frame. **Tip:** When you see that a curve is reversed from what you expected, you can invert the point order by selecting the target curve in one of the component modes and pressing **F**. Now, what was the start is the end and vice versa. (This is the same action as flipping a polygon's normal face).
Align To Spline	When enabled, aligns the deformation of the points to the tangent of the spline.
Anti-Twist Threshold	Defines the size of the region on the curve where the rotation axis is interpolated. **Tip:** You can avoid using the **Anti-Twist threshold** when creating circles or loops, by making the setup spline perpendicular to the deformed curve. So for a deformed loop lying in the XZ plane, it works best if the setup curve is along the Y axis.
Use Twist	When enabled, uses the rotation of the control handles to twist points around the spline axis.
Use Scale	When enabled, uses the scale of the control handles to scale the points.
Orbit Twist	When enabled, adds twist to rotate the deformed spline node around the rest position.
Curve Axis	Determines how Modo uses the Locator transform channels. Select **X**, **Y**, or **Z**. Rotations along the selected axis add twist; Modo ignores the other rotation axes. Scaling perpendicular to the axis stretches the mesh perpendicular to the **Curve Axis**.
Y is Up	Sets where the twist occurs (the twist coordinates' zero-twist value) if you have selected **Curve Axis**. In certain cases, you may want to clear this option to avoid kinks in a deformed mesh when twisting.

Bezier Node Item


Items
Mesh Ops
Shading
Groups
Images
+
Add Item
Select
Filter
Name
Properties
Channels
Lists
+
Transform
Bezier Node
Reset
Zero
Add
Bezier Node
Display
Assembly
User Channels
Tags

bezier.lxo
Sphere
Bezier (Full Mesh)
Directional Light
Bezier Node
Bezier Node (2)

Name: Bezier Node

Position X: -480 mm
Y: 1 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable
☐ Break Tangents

In Tangent X: 199.9998 mm
Y: 680 mm
Z: 0 m

Out Tangent X: -199.9998 mm
Y: -680 mm
Z: 0 m

To control the deformation of the target geometry, you actually manipulate a **Bezier Node** item in the scene. Multiple nodes define a virtual Bezier curve that deforms the target mesh. Each **Bezier Node** has settings for its **Position** as well as the position of the control handles.

Use the **Scale** and **Rotation** values to modify the manipulator handles, but these do not influence the actual deformations of the target. The position of the **Incoming** and **Outgoing Tangent** control settings control the deformations. You set these values manually in the Item's **Properties** panel after you select the item, or you can left-click the item in the viewport and transform it directly. Moving the handles always moves two axes at once, and these axes always move relative to the current Work Plane position.

Name: Displays the current item's name. You may change the name by left-clicking the current name and typing a new name.

Transform

Position X/Y/Z: Define the location in 3D space for the Bezier Node. These positions relative to the other nodes defines the curve that Modo uses to deform the target mesh.

Rotation X/Y/Z: Have no influence over the deformations of the target mesh. Adjust the **Tangent Handles** instead to twist the incoming or outgoing angles of the Bezier curve.

Scale X/Y/Z: Have no influence over the deformations of the target mesh.

Reset: Resets the selected transform values to (0,0,0) and returns the items back to the World origin. As a deformer, resetting changes the location of the Bezier Node influencing the deformation of the target mesh unless you apply it to the items in the **Setup** action state.

Zero: Resets the selected transform property values to **0** and leaves the Center and Mesh positions intact. (Modo adds a negative transform item to the Mesh Item's Channels.)

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

Bezier Node

Enable: Toggles the selected Bezier handle on and off for the deformation. When selected, the handle influences the deformation of the target mesh. When cleared, the handle's influence is ignored.

Break Tangents: If you select **Break Tangents**, Modo treats the handles independently of each other. Use this to create a corner or a sharply angled curve (joint). Bezier control handles are, by default, tangent (straight or linear) on opposing sides of the node position to produce a smooth curve across the Bezier node's position.

In Tangent X/Y/Z: Represent the X,Y, and Z position values of the **Incoming Tangent** handle relative to the actual position of the **Bezier Node**. To manipulate the curve's shape, you can manually enter locations or left-click the handles in the 3D viewport and adjust two axes simultaneously relative to the position of the Work Plane.

Out Tangent X/Y/Z: Represent the X,Y, and Z position values of the **Outgoing Tangent** handle relative to the actual position of the **Bezier Node**. To manipulate the curve's shape, you can manually enter locations or left-click the handles in the 3D viewport and adjust two axes simultaneously relative to the position of the Work Plane.

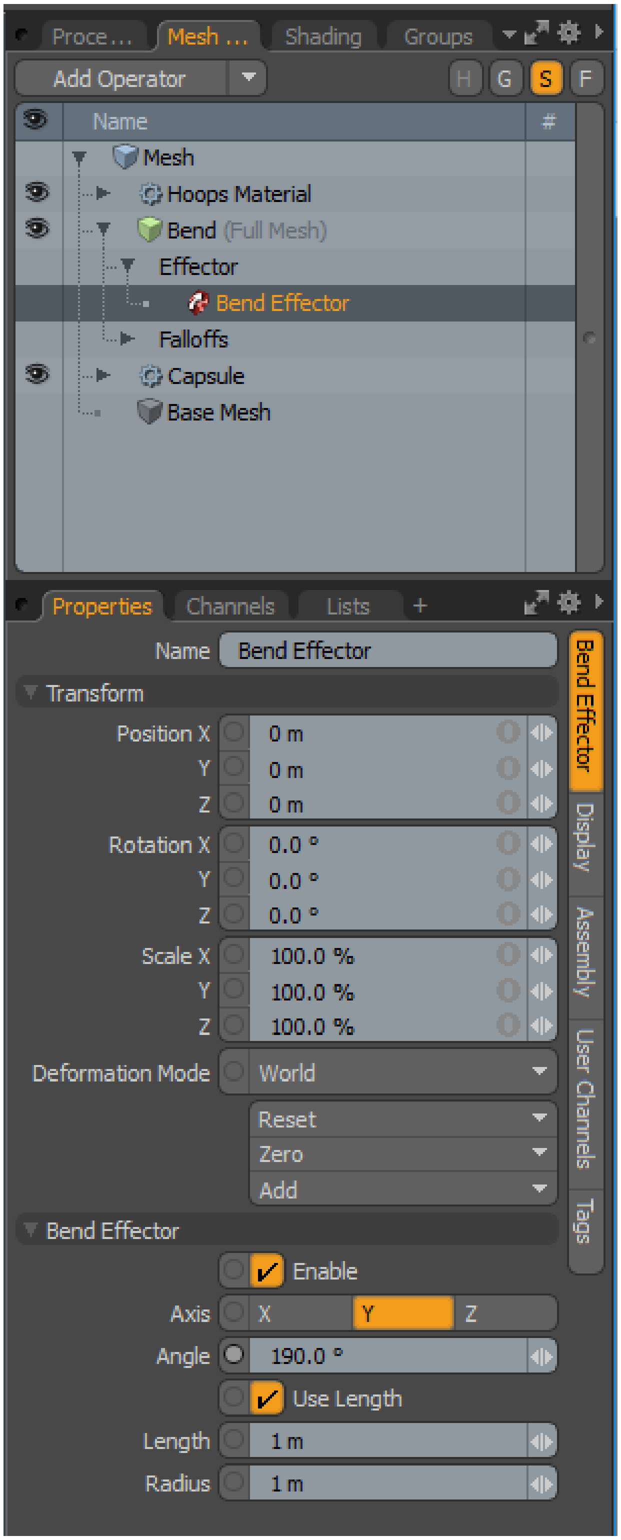
Bend Effector

Effectors are a means to affect a target item, generally a mesh layer, to deform it in specific ways. The **Bend Effector**, when assigned to an item, deforms the object in a curving or arcing fashion. As you increase the **Angle** value, the bending becomes more pronounced. Carefully consider the **Bend Effector**'s position relative to the Mesh Item it is deforming because the position of its origin provides the location to determine the Effector's influence.

Applying an Effector

You can add an Effector by using the **Item List**'s **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer by way of a General Influence. The General Influence acts upon the mesh, and the Effector produces the instructions on how the Influence should modify its target.

If you add the Effector in **Items** mode, Modo connects the Effector and the mesh layer automatically. First, select the target item(s) either in the **Items** list or any 3D viewport and then right-click the item and choose **Add Mesh Operation**, then under **Effectors**, double-click **Bend Effector**. Also, you can select the target item(s) and click the **Effector** button in the **Deformers** sub-tab of the **Layout** interface. Once you create the Effector, you can position it and modify its values to produce the desired results. When you apply an Effector to multiple items, Modo deforms each affected item based on the Effector's relative position to the mesh.


Proce…
Mesh …
Shading
Groups
Add Operator
H
G
S
F
Name
#
Mesh
Hoops Material
Bend (Full Mesh)
Effector
Bend Effector
Falloffs
Capsule
Base Mesh
Properties
Channels
Lists
+
Bend Effector
Display
Assembly
User Channels
Tags
Reset
Zero
Add

Name: Bend Effector

Transform

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %
Deformation Mode: World

Bend Effector

☑ Enable
Axis: X [✓ Y] Z
Angle: 190.0 °
☑ Use Length
Length: 1 m
Radius: 1 m

Name: Displays the current item's name. You may change the name by left-clicking the current name and typing a new name.

Transform

Position XYZ: Indicates the numerical position of the Effector item in XYZ space. These values define the origin of influence for the Effector.

Rotation XYZ: Indicates the rotation of the Effector item numerically. These values define the direction of influence for the Effector.

Scale XYZ: Sets the size of the item's representation in the 3D viewport numerically. To adjust the scale of the arc's influence, clear **Use Length** and adjust the **Radius** value.

Deformation Mode: Determines how the transforms of the Effector translate into a deformation.

- **World**- This is the default setting and determines the deformation based on the World Position of the Effector. In most cases, this provides the expected results.
- **Local**- This applies the deformation relative to the **Setup** position of the item as if the item were still in the rest state regardless of the position of the Effector. This is useful for the order of operation workflow when rigging.
- **Local ** Rotation Modes**- This is identical to the **Local** mode, but Modo only applies the specified rotation. This creates a deformation that is more like a **Rotation Effector** with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position and the mesh position intact. Modo does this by adding a negative transform item to the Mesh Item's channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

Bend Effector

Enable: Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Axis: Determines the general directional **Axis** for the Effector's deformation without having to directly rotate the Effector item.

Angle: Determines the amount of bending applied the target mesh as an arc. For longer or shorter bend lengths, change the Effector's position relative to its target or clear **Use Length** and adjust the **Radius** value.

Use Length: If you select **Use Length**, Modo automatically determines the **Radius** value based on the **Length** value.

Length: Determines the maximum range of influence along the specified **Axis** for attenuation along the curve, but Modo deforms the mesh beyond the defined Length at **100%** strength.

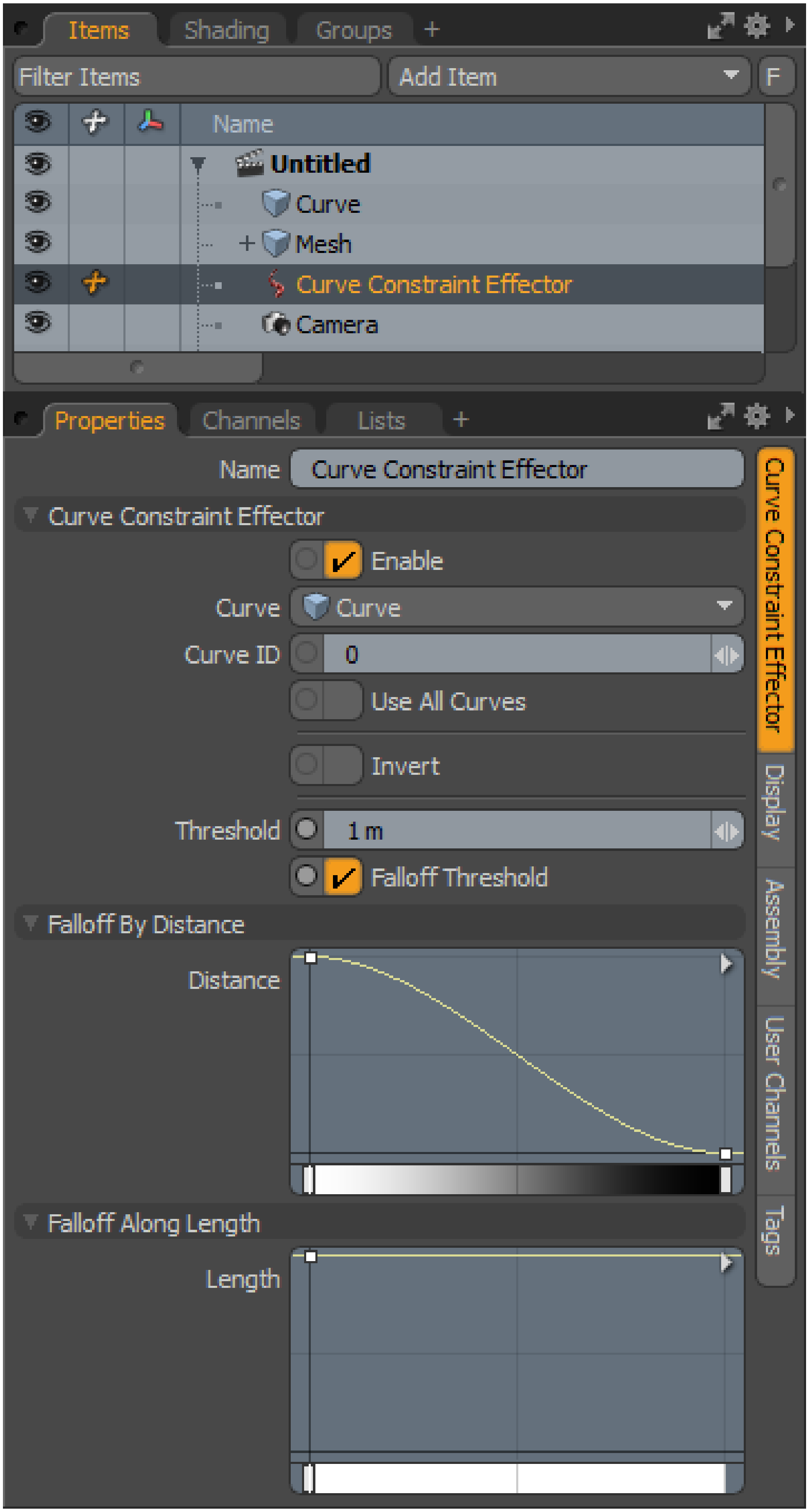
Radius: Determines the size of the bending arc and, consequently, its scale of influence.

Curve Constraint Effector

The **Curve Constraint Effector** is a special kind of deformer that will attract nearby vertices, constraining them to the nearest position along the defined curve. The points that are constrained are determined by the Threshold distance value, defining an area of influence around the curve, and the strength of that attraction outward from the curve is controlled by the **Falloff by Distance** gradient. Using the default gradient as vertices are further away from the curves influence core, the less they are pulled toward that curve itself.

Applying an Effector

The Curve Constraint Effector can be created directly from the **Setup** layouts tool palette under the **Deformers** subtab. Select the Mesh Item you wish to constrain and the left mouse click the **Curve Constraint** button in the tool palette to open the setup dialog asking for a Curve. Select the appropriate curve item and click **OK** to create the constraint, with all the graph connections being automatically wired up. Selecting the newly created Curve Constraint Effector item in the Items to display its attributes in the Properties panel.


Items
Shading
Groups
+
Filter Items
Add Item
F
Name
Untitled
Curve
Mesh
Curve Constraint Effector
Camera
Properties
Channels
Lists
+
Curve Constraint Effector
Display
Assembly
User Channels
Tags

Name: Curve Constraint Effector

Curve Constraint Effector

☑ Enable

Curve: Curve

Curve ID: 0

☐ Use All Curves

☐ Invert

Threshold: 1 m

☑ Falloff Threshold

Falloff By Distance

Distance

Falloff Along Length

Length

Name: Displays the current item's name. You may change the name by left-clicking the current name and typing a new name.

Curve Constraint Effector-

Enable: Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Curve: This menu option allows you to select the particular curve layer that will constrain the associated Mesh Item, allowing alternate curve selection after the constraint has been created.

Curve ID: In defined Mesh Items where there are more than a single curve within the same layer, this option allows users to define specific curves within the layer by their Curve ID, an automatically generated identifier based on the order of curve creation.

Use All Curves: This option, when enabled will use all curve contained in the specified layer as a constraint source. **Falloffs Along Length** will still be applied independently to each individual curve.

Invert: This option, when enabled will invert the influence of the Curve Constraint, pushing nearby vertices away from the curve, rather than pulling them toward it.

Threshold: The Threshold defines a circular area of influence around the curve generated at each position along the curve outward.

Falloff Threshold: The Falloff Threshold option allows the threshold to be modulated based on the distance from the curve. For example, if the element being deformed already exists on the curve, then it will be given a weight of 1.0, if the element being deformed is at a distance that is greater than or equal to the threshold channel, then it will be given a weight of 0.0. Any elements between 0.0 and the threshold distance, will be given a smoothed value between 0.0 and 1.0. The offset vector that is applied to the element position, is scaled by this weight value. The result is a deformation that falls off, the further away from the curve the element is positioned.

Falloff by Distance: This mini gradient editor controls the falloff outward from the curve along its defined **Threshold** distance. By using the mini-gradient editor, you can modify the falloff with gradient keys that represent the strength of influence. The mini gradient width represents the Falloff Distance with the left side representing the interior curve position and the right side the outer most limit of the Threshold. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust its value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black has no influence attenuating toward white with maximum influence.

Falloff Along Length: This mini gradient editor controls the total influence of the curve along its length, with the left side of the gradient representing the base position of the curve (the first vertex when created). To invert the base and tip positions of the curve, select it in Polygons mode and press the 'F' key to flip its orientation. By using the mini-gradient editor, you can modify the falloff with gradient keys that represent the strength of influence. The gradient input's width represents the curves overall length with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust its value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black has no strength attenuating toward white with maximum strength.

Lag Effector

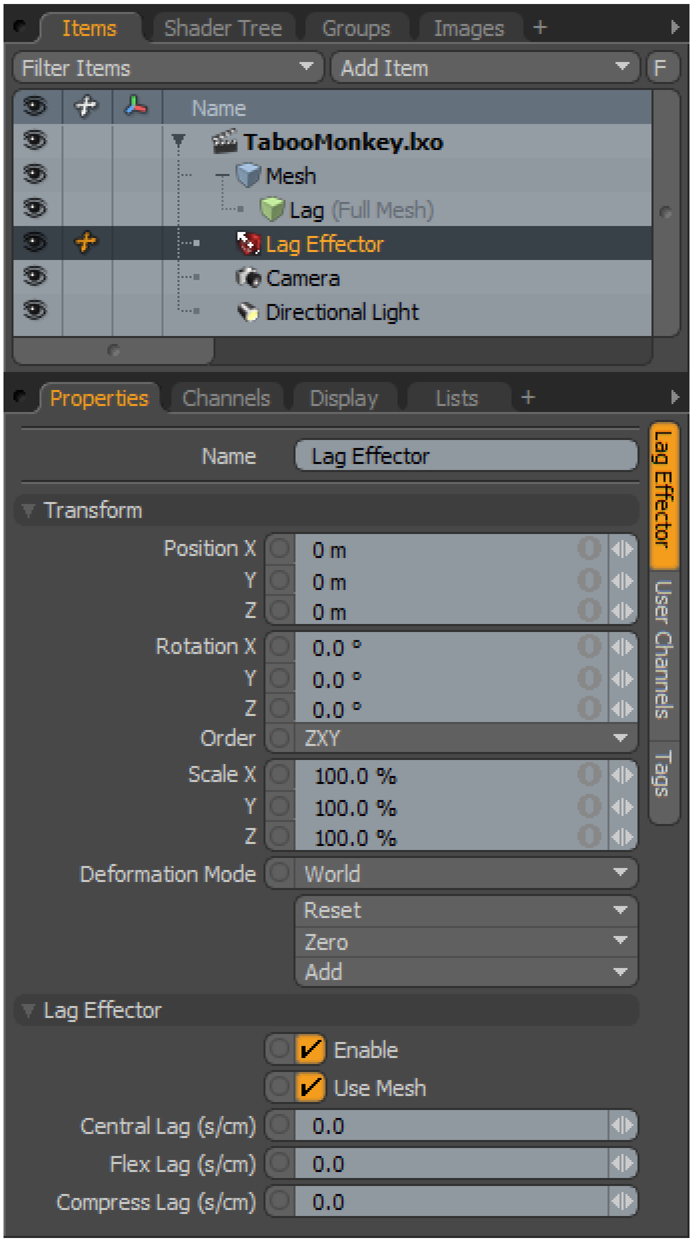
Effectors are a means to affect a target item, specifically a mesh layer, to deform it in specific ways. When you assign the **Lag Effector** to an item, it deforms the object along a directional vector that is determined by the target item's animation. (It is deformed along the direction it is traveling.) Points lag more the further they are from the **Lag Effector**'s position or, if you select **Use Mesh**, from the Mesh Item's center point. If you don't animate the target item, you won't see any effect. As the Lag values increase, the deformation becomes greater.

Applying an Effector

You can add an Effector by using the **Item List**'s **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer by way of a General Influence. The General Influence acts upon the mesh, and the Effector, itself, produces the instructions on how the Influence should modify its target.

Modo can make this connection automatically (while in **Items mode**) if you first select the target item either in the Items list or in any 3D viewport, and then right-click the item and choose choose **Add Mesh Operation**, then under **Effectors**, double-click **Lag Effector**.

Once you apply this Effector, you can see the results from the Deformer by selecting **Use Mesh** and assigning some animation to the target item. Next, assign values (appropriate to the scale of the target) to **Central Lag** or **Flex Lag**. (Lag values are in centimeters per second of time.) Now, when you play the Timeline, the vertices farther away from the item's Center point lag behind the motion of the item itself. If you select the **Lag Effector**, its attributes appear in the **Properties** panel.


Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
TabooMonkey.lxo
Mesh
Lag (Full Mesh)
Lag Effector
Camera
Directional Light
Properties
Channels
Display
Lists
Lag Effector
User Channels
Tags
Transform
Reset
Zero
Add
Lag Effector

Name: Lag Effector

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %
Deformation Mode: World

☑ Enable
☑ Use Mesh
Central Lag (s/cm): 0.0
Flex Lag (s/cm): 0.0
Compress Lag (s/cm): 0.0

Name: Displays the current item's name. You may change the name by clicking the current name and typing a new name.

Transform

Position: Indicates the numerical position of the Effector item in XYZ space. These values define the origin of influence for the Effector.

Rotation: Indicates the rotation of the Effector item numerically. These values define the direction of influence for the Effector.

Order: Specifies the order that Modo applies rotations to the Effector item. Changing this order can sometimes reduce or eliminate gimbal lock.

Scale: Sets the size of the item's representation in the 3D viewport numerically.

Deformation Mode: Determines how Modo translates the transforms of the Effector into a deformation.

- **World**- This is the default setting and determines the deformation based on the World Position of the Effector. In most cases, this provides the expected results.
- **Local**- This applies the deformation relative to the Setup position of the item as if the item were still in the rest state regardless of the position of the Effector. This is useful for the order of operation workflow when rigging.
- **Local ** Rotation Modes**- This is identical to the **Local** mode, but Modo only applies the specified rotation. This creates a deformation that is more like a Rotation Effector with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position intact. Modo does this by adding a negative transform item to the item's transform channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

Lag Effector

Enable: Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Use Mesh: When selected, this sets the effector to use the Center of the mesh it is deforming, instead of the position of the effector as the origin. It is similar to parenting the effector to the mesh, but selecting **Use Mesh** is easier.

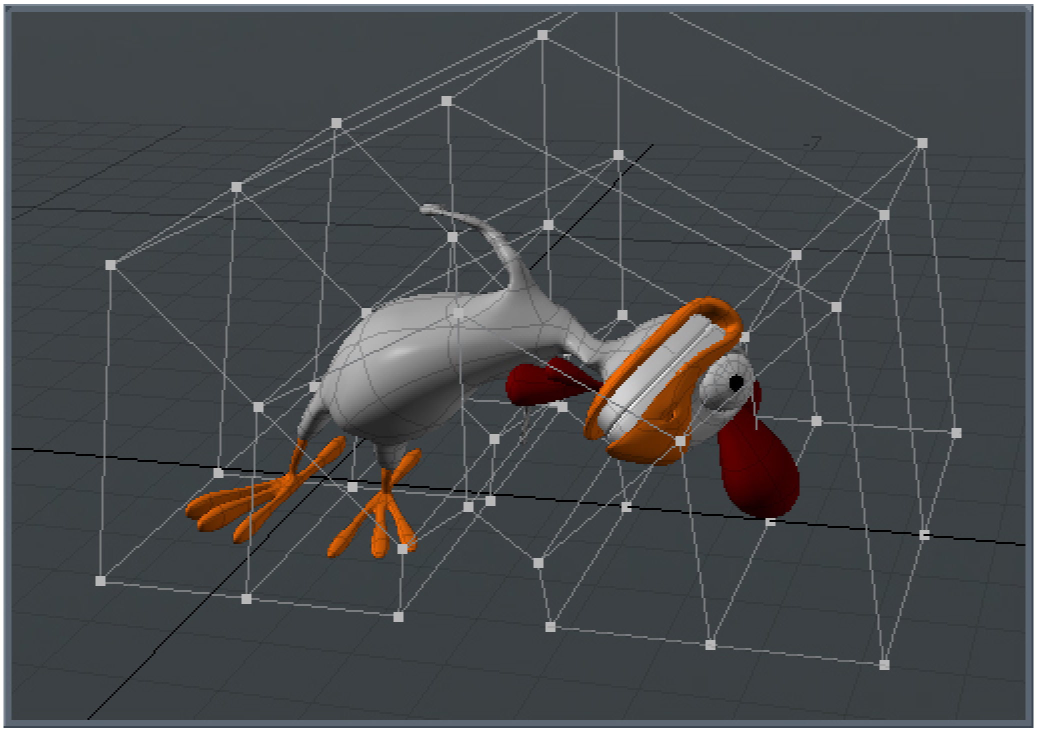
Central Lag: This is the amount of time by which a point 1 cm from the Center lags behind the center (the amount of time lag per cm distance from the center). The distance is the absolute distance from the point to the origin location, whether it is the **Lag Effector** position or the mesh's center.

Flex Lag: Similar to **Central Lag**, this is also a lag amount per cm, but the distance is the distance perpendicular to the line of motion (vector).

Compress Lag: This is the lag amount per cm, but the distance is measured only along the line of motion to stretch along the path of motion. The **Flex Lag** and **Compress Lag** amounts are components of the **Central Lag**. By combining them at different levels, you can get better control over how a motion deforms the mesh.

Lattice Effector

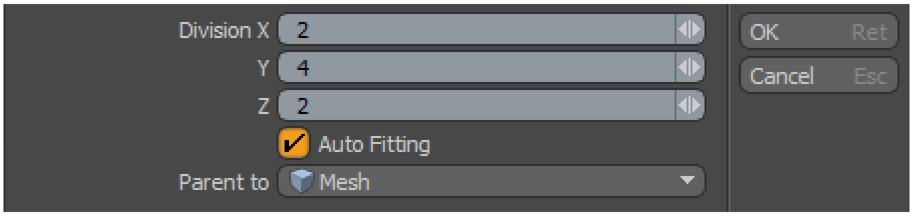
The Lattice Effector, when assigned, creates a structured array of points across the mesh similar to a cage. You can moved each individual vertex of this cage to deform the underlying mesh and smoothly interpolate the deformation between the vertex locations. You can then animate these cage handles to produce very interesting deformation effects.



Applying an Effector

You can add an Effector by using the **Items** list's **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer with a General Influence. The General Influence acts upon the mesh, and the Effector, itself, produces the instructions on how the Influence should modify its target.

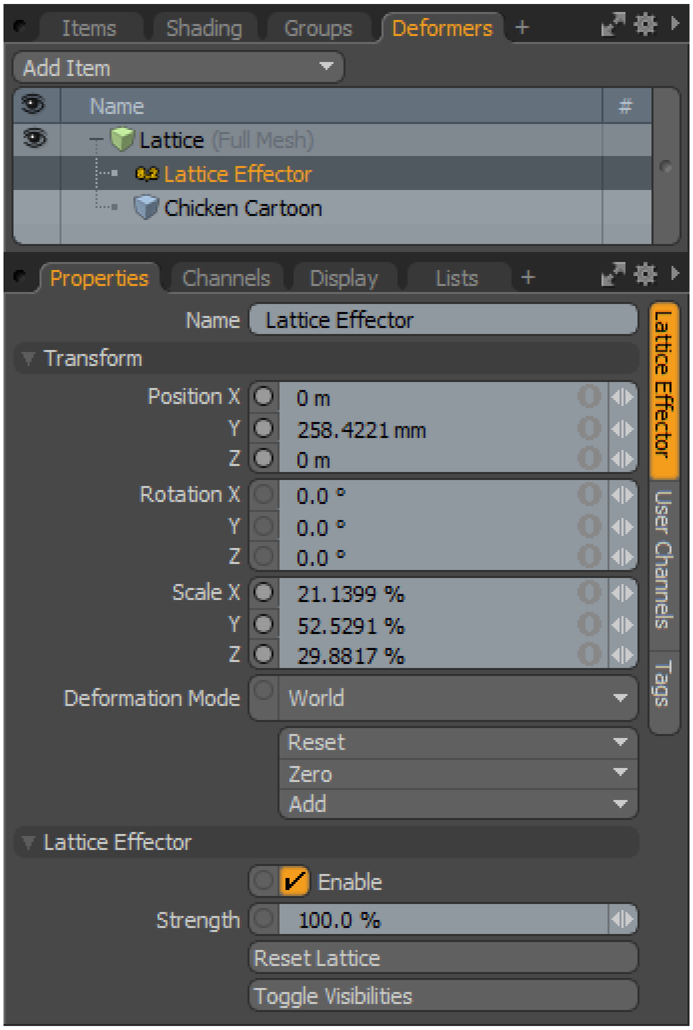
You properly apply the Lattice Effector to a mesh in the **Setup** workspace layout. In the **Deformers** toolbox sub-tab, left-click the **Lattice** deformer button to open the **Create Lattice Deformer** dialog box.



Division X 2
Y 4
Z 2
☑ Auto Fitting
Parent to Mesh

In this dialog box you define the number of lattice handles along each axis. Select the **Auto Fitting** option to fit the cage to the bounding box volume of the target mesh automatically. In the **Parent to** list, select the target mesh for deforming. (If you select the target item prior to opening the dialog box, Modo automatically selects the target item.) Once you have assigned everything properly, click **OK** to actually generate the Lattice.

When assigned as such, a wireframe cage appears around the Mesh Item layer with the number of divisions specified. At each division there is a handle (a Locator actually). You can select these handles then in **Items mode** and position them with the standard Transform tools to deform the underlying Mesh Item. You can modify the handles individually or select them in groups and move them together. You can also select the Lattice Effector item, itself, in the Groups Viewport or Deformers viewport and adjust its attributes in the **Properties** panel. This viewport also has options for you to reset the Lattice and update the number of divisions.


Items
Shading
Groups
Deformers
Add Item
Name
#
Lattice (Full Mesh)
Lattice Effector
Chicken Cartoon
Properties
Channels
Display
Lists
Lattice Effector
User Channels
Tags
Transform
Reset
Zero
Add
Lattice Effector
Reset Lattice
Toggle Visibilities

Name: Lattice Effector

Position X: 0 m
Y: 258.4221 mm
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 21.1399 %
Y: 52.5291 %
Z: 29.8817 %
Deformation Mode: World

☑ Enable
Strength: 100.0 %

Name: Displays the current item's name. You may change the name by clicking the current name and typing a new name.

Transform

Position XYZ: Indicates the numerical position of the Effector item in XYZ space. The Effector cage's position relative to the geometry has a dramatic effect on how it deforms the target. Normally, center the cage on the target bounding box's volume. (Moving the cage is not the same thing as moving all the handles of the cage.)

Rotation: Indicates the rotation of the Effector item numerically. The Effector cage's rotation relative to the geometry has a dramatic effect on how it deforms the target. (Rotating the cage is not the same thing as moving all the handles of the cage.)

Scale: Sets the size of the item's representation in the 3D viewport numerically. The Effector cage's size relative to the geometry has a dramatic effect on how it deforms the target. (Scaling the cage is not the same thing as moving all the handles of the cage.)

Deformation Mode: Determines how the transforms of the Effector translate into a deformation.

- **World**- This is the default setting and determines the deformation based on the World Position of the Effector. In most cases, this provides the expected results.
- **Local**- This applies the deformation relative to the **Setup** position of the item as if the item were still in the rest state regardless of the position of the Effector. This is useful for the order of operation workflow when rigging.
- **Local ** Rotation Modes**- This is identical to the **Local** mode, but Modo only applies the specified rotation. This creates a deformation that is more like a **Rotation Effector** with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position intact. Modo does this by adding a negative transform item to the item's transform channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

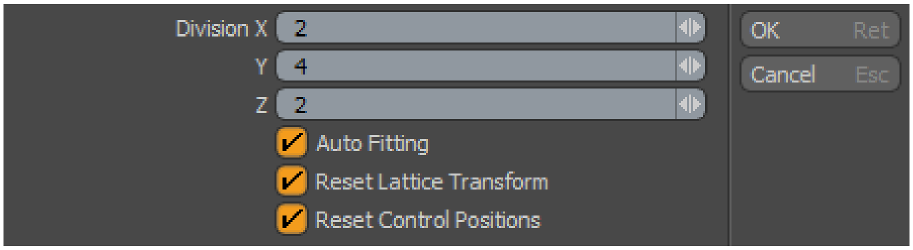
Lattice Effector

Enable: Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Strength: Determines the amount of influence the Effector has on the target mesh. You can switch between a deformed version and an undeformed version.

Reset Lattice: Opens the **Reset Lattice Deformer** dialog box for you to re-define the number of control handles.

Toggle Visibilities: Toggles the visibility (and therefore, the selectability) of the control handles in the 3D viewport.



Division X 2

Y 4

Z 2

☑ Auto Fitting

☑ Reset Lattice Transform

☑ Reset Control Positions

Division X/Y/Z: These options specify the number of control handles given to the Effector. Adjusting the number of divisions after the fact without resetting the Control positions may produce undesirable results.

Auto Fitting: Selecting this option scales the size of the control cage to the bounding box of the target mesh. When disabled, the resulting cage is, by default, 1 meter square.

Reset Lattice Transform: This option resets any transform values applied to the Lattice Effector, itself.

Reset Control Positions: This option removes any transforms applied to the control handles and returns the Effector to its initial, base shape.

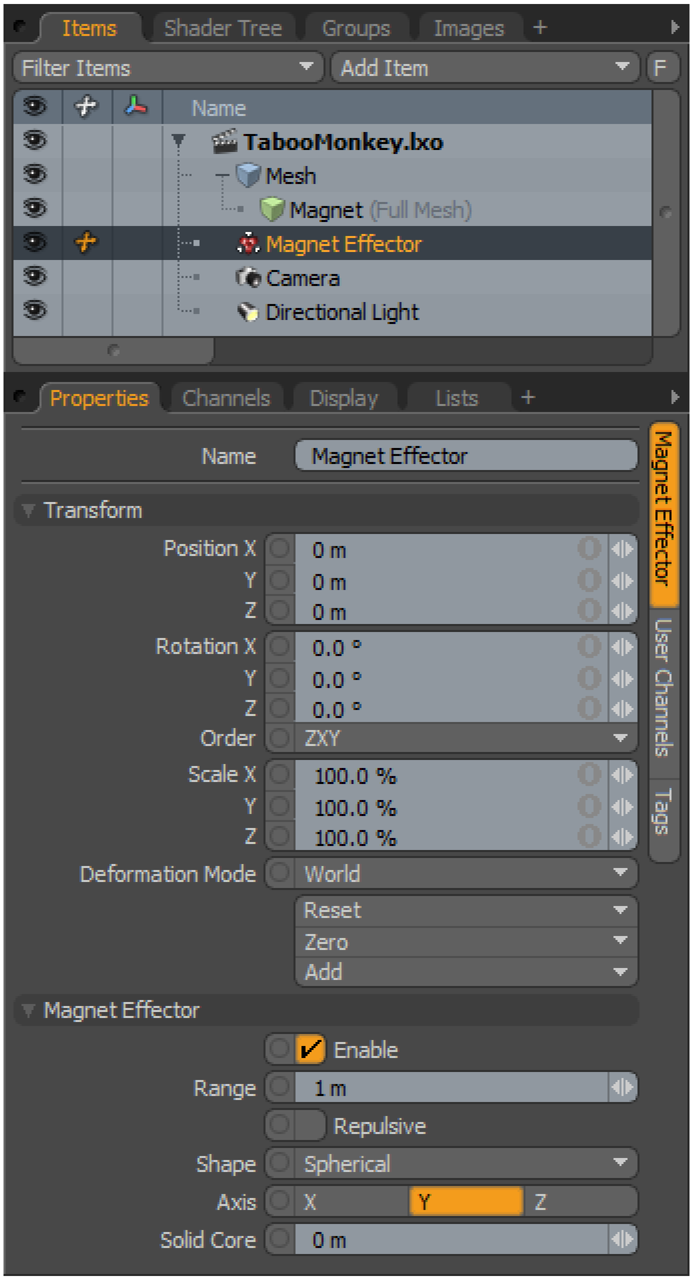
Magnet Effector

Effectors are a means to affect a target item, specifically a mesh layer, deforming it in specific ways. When you assign the **Magent Effector** to an item, Modo draws points within the defined range toward the center of the Magnet itself. Selecting the **Repulsive** option pushes points outward.

Applying an Effector

You can add an Effector by using the **Item List**'s **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer by way of a General Influence. The General Influence acts upon the mesh, and the Effector, itself, produces the instructions on how the Influence should modify its target.

If you add the Effector in **Items** mode, Modo connects the Effector and the mesh layer automatically. First, select the target item(s) either in the **Items** list or any 3D viewport and then right-click the item and choose choose **Add Mesh Operation**, then under **Effectors**, double-click **Magnet Effector**. Also, you can select the target item(s) and click the **Effector** button in the **Deformers** sub-tab of the **Layout** interface. Once you create the Effector, you can position it and modify its values to produce the desired results. When you apply an Effector to multiple items, Modo deforms each affected item based on the Effector's relative position to the mesh.


Items
Shader Tree
Groups
Images
+
Filter Items
Add Item
F
Name
TabooMonkey.lxo
Mesh
Magnet (Full Mesh)
Magnet Effector
Camera
Directional Light
Properties
Channels
Display
Lists
+
Magnet Effector
User Channels
Tags
Transform
Reset
Zero
Add
Magnet Effector

Name: Magnet Effector

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %
Deformation Mode: World

☑ Enable
Range: 1 m
☐ Repulsive
Shape: Spherical
Axis: X [✓ Y] Z
Solid Core: 0 m

Name: Displays the current item's name. You may change the name by left-clicking the current name and typing a new name.

Transform

Position XYZ: Indicates the numerical position of the Effector item in XYZ space. These values define the origin of influence for the Effector.

Rotation XYZ: Indicates the rotation of the Effector item numerically. These values define the direction of influence for the Effector.

Order: Indicates the order that Modo applies rotations to the Effector item. By changing this order, you can sometimes reduce or eliminate gimbal lock.

Scale XYZ: Sets the size of the item's representation in the 3D viewport numerically. To adjust the scale of the arc's influence, clear the **Use Length** option and specify the **Radius** value.

Deformation Mode: Determines how the transforms of the Effector translate into a deformation.

- **World** - This is the default setting and determines the deformation based on the World Position of the Effector. In most cases, this provides the expected results.
- **Local** - This applies the deformation relative to the **Setup** position of the item as if the item were still in the rest state regardless of the position of the Effector. This is useful for the order of operation workflow when rigging.
- **Local ** Rotation Modes** - This is identical to the **Local** mode, but Modo only applies the specified rotation. This creates a deformation that is more like a **Rotation Effector** with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position and the mesh position intact. Modo does this by adding a negative transform item to the Mesh Item's channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

Magnet Effector

Enable: Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Range: Determines the range of influence. Modo deforms any value within the range but does not deform anything outside the range.

Repulsive: Select to have Modo push points from the Locator position. (By default, the **Magnet Effector** pulls points toward the Locator position.)

Shape: Sets the shape of the Effector's field and determines how points are moved to (or away from) the Effector.

- **Planar** - With this shape, points are attracted linearly to a directional axis based on the **Range** value.
- **Spherical** - With this shape, points within the **Range** value are attracted to the center of a round area of influence.

- **Cylindrical** - With this shape, points within the **Range** value are attracted to the centerline of a cylindrical shape.

Axis: Determines the axis direction for the **Shape** option. This only affects the **Planar** and **Cylindrical** shapes. (If you select **Spherical**, Modo produces the same results on all axes.)

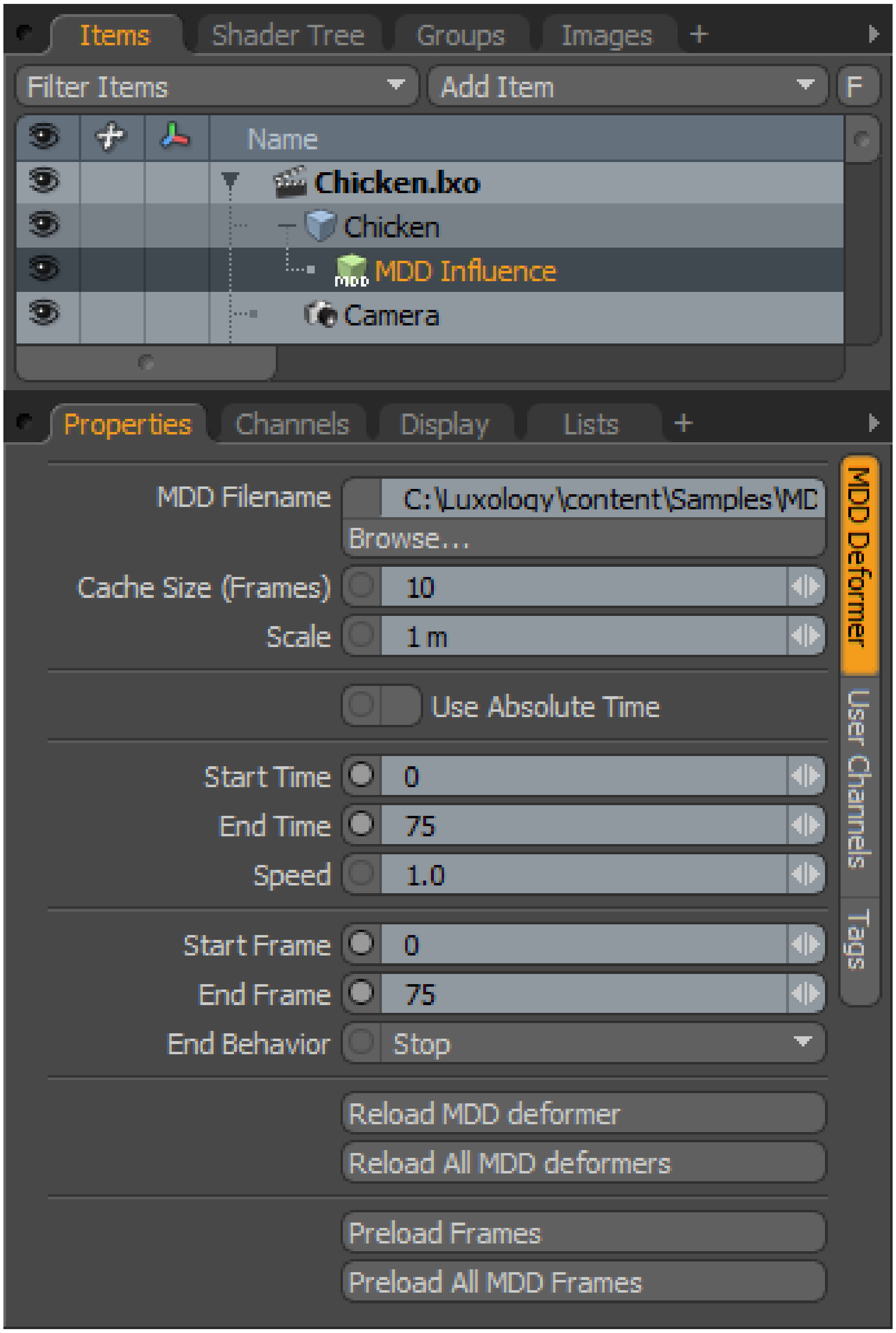
Solid Core: Specifies the distance from the center of the magnet that is considered solid. Modo moves any points inside this range to the surface of the shape.

MDD Deformer

MDD files were invented by Daisuke Ino during his work on a dynamics engine built as a plug-in to a host 3D application. These files are a very simple point cache that simply stores a location in space for every vertex in a model for every frame in the animation. The original intent for these files was to simply allow you to freeze a dynamics simulation so that the result could be saved to disk, reloaded and played back without having to simulate again. Soon Hollywood visual FX facilities realized an interesting benefit to these simple point cache files. They can be generated very easily for any kind of deformation animation. This not only allowed them to disable complex IK and bones calculations, but also by creating MDD deformers in multiple applications, they could transport animation done in one application into a completely different one for final rendering.

Today there are MDD exporters for almost every major 3D application on the market. This means that with MDD support inside Modo, you can model in Modo, perform complex animation or simulations in another host application and return that model with MDD files to Modo for final frame out rendering. Using MDD files is very simple in Modo as it is implemented as a deformer Item type. All you need to do is load your model, add the MDD deformer to the Mesh Item and choose the MDD file.

MDD Deformer


Items
Shader Tree
Groups
Images
+
Filter Items
Add Item
F
Name
Chicken.lxo
Chicken
MDD Influence
Camera
Properties
Channels
Display
Lists
+
MDD Deformer
User Channels
Tags
Browse...
Reload MDD deformer
Reload All MDD deformers
Preload Frames
Preload All MDD Frames

MDD Filename: C:\Luxology\content\Samples\MC
Cache Size (Frames): 10
Scale: 1 m
Use Absolute Time
Start Time: 0
End Time: 75
Speed: 1.0
Start Frame: 0
End Frame: 75
End Behavior: Stop

Enable: This toggle simply activates or deactivates the deformer. This is a quick way to turn off an MDD temporarily. This can also be accomplished by toggling the eye icon in the visibility column of the item list.

Blend Mode: This dropdown choice has options for **Replace**, **Add** or **Multiply**. This dictates how the current MDD with interact with other deformer items on the same mesh. The default, **Replace**, simply writes over any other deformer beneath in the Item List. **Add** and **Multiply** are two blend mode options that either sum the deformation values or multiply them.

Opacity: This control allows you to modulate the strength of the deformer item. This can be particularly useful when you are blending multiple deformer items together.

MDD File Name: This field shows you the path and file name of the MDD that was applied at the time of creation.

Scale: The default scale is 1 meter. By changing this scale value you can increase or decrease the size of the item as it is deformed via the MDD.

Use Absolute Time: This option uses the times indicated in the file header of the MDD file rather than the Modo time settings. This way you can export an animation with precise timing and replay it in Modo without having to adjust any of the MDD time values.

Start/End Time: These values indicate when the MDD animation should be applied during your scene. By default the MDD begins at frame 1 of your animation. However, if you want to delay the start frame you can simple change the Start Time. For instance, changing the Start Time to 2 forces the MDD file to delay until 2 seconds into the animation at which point the MDD begins at its initial frame. This is a very convenient method for time shifting your MDD.

Speed: This parameter adjusts the playback rate of the selected MDD file. By default, Modo uses a one to one match of MDD frame to Modo frame. To slow down the MDD playback, lower the value below 1. To increase playback speed increase the number.

Start/End Frame: These values indicate the frames within the MDD that are to be used. By default when an MDD is applied Modo reads the values from the file directly. If you wish to use only subset of the MDD data this is where you can set the desired in and out points.

End Behavior: If the MDD data is shorter than the Modo animation the deformer needs to know what to do when it reaches the end of the MDD file. The default behavior of Stop simply holds the last frame of the MDD. There are also options to Repeat and PingPong the playback.

Reload MDD (ALL): These options allow you to reload the selected MDD or all MDD files in a given project. With these controls, you can load a project into Modo, return to another application to modify the simulation or character animation, export the MDD file and then force Modo to update to the most recent data without having to reload the entire project.

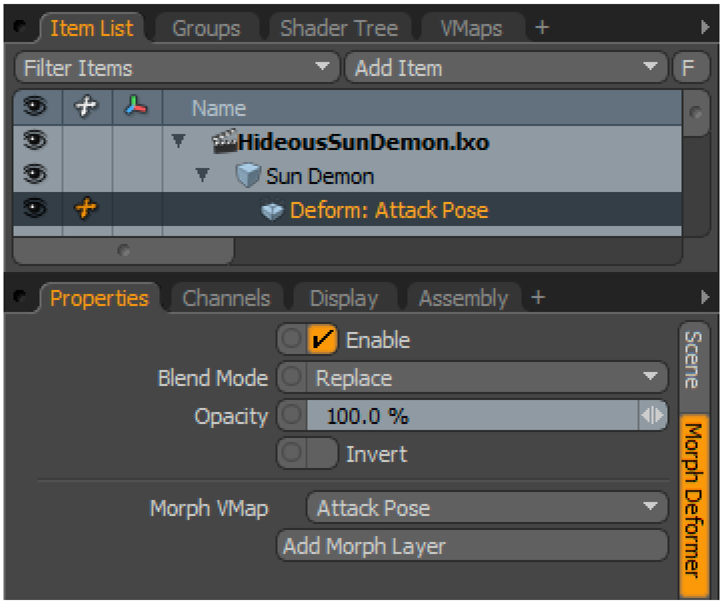
Add Morph

In Modo, you add deformers, such as the Morph deformer, as items. All items have channels, and you can animate all channels. So, to animate morphs, you add a Morph deformer to your Mesh Item and then animate the **Opacity** channel. (Specifying 100% produces the morph in full.) You can add multiple deformers to a single Mesh Item. To animate more than one morph target, add as many Morph deformers as you wish. These deformers naturally blend together so that all your morphs can animate synchronized with one another.

To animate a Morph deformer:

1. Add the deformer to a Mesh Item.
2. Specify the desired morph Vertex Map.
3. Animate the **Opacity** setting.

Morph Deformer



HideousSunDemon.lxo
Sun Demon
Deform: Attack Pose

☑ Enable
Blend Mode: Replace
Opacity: 100.0 %
☐ Invert
Morph VMap: Attack Pose

Enable: Activates or deactivates the deformer. Use as a quick way to turn off a morph temporarily. You can also turn off a morph by toggling the **eye** icon in the visibility column of the **Item List** tab.

Blend Mode: Controls how the current morph interacts with other deformer items on the same mesh. This has options for **Replace**, **Add**, or **Multiply**. The default, **Replace**, writes over any other deformer beneath it in the **Item List** tab. **Add** and **Multiply** are two blend mode options that either sum the deformation values or multiply them.

Opacity: Modulates the strength of the deformer item. **100%** produces the full effect of the morph. This is useful when you are blending multiple deformer items together.

Morph VMap: Use this to select the Morph Vertex Map to apply to the geometry.

Add Morph Layer: Use this to add additional morph deformers to the model.

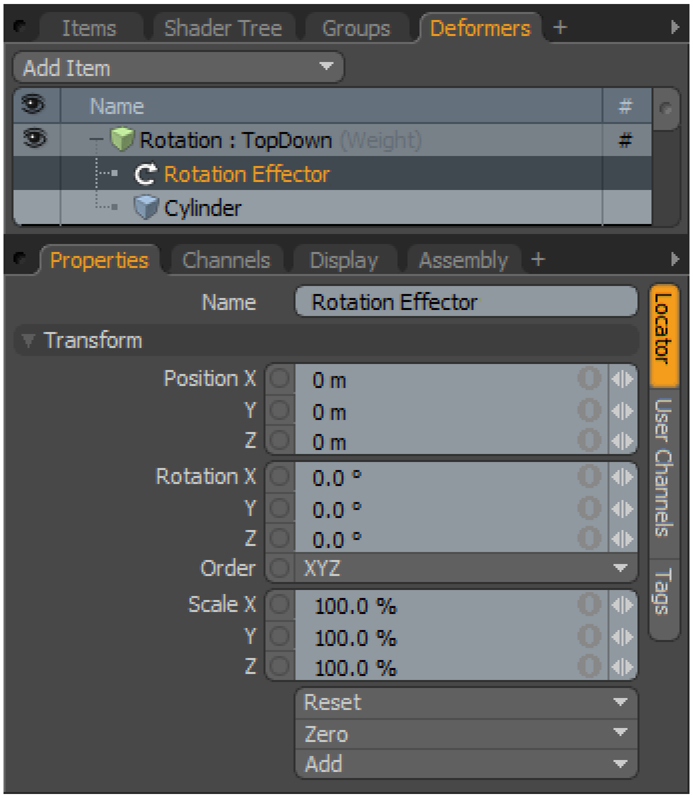
Rotation Effector

Effectors are a means to affect a target item, generally a mesh layer, to deform it in specific ways. The **Rotation Effector**, when assigned to an item, rotates the target item. This is the same as a regular item transform, but you can link this to rigging and control it through the **Schematic** view because it is an item with channels. With a regular tool, you can only control it when it is active.

Applying an Effector

You can add an Effector by using the **Item List**'s **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer by way of a General Influence. The General Influence acts upon the mesh, and the Effector itself, produces the instructions on how the Influence should modify its target.

If you add the Effector in **Items** mode, Modo connects the Effector and the mesh layer automatically. First, select the target item(s) either in the **Items** list or any 3D viewport and then right-click the item and choose choose **Add Mesh Operation**, then under **Effectors**, double-click **Rotate Effector**. Also, you can select the target item(s) and click the **Effector** button in the **Deformers** sub-tab of the **Layout** interface. Once you create the Effector, you can position it and modify its values to produce the desired results. When you apply an Effector to multiple items, Modo deforms each affected item based on the Effector's relative position to the mesh.


Items
Shader Tree
Groups
Deformers
Add Item
Name
#
Rotation : TopDown (Weight)
Rotation Effector
Cylinder
Properties
Channels
Display
Assembly
Locator
User Channels
Tags
Transform
Reset
Zero
Add

Name: Rotation Effector

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Order: XYZ

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Name: Displays the current item's name. You may change the name by clicking the current name and typing a new name.

Transform

Position XYZ: Indicates the numerical position of the Effector item in XYZ space. These values define the origin of influence for the Effector.

Rotation XYZ: Indicates the rotation of the Effector item numerically. Changes in these values from the rest position transform the deformed target that amount. (Modo determines the rest position by the item's position in the **Setup** action state.)

Order: Indicates the order that Modo applies rotations to the Effector item. By changing this order, you can sometimes reduce or eliminate gimbal lock.

Scale XYZ: Sets the size of the item's representation in the 3D viewport numerically. (This does not transform the target item.)

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position and the mesh position intact. Modo does this by adding a negative transform item to the Mesh Item's channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

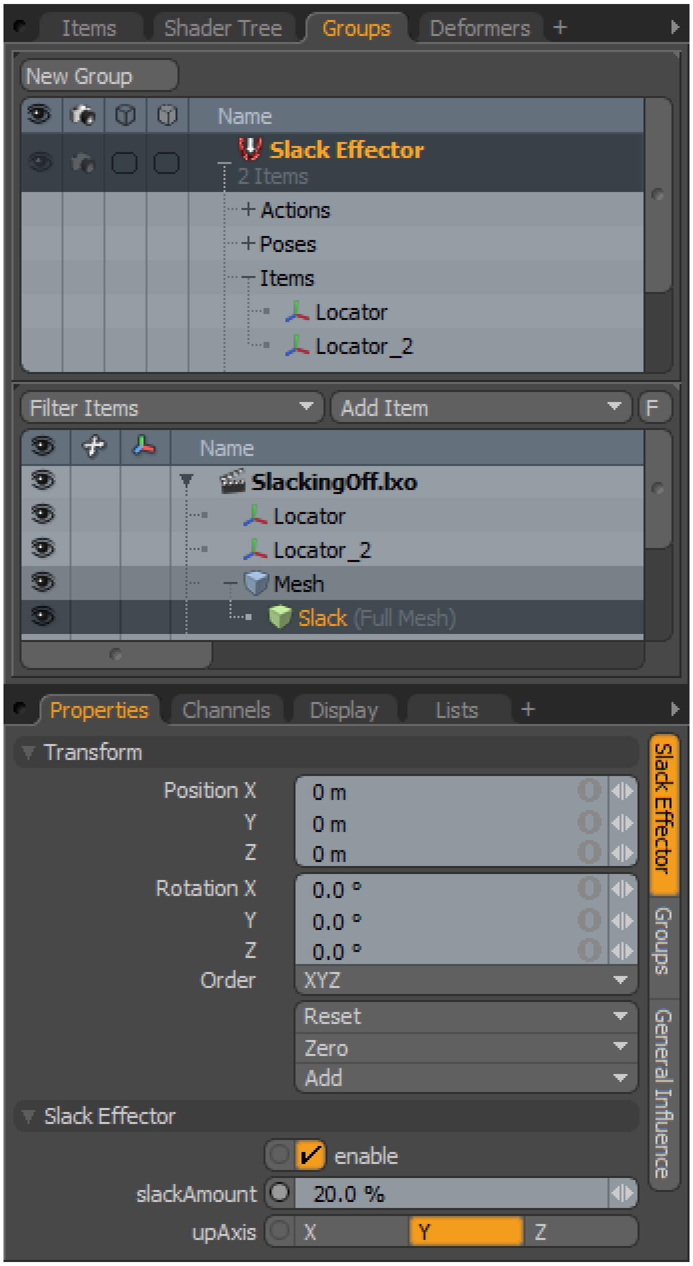
Slack Effector

Effectors are a means to affect a target item, generally a mesh layer, to deform it in specific ways. The **Slack Effector**, when assigned to an item, deforms the target as if it were suspended between two points. As the two points move closer together, the deforming element in between sags and droops but retains its overall initial length. For example, consider a rope holding a boat to a dock. As the boat drifts back and forth the rope droops and stretches between the boat and the dock, but it always retains its overall length.

Applying an Effector

You can add an Effector by using the **Item List**'s **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer by way of a General Influence. The General Influence acts upon the mesh, and the Effector, itself, produces the instructions on how the Influence should modify its target.

To deform items with a **Slack Effector**, you need to set it up properly beforehand to produce the intended result. First, the target item should be long and skinny in nature (such as a rope, a bar, or a chain with links) and be a single Mesh Item layer aligned with one of the three axes (X, Y, or Z). Second, position two additional Locator Items at each end to represent the start and end of the mesh. To automatically connect the Effector to the target, select it in the **Items** list and then right-click and choose **Add Mesh Operation,** then under **Effectors**, double-click **Slack Effector**. Modo creates the Effector as a special group (not as an Item in the **Items** list). Next, select the **Slack Effector** Group and the two previously created Locators. right-click the **Slack Effector** Group name and choose **Add Items**. Once you add the Locators to the Group, the **Slack Effector** immediately affects the target mesh. You can adjust the attributes in the **Properties** viewport panel.


Items
Shader Tree
Groups
Deformers
New Group
Name
Slack Effector
2 Items
Actions
Poses
Items
Locator
Locator_2
Filter Items
Add Item
F
Name
SlackingOff.lxo
Locator
Locator_2
Mesh
Slack (Full Mesh)
Properties
Channels
Display
Lists
Transform
Slack Effector
Groups
General Influence
Reset
Zero
Add
Slack Effector

Position X 0 m
Y 0 m
Z 0 m
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Order XYZ
☑ enable
slackAmount 20.0 %
upAxis X [✓ Y] Z

Name: Displays the current item's name. You may change the name by clicking the current name and typing a new name.

Transform

Position XYZ: Indicates the numerical position of the Effector item in XYZ space. These have no effect upon the resulting deformations.

Rotation XYZ: Indicates the rotation of the Effector item numerically. These have no affect upon the resulting deformations.

Order: Indicates the order that Modo applies rotations to the Effector item.

Scale XYZ: Sets the size of the item's representation in the 3D viewport numerically. These have no affect upon the resulting deformations.

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position and the mesh position intact. Modo does this by adding a negative transform item to the Mesh Item's channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

Slack Effector

Enable: Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Slack Amount: Determines the amount of droop or sagging between the two Locators. You can move the end Locators to increase or reduce the amount of drooping or sagging.

Up Axis: Specifies which axis (and direction) that Modo should use to apply the deformations. Gravity always pulls downward in the real world (the **Y** axis in Modo space), but not all scenes are meant to simulate the real world.

Spline Effector

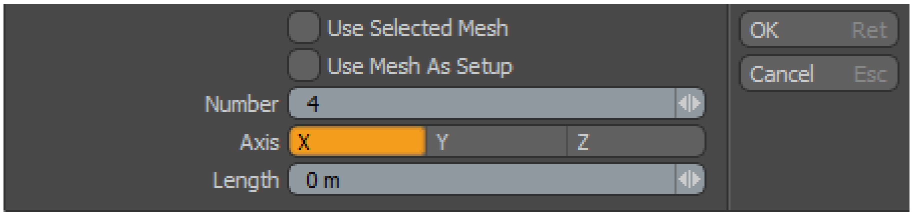
With the **Spline Deformer** you can apply deformations in a continuously smooth manner. This can be useful for long tube-like structures (such as tentacles, tails, and whips). The **Spline Effector** is functionally identical to the **Bezier Effector**, but it uses a hermite spline curve as the deformation base.

The Spline Deformer works as a group of Locators that define a Spline-style curve through their collective locations. Their initial positions (defined as the **Setup Position**) determine the undeformed state of the target mesh, and as Modo moves each Locator, it calculates the difference from the unmodified curve and applies it to the mesh as the deformation. (This is why Curve Deformers always appear to have two curves associated with them.) As with many functions of Modo, there are several ways for you assign and apply a spline curve for deformations.

No Initial Curve

You don't need to create a Spline curve prior to applying a Spline Deformer. Modo opens the **Create Spline** dialog box whenever you apply the Spline Deformer to a Mesh Item layer. The resulting curve aligns along one of the three axes in a straight line and centers on the specified axis. You either model the target geometry to match this requirement or enter **Setup Mode** after the fact and reposition the generated curve to coincide with appropriate locations relative to the geometry it's meant to deform.

To apply the Deformer itself, you first select the target Mesh Item layer either in the **Items** list or in the 3D viewport. With the target item selected, you then either click the **Spline** button in the **Deformers** sub-tab in the **Setup** interface layout, or you right-click the target layer in the **Items** list and choose **Add Mesh Operation**, then under **Effectors**, double-click **Spline Effector**.


○ Use Selected Mesh
○ Use Mesh As Setup
Number 4
Axis X Y Z
Length 0 m

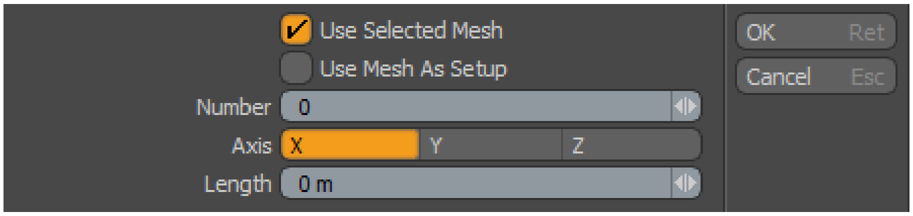
In the **Create Spline** dialog box, you only need to modify three values: **Number**, **Axis**, and **Length**. **Number** indicates the number of Locator control positions along the length of the curve. More control points provide finer control over the deformation, but they are also more difficult to deal with. Because the deformation smoothly attenuates between each location, you should define the minimum number of points necessary to obtain the desired effect. The **Axis** value indicates to Modo which axis it should use to generate the resulting spline. **Length** determines overall length of the resulting curve when centered at the World origin. Modo places the base of the generated curve on the negative side of the specified axis and draws it towards the positive direction. By clicking **OK** or pressing **Enter**, you save the values, and Modo generates the Spline Deformer.

If you need to make modifications, press the **Setup** button in the **Setup** interface layout to enter the **Setup** action state. In this state, you can modify the curve relative to the target geometry without generating any deformations. (This also temporarily disables any position changes applied to the curve outside of **Setup**.) When you properly position the curve, you can exit the **Setup** action state and change the positions of any Locators to deform the target mesh. To do this, select the associated Locator and move it with a transform tool. Set keyframes as necessary for a desired animation.

User Created Curve

To create a custom Spline curve to use as a Deformation controller, generate the curve in the same **Item** layer as the target mesh it's meant to deform. (You can also do this by cutting and pasting prior to invoking the add **Spline** deformer command.) You can draw the spline with the Bezier tool.

Once you create the Spline curve, you add it as a deformer to the mesh. To do so, you must be in the **Polygons** component mode with the target curve as the only curve selected. With the target Spline curve selected, you can then click the **Spline** interface button in the **Deformers** sub-tab in the **Setup** interface layout, or you can right-click the target layer in the **Items** list and choose **Add Mesh Operation**, then under **Effectors**, double-click **Spline Effector**.



☑ Use Selected Mesh

☐ Use Mesh As Setup

Number 0

Axis X Y Z

Length 0 m

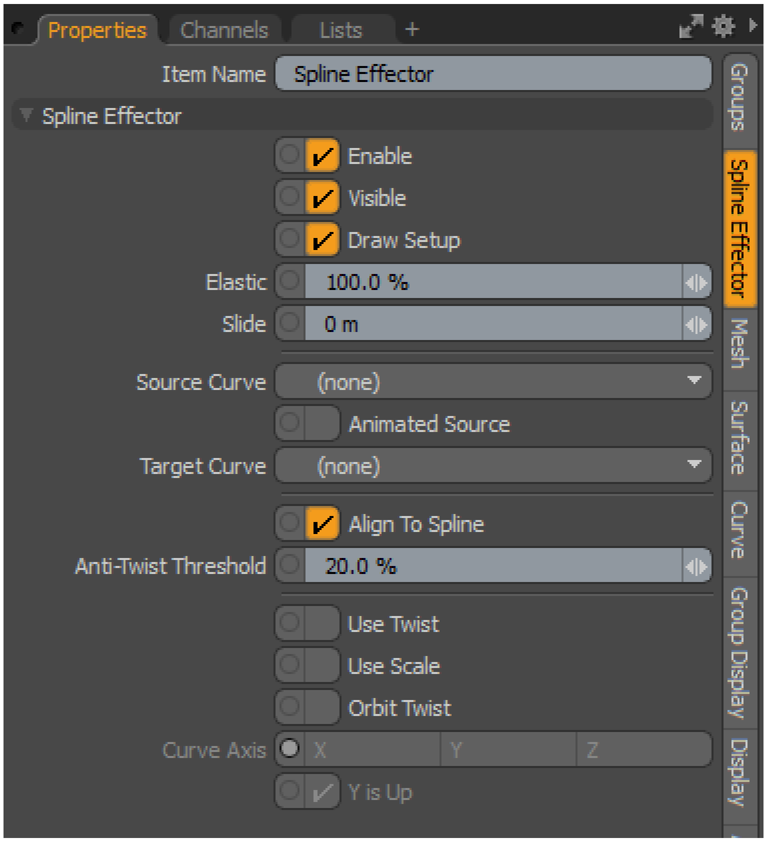
For a user-created curve, you only need to select **Use Selected Mesh** to have the tool use the selected curve. Everything else keys off the curve itself.

If you enable **Use Mesh As Setup**, Modo sets up the spline deformer so that it uses a curve polygon in the mesh to define the setup position for the deformer, then creates locators in appropriate places for the animated deformation.

After you click **OK** or press **Enter**, Modo accepts the values and generates the Spline Deformer. Once you apply the Spline Deformer, to deform the target mesh you only need to change the positions of any of the resulting Locators generated at each curve control location. Select the associated Locator and move it with any of the transform tools. You can set keyframes as necessary for a desired animation.

Spline Effector

To access the options related to the Spline Effector, select the **Spline Effector** item itself within the Groups Viewport as the parent of the Locators, or in the Deformers palette and view the associated attributes in the **Properties** panel.

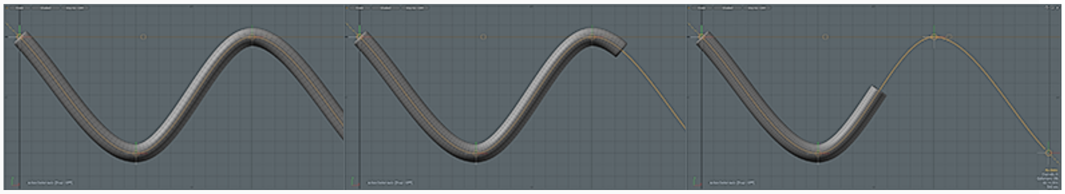
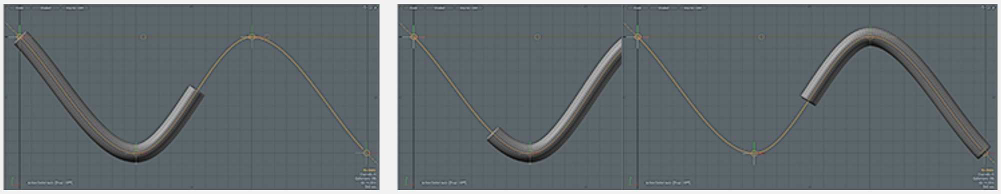


Item Name: Spline Effector

Spline Effector

☑ Enable
☑ Visible
☑ Draw Setup
Elastic: 100.0 %
Slide: 0 m
Source Curve: (none)
☐ Animated Source
Target Curve: (none)
☑ Align To Spline
Anti-Twist Threshold: 20.0 %
☐ Use Twist
☐ Use Scale
☐ Orbit Twist
Curve Axis: X Y Z
☑ Y is Up

Spline Effector	
Enable	Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.
Visible	Toggles the visibility of the curve representations in the 3D viewport.
Draw Setup	Toggles display of the setup curve in the 3D viewport.
Elastic	Determines to what degree Modo deforms the item as Modo stretches it along the curve to match its overall length. A setting of **100%** (the default) stretches the deformed area to the same length as the target curve. With a setting of **0%**, the item retains its original undeformed length along the curve. Deforming always originates from the start of the curve itself. For items that are shorter than the target deformation curve, use the **Slide** option to position the item along the curve.

Spline Effector	
	 Elastic Value 100% Elastic Value 50% Elastic Value 0%
Slide	Moves the target item along the length of the curve. Use this especially when the target item is shorter than the deforming curve. For example, use this to animate a snake along a predetermined path by setting the **Elastic** value to **0%** and then using the **Slide** option to deform the snake along the curve.  Slide Value 0m Slide Value 1m Slide Value 2m
Source Curve/Target Curve	Provides an additional way for you to deform a target mesh using a custom-created curve that is itself deformed by another Deformer. (This bypasses the need to create handles for manipulating the curve.) First, create the curve and then duplicate the curve as a new item to have two identical versions in separate item layers. (Each item layer should have only a single curve.) Next, assign the deformation to the **Target Curve**. (For example, create a Morph for the curve.) Next, add a **Spline Curve Deformer** to an **Item** but do not indicate a specific curve to have Modo create the Effector. You can then specify the deforming curve as the **Target Curve** and the undeformed version as the **Source Curve**.
Animated Source	Indicates when you animate the **Source Curve**. When you select this, Modo constantly evaluates the **Source Curve**. If clear, Modo only evaluates the **Source Curve**'s position on the first frame.

Spline Effector	
	Tip: When you see that a curve is reversed from what you expected, you can invert the point order by selecting the target curve in one of the component modes and pressing **F**. Now, what was the start is now the end and vice versa. (This is the same action as flipping a polygon's normal face).
Align To Spline	When enabled, aligns the deformation of the points to the tangent of the spline.
Anti-Twist Threshold	Defines the size of the region on the curve where the rotation axis is interpolated. **Tip:** You can avoid using the **Anti-Twist threshold** when creating circles or loops, by making the setup spline perpendicular to the deformed curve. So for a deformed loop lying in the XZ plane, it works best if the setup curve is along the Y axis.
Use Twist	When enabled, uses the rotation of the control handles to twist points around the spline axis.
Use Scale	When enabled, uses the scale of the control handles to scale the points.
Orbit Twist	When enabled, adds twist to rotate the deformed spline node around the rest position.
Curve Axis	Determines how Modo uses the Locator transform channels. Select **X**, **Y**, or **Z**. Rotations along the selected axis add twist; Modo ignores the other rotation axes. Scaling perpendicular to the axis stretches the mesh perpendicular to the **Curve Axis**.
Y is Up	Sets where the twist occurs (the twist coordinates' zero-twist value) if you have selected **Curve Axis**. In certain cases, you may want to clear this option to avoid kinks in a deformed mesh when twisting.

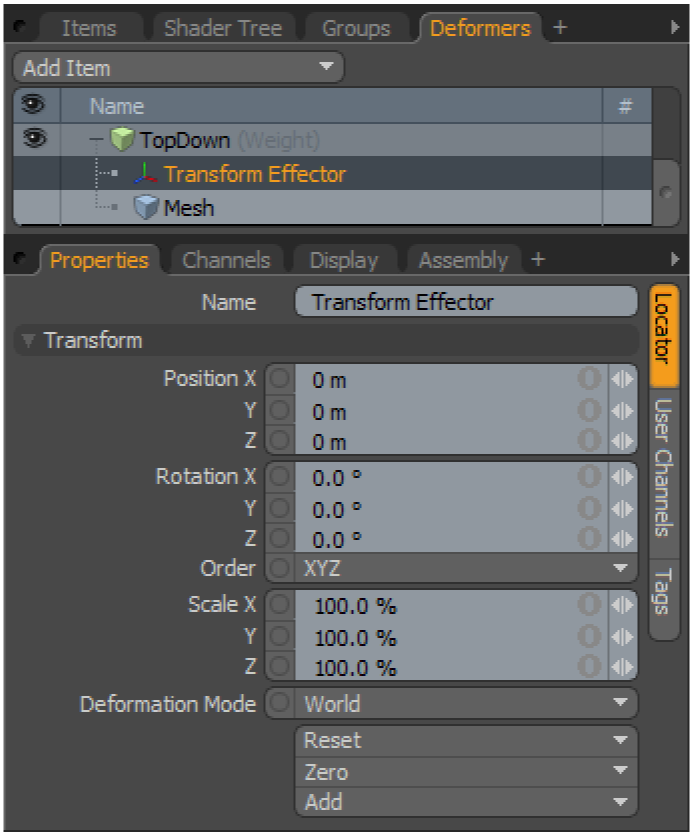
Transform Effector

Effectors are a means to affect a target item, specifically a mesh layer, deforming it in specific ways. When you assign the **Transform Effector** to an item, Modo moves, rotates, and scales the target item in the same way as regular item transforms would. With the **Transform Effector**, you can link it up to rigging and control it through the **Schematic** view links. When combined with various modifiers as an item with channels, this is different from a regular tool that you can only control when it is active.

Applying an Effector

You can add an Effector by using the **Item List**'s **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer by way of a General Influence. The General Influence acts upon the mesh, and the Effector, itself, produces the instructions on how the Influence should modify its target.

If you add the Effector in **Items** mode, Modo connects the Effector and the mesh layer automatically. First, select the target item(s) either in the **Items** list or any 3D viewport and then right-click the item and choose choose **Add Mesh Operation**, then under **Effectors**, double-click **Transform Effector**. Also, you can select the target item(s) and click the **Effector** button in the **Deformers** sub-tab of the **Layout** interface. Once you create the Effector, you can position it and modify its values to produce the desired results. When you apply an Effector to multiple items, Modo deforms each affected item based on the Effector's relative position to the mesh.



Name: Transform Effector

Transform

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Order: XYZ

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Deformation Mode: World

Reset

Zero

Add

Name: Displays the current item's name. You may change the name by clicking the current name and typing a new name.

Transform

Position XYZ: Indicates the numerical position of the Effector item in XYZ space. Changes in these values from the rest position transform the deformed target that amount. (Modo determines the rest position by the item's position in the **Setup** action state.)

Rotation XYZ: Indicates the rotation of the Effector item numerically. Changes in these values from the rest position transform the deformed target that amount.

Order: Indicates the order that Modo applies rotations to the Effector item. By changing this order, you can sometimes reduce or eliminate gimbal lock.

Scale XYZ: Sets the size of the item's representation in the 3D viewport numerically. Changes in these values from the rest position transform the deformed target that amount.

Deformation Mode: Determines how the transforms of the Effector translate into a deformation:

- **World** - This is the default setting and determines the deformation based on the World Position of the Effector. In most cases, this provides the expected results.
- **Local** - This applies the deformation relative to the **Setup** position of the item as if the item were still in the rest state regardless of the position of the Effector. This is useful for the order of operation workflow when rigging.
- **Local ** Rotation Modes** - This is identical to the **Local** mode, but Modo only applies the specified rotation. This creates a deformation that is more like a **Rotation Effector** with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position and the mesh position intact. Modo does this by adding a negative transform item to the Mesh Item's channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

Twist Effector

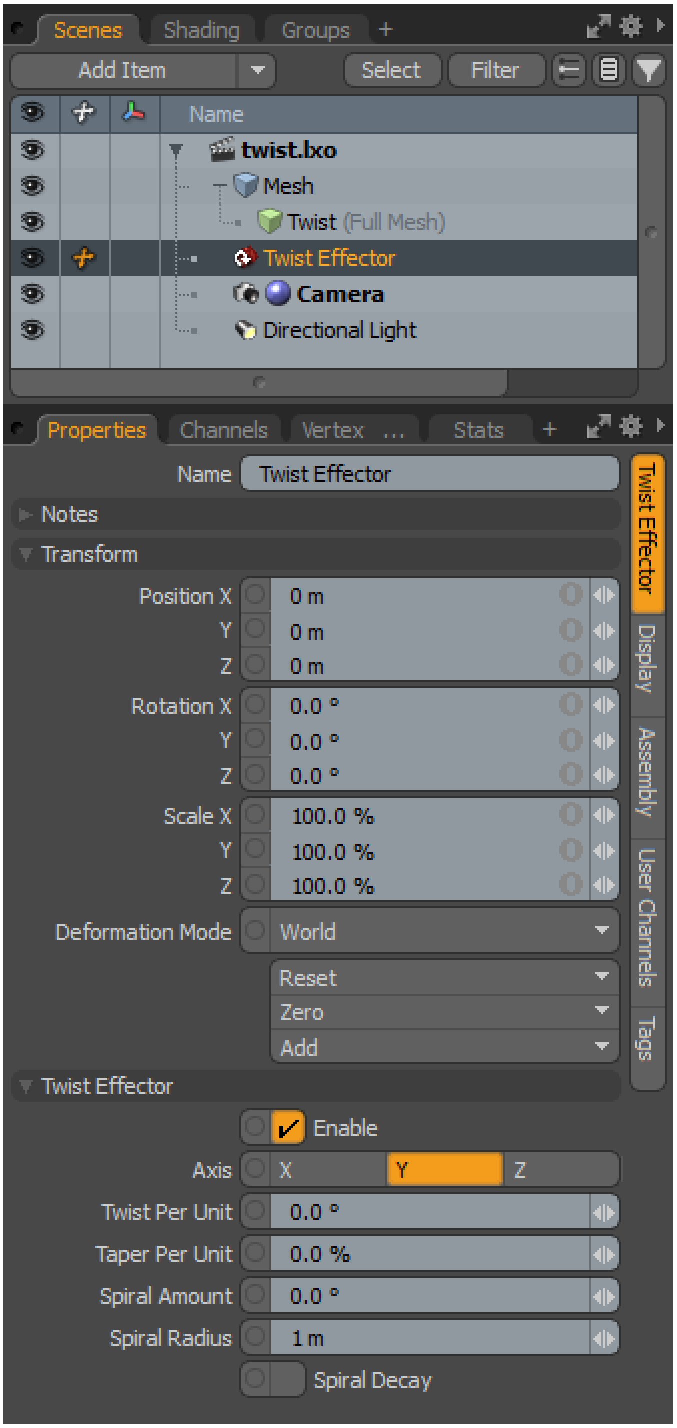
Effectors are a means to affect a target item, generally a mesh layer, to deform it in specific ways. The **Twist Effector**, when assigned to an item, deforms the object in a twisting motion. It has options for **Decay** to apply a spinning-tornado type of deformation to the target item.

Note: Before Modo 14.0v1, Twist Effector was called Vortex Effector.

Applying an Effector

You can add an Effector by using the **Item List**'s **Add Items** command, but it won't produce the intended result because it only places the Effector in the **Items** list. For proper results, you must connect the Effector to the mesh layer by way of a General Influence. The General Influence acts upon the mesh, and the Effector, itself, produces the instructions on how the Influence should modify its target.

If you add the Effector in **Items** mode, Modo connects the Effector and the mesh layer automatically. First, select the target item(s) either in the **Items** list or any 3D viewport and then right-click the item and choose **Add Mesh Operation**, then under **Effectors**, double-click **Twist Effector**. Also, you can select the target item(s) and click the **Effector** button in the **Deformers** sub-tab of the **Layout** interface. Once you create the Effector, you can position it and modify its values to produce the desired results. When you apply an Effector to multiple items, Modo deforms each affected item based on the Effector's relative position to the mesh.


Scenes
Shading
Groups
Add Item
Select
Filter
Name
Properties
Channels
Vertex ...
Stats
Twist Effector
Display
Assembly
User Channels
Tags

twist.lxo
Mesh
Twist (Full Mesh)
Twist Effector
Camera
Directional Light

Name: Twist Effector

Notes

Transform

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %
Deformation Mode: World
Reset
Zero
Add

Twist Effector

☑ Enable
Axis: X [✓ Y] Z
Twist Per Unit: 0.0 °
Taper Per Unit: 0.0 %
Spiral Amount: 0.0 °
Spiral Radius: 1 m
☐ Spiral Decay

Name: Displays the current item's name. You may change the name by clicking the current name and typing a new name.

Transform

Position XYZ: Indicates the numerical position of the Effector item in XYZ space. These values define the origin of influence for the Effector.

Rotation XYZ: Indicates the rotation of the Effector item numerically. These values define the direction of influence for the Effector.

Order: Indicates the order that Modo applies rotations to the Effector item. By changing this order, you can sometimes reduce or eliminate gimbal lock.

Scale XYZ: Sets the size of the item's representation in the 3D viewport numerically.

Deformation Mode: Determines how the transforms of the Effector translate into a deformation:

- **World** - This is the default setting and determines the deformation based on the World Position of the Effector. In most cases, this provides the expected results.
- **Local** - This applies the deformation relative to the **Setup** position of the item as if the item were still in the rest state regardless of the position of the Effector. This is useful for the order of operation workflow when rigging.
- **Local ** Rotation Modes** - This is identical to the **Local** mode, but Modo only applies the specified rotation. This creates a deformation that is more like a **Rotation Effector** with Parametric interpolation.

Reset: Resets the selected transform values to (0,0,0) and returns the item to its default state.

Zero: Resets the selected transform property values to **0** but leaves the item's position and the mesh position intact. Modo does this by adding a negative transform item to the Mesh Item's channels.

Add: Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)

Twist Effector

Enable: Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.

Axis: Determines the general directional **Axis** for the twisting motion.

Twist Per Unit: Indicates the amount of rotation that Modo adds for every unit of distance from the base of the Twist.

Taper Per Unit: Specifies the change in the radius for every unit of distance from the base of the Twist.

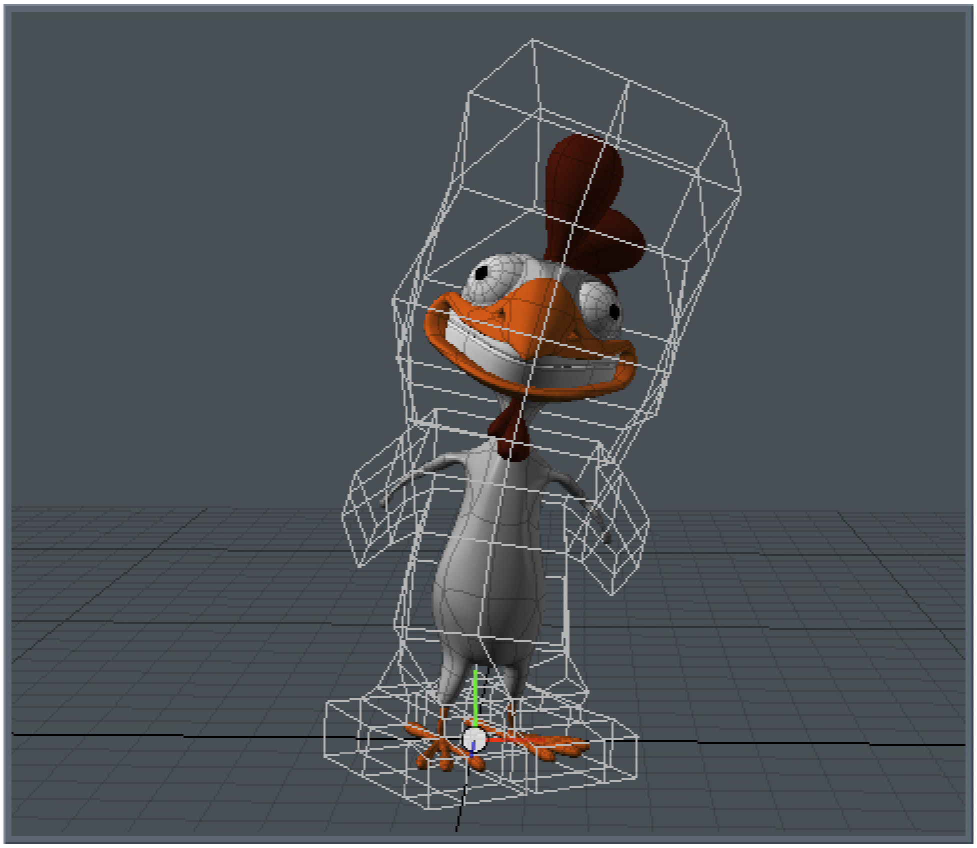
Spiral Amount: Indicates the rotation amount that Modo adds as the effect moves away from the center axis of the cylinder. The angle equals the **Spiral Amount** when the distance equals the **Spiral Radius**.

Spiral Radius: Specifies the distance from the axis at which the spiral rotation that Modo adds is equal to the **Spiral Amount**.

Spiral Decay: Sets whether the spiral increases or decreases with distance along the axis.

Wrap Influence

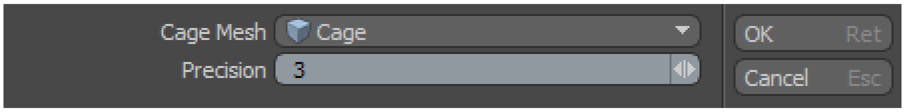
The Wrap Influence (also known as Wrap Deformer), when assigned to a mesh, creates a custom, low-resolution cage as a proxy to deform the associated mesh. Typically the cage approximates the shape of the target mesh layer to make it easier to manipulate and control. Once assigned, you can animate the Control cage by using other deformers (such as Joints or Morphs) to produce animated results.



Applying the Wrap Influence

Before applying a Wrap Influence, you need to create both a target mesh that is to be deformed in its own Item layer and a secondary low-resolution proxy mesh, also in its own layer, to act as the deformation cage. To apply the Wrap Influence follow these steps:

1. Select the target mesh.
2. Open the **Setup** workspace layout, and select the **Deformers** sub-tab.
3. Click the **Wrap** button. This opens the **Create Wrap Deformer** dialog box.



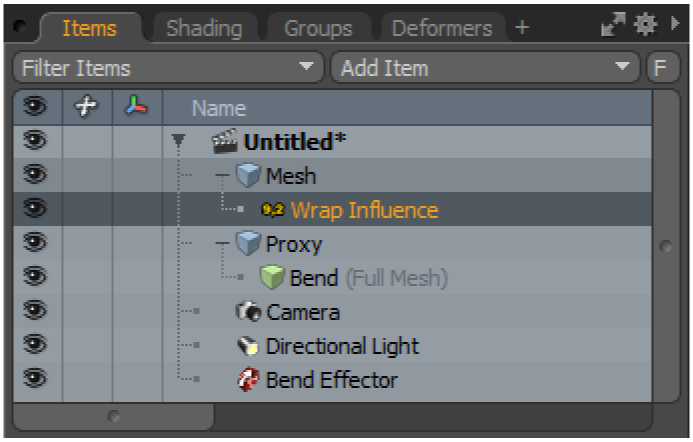
Cage Mesh: Cage
Precision: 3

OK Ret
Cancel Esc

4. Select the low resolution mesh to use as the influence as **Cage Mesh**.
5. Set the **Precision**.

   Modo uses this to compute the weight values for binding the cage to the target. Higher precision values generate more accurate weights for deforming the target, but they also take longer to calculate; therefore, don't set this value higher than necessary.
6. Click **OK**.

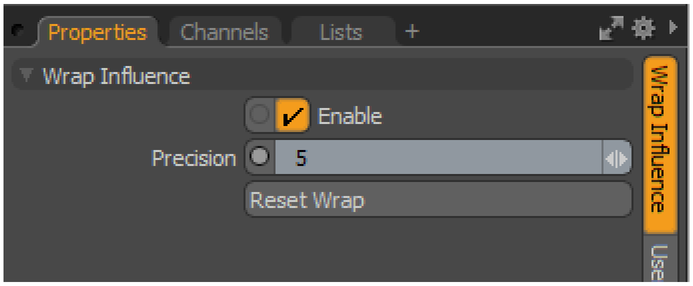
   You can now manipulate the cage mesh to deform the target mesh.



Items
Shading
Groups
Deformers
Filter Items
Add Item
F
Name
Untitled*
Mesh
Wrap Influence
Proxy
Bend (Full Mesh)
Camera
Directional Light
Bend Effector

Once you assign the cage deformer, you can add additional deformer types to the proxy cage Mesh Item and animate them to deform the high-resolution target mesh. You can select the **Wrap Effector** item, itself, (in the Items (Scenes) List or Deformers viewport) and adjust its attributes in the **Properties** viewport.

Wrap Influence Properties



- **Enable** - Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.
- **Precision** - Determines the accuracy of the Weight Maps that connect the low-resolution proxy mesh to the high-resolution target, which in turn affects the deformation quality. If you change this value, Modo re-evaluates the weights for the associated items. Higher values may take longer to calculate (possibly much longer) depending on the number of vertices in the proxy mesh.
- **Reset Wrap** - Restores the mesh to its original state (without the deformation) but preserves the wrap geometry in its current state. This lets you use the newly transformed wrap geometry on the original target mesh. You can also alter the **Precision**.

Wrap Influence Procedural

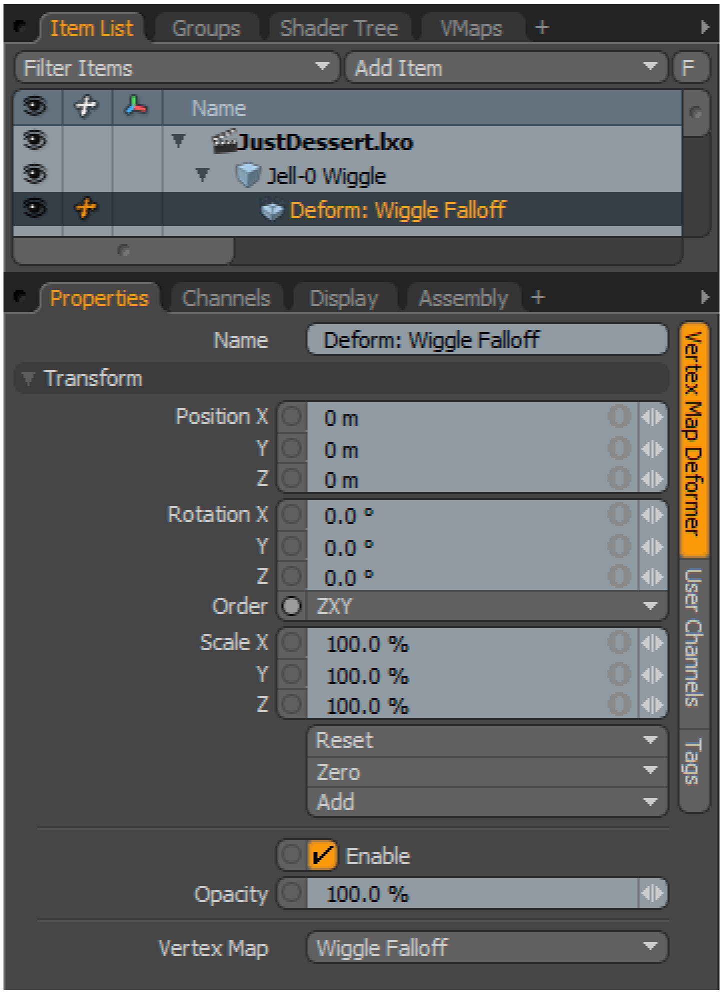
The procedural version of the Wrap Influence has the following sub-tools:

- **Falloffs** - This allows you to add one or more falloffs to attenuate the effect of the tool. Click **(Add Falloff)** to open the Preset Browser. From there you can select an existing falloff (from your scene) or add a new falloff.
- **Cage Mesh** - This holds a reference to the low resolution mesh that is used as a cage to deform the target mesh.
- **Geometry** - This lists any geometry that is affected by the tool. Meshes are connected automatically if they are below the tool in the **Mesh Operations** list. You can connect additional meshes by clicking **(Add Geometry)** to open the Preset Browser. From there you can select an existing mesh or add a new, empty mesh.

Weight Map Deformer

With a Weight Map deformer, you can produce organic deformations based on a weight Vertex Map. Before applying the deformer, generate a weight Vertex Map to represent the affected vertices. Weight Maps can attenuate, so some vertices are more affected than others. This modulates the overall effect. Once you create a weight Vertex Map, apply the Vertex Map deformer to the Item and select the target Weight Map with the **Vertex Map** option. Once you assign a deformer, Modo positions a locator at the center of the equivalent bounding box of the Weight Map. This locator now controls the geometry through the Weight Map, so transforms applied to the locator deform the geometry appropriately. You can adjust the position of the locator by moving its Pivot position. Animating the locator produces animated deformations otherwise not possible in Modo.

VMap Deformer


Item List
Groups
Shader Tree
VMaps
+
Filter Items
Add Item
F
Name
JustDessert.lxo
Jell-0 Wiggle
Deform: Wiggle Falloff
Properties
Channels
Display
Assembly
+
Transform
Reset
Zero
Add
Vertex Map Deformer
User Channels
Tags

Name: Deform: Wiggle Falloff
Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %
☑ Enable
Opacity: 100.0 %
Vertex Map: Wiggle Falloff

Name: This specifies the name of the deformer item.

The following **Transform** options are available for the **Weight Map Deformer**:

Position: Use for an item transform to numerically position the deformer item in XYZ space. By default, position transforms originate from an item's pivot position.

Rotation: Use for an item transform to numerically set the rotation of the deformer item. By default, rotation transforms originate from an item's pivot position.

Order: Use to set the order that Modo applies rotations to the deformer controller item. Changing the order that rotations are applied can help to reduce or eliminate gimbal lock.

Scale: Use to define the amount of scale applied to the deformer item. By default, scale transforms originate from an item's pivot position.

Reset: Resets the selected transform values to default Worldspace center (0,0,0) and returns the controller item back to its initial position.

Zero: This is disabled for weight deformers.

Add: This adds the selected set of transforms to the channel list. When animating complex motions, this provides a layered approach to the transforms.

Enable: Use to activate or deactivate the deformer. You can use this as a way to turn off a weight deformer temporarily. You can also click the **eye** icon in the visibility column of the **Item List** tab.

Opacity: Use to modulate the strength of the deformer item particularly when you are blending multiple deformer items together.

Vertex Map: Use to select the Weight Vertex Map to apply to the geometry for deforming.

Using Deformer Falloffs

Falloffs have a multitude of uses, mainly:

- As Deformers, where their area of influence can be assigned to deform the target mesh (similar to having a magnet that pushes and pulls a surface).
- To attenuate the effects of other Influences upon an item or mesh. For example, a modeling falloff can control the amount of influence a tool has on a surface.
- To control the degree to which a Force influences a Dynamic item (such as a Rigid or Soft Body). How the falloff affects other items depends on how you add it to the scene. The influence shape is related to its type.

Adding a Falloff as a Deformer

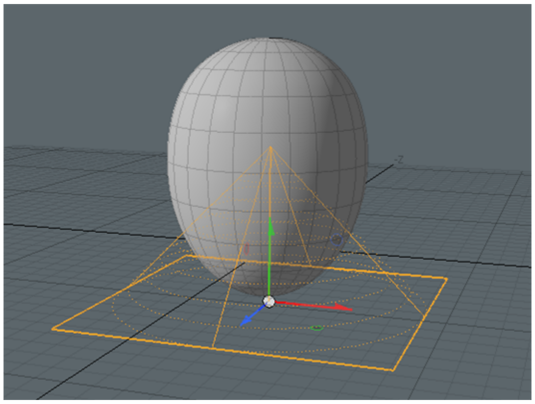
You can add a falloff item to a scene by using the **Add Item** function of the **Items** list, but the falloff has no direct effect upon any items in the scene until you link it to the target item. You can do this manually by using the Schematic view.

Modo offers a few ways to create these links automatically. For example:

1. In **Items mode**, select the target mesh item.
2. In the **Setup** interface tab in the **Deformers** sub-tab of the toolbox, select the target falloff type in the **Add Falloff Deformer** list.

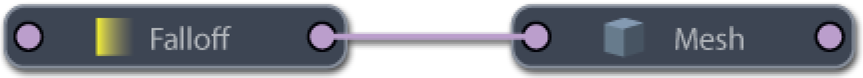
   This adds the falloff and links it to the selected mesh using the falloff's **Deform Meshes** input.

If you move the falloff now, Modo also moves any points within the influence of the falloff.



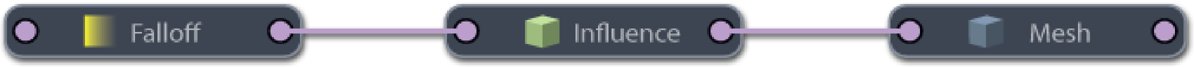
A sphere primitive deformed by a linear falloff.

You can adjust the initial **Rest** (undeformed) state by clicking the **Setup** button. (In Setup mode, the viewport frame has a yellow outline.) Position the falloff to where it should be in its undeformed state. After exiting Setup, you can animate or adjust the position of the falloff and deform the target mesh.



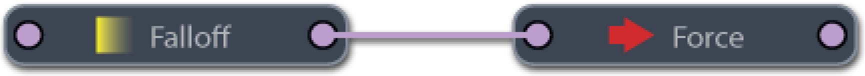
Adding a Falloff to an Influence

To have a falloff attenuate an Influence, you use the Deformers viewport where you can automatically create and link a falloff to the Influence in a single step. Within the viewport, select the target Influence item, right-click the item, and in the shortcut menu click **Add Override > Falloff > Falloff Type** to add it to the scene. Once added, you can animate the Position's Size and Range. The Influence only affects the target mesh within the bounds of the falloff. (You can invert this behavior, if desired.)



Adding a Falloff to a Force

You can also automatically link a falloff to a Dynamics Forces item to attenuate the effects of the force on dynamic items. To do this, in **Items mode** select the **Force Item** and use the **Add Falloff** option found in the **Dynamics** sub-tab of the same toolbox. With the falloff active, dynamic items within the bounds of the falloff are not affected to the full degree of the Force and attenuate to the core of the falloff where Items aren't affected at all.



Adding Falloffs in Procedural Modeling

You can add the procedural versions of falloffs in the **Mesh Operations** list. In the **Model** layout, you can find this in the lower half of the **Mesh Ops** tab, on the right side of the interface.

Note: For more information on the **Mesh Ops** tab, see Using the Mesh Ops Tab.

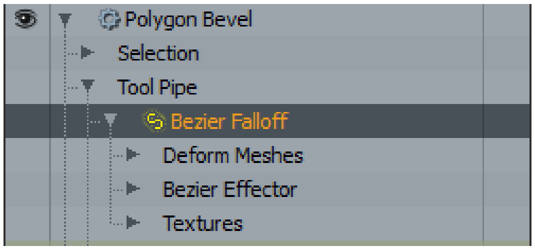
Most of the time, you add a falloff to the Tool Pipe of a mesh operation to modulate the effect of the operation.

You can add falloffs to procedural mesh operations the following way:

1. In the **Mesh Operations** list, expose the tool's **Tool Pipe** input and click **(Add Tool Pipe)**.

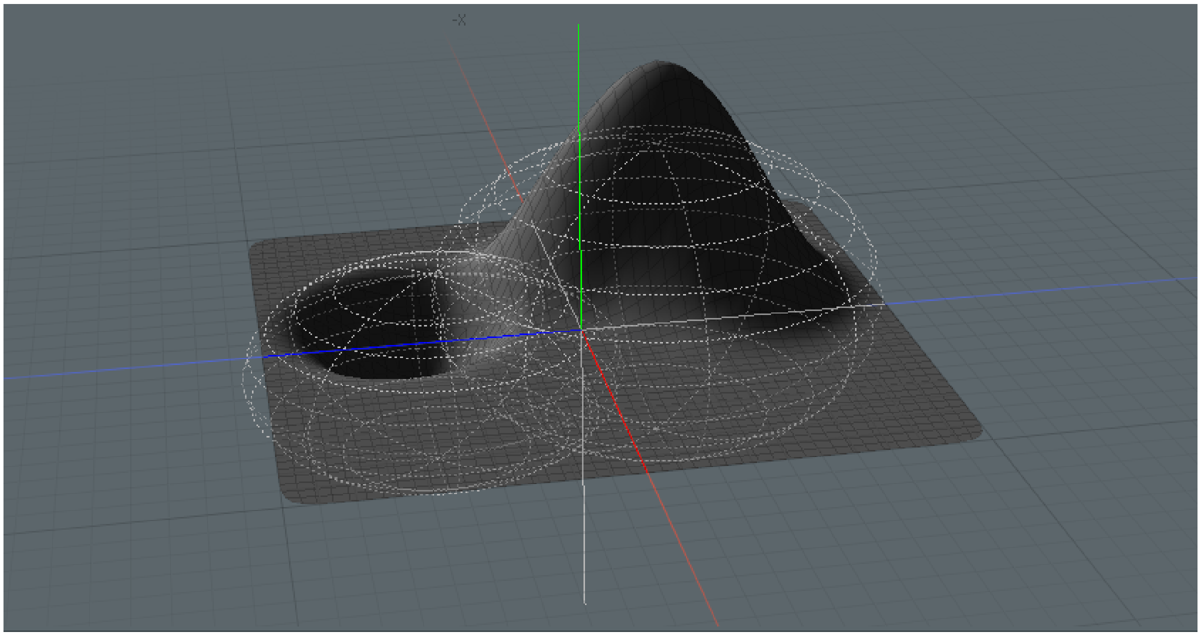
   This opens the **Preset Browser**, where you can select the falloff you need.
2. Double-click a falloff to add it to the list.

The image below shows a Bezier Falloff added to the Tool Pipe of a Polygon Bevel operation in the **Mesh Operations** list:


Polygon Bevel
  Selection
  Tool Pipe
    Bezier Falloff
      Deform Meshes
      Bezier Effector
      Textures

Blending Falloffs

When using multiple falloffs to attenuate the effect of a tool, falloffs behave additively and are applied one after the other in the order you add them. However, to achieve a different effect, you can also combine multiple falloffs into one by using a Blend Falloff. A Blend Falloff is a falloff that takes other falloffs as inputs and combines them based on the math operation you select in its properties.



You can work with Blend Falloffs in the Schematic Viewport or the Mesh Operations List.

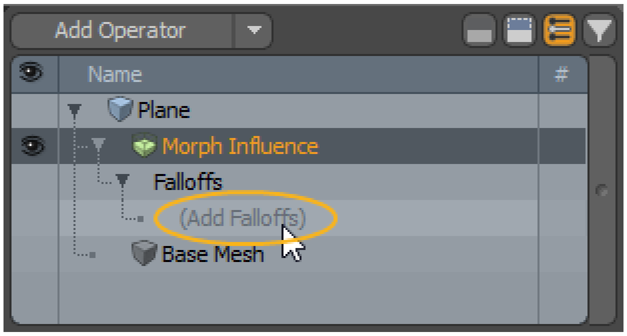
Blending Falloffs in the Mesh Operations List

Make sure you have a mesh and an operation affecting it, that you can add falloffs to. Here the Morph Influence is used as an example.

1. On the right panel, open the **Mesh Ops** tab.

> **Note:** If you're working in a layout where the **Mesh Ops** tab is not visible by default, click the **+** button on the right of the tab names, and select **Data Lists** > **Mesh Ops**.
> For more information about the **Mesh Ops** tab and how mesh operations work, see Procedural Modeling with MeshOps.

2. In the Mesh Operations list, expand the operation to which you want to add the falloff.
3. Expand the operation's **Falloff** input and click **(Add Falloff)**.


Add Operator
Name
#
Plane
Morph Influence
Falloffs
(Add Falloffs)
Base Mesh

4. In the **Add Operator window**, navigate to **New** > **Falloffs**, then double-click **Blend Falloff**.

   The Blend Falloff is added to the selected operation in the Mesh Operations list.

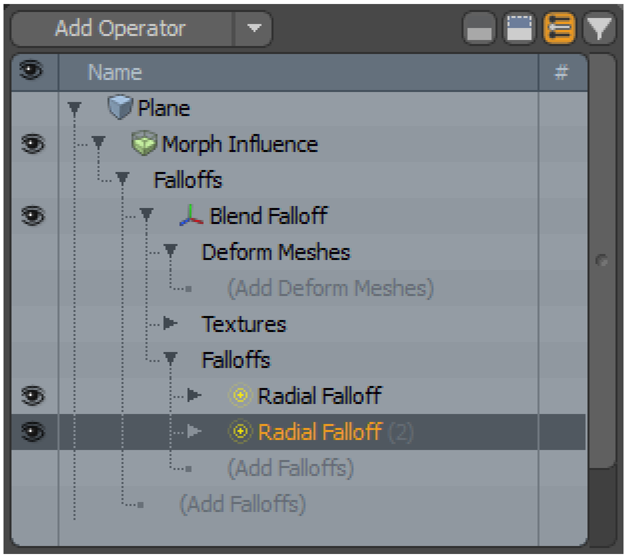
5. To add the falloffs to blend, expand the **Blend Falloff** item in the list.

   This reveals its **Falloff** input.

6. Expand the operation's **Falloff** input and click **(Add Falloff)**.

7. In the **Add Operator window**, navigate to **New** > **Falloffs**, then double-click the falloff you want to add. Repeat this to add the required falloffs.

   The example below uses a Radial Falloffs.


Add Operator
Name
#
Plane
Morph Influence
Falloffs
Blend Falloff
Deform Meshes
(Add Deform Meshes)
Textures
Falloffs
Radial Falloff
Radial Falloff (2)
(Add Falloffs)
(Add Falloffs)

8. Adjust each falloff by selecting it in the list, and editing its properties in the **Properties** tab in the lower right panel.

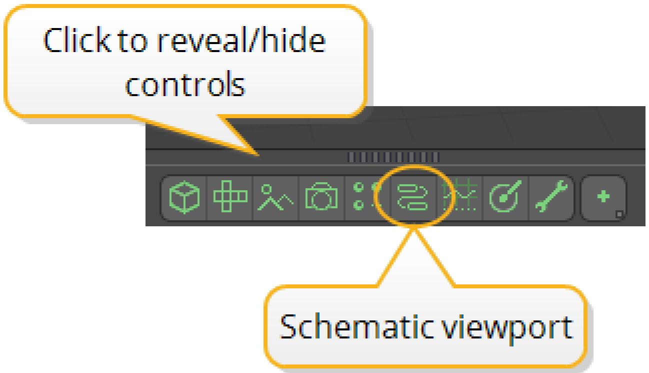
9. Select the Blend Falloff in the Mesh Operations list to open its properties in the lower right panel. Edit the properties to achieve different effects. For more information, see Blend Falloff Properties.

Blending Falloffs in the Schematic

Make sure you have a mesh and an operation affecting it, that you can add falloffs to. Here the Morph Influence is used as an example.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport button.


Click to reveal/hide controls
Schematic viewport

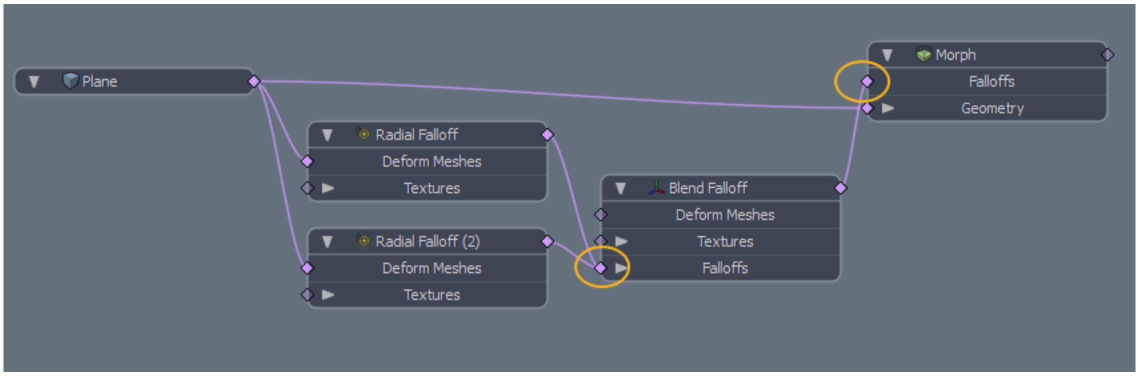
>

MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

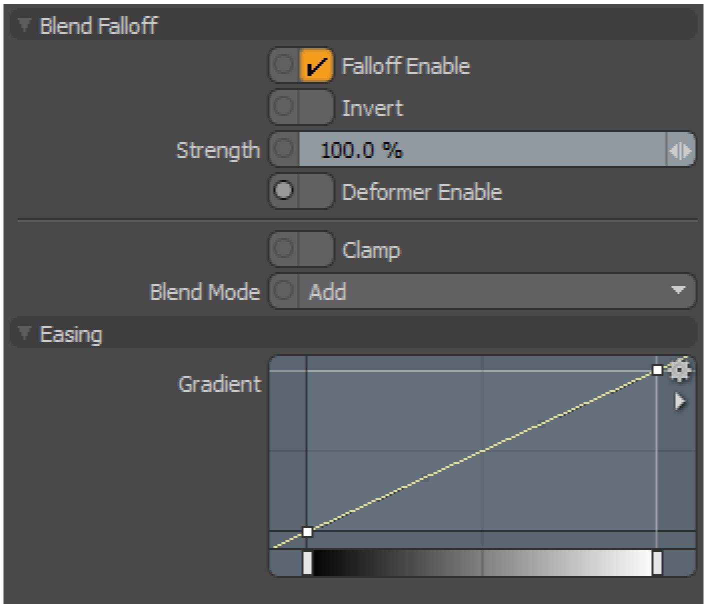
1. Click **Add...**, navigate to **Mesh Operations** > **Tool Pipe** > **Falloffs**, and double-click **Blend Falloff**.
2. Add the other required falloffs the same way.
3. Connect the **Blend Falloff** node to the required operation's **Falloff** input, and the falloffs to the **Blend Falloff** node's **Falloff** input.

   In the example below, the **Blend Falloff** is connected to the Morph Influence node (**Morph** in this case) and combines the two Radial Falloffs.


Plane
Morph
Falloffs
Geometry
Radial Falloff
Deform Meshes
Textures
Blend Falloff
Deform Meshes
Textures
Falloffs
Radial Falloff (2)
Deform Meshes
Textures

4. Adjust each falloff by selecting it and editing its properties in the **Properties** tab in the lower right panel.
5. Select the Blend Falloff node in the Schematic to open its properties in the lower right panel. Edit the properties to achieve different effects. For more information, see Blend Falloff Properties.

Blend Falloff Properties



Blend Falloff

☑ Falloff Enable

☐ Invert

Strength 100.0 %

○ Deformer Enable

☐ Clamp

Blend Mode Add

Easing

Gradient

Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. For more information, see Applying Deformers.
Clamp	When disabled, the falloffs are applied additively, one after the other. When enabled, they are clamped.
Blend Mode	Determines the math operation used to combine the falloffs. You can choose between **Add**, **Subtract**, and **Multiply**.
Gradient	Allows you to re-map the weight of a falloff by creating a new custom falloff shape. In the Graph Editor, press **Ctrl**/**Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create a custom falloff shape. For more information, see Key Creation.

Falloffs

Click any link below to view the topic for a specific falloff.

Area Falloff

Bezier Falloff

Capsule Falloff

Curve Falloff

Falloff Operator

Incidence Falloff

Linear Falloff

Radial Falloff

Spline Falloff

Texture Falloff

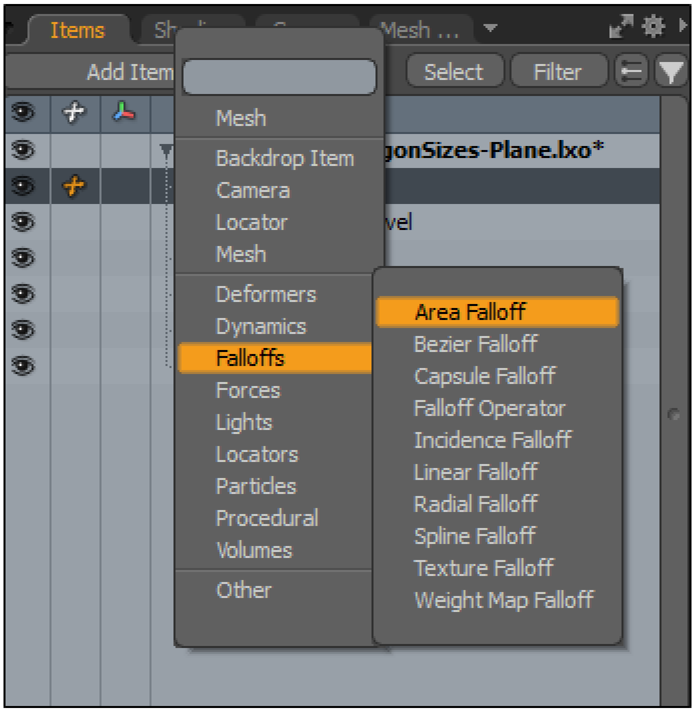
Weight Map Falloff

Area Falloff

The Area Falloff returns a weight value determined by the area of the polygon being evaluated. By applying a Mesh Operation, such as Polygon Bevel, it bevels the mesh based on the polygon area. This is useful when you have meshes which have polygon using different sizes and you want to control the shape of the mesh.

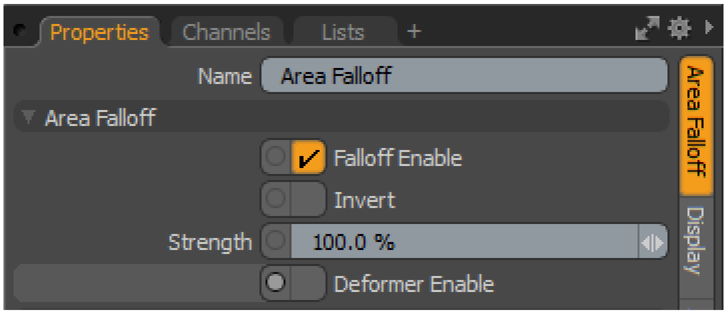
To activate the Area Falloff tool:

1. Open the **Model** layout, on the right panel, open the **Items** tab.
2. Click the down arrow beside **Add Item** and select **Falloffs** > **Area Falloff**.


Items
Sh
Mesh ...
Add Item
Select
Filter
gonSizes-Plane.lxo*
vel
Mesh
Backdrop Item
Camera
Locator
Mesh
Deformers
Dynamics
Falloffs
Forces
Lights
Locators
Particles
Procedural
Volumes
Other
Area Falloff
Bezier Falloff
Capsule Falloff
Falloff Operator
Incidence Falloff
Linear Falloff
Radial Falloff
Spline Falloff
Texture Falloff
Weight Map Falloff

- Alternatively, open the **Setup** layout, on the bottom panel, click **Add**, and select **Mesh Operations** > **Tool Pipe** > **Falloffs** > **Area Falloff**.

Area Falloff Properties


Properties
Channels
Lists
+
Area Falloff
Display

Name: Area Falloff

Area Falloff

☑ Falloff Enable

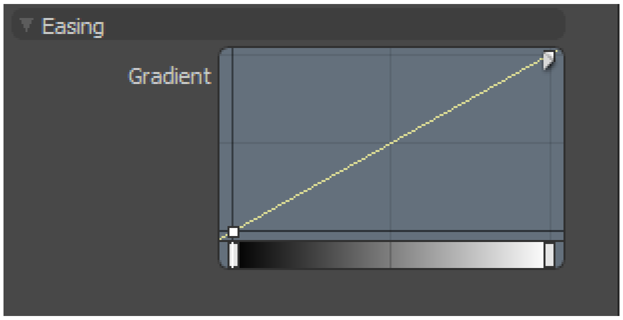
☐ Invert

Strength: 100.0 %

○ Deformer Enable

Name	Displays the Weight Map Falloff item name.
Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. For more information, see Applying Deformers.

Easing Properties



Gradient	Allows you to re-map the weight of a falloff by creating a new custom falloff shape. In the Graph Editor, press **Ctrl**/**Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shape. For more information, see Key Creation.

Bezier Falloff

As with modeling, you use a falloff to attenuate the effect that any deformer has on a target item and to provide additional control over how deformers influence a mesh. The Bezier Falloff is almost identical to the Spline Falloff, but works with Bezier curves instead of the simpler Spline curves. When working with the Bezier Falloff, you use special **Bezier Node** Locators that shape the resulting curve and, with the additional control handles, you can shape the resulting Bezier curve.

Using the Bezier Falloff

To add a Bezier Falloff to a scene, you must first add a Bezier Deformer. The falloff item uses this deformer and its resulting curve to produce the falloff. The deformer itself does not have to be active for the falloff to function, but the falloff needs it to be present as a container for the Bezier Node handles. Also, it is easier to use the **Bezier Deformer** and Bezier Falloff together without duplicating the actual curve.

Once you define a Bezier Deformer, you add the Bezier Falloff to it.

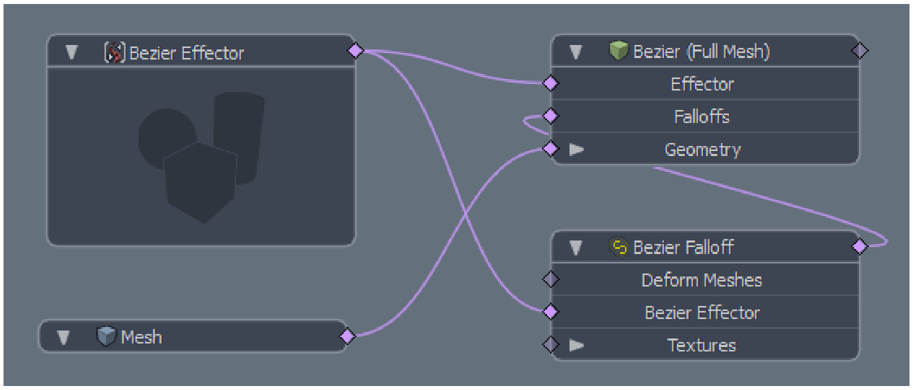
1. Right-click the **Bezier (Full Mesh)** layer within the Deformer viewport or the Items (Scenes) List and choose **Add Override > Falloffs > Bezier Falloff**.
2. Once created, connect the resulting Bezier Effector manually to the the Bezier Falloff's **Bezier Effector** connector.

> **Note:** If you're not familiar with the **Schematic** viewport, see Schematic viewport for more information.

3. To create the link, drag a connector between the Bezier Effector's relationship output (the diamond connector) and the **Bezier Effector** input of the Bezier Falloff.

> **Note:** For the falloff to adjust the range of the deformation rather than its shift along the length of the curve, you must set the **Bezier (Full Mesh)** item's **Interpolation** mode to **Linear**.

Once defined, the falloff properly attenuates the effects of the Bezier Deformer.



4. At this point, select the **Bezier Effector** item, and disable it in its **Properties** panel.
5. Then add any additional Deformers to the item and link the **Bezier Falloff** output to the resulting Deformer Item's **Falloffs** input.

   The falloff attenuates the influence of the new deformer. The falloff's position relative to the undeformed item determines which parts of the target item Modo affects.

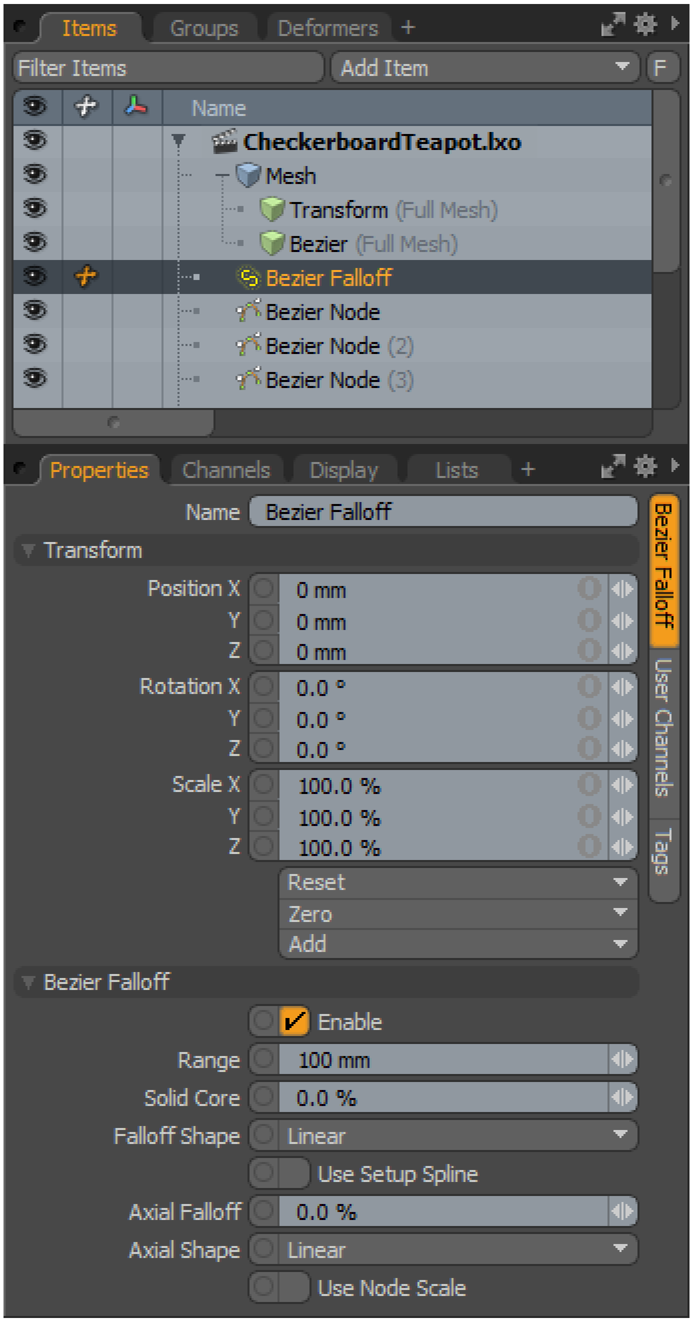
Note: See the Schematic viewport topic for more information about adding items to the viewport and linking the items together.

Bezier Falloff in the Procedural

Note: For details on how to add falloffs to the **Mesh Operations** list, see Adding Falloffs in Procedural Modeling.

In the **Mesh Operations** list, the Bezier Falloff has the following inputs:

- **Deform Meshes** - This allows you to use the falloff as a deformer on a mesh. You can connect a mesh to the falloff to transform the vertices, modulated by the falloff.
- **Textures** - You can specify a texture to be used by the falloff to attenuate its effect.
- **Bezier Effector** - The Bezier Falloff operation requires a Bezier Effector's curve to produce the falloff. For more information on the Bezier Effector, see Bezier Effector.


Items
Groups
Deformers
Filter Items
Add Item
F
Name
CheckerboardTeapot.lxo
Mesh
Transform (Full Mesh)
Bezier (Full Mesh)
Bezier Falloff
Bezier Node
Bezier Node (2)
Bezier Node (3)
Properties
Channels
Display
Lists
Bezier Falloff
User Channels
Tags
Transform
Reset
Zero
Add
Bezier Falloff

Name: Bezier Falloff

Position X: 0 mm
Y: 0 mm
Z: 0 mm

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable

Range: 100 mm

Solid Core: 0.0 %

Falloff Shape: Linear

☐ Use Setup Spline

Axial Falloff: 0.0 %

Axial Shape: Linear

☐ Use Node Scale

Bezier Falloff Properties	
Name	Displays the current item's name. You may change the name by left-clicking the current name and typing a new name.
Transform	
Position XYZ	Indicates the numerical position of the Effector item in XYZ space. Position

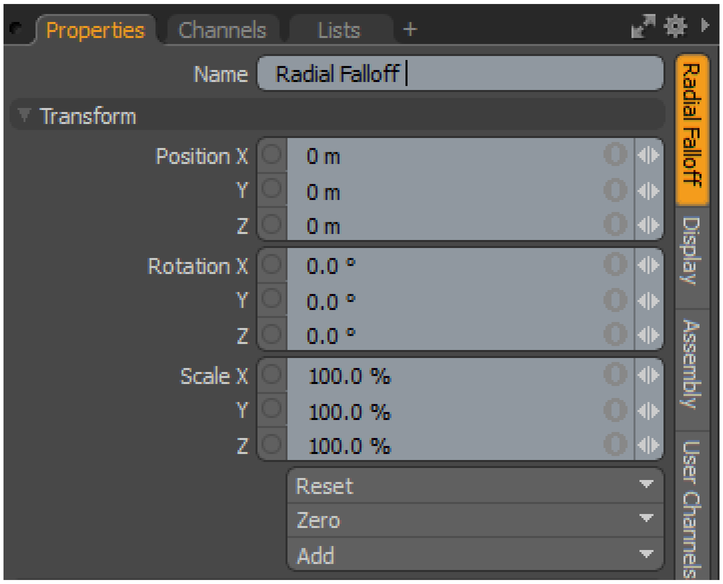
Bezier Falloff Properties	
	transforms originate from the Center position. Because the Bezier Falloff is dependent on its associated Locators, this option produces no useful results.
Rotation	Indicates the rotation of the Effector item numerically. Because the Bezier Falloff is dependent on its associated Locators, this option produces no useful results.
Scale	Sets the size of the item's representation in the 3D viewport numerically. Because the Bezier Falloff is dependent on its associated Locators, this option produces no useful results. Scaling the Locators only has an effect when you select the **Use Node Scale** option.
Reset	Resets the selected transform values to (0,0,0) and returns the item to its default state.
Zero	Resets the selected transform property values to **0** but leaves the Center position and the item's position intact.
Add	Adds a Transform item to control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Only add the necessary transforms on an as-needed basis to reduce scene overhead. You can also use this **Add** function to add a selected set of transforms to the Channel list while keeping the default 0,0,0 values, which is necessary for Referencing. (To override channels, they must exist.)
Bezier Falloff	
Enable	Toggles the effect of the layer on or off. When clear, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.
Range	Defines how far away from the Bezier curve that the falloff influences its target.
Falloff Shape	Controls the rate of attenuation across the falloff with four options. • **Linear** - Provides a straight attenuation of the falloff's influence across its total range. • **Ease In** - Gives a smooth attenuation with greater influence toward the Start of the falloff's range. • **Ease Out** - Gives a smooth attenuation with greater influence toward the End of the falloff's range. • **Smooth** - Provides a smooth attenuation with a slightly stronger influence in the

Bezier Falloff Properties	
	middle of the falloff's range.
Use Setup Spline	Select when the Spline is both a deformer and a falloff. When selected, Modo uses the Setup Spline shape to compute distances between the vertices and spline. (The Setup Spline is how it looks in its Setup state.) When you use the Spline as both a deformer and a falloff, Modo determines the falloff on the undeformed vertex positions. Then, the deformer can pull the mesh out the reach of the falloff and cause deformation artifacts. With **Use Setup Spline** selected, Modo produces a more desirable result.
Axial Falloff	Controls the strength of the falloff along the length of the curve. A value of **0%** generates no axial falloff. A value of **100%** attenuates the falloff's strength along the curve with the maximum influence at the base of the curve and no influence at its termination point; a value of **-100%** produces the inverse: no influence at the base and the maximum strength at the terminator. (You can use values of greater than 100% to achieve unusual results.)
Axial Shape	Controls the attenuation amount along the length of the curve when you set **Axial Falloff** to a value greater than **0%**. This duplicates the functionality of the **Falloff Shape** option, but Modo applies it along the length instead of outward from the Bezier Falloff's core.
Use Node Scale	When selected, you can scale the Locator(s) that make up the Bezier Falloff's curve that then scales the **Range** value. This acts as a multiplier on a per node basis. In the unselected, default state, node scaling has no effect on the falloff's **Range**.

Capsule Falloff

The Capsule Falloff produces an attenuation area shaped as a cylinder with rounded ends. It has options for you to adjust the size, the shape, and the transition of the falloff itself.

Transform Properties

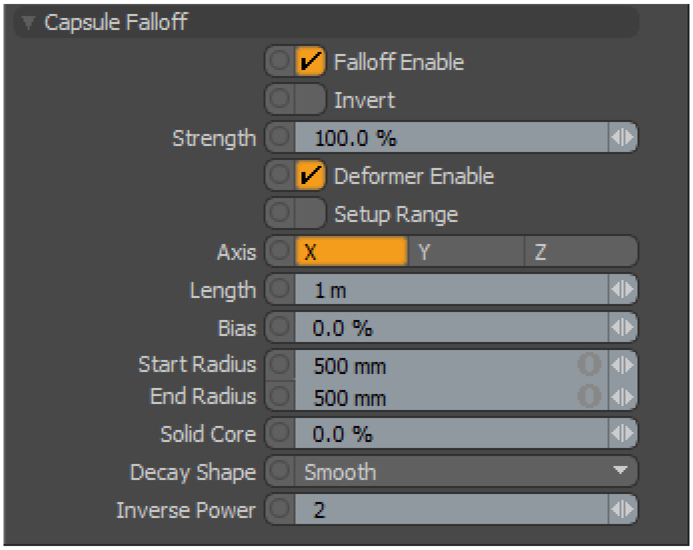


Name: Radial Falloff
Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Name	This displays the current item's name. You may change it by clicking the current name and typing a new name.
Position XYZ	Specify to position the item numerically in XYZ space. Position transforms originate from the Center position.
Rotation XYZ	Specify to set the rotation of the item numerically. Rotation transforms originate from the Center position.
Order	Sets the order that Modo applies rotations to the falloff item. By changing the order that Modo applies rotations you can sometimes reduce or eliminate gimbal lock.
Scale XYZ	Sets the size of the item numerically. Scale transforms originate from the Center position.
Reset	Resets the selected transform values to (0,0,0) and returns the items back to their default state.
Zero	Resets the selected transform property value to **0** and leaves the Center position and Item position intact.
Add	Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for Referencing. To be able to override a channel, it must first exist.)

Note: Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function (as given here), you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Capsule Falloff Properties



Capsule Falloff

☑ Falloff Enable

☐ Invert

Strength: 100.0 %

☑ Deformer Enable

☐ Setup Range

Axis: X (Y, Z)

Length: 1 m

Bias: 0.0 %

Start Radius: 500 mm

End Radius: 500 mm

Solid Core: 0.0 %

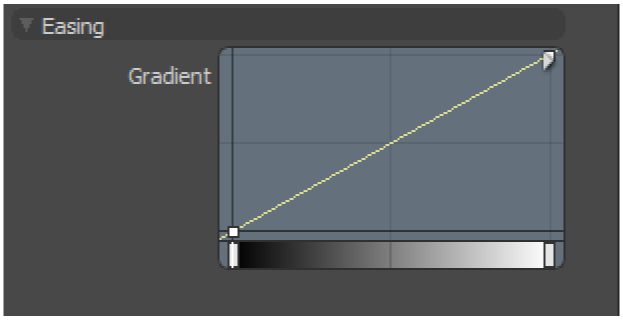
Decay Shape: Smooth

Inverse Power: 2

Falloff Enable	Toggles the influence of the falloff on or off. When clear (disabled), the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When selected, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff, would only be influenced half as much.
Deformer Enable	When selected (enabled), you can connect the falloff to a mesh and use it as a mesh deformer. You can specify the initial rest state for the deformer if you select **Setup**

	Range as well and position the falloff appropriately.
Setup Range	Determines how the **Deformer Enable** option affects the target. When using the falloff as a deformer, use the **Setup Range** to determine the undeformed position for the falloff by using the Setup state. You enter the Setup state and position the falloff where you want the Deformation to originate from. Once you exit Setup, moving the falloff deforms only those vertices within its area of influence from the Setup position. When you disable Setup Range and animate the falloff or mesh, the vertices within its influence are affected, but they slide through the area of influence instead of being affected by it directly.
Axis	Determines the initial axis direction of the Capsule shape.
Length	Determines the extent of the capsule shape from end to end.
Bias	Normally, the Center of influence is directly between the start and end points of the Capsule. Modo uses the **Bias** value to shift the center toward the Start side of the Capsule with positive values and toward the End side with negative ones.
Start Radius	Specifies the radius of the beginning section of the Capsule from its center line to its outside perimeter.
End Radius	Specifies the radius of the terminating section of the Capsule from its center line to its outside perimeter.
Solid Core	Generally, the falloff's influence attenuates from the Capsule's center line toward its outer edge. The **Solid Core** value, as a percentage of the total size of the Capsule's shape, defines an inner range where the falloff's influence is at a maximum amount.
Decay Shape	Controls the rate of attenuation across the falloff with three possible choices. • **Smooth** - Smoothly fades the strength of the falloff between the **Solid Core** setting and the **Radius** of the Capsule Falloff item. • **Inverse Power** - Applies a user-controllable curve function to attenuate the value between a linear type falloff to an abrupt, hard-edged one. This decay shape uses the **Inverse Power** value. • **Gaussian** - Smoothly applies the falloff with the greatest influence in the center, which attenuates and terminates outside the falloff's radius.
Inverse Power	If you select **Inverse Power** as the **Decay Shape**, this value determines the sharpness of the falloff's attenuation. A low value is similar to an almost linear type of falloff.

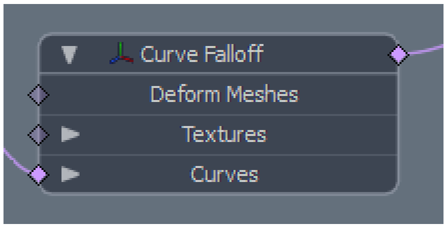
Easing Properties



Gradient	Allows you to re-map the weight of a falloff by altering the strength to create a new custom falloff shape. In the Graph Editor, press **Ctrl**/**Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shapes. For more information, see Key Creation.

Curve Falloff

The Curve falloff determines the effect of a tool on your geometry based on a user-defined curve.



Using Curve Falloff

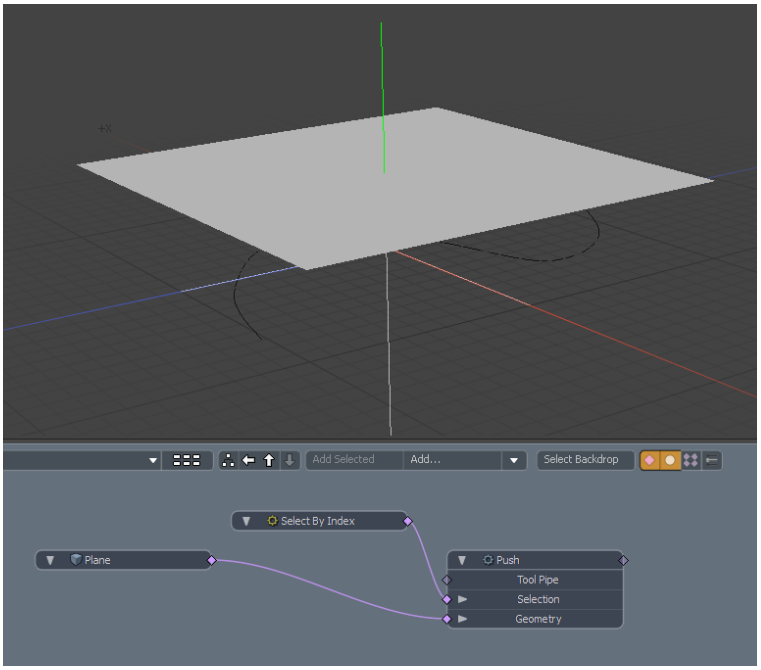
Before using Curve Falloff, make sure you have the following set up in the Schematic viewport:

- A mesh containing the geometry you want to modify
- One or more separate meshes with your curve(s)
- A deformer or mesh operation

Note: For more information on working in the Schematic, see Schematic Viewport.

This process walks you through adding and connecting a Curve Falloff and editing its gradients to achieve the required result.

In this example, a Push mesh operation is applied to a plane.

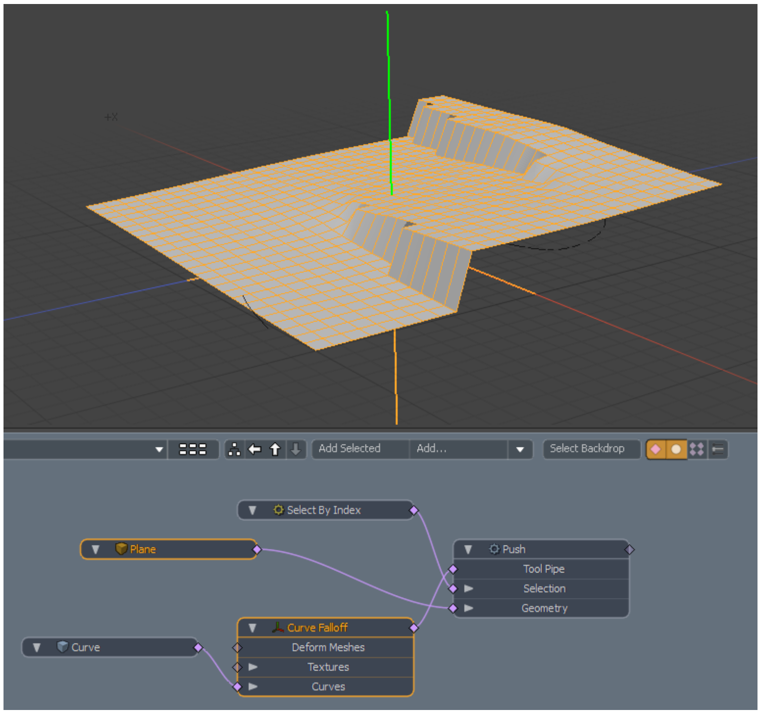

+X
Add Selected
Add...
Select Backdrop
Select By Index
Plane
Push
Tool Pipe
Selection
Geometry

1. In the Schematic, click **Add...**, then under **Mesh Operations** > **Tool Pipe** > **Falloffs**, double-click **Curve Falloff**.

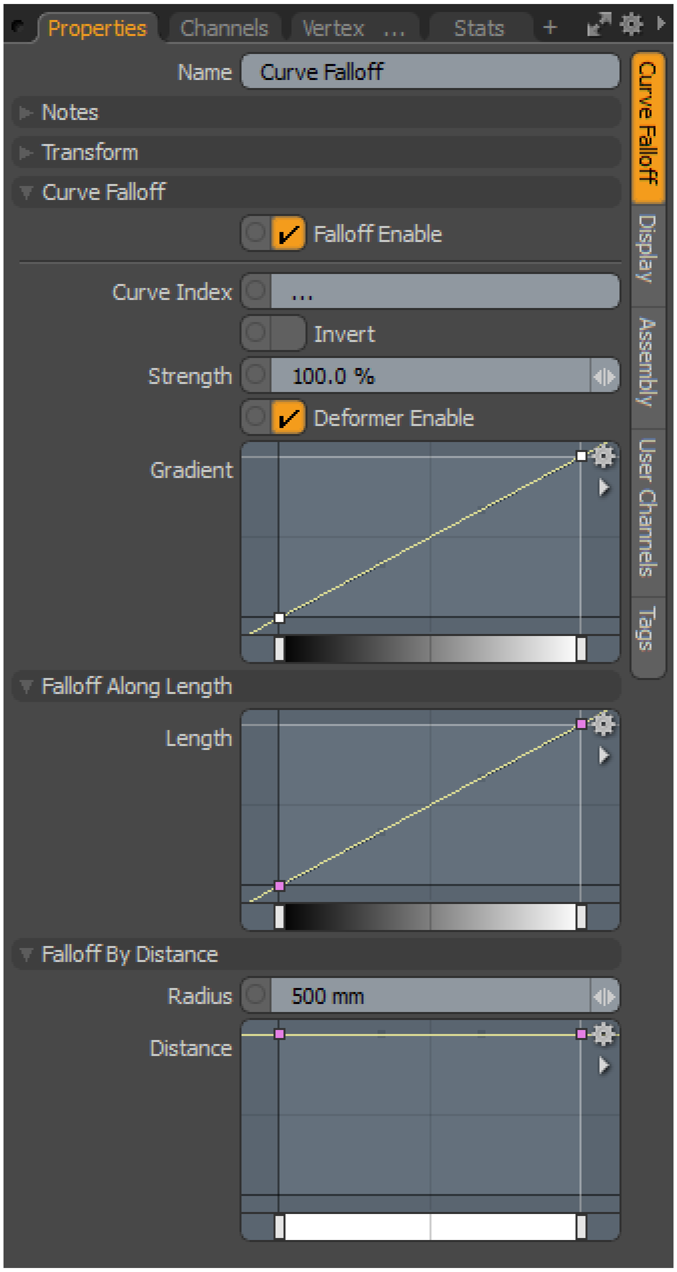
   This adds the Curve Falloff node to the Schematic.

2. Connect the mesh containing your curve to the Curve Falloff's **Curve** channel, and connect the **Curve Falloff** node to the deforming tool's **Falloff** or **Tool Pipe** channel, depending on which is present.

   The tool's effect on the geometry is attenuated along the curve.


+X
Add Selected
Add...
Select Backdrop
Select By Index
Plane
Push
Tool Pipe
Selection
Geometry
Curve Falloff
Curve
Deform Meshes
Textures
Curves

3. To specify the gradient used for determining weights along the curve, select the **Curve Falloff** node. This opens the tool's **Properties** tab on the lower right panel.


Properties
Channels
Vertex ...
Stats
+
Curve Falloff
Display
Assembly
User Channels
Tags
Notes
Transform
Curve Falloff
Gradient
Falloff Along Length
Length
Falloff By Distance
Distance

Name: Curve Falloff

☑ Falloff Enable

Curve Index: ...

☐ Invert

Strength: 100.0 %

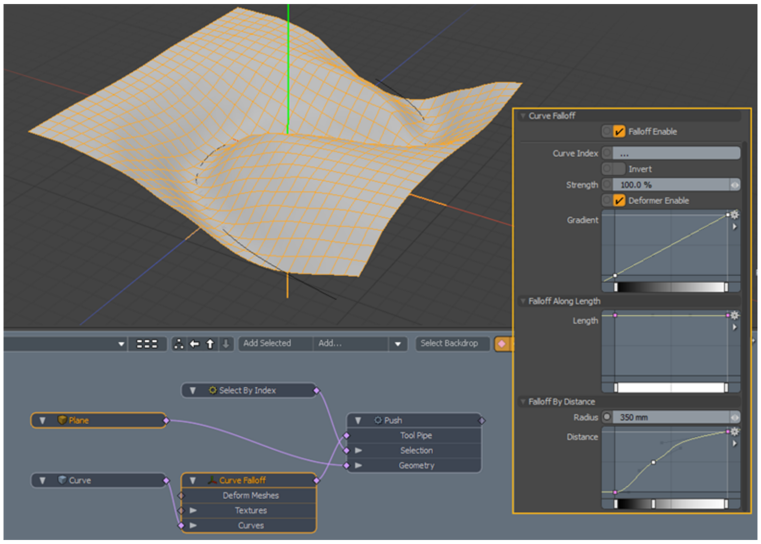
☑ Deformer Enable

Radius: 500 mm

Here you can edit the falloff's strength according to length or distance.

4. Edit the gradient under **Falloff Along Length** to change the strength of the falloff along the curve. Edit the gradient under **Falloff By Distance** to change the strength of the falloff depending on how far it is from the curve in a specified radius. You can edit the **Radius** value to modify the distance.

   In this example, the **Distance** gradient is used to determine the falloff weight.


Add Selected
Add...
Select Backdrop
Select By Index
Plane
Push
Tool Pipe
Selection
Geometry
Curve
Curve Falloff
Deform Meshes
Textures
Curves

Curve Falloff

☑ Falloff Enable

Curve Index ...

☐ Invert

Strength 100.0 %

☑ Deformer Enable

Gradient

Falloff Along Length

Length

Falloff By Distance

Radius 350 mm

Distance

Curve Falloff Properties

The Curve Falloff has the following properties.

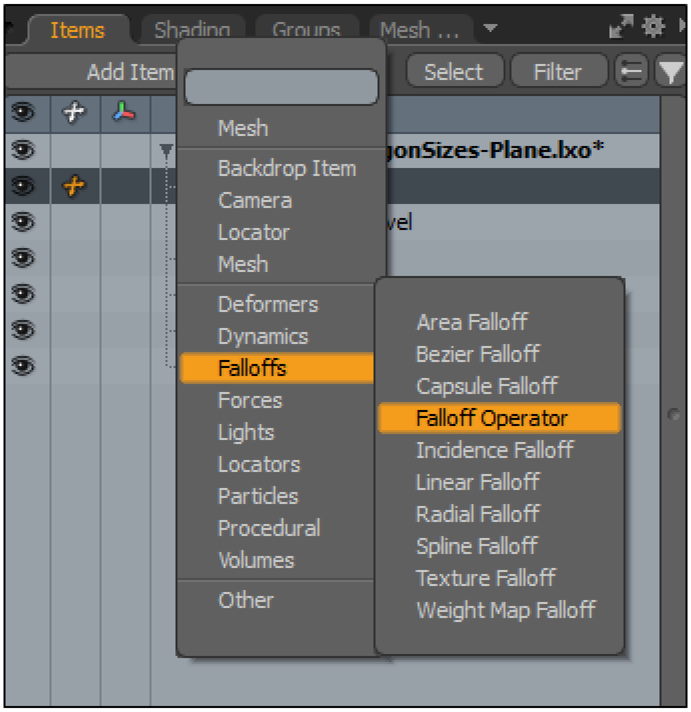
Falloff Enable	Enables or disables the falloff.
Curve Index	If you have multiple curves in your mesh layer, this specifies which one is used by the falloff. You can change the number to use a different curve.
Invert	When enabled, reverses the effect of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. For more information, see Applying Deformers.
Gradient	Specifies how the **Length** and **Distance** gradient values are blended.
Falloff Along Length	
Length	Determines the weight of the falloff along the length of the curve.
Falloff By Distance	
Radius	Specifies the radius used by the **Distance** gradient.
Distance	Determines the weight of the falloff by the points' distance from the curve in a specific **Radius**.

Falloff Operator

The Falloff Operator allows you to rig a falloff value using the position or the properties of a point or polygon being evaluated. The properties are provided as output channels and the rig returns an evaluated weight.

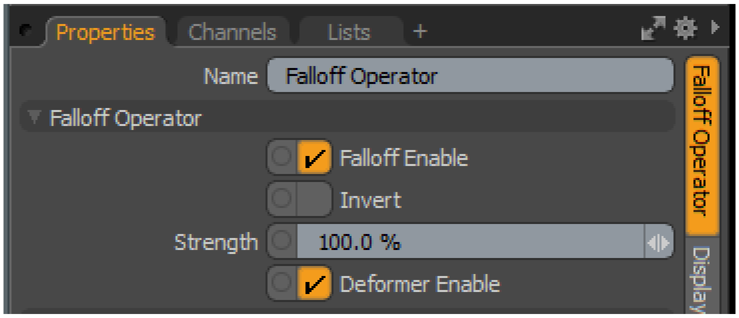
To activate the Falloff Operator tool:

1. Open the **Model** layout, on the right panel, open the **Items** tab.
2. Click the down arrow beside Add Item and select **Falloffs** > **Falloff Operator**.


Items
Shading
Groups
Mesh ...
Add Item
Select
Filter
Mesh
Backdrop Item
Camera
Locator
Mesh
Deformers
Dynamics
Falloffs
Forces
Lights
Locators
Particles
Procedural
Volumes
Other
Area Falloff
Bezier Falloff
Capsule Falloff
Falloff Operator
Incidence Falloff
Linear Falloff
Radial Falloff
Spline Falloff
Texture Falloff
Weight Map Falloff
gonSizes-Plane.lxo*
vel

- Alternatively, open the **Setup** layout, on the bottom panel, click **Add**, and select **Mesh Operations** > **Tool Pipe** > **Falloffs** > **Falloff Operator**.

Falloff Operator Properties



Name: Falloff Operator

Falloff Operator

☑ Falloff Enable

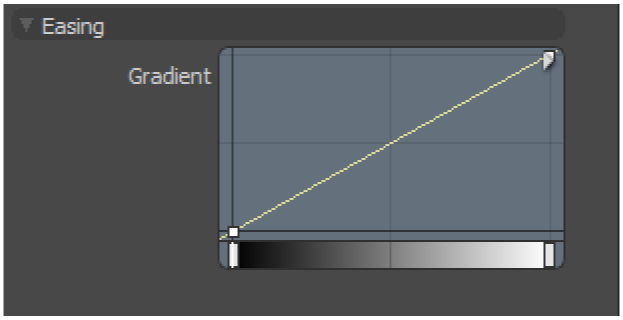
☐ Invert

Strength: 100.0 %

☑ Deformer Enable

Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. For more information, see Applying Deformers.

Easing Properties



Gradient	Allows you to re-map the weight of a falloff by altering the strength to create a new custom falloff shape. In the Graph Editor, press **Ctrl/Cmd** + **Alt** and click on the

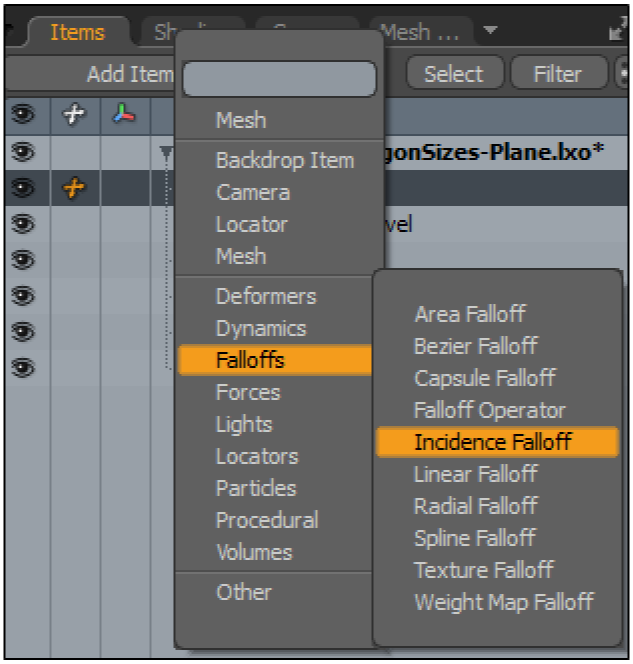
	graph line to add a new key point. Move the key points to create custom falloff shapes. For more information, see Key Creation. Watch this video for an overview: Gradient Editor.

Incidence Falloff

The Incidence Falloff calculates a weight value determined by the incidence between the vertex normal and the falloff locator. If no vertex is provided, the default value of 1.0 is applied.

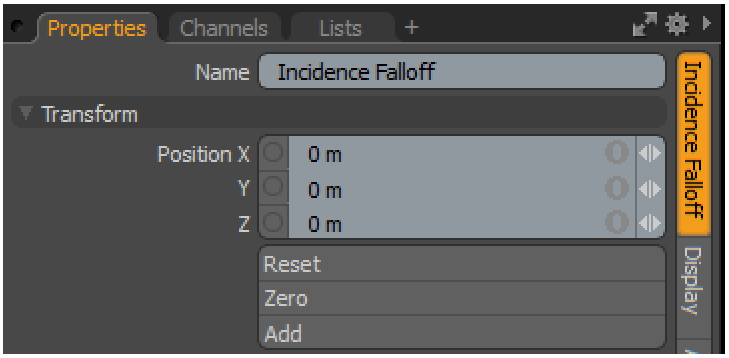
To activate the Incidence Falloff tool:

1. Open the **Model** layout, on the right panel, open the **Items** tab.
2. Click the down arrow beside **Add Item** and select **Falloffs** > **Incidence Falloff**.


Items
Mesh ...
Add Item
Select
Filter
gonSizes-Plane.lxo*
vel
Mesh
Backdrop Item
Camera
Locator
Mesh
Deformers
Dynamics
Falloffs
Forces
Lights
Locators
Particles
Procedural
Volumes
Other
Area Falloff
Bezier Falloff
Capsule Falloff
Falloff Operator
Incidence Falloff
Linear Falloff
Radial Falloff
Spline Falloff
Texture Falloff
Weight Map Falloff

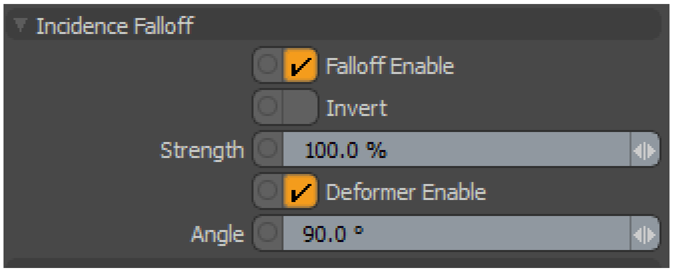
- Alternatively, open the **Setup** layout, on the bottom panel, click **Add**, and select **Mesh Operations** > **Tool Pipe** > **Falloffs** > **Incidence Falloff**.

Transform Properties



Name	Displays the current item's name. You may change it by clicking the current name and typing a new name.
Position XYZ	Specify to position the item numerically in XYZ space. Position transforms originate from the Center position.
Reset	Resets the selected transform values to (0,0,0) and returns the items back to their default state.
Zero	Resets the selected transform property value to **0** and leaves the Center position and Item position intact.
Add	Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for Referencing. To be able to override a channel, it must first exist.) **Note:** Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function (as given here), you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Incidence Falloff Properties



☑ Falloff Enable

☐ Invert

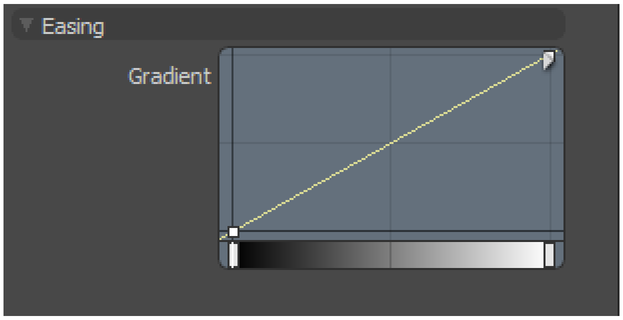
Strength 100.0 %

☑ Deformer Enable

Angle 90.0 °

Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. For more information, see Applying Deformers.
Angle	Defines the threshold angle used to modulate the weight.

Easing Properties



Gradient	Allows you to re-map the weight of a falloff by creating a new custom falloff shape.

	In the Graph Editor, press **Ctrl**/**Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shape. For more information, see Key Creation.

Linear Falloff

The Linear Falloff produces an attenuation along its length (range) that moves outward toward infinity and parallel to its axis. It has options for you to adjust the axis, range, and transition of the falloff itself.

To activate the Linear Falloff tool:

- Open the **Model** layout, under the layout menu, click ◎ Falloff, and select **Linear**.
- Alternatively, open the **Setup** layout, on the bottom panel, click **Add**, and select **Mesh Operations** > **Tool Pipe** > **Falloffs** > **Linear Falloff**.

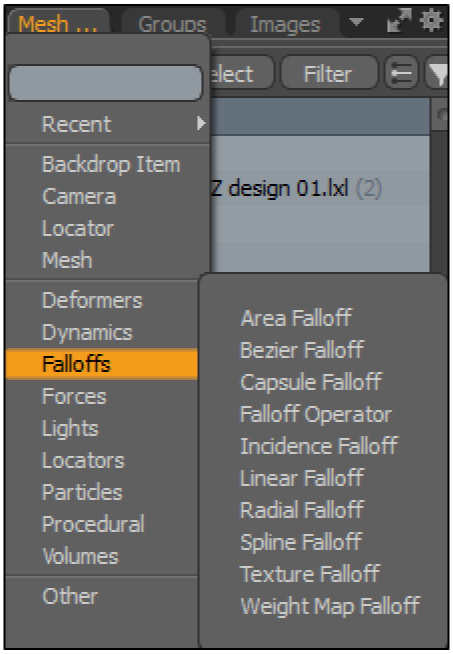
Adding a Linear Falloff

A Linear Falloff can be used with Deforms or a Mesh Operations item. You can use the **Model** layout or the **Setup** layout to add a Linear Falloff.

For more information, see Applying Deformers and Adding Falloffs in Procedural Modeling.

1. In **Model** layout, select a mesh item in the **Items** list.
2. With the Polygon Selection Mode enabled, select the polygons to apply the falloff to.
3. Under the layout menu, click ◎ Falloff, and select **Linear**.

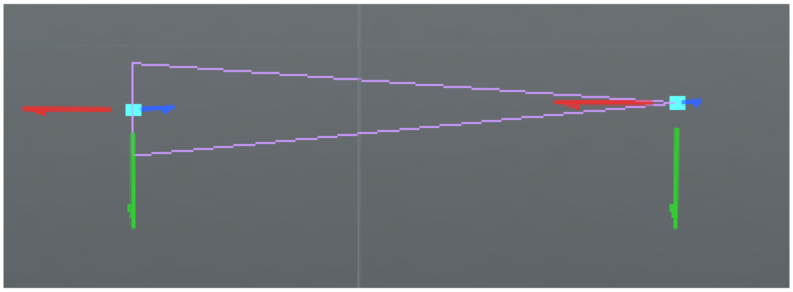
   Alternatively, on the right panel, open the **Mesh Ops** tab, and click the down arrow beside **Add Item** and select **Falloff** > **Linear Falloff**.


Mesh ...
Groups
Images
elect
Filter
Recent
Backdrop Item
Camera
Locator
Mesh
Deformers
Dynamics
Falloffs
Forces
Lights
Locators
Particles
Procedural
Volumes
Other
Z design 01.lxl (2)
Area Falloff
Bezier Falloff
Capsule Falloff
Falloff Operator
Incidence Falloff
Linear Falloff
Radial Falloff
Spline Falloff
Texture Falloff
Weight Map Falloff

By default, the falloff aligns to the selection bounding box.

4. Click and drag from right to left in the viewport to position the falloff.

   The small end of the falloff is the front and the large end is the back.

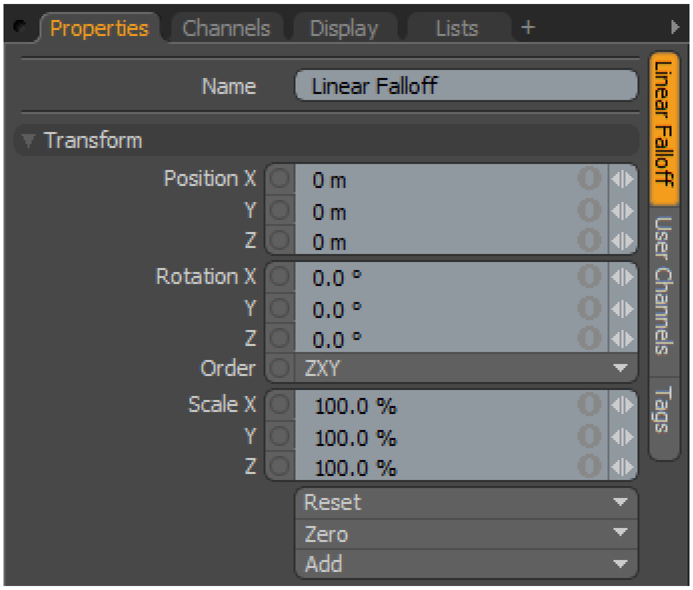


5. On the left panel, select a tool to apply the falloff to. For example, open the **Basic** tab and click **Bevel** (Polygon Bevel) and apply the tool.
6. Click + drag the Polygon Bevel handles to change the shape of the selected polygons.

Note: The backside of the selected polygons move more than the front. This is based on the position of the falloff.

7. Click and drag on the end and start falloff handles to change how the falloff is applied.

Transform Properties



Name: Linear Falloff

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

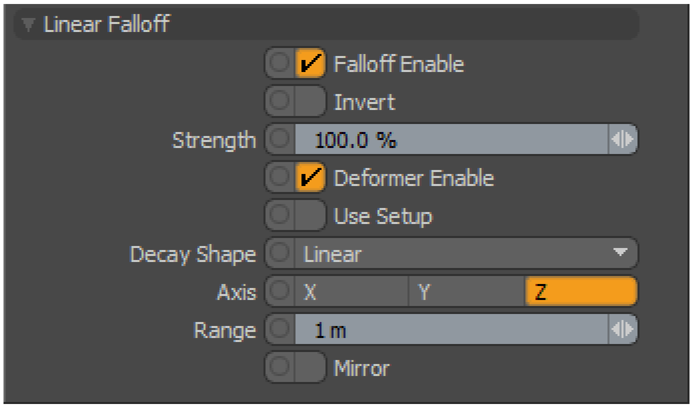
Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Name	This displays the current item's name. You may change it by clicking the current name and typing a new name.
Position XYZ	Specify to position the item numerically in XYZ space. Position transforms originate from the Center position.
Rotation XYZ	Specify to set the rotation of the item numerically. Rotation transforms originate from the Center position.
Order	Sets the order that Modo applies rotations to the falloff item. By changing the order that Modo applies rotations you can sometimes reduce or eliminate gimbal lock.
Scale XYZ	Sets the size of the item numerically. Scale transforms originate from the Center position.
Reset	Resets the selected transform values to (0,0,0) and returns the items back to their default state.
Zero	Resets the selected transform property value to **0** and leaves the Center position and Item position intact.
Add	Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for Referencing. To be able to override a channel, it must first exist.)

Note: Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function (as given here), you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Linear Falloff Properties



Linear Falloff

☑ Falloff Enable

☐ Invert

Strength 100.0 %

☑ Deformer Enable

☐ Use Setup

Decay Shape Linear

Axis X Y Z

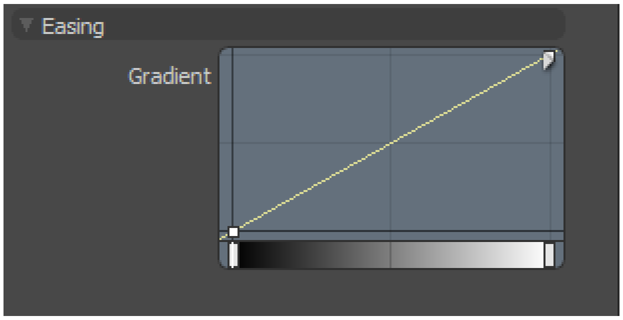
Range 1 m

☐ Mirror

Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. For more information, see Applying Deformers.
Use Setup	Determines how the **Deformer Enable** option affects the target. When using the falloff as a Deformer, use the **Use Setup** determines the undeformed position for the falloff by using the Setup state.

	You enter the **Setup** state and position the falloff where you want the Deformation to originate from. Once you exit **Setup**, moving the falloff deforms only those vertices within its area of influence from the Setup position. When you disable **Use Setup** and animate the falloff or mesh, the vertices within its influence are affected, but they slide through the area of influence instead of being affected by it directly.
Decay Shape	Controls the rate of attenuation across the falloff with four possible choices. • **Linear**: This gives a straight attenuation of the falloff influence across its total range. • **Ease In**: This gives a smooth attenuation with greater influence toward the start of the falloff's range. • **Ease Out**: This gives a smooth attenuation with greater influence toward the end of the falloff's range. • **Smooth**: This gives a smooth attenuation with a slightly greater influence in the middle of the falloff's range.
Axis	Determines the initial axis direction of the Linear Falloff's shape.
Range	Determines the overall extent of the falloff's influence as a length originating from the falloff's center.
Mirror	Produces an inverted duplicate of the falloff's influence across its origin and axis.

Easing Properties



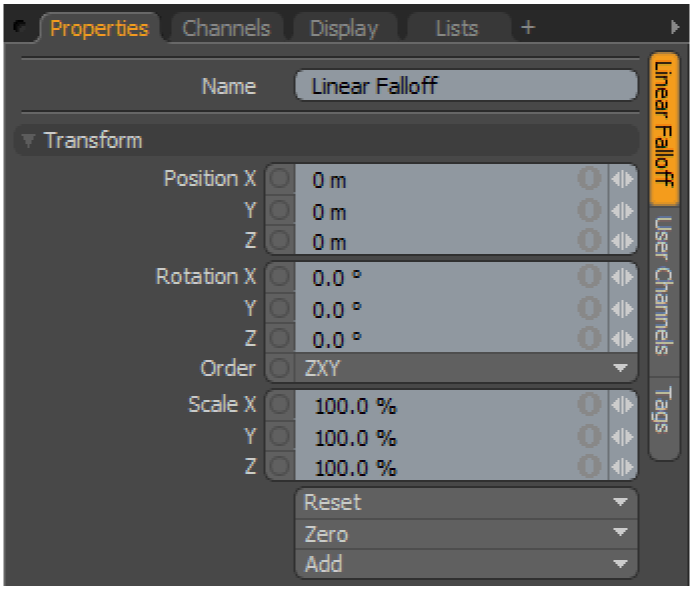
Gradient	Allows you to re-map the weight of a falloff by creating a new custom falloff

	shape. In the Graph Editor, press **Ctrl**/**Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shape. For more information, see Key Creation. Watch this video for an overview: Gradient Editor.

Radial Falloff

The Radial Falloff produces a spherical or round attenuation area and has controls to adjust the size, the shape, and the transition of the falloff, itself.

Transform Properties


Properties
Channels
Display
Lists
+
Linear Falloff
User Channels
Tags
Transform
Reset
Zero
Add

Name Linear Falloff

Position X 0 m
Y 0 m
Z 0 m

Rotation X 0.0 °
Y 0.0 °
Z 0.0 °

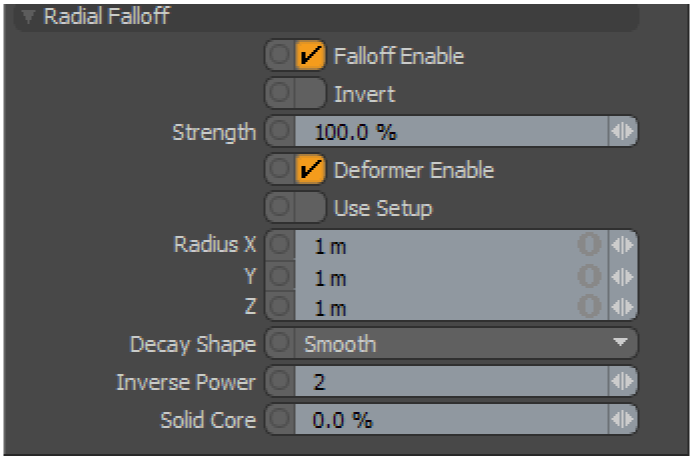
Order ZXY

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Name	This displays the current item's name. You may change it by clicking the current name and typing a new name.
Position XYZ	Specify to position the item numerically in XYZ space. Position transforms originate from the Center position.

Rotation XYZ	Specify to set the rotation of the item numerically. Rotation transforms originate from the Center position.
Order	Sets the order that Modo applies rotations to the falloff item. By changing the order that Modo applies rotations you can sometimes reduce or eliminate gimbal lock.
Scale XYZ	Sets the size of the item numerically. Scale transforms originate from the Center position.
Reset	Resets the selected transform values to (0,0,0) and returns the items back to their default state.
Zero	Resets the selected transform property value to **0** and leaves the Center position and Item position intact.
Add	Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for Referencing. To be able to override a channel, it must first exist.) **Note:** Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function (as given here), you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Radial Falloff Properties



Radial Falloff

☑ Falloff Enable

☐ Invert

Strength: 100.0 %

☑ Deformer Enable

☐ Use Setup

Radius X: 1 m

Y: 1 m

Z: 1 m

Decay Shape: Smooth

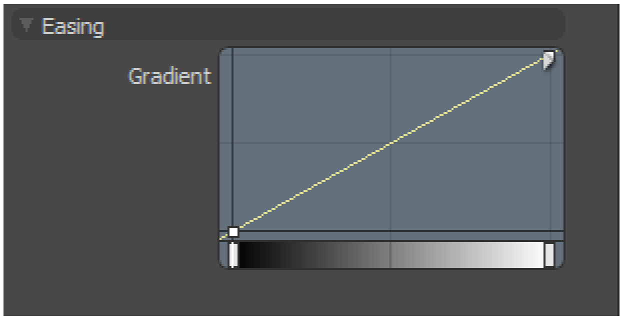
Inverse Power: 2

Solid Core: 0.0 %

Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. For more information, see Applying Deformers.
Setup Range	Determines how the **Deformer Enable** option affects the target. When using the falloff as a deformer, use the **Setup Range** to determine the undeformed position for the falloff by using the **Setup** state. You enter the **Setup** state and position the Falloff where you want the deformation to originate from. Once you exit **Setup**, moving the falloff deforms only those vertices within its area of influence from the **Setup** position. When you disable **Setup Range** and animate the falloff or mesh, the vertices within its influence are affected, but they slide through the area of influence instead of being affected by it directly.
Radius XYZ	Determine the size of the sphere as a radius (from the sphere's center to its outside perimeter) for each axis specified.
Decay Shape	Controls the rate of attenuation across the falloff with three possible choices:

	• **Smooth** - This smoothly fades the strength of the falloff between the **Solid Core** setting and the **Radius** of the **Spherical Falloff** item. • **Inverse Power** - This attenuates the value between a linear type falloff to an abrupt, hard-edged one based on the **Inverse Power** value. • **Gaussian** - This provides a very smooth falloff with the greatest influence in the center that attenuates and terminates outside the falloff's radius.
Inverse Power	If you select **Inverse Power** as the **Decay Shape**, this value determines the falloff's attenuation sharpness. Low values are similar to an almost linear type of falloff.
Solid Core	Defines an inner range where the falloff's influence is at a maximum amount as a percentage of the total size of the spherical shape. (Generally, the falloff's influence attenuates from its center line toward its outer edge.)

Easing Properties



Gradient	Allows you to re-map the weight of a falloff by altering the strength to create a new custom falloff shape. In the Graph Editor, press **Ctrl/Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shapes. For more information, see Key Creation.

Radial Falloff in the Procedural

Note: For details on how to add falloffs to the **Mesh Operations** list, see Adding Falloffs in Procedural Modeling.

In the **Mesh Operations** list, the Radial Falloff has the following inputs:

- **Deform Meshes** - This allows you to use the falloff as a deformer on a mesh. You can connect a mesh to the falloff to transform the vertices, modulated by the falloff.
- **Textures** - You can specify a texture to be used by the falloff to attenuate its effect.

Spline Falloff

Like with modeling, you use a falloff to attenuate the effect any deformer has on a target item and to provide additional control about how deformers influence a mesh.

For more information, see Applying Deformers.

Using the Spline Falloff

To add a Spline Falloff to a scene, you must first add a Spline Deformer. The falloff item uses this deformer and its resulting curve to produce the falloff. The deformer, itself does not have to be active for the falloff to function but the falloff needs the deformer to be present as a container for the locators. Then, you can use both the Spline Deformer and Spline Falloff together without duplicating the spline.

Once you define a Spline Deformer, you add the Spline Falloff to it.

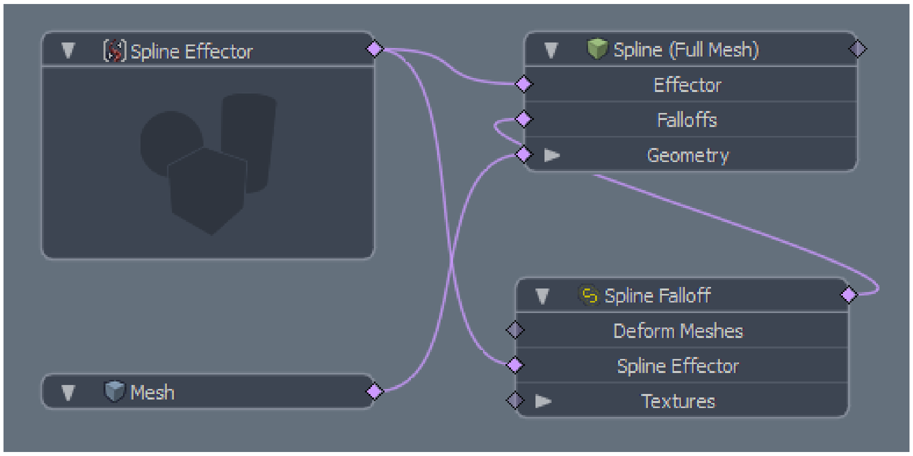
1. Right-click the **Spline (Full Mesh)** layer within the Deformers viewport or the Items (Scenes) List and choose **Add Override** > **Falloffs** > **Spline Falloff**.
2. Once you create the Spline Falloff, in the **Schematic**, connect the resulting Spline Effector to the Spline Falloff's **Spline Effector** connector.

Note: If you're not familiar with the **Schematic** viewport, see Schematic viewport for more information.

3. Create a link by dragging a connector between the Spline Effector's relationship output to the Spline Falloff's **Spline Effector** input.

Note: For the falloff to adjust the range of the deformation rather than shift along the length of the spline, set the **Spline (Full Mesh)** item **Interpolation** mode to **Linear**.

Once defined, the falloff properly attenuates the effects of the Spline Deformer.



4. At this point, select the **Spline Effector** item and disable it.
5. Add an additional Deformer to the item and link the **Spline Falloff** output to the resulting item's **Falloffs** input to have the falloff attenuate the influence of the new Deformer.

   The falloff's position relative to the undeformed item determines which parts of the target item Modo affects.

Note: See the **Schematic** viewport topic for more information about adding items to the viewport and linking them together.

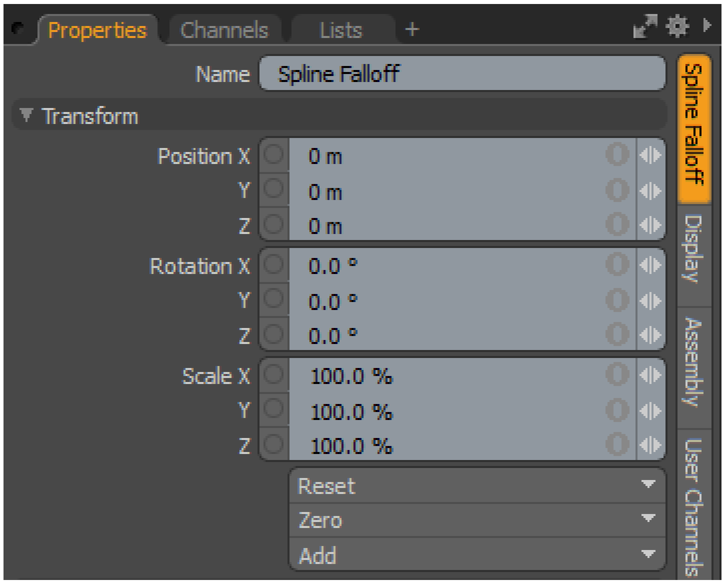
Spline Falloff in the Procedural

Note: For details on how to add falloffs to the **Mesh Operations** list, see Adding Falloffs in Procedural Modeling.

In the **Mesh Operations** list, the Spline Falloff has the following inputs:

- **Deform Meshes** - This allows you to use the falloff as a deformer on a mesh. You can connect a mesh to the falloff to transform the vertices, modulated by the falloff.
- **Textures** - You can specify a texture to be used by the falloff to attenuate its effect.
- **Spline Effector** - The Spline Falloff operation requires a Spline Effector's curve to produce the falloff. For more information on the Spline Effector, see Spline Effector.

Transform Properties



Name: Spline Falloff

Position X: 0 m
Y: 0 m
Z: 0 m

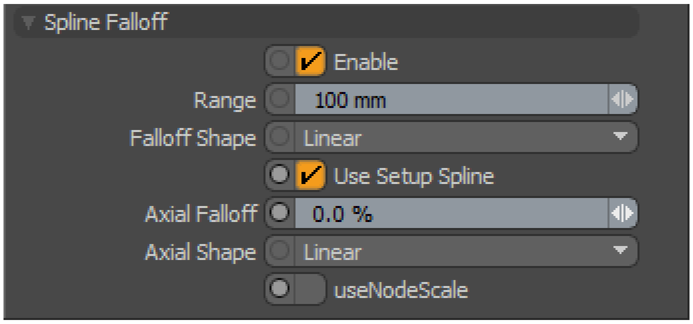
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Name	This displays the current item's name. You may change it by clicking the current name and typing a new name.
Position XYZ	Specify to position the item numerically in XYZ space. Position transforms originate from the Center position.
Rotation XYZ	Specify to set the rotation of the item numerically. Rotation transforms originate from the Center position.
Order	Sets the order that Modo applies rotations to the falloff item. By changing the order that Modo applies rotations you can sometimes reduce or eliminate gimbal lock.
Scale XYZ	Sets the size of the item numerically. Scale transforms originate from the Center position.
Reset	Resets the selected transform values to (0,0,0) and returns the items back to their default state.
Zero	Resets the selected transform property value to **0** and leaves the Center position and Item position intact.
Add	Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for Referencing. To be able to override a channel, it must first exist.)

Note: Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function (as given here), you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Spline Falloff Properties


Spline Falloff

☑ Enable

Range 100 mm

Falloff Shape Linear

☑ Use Setup Spline

Axial Falloff 0.0 %

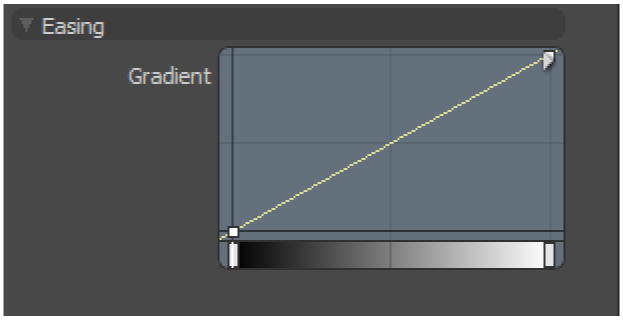
Axial Shape Linear

☐ useNodeScale

Enable	Toggles the falloff on or off. When clear, the falloff has no influence on the scene. However, Modo saves disabled layers with the scene and their values are persistent across Modo sessions.
Range	Defines the distance away from the Spline that the falloff influences its target.
Falloff Shape	Controls the rate of attenuation across the falloff with four options. • **Linear** - Provides a straight attenuation of the falloff's influence across its total range. • **Ease In** - Gives a smooth attenuation with greater influence toward the start of the falloff's range. • **Ease Out** - Gives a smooth attenuation with greater influence toward the end of the falloff's range. • **Smooth** - Provides a smooth attenuation with a slightly stronger influence in the middle of the falloff's range.

Use Setup Spline	Select when the Spline is both a deformer and a falloff. When selected, Modo uses the **Setup Spline** shape to compute distances between the vertices and the spline. (The **Setup Spline** is how it looks in its Setup state.) When you use the Spline as both a deformer and a falloff, Modo determines the falloff on the undeformed vertex positions. Then, the deformer can pull the mesh out the reach of the falloff and cause deformation artifacts. With **Use Setup Spline** selected, Modo produces a more desirable result.
Axial Falloff	Controls the strength of the falloff along the length of the curve. A value of **0%** generates no axial falloff. A value of **100%** attenuates the falloff's strength along the curve with the maximum influence at the base of the curve and no influence at its termination point; a value of **-100%** produces the inverse: no influence at the base and the maximum strength at the terminator. (You can use values of greater than 100% to achieve unusual results.)
Axial Shape	Controls the attenuation amount along the length of the curve when you set **Axial Falloff** to a value greater than **0%**. This duplicates the functionality of the **Falloff Shape** option, but Modo applies it along the length instead of outward from the Spline Falloff's core.
Use Node Scale	When selected, you can scale the Locator(s) that make up the Spline Falloff's curve that then scales the **Range** value. This acts as a multiplier on a per node basis. In the unselected, default state, node scaling has no effect on the falloff's **Range**.

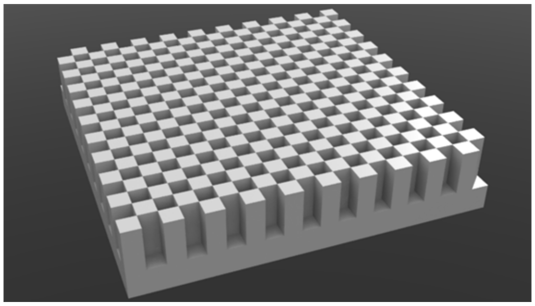
Easing Properties



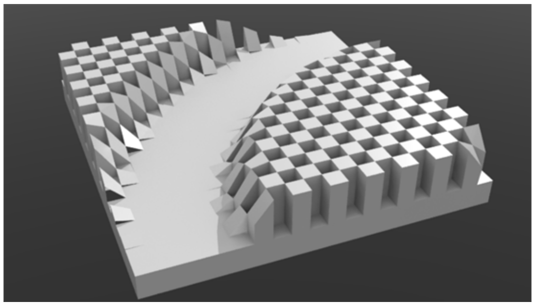
Gradient	Allows you to re-map the weight of a falloff by creating a new custom falloff shape. In the Graph Editor, press **Ctrl/Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shape. For more information, see Key Creation.

Texture Falloff

A Texture Falloff allows you to use a texture to attenuate the effect of a tool, such as a bevel.



A cuboid using a polygon bevel to create a raised check pattern.



A texture falloff has been added to the bevel, using the Ring texture.

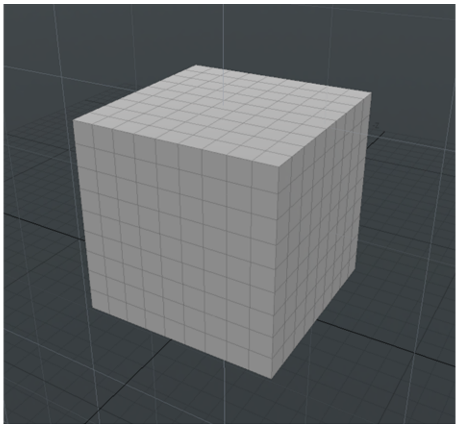
The Texture Falloff has a constant value of 1 across its influence so as to affect a tool using only a given texture, and not by another means of attenuation. This is designed to make texturing falloffs easier when using the Mesh Operations List by making use of the falloff's texture input.

The Texture Falloff allows you to add a texture using the Mesh Operations List without using the Shader Tree.

Note: All falloffs have a texture input. You can use a different falloff for applying textures but it is less straightforward. For example, you could use the schematic texture input of the Linear Falloff, and position the linear falloff so that the geometry is only affected by the widest area of the falloff, and so remains unaffected.

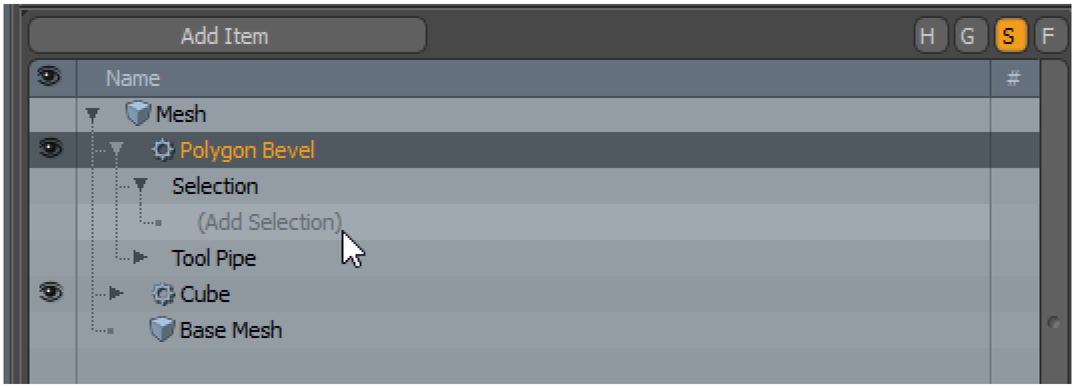
Using the Texture Falloff

The Texture Falloff can be added using the Preset Browser. The instructions below show how to use the falloff to apply a Dimples texture to a polygon bevel tool in the Mesh Operations List. This indents the beveled polygons across the surface of the mesh. The steps assume a mesh already exists, and in this case we'll use a procedural cube with 9 segments across each axis as pictured below.



Note: If you already have a tool applied to a mesh that can be textured, skip to step 4.

1. On the right panel open the Mesh Ops tab, click **Add Item**, and double-click **Procedural** > **Polygon** > **Polygon Bevel**.
2. Select polygons to bevel: Expand **Selection** in the **Polygon Bevel** and click **(Add Selection)**.

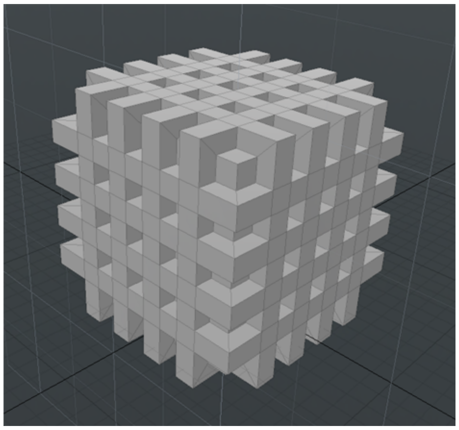


In this case, let's use **Select by Index**. Double-click on **Select by Index** in the **Preset Browser** to add it to the **Polygon Bevel** selection.

Select **Polygons** selection mode and click-select a number of polygons. When finished, click **Set** in the **Select by Index** properties. This populates the selection.

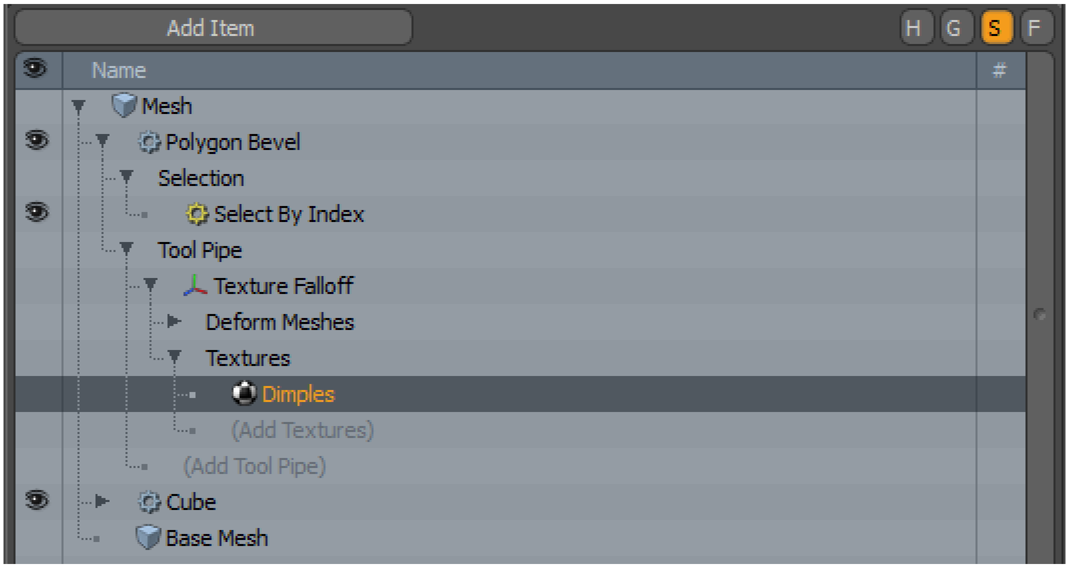
Note: In this example, a checkerboard pattern of polygons is selected across the surface.

3. Apply the bevel by clicking on **Polygon Bevel** in the list and adjusting the properties. In this case, we'll shift the polygons out by setting **Shift** to **90 mm**.



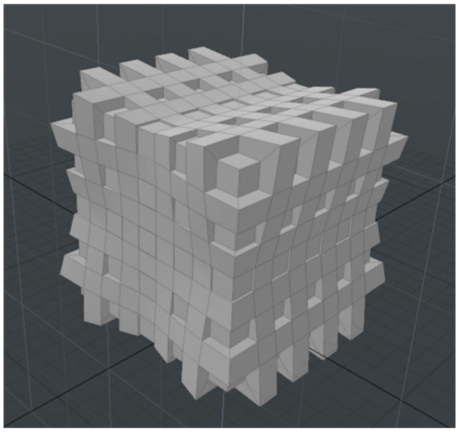
4. You're now ready to apply a texture to the bevel tool. In the Mesh Operations List, expand the **Tool Pipe** for the **Polygon Bevel** and click **(Add Tool Pipe)**.
5. Double click on **Texture Falloff** to add it to the stack.
6. Now choose a texture to apply to the falloff. In the Mesh Operations List, expand **Textures** under **Texture Falloff** and click **(Add Textures)**.

   Choose a texture from the browser. In this case, let's use the **Dimples** texture.



Add Item
H
G
S
F
Name
#
Mesh
Polygon Bevel
Selection
Select By Index
Tool Pipe
Texture Falloff
Deform Meshes
Textures
Dimples
(Add Textures)
(Add Tool Pipe)
Cube
Base Mesh

7. Set the **Projection Type** set to **Cubic** to indent the sides of the mesh. Edit the texture (Dimples) properties to adjust how the bevel is affected.



8. Edit the Texture Falloff Properties to customize the falloff behavior.

Texture Falloff Procedural

The following procedural sub-tools are available:

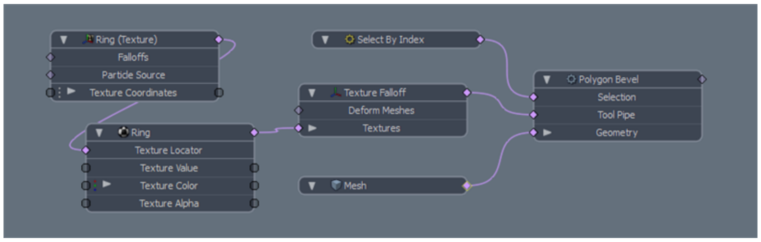
- **Deform Meshes** - This allows you to use the falloff as a deformer on a mesh. You can connect a mesh to the falloff to transform the vertices, modulated by the falloff. SeeAdding a Falloff as a Deformer for more details.

Note: This offers a simple way to set up a deform with a falloff, without having a Transform Locator combined with the falloff.

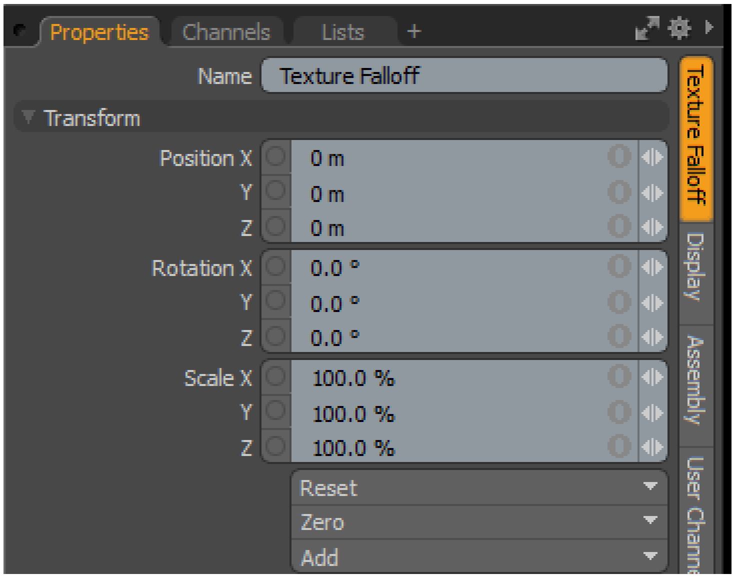
- **Textures** - The source texture(s) used to attenuate the affected tool.

Texture Falloff Schematic

The schematic node has the same inputs as the procedural node.


Ring (Texture)
Falloffs
Particle Source
Texture Coordinates
Select By Index
Texture Falloff
Deform Meshes
Textures
Polygon Bevel
Selection
Tool Pipe
Geometry
Ring
Texture Locator
Texture Value
Texture Color
Texture Alpha
Mesh

Transform Properties



Name: Texture Falloff

Transform

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

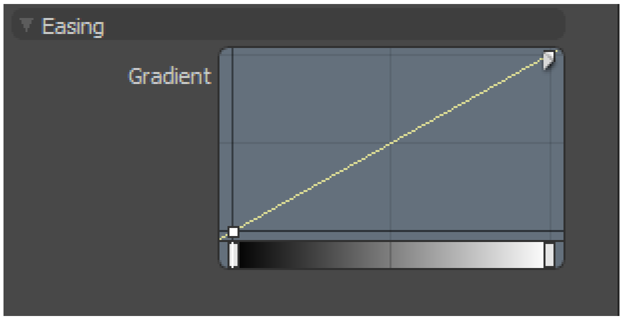
Name	This displays the current item's name. You may change it by clicking the current name and typing a new name.
Position XYZ	Specify to position the item numerically in XYZ space. Position transforms originate from the Center position.
Rotation XYZ	Specify to set the rotation of the item numerically. Rotation transforms originate from the Center position.
Order	Sets the order that Modo applies rotations to the falloff item. By changing the order that Modo applies rotations you can sometimes reduce or eliminate gimbal lock.
Scale XYZ	Sets the size of the item numerically. Scale transforms originate from the Center position.
Reset	Resets the selected transform values to (0,0,0) and returns the items back to their default state.
Zero	Resets the selected transform property value to **0** and leaves the Center position and Item position intact.
Add	Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for referencing. To be able to override a channel, it must first exist.)

Note: Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function (as given here), you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Texture Falloff Properties

Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.
Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	Connects the falloff to a mesh and use it as a mesh deformer. You can specify the initial rest state for the deformer if you select Setup Range as well and position the falloff appropriately.

Easing Properties

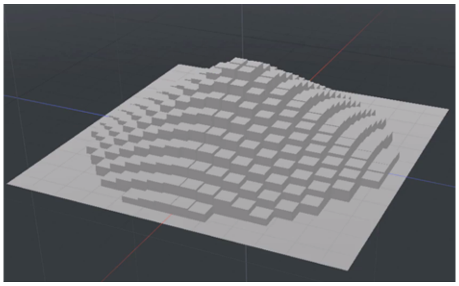


Gradient	Allows you to re-map the weight of a falloff by creating a new custom falloff shape. In the Graph Editor, press **Ctrl**/**Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shape. For more information, see Key Creation.

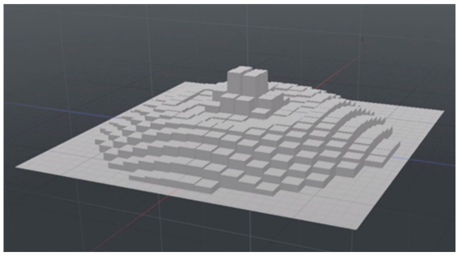
Weight Map Falloff

The Weight Map Falloff evaluates a weight map applied to a Mesh Operation. You can choose which weight map to evaluate. Weight maps define the area of influence upon a mesh. A Mesh Operation, such as Polygon Bevel, drives the Weight Map Falloff. The Easing Properties allows you to quickly create a new falloff shape.

For more information about Weight Maps, see Working with Weight Maps.



Before



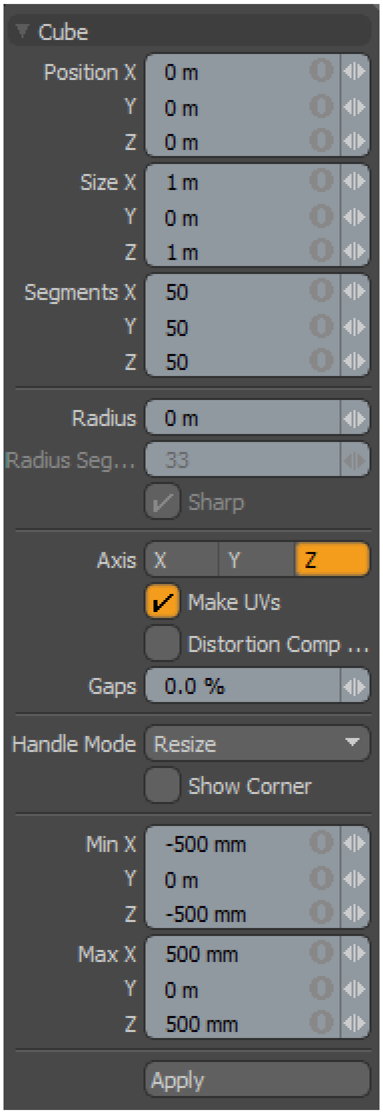
After

Adding a Weight Map Falloff

In this example, create a subdivided plane with a weight map, and apply a mesh operation to drive the Weight Map Falloff.

To create a plane with a weight map:

1. In **Modo** layout, on the left panel, click **Cube**.
2. Set the cube properties to the following and click **Apply**.



Cube

Position X 0 m
Y 0 m
Z 0 m

Size X 1 m
Y 0 m
Z 1 m

Segments X 50
Y 50
Z 50

Radius 0 m
Radius Seg… 33
☑ Sharp

Axis X Y Z (Z selected)
☑ Make UVs
☐ Distortion Comp …
Gaps 0.0 %

Handle Mode Resize
☐ Show Corner

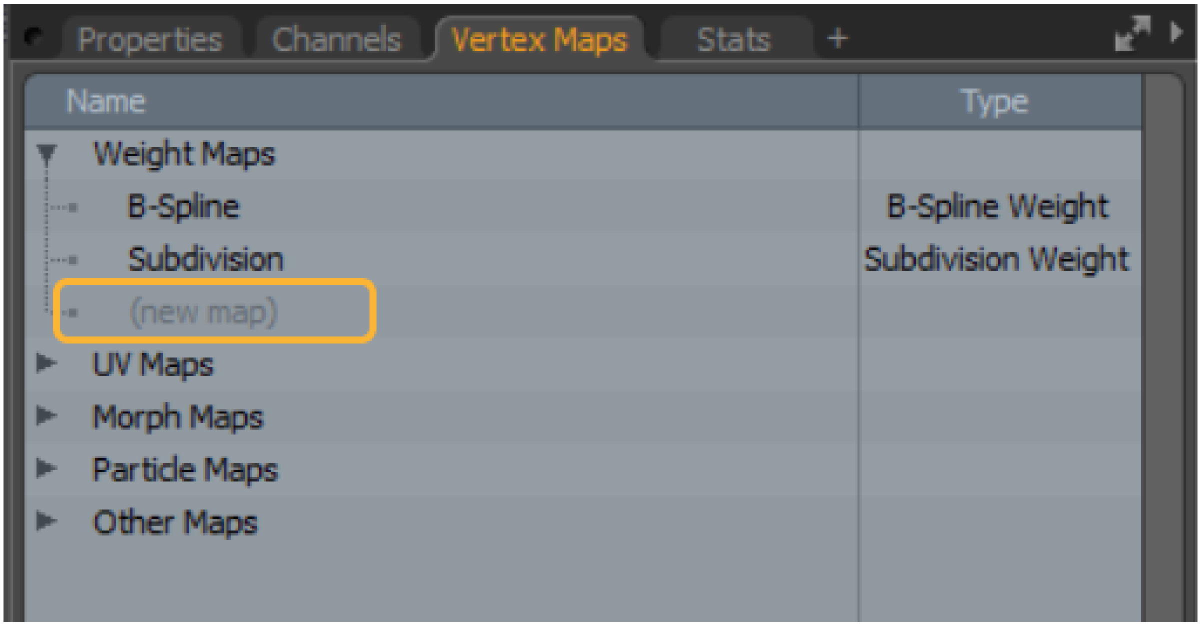
Min X -500 mm
Y 0 m
Z -500 mm

Max X 500 mm
Y 0 m
Z 500 mm

3. On the right panel, open the **Shading** tab.

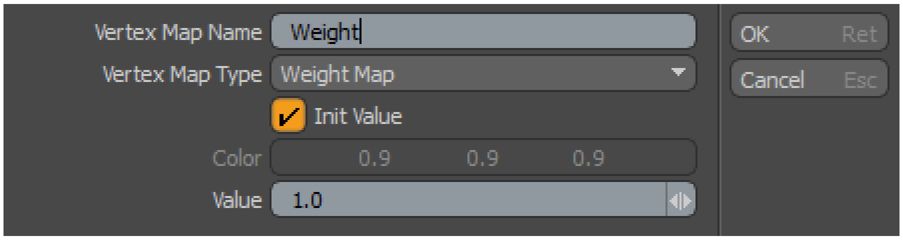
> **Tip:** If the **Shading** tab is not visible, below the menu bar double-click **Model** to reset the layout.

4. On the bottom of the panel, open the **Lists** tab, expand **Vertex Maps**, and click **(new map)**.


Properties
Channels
Vertex Maps
Stats
+

Name	Type
Weight Maps	
B-Spline	B-Spline Weight
Subdivision	Subdivision Weight
(new map)	
UV Maps	
Morph Maps	
Particle Maps	
Other Maps	

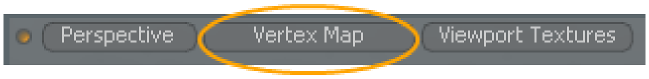
5. Ensure that the following weight map properties are set and click **OK**.



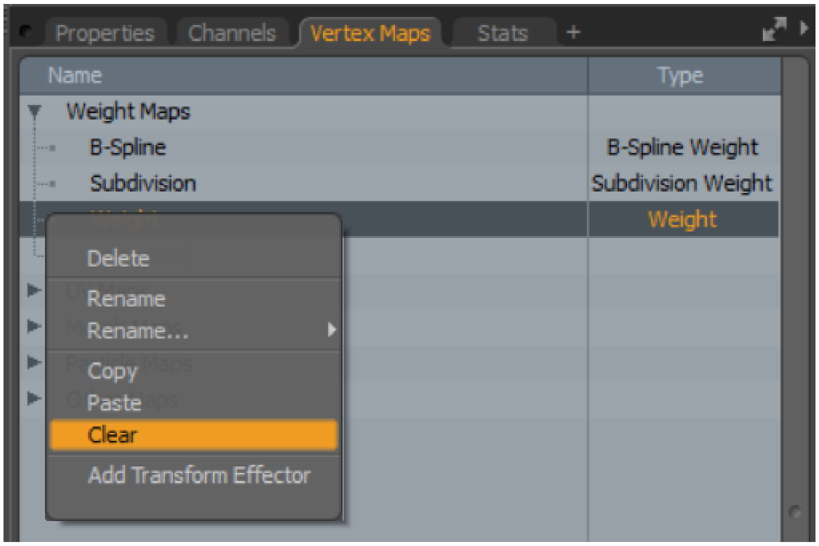
Vertex Map Name: Weight
Vertex Map Type: Weight Map
☑ Init Value
Color: 0.9 0.9 0.9
Value: 1.0

OK Ret
Cancel Esc

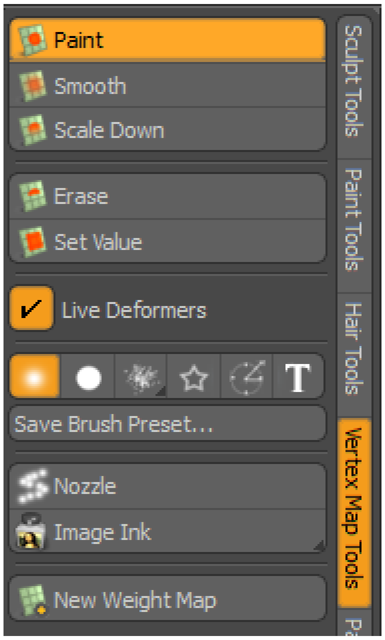
6. Above the Create tab, click the **Brush Toolbar** button and set the viewport display property to **Vertex Map**.


Perspective
Vertex Map
Viewport Textures

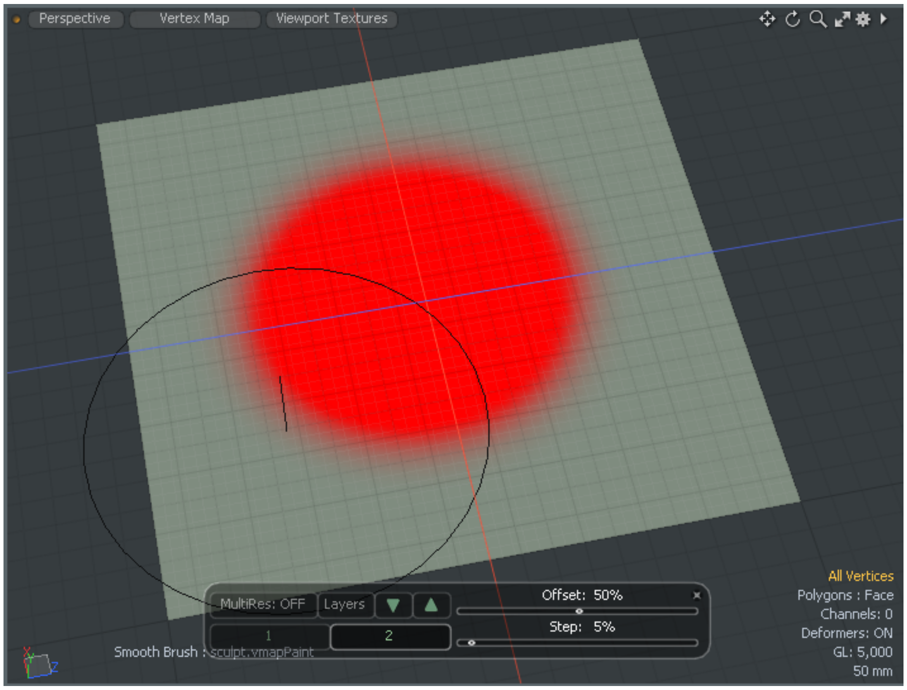
7. On the right panel, under **Vertex Maps**, right-click on **Weight**, and click **Clear**.


Properties
Channels
Vertex Maps
Stats
Name
Type
Weight Maps
B-Spline
B-Spline Weight
Subdivision
Subdivision Weight
Weight
Delete
Rename
Rename...
Copy
Paste
Clear
Add Transform Effector

8. On the left panel, open the **Vertex Map Tools** tab, and click **Paint**.

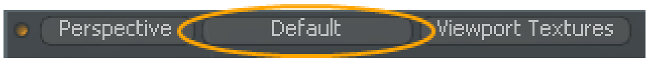

Paint
Smooth
Scale Down
Erase
Set Value
Live Deformers
T
Save Brush Preset...
Nozzle
Image Ink
New Weight Map
Sculpt Tools
Paint Tools
Hair Tools
Vertex Map Tools

9. Left-click on the center of the cube and drag to paint a circle.


Perspective
Vertex Map
Viewport Textures
MultiRes: OFF
Layers
1
2
Offset: 50%
Step: 5%
Smooth Brush : sculpt.vmapPaint
All Vertices
Polygons : Face
Channels: 0
Deformers: ON
GL: 5,000
50 mm

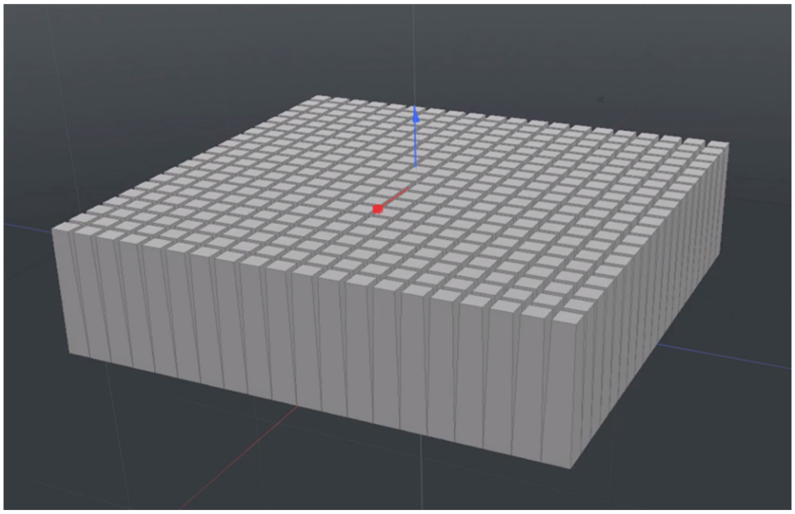
Tip: You can scale the paint brush size by right-clicking and dragging in the 3D viewport.

10. Press **Q** to drop the tool.
11. In the viewport, set the viewport display property to **Default**.

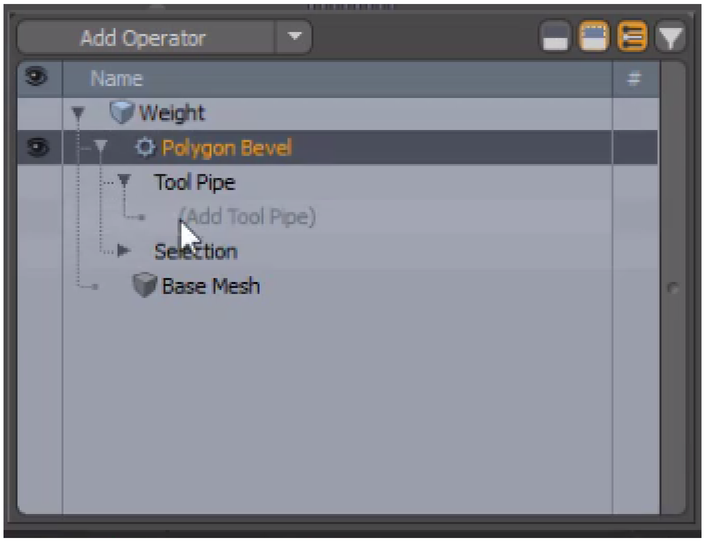

Perspective
Default
Viewport Textures

To add a Mesh Operation and Weight Map Falloff:

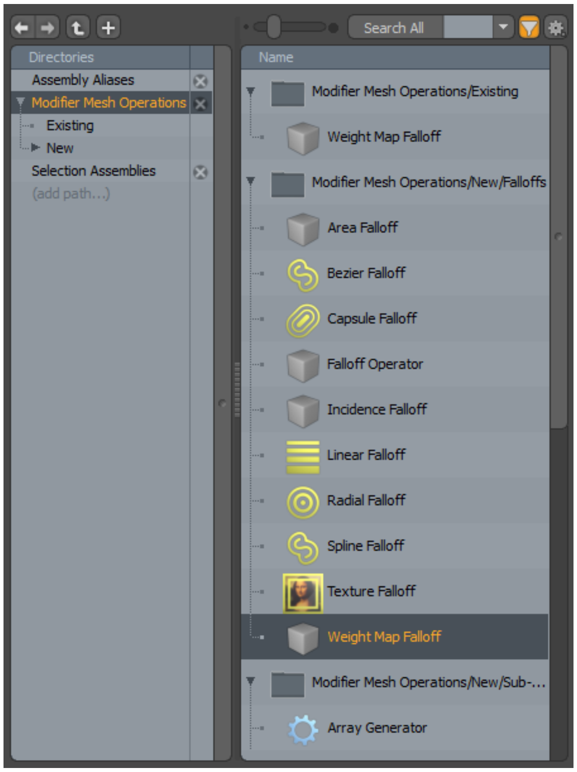
1. Next to the Items List, click the **Mesh Operations Viewport** button to open the **Mesh Operations** viewport to the left of the Items List.
2. Click **Add Operator**, expand **Mesh Operations** > **Polygon**, and double-click **Polygon Bevel**.
3. In the Polygon Bevel Properties, on the bottom of the right panel, disable **Group Polygons**.
4. Click and drag both the red and blue handles to bevel the plane.



5. On the Mesh Ops tab, expand **Polygon Bevel** > **Tool Pipe**, and click **(Add Tool Pipe)**.


Add Operator
Name
Weight
Polygon Bevel
Tool Pipe
(Add Tool Pipe)
Selection
Base Mesh

6. In the Add Operation dialog, select **Modifier Mesh Operations** > **New** > **Falloffs** and double-click **Weight Map Falloff**.



Directories

Assembly Aliases

Modifier Mesh Operations

Existing

New

Selection Assemblies

(add path...)

Name

Modifier Mesh Operations/Existing

Weight Map Falloff

Modifier Mesh Operations/New/Falloffs

Area Falloff

Bezier Falloff

Capsule Falloff

Falloff Operator

Incidence Falloff

Linear Falloff

Radial Falloff

Spline Falloff

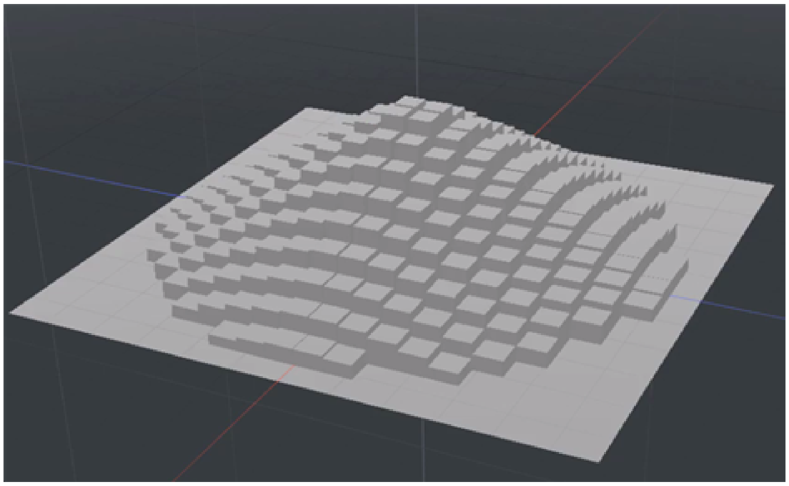
Texture Falloff

Weight Map Falloff

Modifier Mesh Operations/New/Sub-...

Array Generator

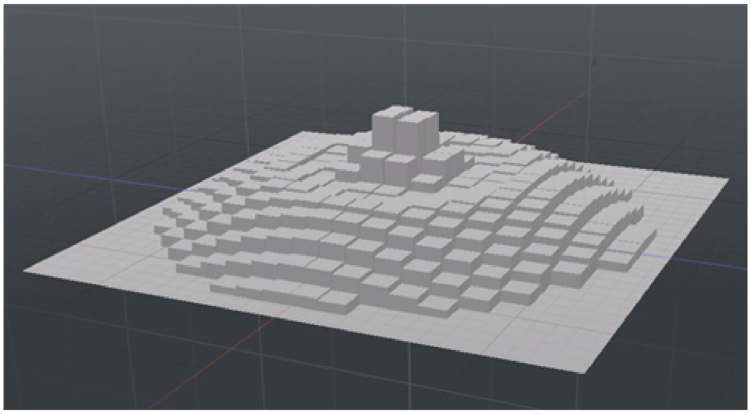
7. In the **Properties** tab, set **Weight Map** to use the plane's weight map. In this example, select **Weight**.



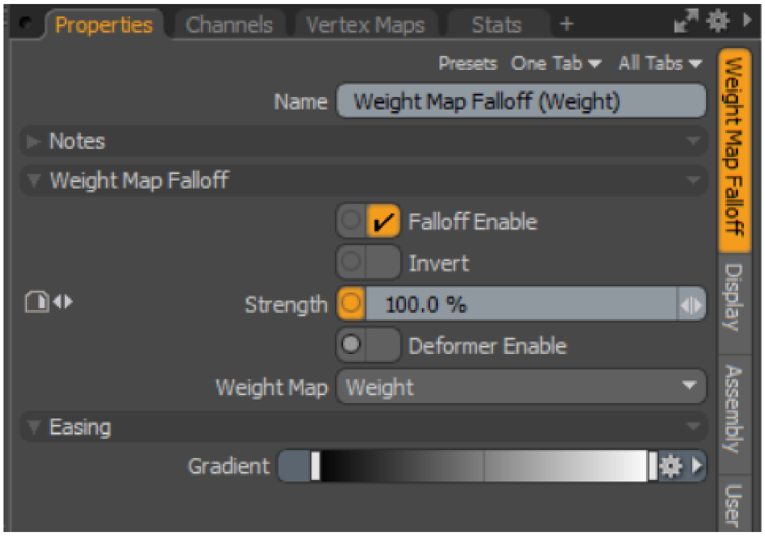
8. In the Easing Properties, press **Ctrl**/**Cmd** + **Alt** and click on the graph line to add keys.

   For more information, see Key Creation.

9. Click and drag the key points into a new position to create a new falloff shape.



Weight Map Falloff Properties


Properties
Channels
Vertex Maps
Stats
Presets
One Tab
All Tabs
Notes
Weight Map Falloff
Easing
Gradient
Weight Map Falloff
Display
Assembly
User

Name: Weight Map Falloff (Weight)

☑ Falloff Enable

☐ Invert

Strength: 100.0 %

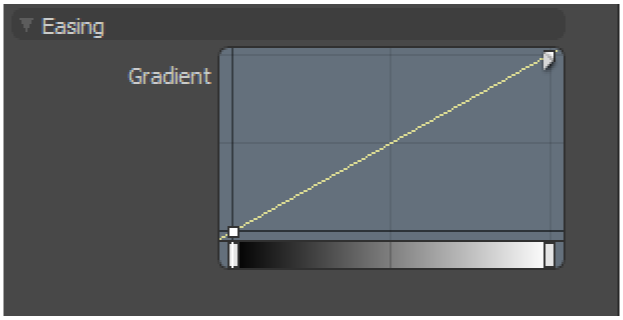
○ Deformer Enable

Weight Map: Weight

Name	Displays the Weight Map Falloff item name. The name of the Weight Map Falloff item inherits the name of the weight map that has been assigned and places it in brackets after Weight Map Falloff. For example, if your weight map is called weight, the name present in the name parameter would be **Weight Map Falloff (Weight)**.
Falloff Enable	Toggles the influence of the falloff on or off. When disabled, the falloff has no effect on its target. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.
Invert	When enabled, reverses the influence area of the falloff.

Strength	Controls the maximum amount of influence the falloff has on its target. At a value of **50%**, even areas that are affected fully by the falloff are only influenced half as much.
Deformer Enable	When enabled, you can connect the falloff to a mesh and use it as a mesh deformer. You can specify the initial rest state for the deformer if you select Setup Range as well and position the falloff appropriately.
Weight Map	Evaluates the selected weight map.

Easing Properties



Gradient	Allows you to re-map the weight of a falloff by creating a new custom falloff shape. In the Graph Editor, press **Ctrl/Cmd** + **Alt** and click on the graph line to add a new key point. Move the key points to create custom falloff shape. For more information, see Key Creation.

Using Influences

Deformers twist, bend, and otherwise distort a target mesh, but sometimes you may wish to affect an item's position, scale, or rotation without distorting the mesh itself. In Modo, deformers have two basic parts: an effector and an influence. These two parts collectively are considered deformers. Effectors provide instructions on how to affect the Target, and the influence determines what gets affected. Having these two parts provides flexibility. By default, each influences the entire Mesh. Learn about the different Influences you can use to transform individual elements in your animated scene.

General Influence

Many types of Deformers (such as the Bend, Lag, or Vortex Deformer) have their own **Effector** type, but you connect it to a target with a **General Influence**. With the **General Influence** you can assign various vertex tagging methods to define a grouping of vertices for the Deformer to influence, which limits their application to specific user-defined areas of a mesh (or the entire mesh). You can then transform these tagged vertices with Locators to control the effect. Doing this, gives you a direct way to influence geometry in a scene.

The most flexible of the tag types is the **Weight Map** type, which you can create by using the Weighting tools and Brushes. When animating with a skeleton, you can deform a mesh with these tools. You assign a **General Influence** with the type set to **Weight Map** for each joint within the hierarchy. Modo assigns the resulting Weight Map from the Bind process to the Influence based on the **Type Name** option. Once defined, any transforms applied to the joint also transform the associated vertices. **General Influences** are also the basis for the transformation Deformers: **Rotation** and **Transform**. These are based on simple Locator Item transforms that mimic the **Weight Map** deformers from previous versions of Modo.

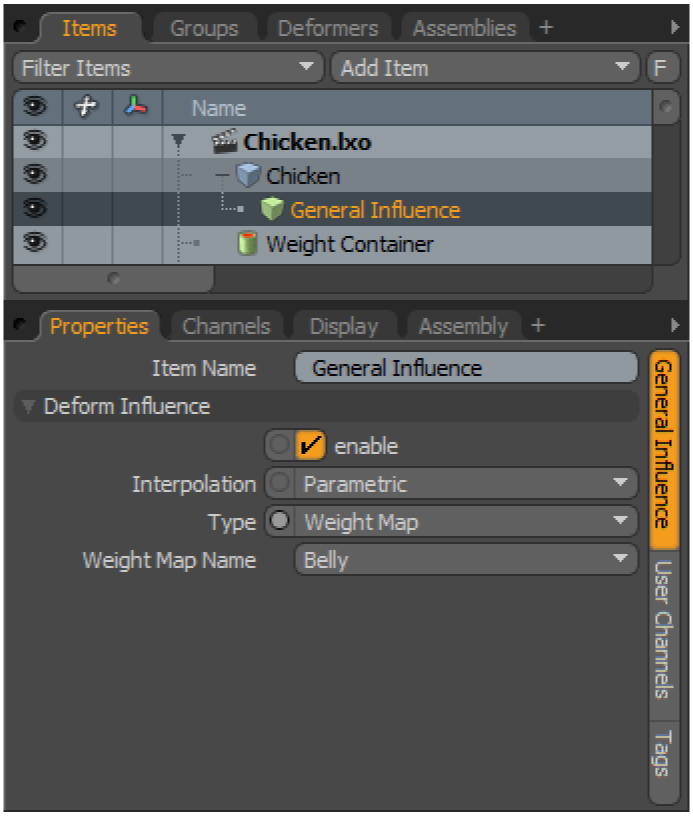
Applying a General Influence

Modo generates **General Influences** automatically when you apply a Bend, Lag, Magnet, Rotation, Slack, Spline, Transform, or Vortex Effector with the method given in the Working with Deformers topic. By default, each influences the **Entire Mesh**, but you can define a polygon tag and then specify it by name in the **Properties** panel to limit the area of influence to specific regions of a model.

Normalization

When binding joints to a mesh, you can find the resulting **General Influences** placed within a **Normalization** folder in the **Deformers** viewport. When two opposing **Weight Maps** influence the same surface, Modo produces smoother deformations when the transitions between the two are soft (such as what the Bind process creates). However, in these transition areas there is an anomaly. If a single vertex is influenced equally by two **Weight Map** values (such as 50% and 50% respectively) applying a 50% transform amount and then another at 50% doesn't mathematically add up to 100% for a full Deformation when they are applied in sequence. You resolve this by using **Normalization**. After Modo places **Influences** within a **Normalization** folder, it normalizes their weight values to adjust all overlapping values to equal to 100% when combined. Additionally, Modo calculates the deformations concurrently to eliminate the anomaly that occurs with sequential calculation.

General Influence


Items
Groups
Deformers
Assemblies
+
Filter Items
Add Item
F
Name
Chicken.lxo
Chicken
General Influence
Weight Container
Properties
Channels
Display
Assembly
+
General Influence
User Channels
Tags

Item Name: General Influence

Deform Influence

☑ enable

Interpolation: Parametric

Type: Weight Map

Weight Map Name: Belly

Item Name: This displays the current item's name. You may change it by clicking the current name and typing a new name.

Deform Influence

Enable: Toggles the Influence item on or off. When selected, the influence affects its target; when cleared, Modo ignores its influence on the target. However, Modo saves disabled influences with the scene, and the values persist across Modo sessions.

Interpolation: Modo provides two ways of calculating the resulting deformation from Influences:

- **Parametric** - Modo passes vertex values directly to the Effector for deformation. This produces smooth-curved interpolations, but it requires more overhead for the calculations.
- **Linear** - Modo calculates vertex values between undeformed and fully deformed items and linearly applies a deformation between the two.

Type: Specifies the type of Vertex Map tag:

- **Entire Mesh**- Influences the entirety of the mesh within the current item layer.
- **Weight Map**- Influences vertices with **Weight** values greater than **0**.

- **Vertex Selection Set**- Influences all vertices within the defined Selection set.
- **Material**- Influences all vertices within the defined Material tag.
- **Part**- Influences all vertices within the defined Part tag.
- **Polygon Selection Set**- Influences all vertices within the defined Polygon Selection set.

<Type Name>: Lists all the tag types assigned to the Mesh Item based on the selected **Type**. (The name for this option depends on the selected **Type**. For example, if you select **Weight Map**, this is the **Weight Map Name**.)

Item Influence

Deformers twist, bend, and otherwise distort a target mesh, but sometimes you may wish to affect an item's position, scale, or rotation without distorting the mesh itself. This can be useful when a rigid item needs to stay in a proper position relative to a Deforming surface or when multiple items need to transform in a uniform way (such as a tower of spheres that bends around on itself). In these cases, you can apply an **Item Influence**. When an **Item Influence** replaces the standard General Influence, the Effector only applies basic Item transform properties.

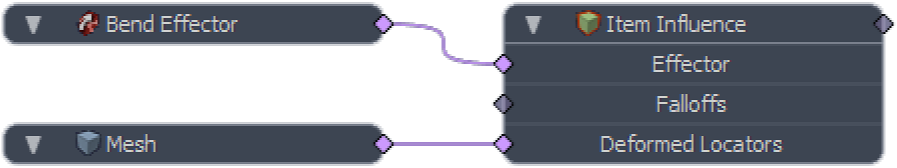
Applying an Item Influence

As a specialty case, Modo does not offer a means to automatically connect the Influence to the target item; therefore, you should be familiar with the Schematic view.

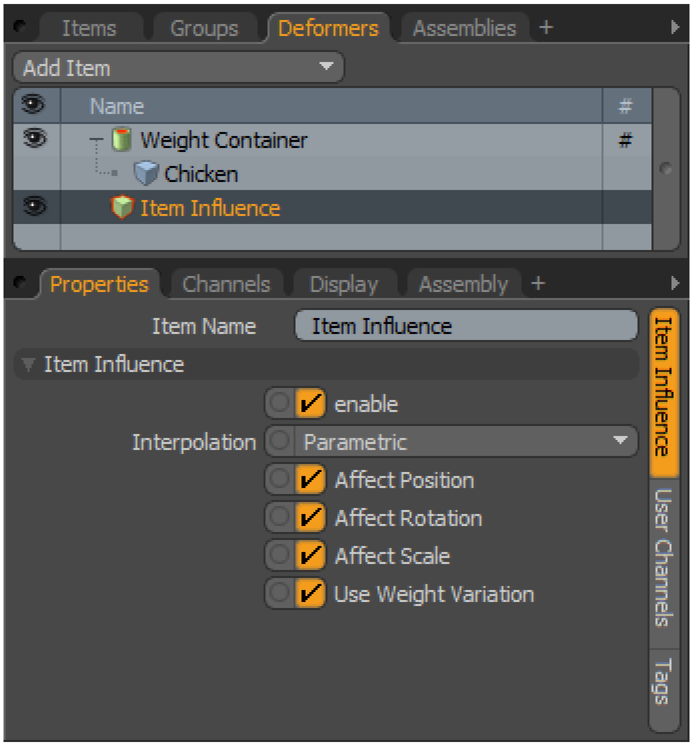
You need to add an Effector, the **Item Influence**, and the target Mesh Item(s) manually to the **Schematic** view. To do this, first switch to the **Setup** interface tab, where the **Schematic** view shows up as a viewport directly below the 3D viewport. Next, add the Mesh Item(s) by selecting all the desired layers in the **Items** list and then dragging the layers into the **Schematic** view area. This creates nodes that represent each item in the view. Next, click the **Add** button on the top of the **Schematic** window, itself, to create the desired **Effector** (such as the Bend Effector) and then create the **Item Influence** the same way.

With the newly created Influence and Effector placed into the **Schematic** view, connect them. You link non-channel connections with relationship connections (small diamond-shaped connectors on the sides of a node). Dragging the diamond from the Effector to the Item Influence creates a line, and Modo defines the connection when you release the mouse button over the **Effector** input. Doing the same action from the Mesh Item to the Item Influence **Deformed Locators** input connects them as well. You can use one Effector and Influence to affect multiple Items; therefore, if there are more Items, drag out additional connections to every Mesh Item that is meant to be influenced. From there you can adjust the settings of the Effector to properly transform the target mesh(s).

Note: These transform operations are relative to the target item's Center position.


Bend Effector
Mesh
Item Influence
Effector
Falloffs
Deformed Locators

Item Influence


Items
Groups
Deformers
Assemblies
Add Item
Name
Weight Container
Chicken
Item Influence
Properties
Channels
Display
Assembly
Item Influence
User Channels
Tags

Item Name: Item Influence

Item Influence

☑ enable

Interpolation: Parametric

☑ Affect Position

☑ Affect Rotation

☑ Affect Scale

☑ Use Weight Variation

Name:This displays the current item's name. You may change it by clicking the current name and typing a new name.

Item Influence

Enable: Toggles the Influence of the layer on or off. When clear (disabled), the Influence has no effect on the scene. However, Modo saves disabled Influences with the scene, and the values persist across Modo sessions.

Interpolation: Modo provides two ways to calculate the resulting deformation from Influences:

- **Parametric**- Modo passes transform values directly to the Effector for the deformation to produce smooth-curved interpolations, but this has more overhead for the calculations.

- **Linear**- Modo calculates transform values between undeformed and fully deformed states and applies values linearly between the two states.

Affect Position: Influences the **Position** transform channels of the target item if selected.

Affect Rotation: Influences the **Rotation** transform channels of the target item if selected.

Affect Scale: Influences the **Scale** transform channels of the target item if selected.

Use Weight Variation: Samples the weighting gradient to determine the item's transform if selected.

MDD Influence

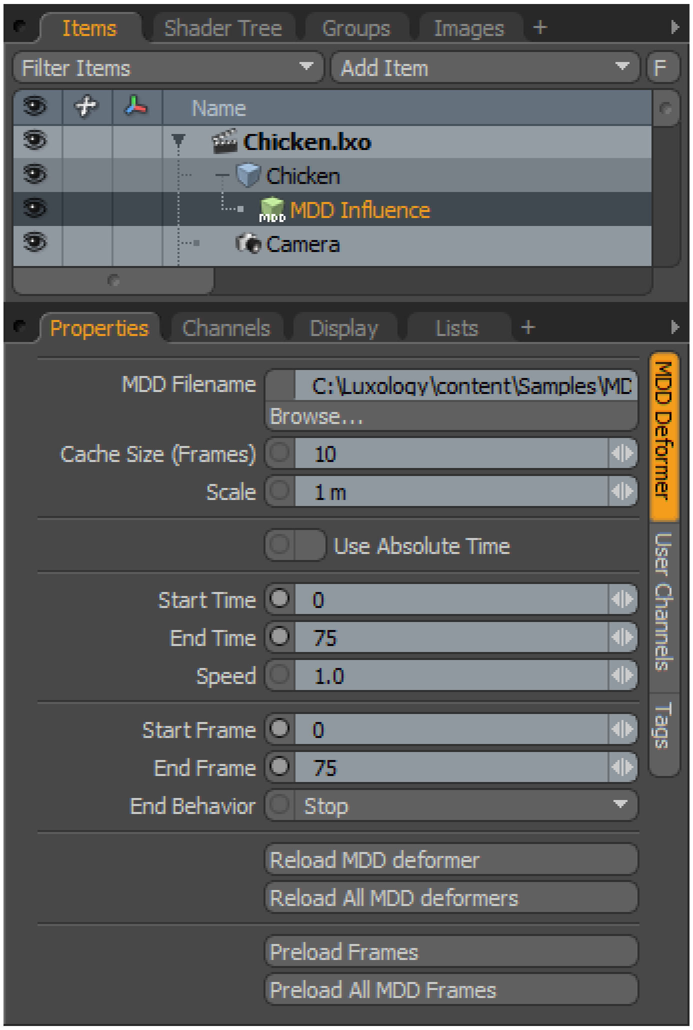
Daisuke Ino invented Motion Designer Document (MDD) files during his work on a dynamics engine built as a plug-in to a host 3D application. These files are a simple point cache that stores a location in space for every vertex in a model for every frame in the animation. The original intent for these files was for you to freeze a dynamic simulation so that the result could be saved to disk, reloaded, and played back without having to simulate again. Soon Hollywood visual FX facilities realized an interesting benefit to these simple point cache files. Software could generate these files very easily for any kind of deformation animation. This not only allowed these files to replace re-doing complex IK and bones calculations, but by creating MDD deformers in multiple applications, these could transfer data about animations done in one application into a completely different program for final rendering.

Today MDD exporters exist for almost every major 3D application on the market. With MDD support inside Modo, you can model in Modo, perform complex animations or simulations in another host application, and return that model with MDD files to Modo for final frame-out rendering.

Applying an MDD

To apply an MDD file to a file, first generate an MDD file from another 3D package. Almost every 3D program has an exporter. (If that program doesn't have a built-in exporter, search online for one.) Create the base geometry for the MDD file in Modo or import it into Modo. With the target item layer selected in the **Items** list, right-click the item, itself, and choose **Add Mesh Operation**, then under **Influences**, double-click **MDD Influence**. In the OS-specific dialog box, locate and select the saved MDD file. After you click **Open**, Modo applies the MDD Influence item to the target mesh. Scrubbing in the viewport displays the results. You can make adjustments to the attributes, if necessary, in the **Properties** panel.

MDD Influence


Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
Chicken.lxo
Chicken
MDD Influence
Camera
Properties
Channels
Display
Lists
MDD Deformer
User Channels
Tags
Browse...
Reload MDD deformer
Reload All MDD deformers
Preload Frames
Preload All MDD Frames

MDD Filename: C:\Luxology\content\Samples\MD

Cache Size (Frames): 10

Scale: 1 m

○ Use Absolute Time

Start Time: 0

End Time: 75

Speed: 1.0

Start Frame: 0

End Frame: 75

End Behavior: Stop

MDD Filename: This shows you the path and file name of the MDD file that Modo applied at the time of creation.

Cache Size (Frames): Determines the number of frames from the sequence held in memory at one time. If this value is exceeded during playback, Modo loads frames from disk, which slows performance. If enough RAM is available, you may prefer to specify **Preload Frames** to increase scene playback stability.

Scale: Increases or decreases the size of the item as it is deformed with the MDD file. The default scale is 1 meter.

Use Absolute Time: Uses the times indicated in the file header of the MDD file rather than Modo time settings. By specifying this, you can export an animation with precise timing and replay it in Modo without having to adjust any MDD time values.

Start Time/End Time: Indicate when the MDD animation should be applied during your scene. By default, Modo begins the MDD file at frame 0 of your animation. However, to delay the start frame you can change the **Start Time**. For example, by changing the **Start Time** to **2**, the MDD file begins its initial frame 2 seconds into the animation. This is a convenient method for time shifting your MDD file.

Speed: Adjusts the playback rate of the selected MDD file. By default, Modo uses a one-to-one match of MDD frame to Modo frame. To slow down the MDD playback, lower the value below 1. To increase playback speed, increase the number.

Start Frame/End Frame: Indicate the frames within the MDD file that Modo should use. By default, when you apply an MDD file, Modo reads the values from the file directly. To use only subset of the MDD data, set the desired in and out points.

End Behavior: If the MDD data is shorter than the Modo animation, Modo needs to know what to do when it reaches the end of the MDD file. The default behavior of **Stop** holds the last frame of the MDD file. You can also **Repeat** (to loop the playback) and **Ping-Pong** (to playback normally and then reverse the playback).

Reload MDD Deformer/Reload All MDD Deformers: Reloads the selected MDD file or all MDD files in a given project. With these buttons, you can load a project into Modo, return to another application to modify the simulation or character animation, export the MDD file, and then force Modo to update to the most recent data without having to reload the entire project.

Preload Frames/Preload All MDD Frames: Loads all the MDD frames into memory for the entire sequence on the selected item. This improves playback speed because the Deformer won't have to load data for every frame. The **All** button loads frames for all MDD Influence items in the scene, but doing this requires more memory than just preloading a single MDD file.

Morph Influence

With morph maps, you can save alternate user-defined deformations for any mesh. For example, you can animate individual morph map deformations over time to animate facial expressions. To animate any morph map, you must first apply a morph deformer to the model.

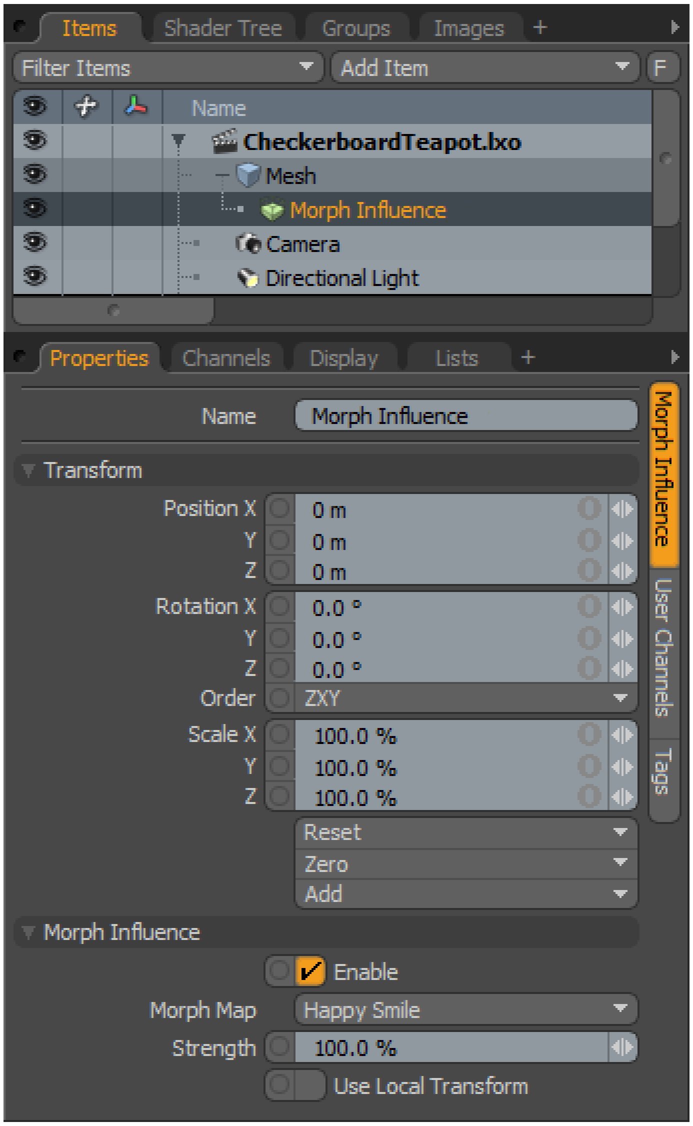
Applying a Morph

Before you can apply a morph, you need to create it. You add new morph maps by using the Vertex Map List viewport or within the **Properties** panel of a Mesh Item, under the **Mesh** sub-tab. Any modifications that you make to the Mesh Item while the specific morph map is selected in either location are stored in that map. For more information on creating morph maps, see the Working with Vertex Maps topic.

To apply a morph to a mesh for animating it, select the target Mesh Item in the Items (Scenes) List. Then right-click the item name and choose **Add Mesh Operation,** then under **Influences**, double-click **Morph Influence**. Modo automatically adds the Morph Influence item and connects it to the mesh. You can also select **Morph** in the **Setup** toolbox on the **Deformers** sub-tab. Once you apply the morph, select the Morph item and view its attributes in the **Properties** panel. Each Morph Influence supports a single morph map. For multiple maps, you must add additional Morph Influences. Combined Morph Maps are additive, and Modo blends all together appropriately to provide predictable results. Additionally, you can specify a morph **Strength** value greater than **100%** (or as a negative value) to produce unusual results.

Note: If you are having unexpected results with the morph such as a bad deformation, the vertex index order between the source and target mesh may be out of sync. See Reordering Vertex Index Order for more information on how to fix this.

Tip: One of the most confusing aspects of working with Morphs is that, when modeling, a Morph appears automatically in the 3D viewport when selected, but it does not appear in the **Animation** and **Render** interface tabs. To actually see and animate a Morph, you need to apply the Morph Influence and set its strength to some value greater than **0**. Also, you must select **Enable Deformers** in the viewport.


Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
CheckerboardTeapot.lxo
Mesh
Morph Influence
Camera
Directional Light
Properties
Channels
Display
Lists
Morph Influence
User Channels
Tags
Transform
Reset
Zero
Add
Morph Influence

Name: Morph Influence

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ Enable

Morph Map: Happy Smile

Strength: 100.0 %

☐ Use Local Transform

Name: This displays the current item's name. You may change it by clicking the current name and typing a new name.

Transform

> **Note:** Transform values changes the position of the Influence item's locator, but they have no effect on the Morph itself.

Position XYZ: Specify to position the Influence item numerically in XYZ space.

Rotation XYZ: Specify to set the rotation of the Influence item numerically.

Order: Sets the order that Modo applies rotations to the Influence item.

Scale XYZ: Sets the size of the Influence item numerically.

Reset: Resets the selected transform values to (0,0,0) and returns the items to their default state.

Zero: Resets the selected transform property value to **0** and leaves the Center position and Item position intact.

Add: Use to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values. (Keeping the default values is necessary for Referencing. To be able to override a channel, it must first exist.)

Note: Transform items are the channel groups associated with an item that store its transform values. These transform values control an item's position, rotation, or scale. By default, new items do not have any transform items associated with them (although they are visible within the **Properties** panel). Add only the necessary transforms on an as-needed basis to reduce scene overhead. If you don't use the **Add** function, you can transform the target item with one of the various transform tools (or by editing the values). When you do so, Modo automatically adds the transform item to the **Channels** viewport list.

Morph Influence

Enable: Toggles the influence of the layer on or off. When clear (disabled), the layer has no effect on the scene. However, Modo saves disabled layers with the scene, and the values persist across Modo sessions.

Morph Map: Specifies the target Morph Vertex Map.

Strength: Modulates the strength of the Morph. At **100**%, Modo deforms the mesh fully at the morph state.
Use Local Transforms: Adjusts the offsets of the Morph so that they deform by using local coordinates. This is useful in creating corrective Morphs when you couple them with other deformers.

Push Influence

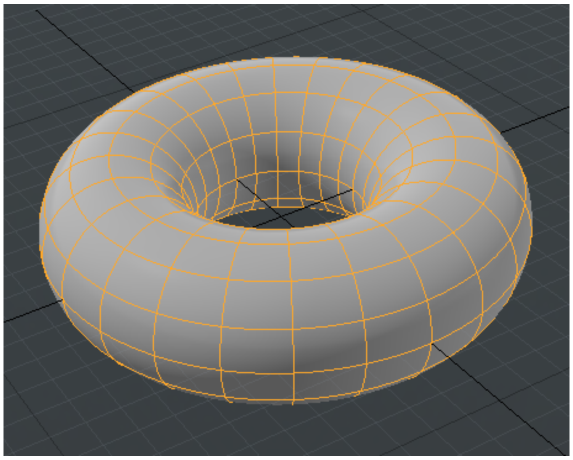
The Push Influence deforms geometry by pushing vertices along their average normals. You can specify the push distance in meters. In addition, you can use the Push Influence with a falloff to further modulate the push distance.

Using the Push Influence

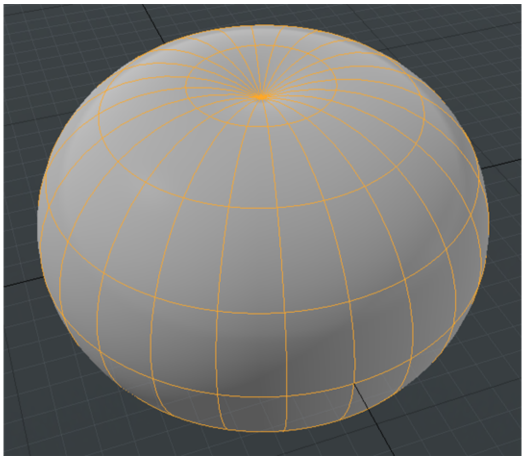
1. In the 3D viewport, right-click the mesh item you want to add the Influence to, and click **Item Operations** > **Add Mesh Operations**.
2. In the **Add Mesh Operations** dialog, click **Deformers** > **Influences** > **Push Influence**.

> **Note:** When working with the **Mesh Ops** tab, you can add the Push Influence by clicking **Add Operator** in the **Mesh Operations** list.

Your mesh is pushed out to a distance of one meter, which is the default setting. You can edit the distance in the **Properties** panel of the Push Influence.



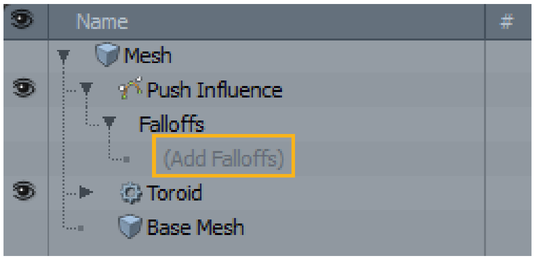
Toroid without any deformers applied.



Toroid with a Push Influence of 1 m.

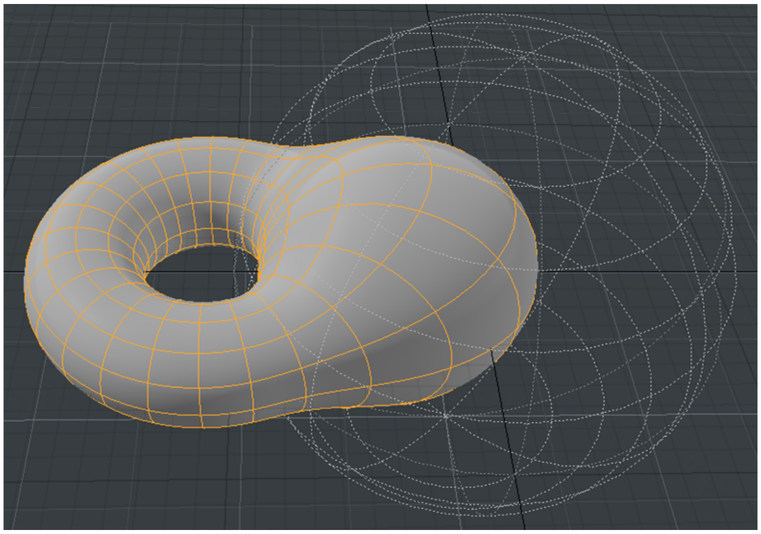
You can also scale the push distance by using a falloff.

1. To add a falloff, expand the Push Influence item in the list to reveal its **Falloff** input.



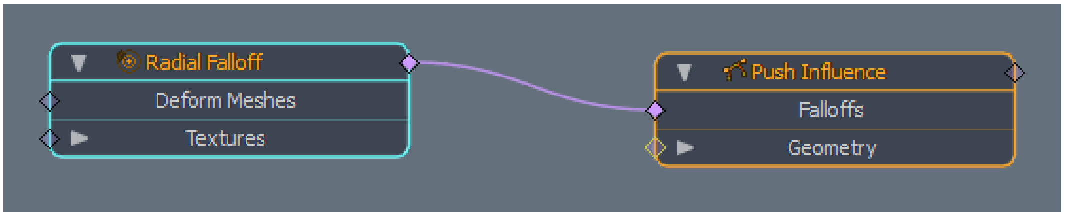
2. Click **Add Falloff**, and select a falloff in the **Preset Browser**.

   The falloff appears in the viewport and you can customize it by editing the settings in its **Properties** panel. You can see the effect of the Push Influence modified by the falloff.



Toroid with Push Influence and Radial Falloff.

You can also use the **Schematic** viewport to add the falloff. Here's how to connect it to the Push Influence.



Creating a Full Body IK Setup

Modo has an Inverse Kinematics (IK) system that is fairly automatic and extremely robust. Its Full Body IK is an automatic IK system that works in a wide variety of rigging situations. It is commonly used for characters, but you can apply it to any hierarchy chain of locator-based items. The following topics below provide information on how to create a Full Body IK setup and details on the options available to customize your animated elements.

- Setting up a Full Body IK
- Positioning and Rotating the Full Body IK

Setting up a Full Body IK

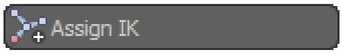
There are two methods for animating hierarchies of elements (for example, having a character's arm reach for an object on a table): Forward Kinematics (FK) and Inverse Kinematics (IK).

With FK, you set up the animation by positioning the root element and then adjusting all the elements down the chain. First, you position the shoulder joint, then the elbow, and finally the wrist. It may take only a few attempts to get the hand properly positioned to the object.

IK works in reverse and is far more intuitive. With IK, select the hand and move it to the object. The wrist, the elbow, and shoulder joints adjust automatically to maintain their proper orientation toward the goal, in this case, the hand.

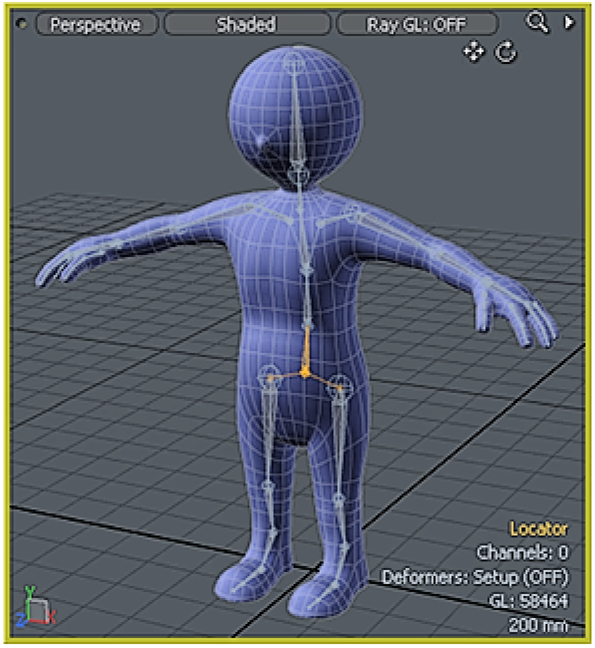
Modo has an IK system that is fairly automatic and extremely robust. Its Full Body IK is an automatic IK system that works in a wide variety of rigging situations. It is commonly used for characters, but you can apply it to any hierarchy chain of locator-based items. You set up the chains by selecting the target elements and applying the commands found in the toolbox of the **Setup** and **Animate** interface tabs in the **Inverse Kinematics** sub-tab.

Assigning IK

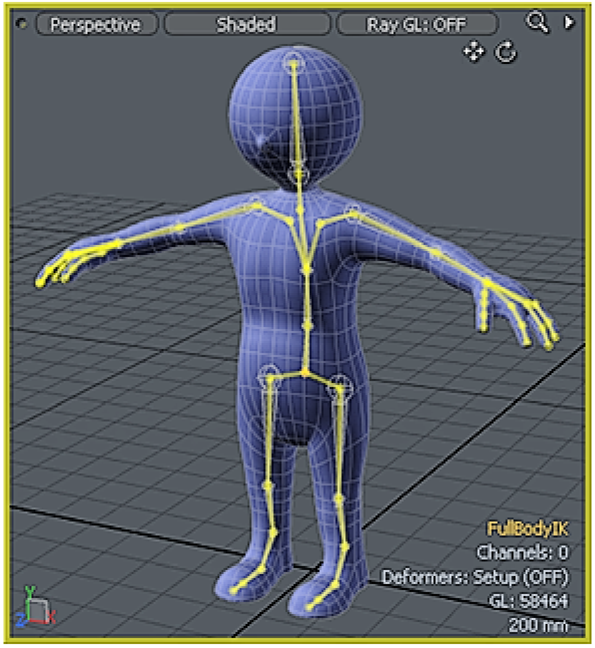


To use IK in Modo, first you must assign IK to the items that you want it to work with. You do this in **Setup** mode to define the rest positions for each joint. You can apply IK to a hierarchy of items or to a full Skeleton — either before or after the binding process.

To assign IK to an entire hierarchy, select the root item of the chain and click the **Assign IK** button. Modo assigns IK to all items in the hierarchy from the root outward.

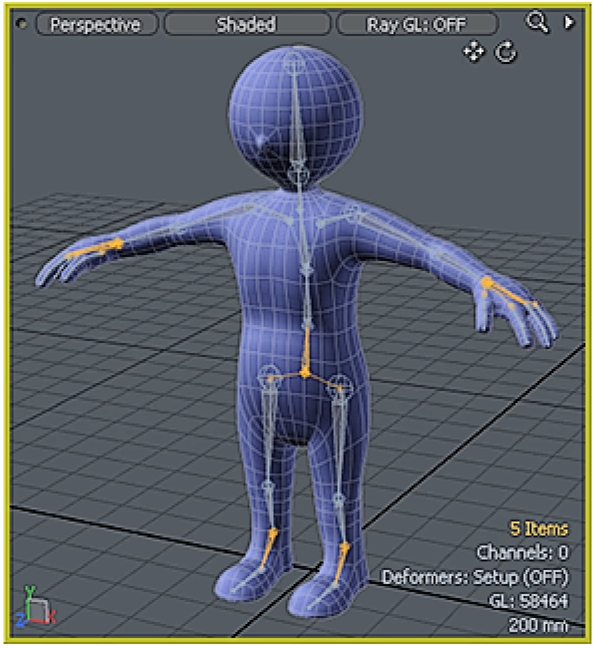

Perspective
Shaded
Ray GL: OFF
Locator
Channels: 0
Deformers: Setup (OFF)
GL: 58464
200 mm

Select the **Root** element of the hierarchy

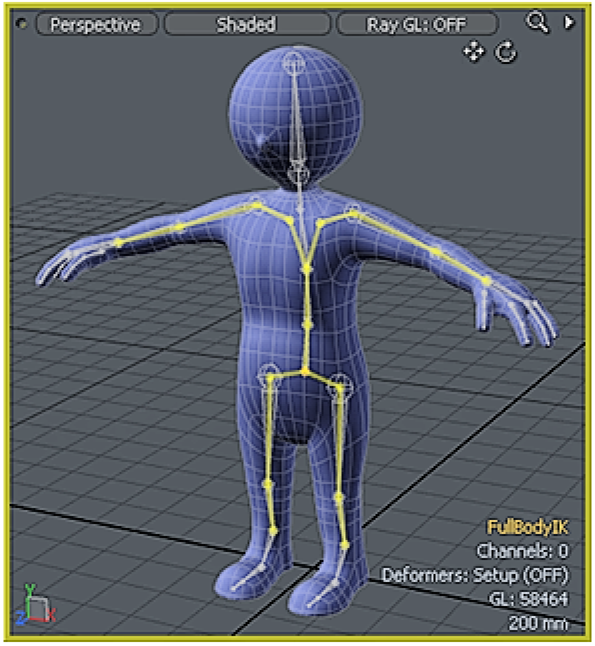

Perspective
Shaded
Ray GL: OFF
FullBodyIK
Channels: 0
Deformers: Setup (OFF)
GL: 58464
200 mm

Apply IK applies IK to the entire chain

To assign IK to just parts of a hierarchy, select the root item first. Then, select the additional item(s) where the IK terminates. Finally, click the **Assign IK** button.

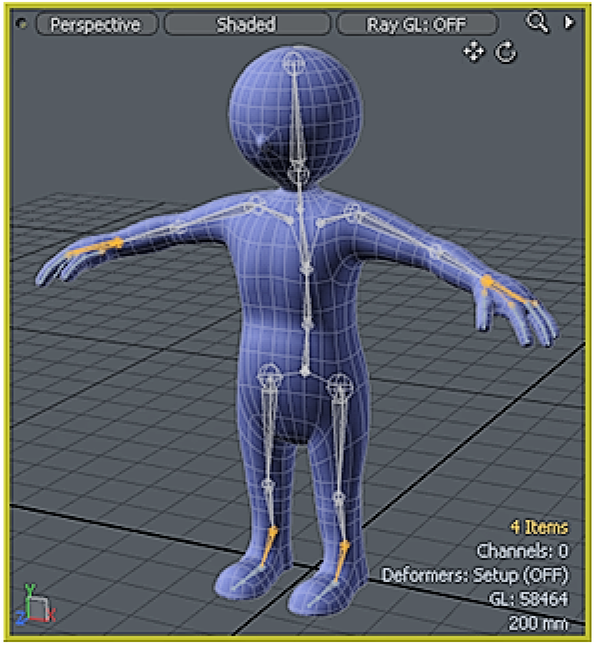

Perspective
Shaded
Ray GL: OFF
5 Items
Channels: 0
Deformers: Setup (OFF)
GL: 58464
200 mm

Select the **Root** and **Terminating** elements

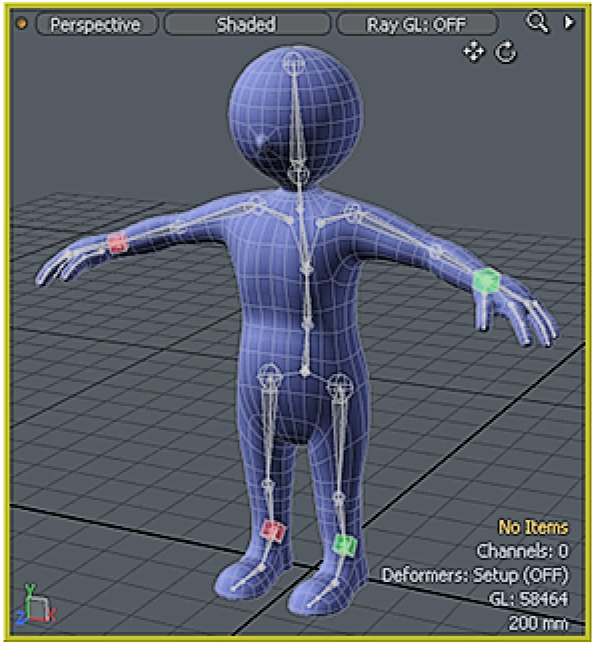

Perspective
Shaded
Ray GL: OFF
FullBodyIK
Channels: 0
Deformers: Setup (OFF)
GL: 58464
200 mm

Modo applies IK between selected elements

After you assign IK this way, Modo retains the selection of the terminating elements. You can then add goal items by clicking the **Add Goal** button. Now, when you transform the goals, the chain adjusts the positions and rotations toward the goal. You can then exit **Setup** and begin to position the chain.



The terminating elements remain selected



The **Add Goal** button adds goals to each terminating element

All the items in the chain must be part of the same hierarchy, and all items must be 3D locator-type items (such as joints created by the Skeleton tool, locators, or Mesh Items). If Modo encounters any other type of item in the hierarchy or if another IK chain is present, the IK process terminates at that item. Any hierarchies that have scaled items need to have their scale frozen before assigning IK to them. With the IK defined, Modo creates an IK solver and parents it to the root item (**FullBodyIK**). This item has the settings that control the overall behavior of the chain, which can be found in the item's **Properties** panel when selected. (For information, see the Full Body IK topic.)

Goals

With the IK chains defined, FK (forward kinematics) still control the target elements directly, and you can still individually transform each item in the chain. By adding **Goals** and **Tasks**, you automatically activate the IK. (A goal is a general term for items that drive IK chains. Manipulating a goal item causes the rotations of items in the IK chain to change to make the position or rotation of the end item match that of the goal item.) You can add goal items to any item belonging to an IK chain (except for the chain root) and can have multiple goals. The IK solver tries to find the best solution to satisfy all the goals.

To add a goal, select the item in the IK chain that represents the part of the chain that should move toward the goal (such as the hand/wrist joint in an arm to give FK control over the fingers). Next, click the **Add Goal** button in the IK toolbar. Modo places a small cube item (as a locator shape) at the position and orientation of the selected chain item, adds position and orientation tasks to the selected joint, and properly connects them together with channel links. The cube has an initial custom color to help identify it: green for the positive side and red for the negative side.

Assign Goal

If you have an existing item to use as a goal, then you can use the **Assign Goal** commands to attach it to a chain. Select the target chain item and the item to be used as a goal (in any order). Choose one of the **Assign Goal** commands by right-clicking the **Assign Goal** button. With these commands you can assign an item as a **Position Goal**, an **Orientation Goal**, or both.

Remove Position Task

You remove a goal item by deleting the goal item and then selecting the chain item that the goal was connected to and removing the **Tasks** by using the buttons in the IK toolbar.

Tasks

IK assigned to a chain only comes into effect when you add **Tasks** to one or more items in the chain. Tasks are related to Goals because they are the instructions to the joint to retain a position, rotation, or both. For a **Position Task** the IK solver attempts to keep the chain item at the World location the item had when the task was added. For a **Rotation Task**, the IK solver attempts to keep the chain item at the same World orientation that it had when the task was added. Modo implements **Tasks** as groups of channels added to the joint. Each task has a set of FK channels and a Matrix channel. You use these channels to set the task transforms (in World space), so modifiers, channel links, or other means can drive them. You can add tasks to multiple items in a chain. If you do so, the IK solver tries to find the best solution to satisfy all the tasks on the chain. You can find settings for each task on a chain item in the item's **Properties** panel.

Add Position Task

To assign a **Position Task**, select the target Joint(s), and click the **Add Position Task** button. Modo adds additional Position Task options to the joint's attributes in the **Properties** panel when you select the joint.

Add Orientation Task

To assign an **Orientation Task**, select the target Joint(s), and click the **Add Orientation Task** button. Modo adds additional Orientation Task options to the joint's attributes in the **Properties** panel when you select the joint.

Adding to the Chain

You can add or remove elements from a chain with the **Add** and **Remove** buttons.

Add to Chain

To add a joint or chain of joints to an existing chain, the new element(s) must be part of the same hierarchy. One way to add elements is by selecting the root item in the target chain and then activating the **Skeleton** tool. You can then select the parent element in the chain and click away from that element to draw the actual joint, additional joints are added down the hierarchy as each is created.

A second way is by parenting the joint (or chain of joints) to the target chain's hierarchy manually with the parenting option or the Drop Action (particularly by using the **Parent** and **Parent in Place** options). Once you've made the item(s) part of the main hierarchy, select the element(s) or the base of the new chain and click the **Add to Chain** button.

Remove from Chain

To remove a joint or a chain of joints, select the base element to remove and click the **Remove from Chain** button.

Adjusting the Chain

Update Rest Pose

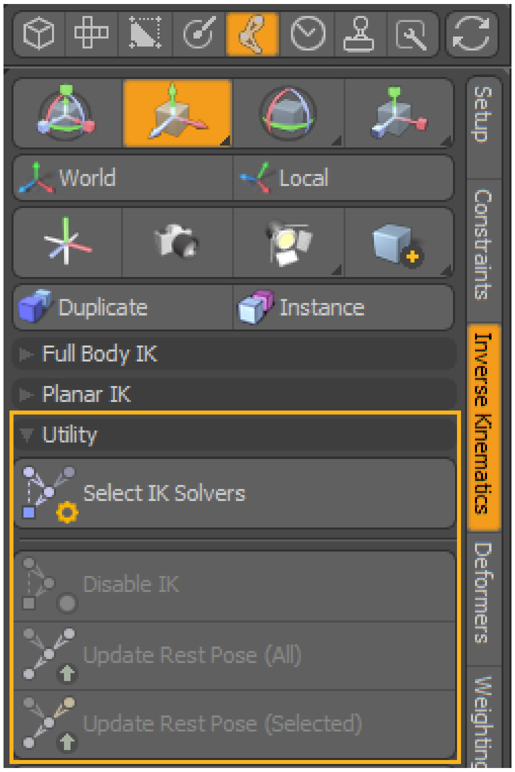
You may need to adjust parts of a chain, such as repositioning a knee or elbow joint. You do this type of adjustment in **Setup** mode. When in **Setup** mode, you can adjust the joints and then click the **Update Rest Pose** button to update the IK chain. You may need to update the location or orientation of any goal items that are driving the tasks as well with the Match Position or Match Rotation buttons in the **Setup** tab of the toolbar. If tasks are not driven by goals, the chain updates itself to match the values of the task channels. You may need to edit these manually depending on your setup.

Removing IK

You can remove IK from the hierarchy by selecting the **FullBodyIK** item and clicking the **Remove IK** button in the Inverse Kinematics toolbox. Modo removes the IK settings from each item and deletes the **FullBodyIK** solver item from the scene.

Utility Options

Utility options have been added to the **Inverse Kinematics** tab under the **Setup** toolbar in Modo 14.1v1. Using these buttons you can now disable both Full Body and Planar Inverse Kinematics in setup mode, as well as update the Rest Pose and quick select IK solvers.


World
Local
Duplicate
Instance
Full Body IK
Planar IK
Utility
Select IK Solvers
Disable IK
Update Rest Pose (All)
Update Rest Pose (Selected)
Setup
Constraints
Inverse Kinematics
Deformers
Weightin

To quick select an IK Solver item, select a Joint or Goal item belonging to the chain from which you want to access the IK Solver item, and click the **Select IK Solvers** button.

To disable an IK from Setup mode, enter Setup mode and click the **Disable IK** button found in the **Setup** toolbar under **Inverse Kinematics** > **Utility**. This toggles the IK off and on. The state is stored as a per scene setting and is saved in the LXO scene file.

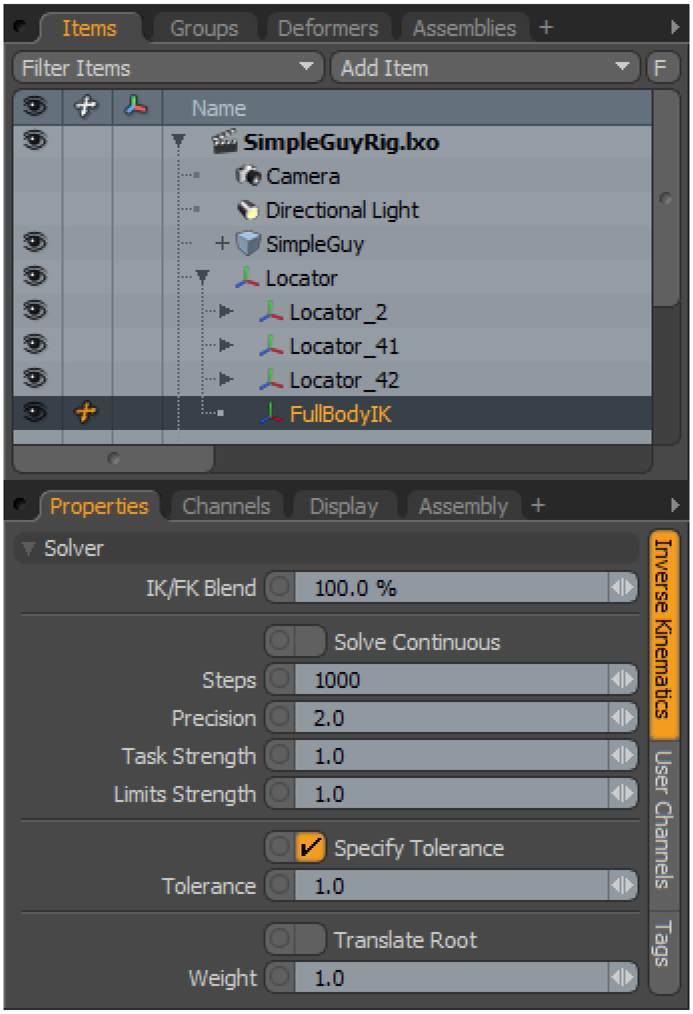
Note: When disabling IK, the Joints and Goals are not automatically returned to their Rest Pose.

Whilst IK is disabled, the **Update Rest Pose (All)** and **Update Rest Pose (Selected)** buttons are active. These allow the Rest Pose of all IK chains in the scene to be updated, or just selected chains by selecting any Joint or Goal item belonging to the chain you want to update.

Positioning and Rotating the Full Body IK

Inverse Kinematics, IK, is a system where you can position and rotate a goal item. The IK hierarchy chain reaches toward that goal to make a faster and more intuitive animation workflow. Modo provides a Full Body IK option that is easy to setup and use. See the Full Body IK Setup topic for information about setting up the chain.

Solver


Items
Groups
Deformers
Assemblies
Filter Items
Add Item
F
Name
Properties
Channels
Display
Assembly
Inverse Kinematics
User Channels
Tags

SimpleGuyRig.lxo
Camera
Directional Light
SimpleGuy
Locator
Locator_2
Locator_41
Locator_42
FullBodyIK

Solver
IK/FK Blend 100.0 %
☐ Solve Continuous
Steps 1000
Precision 2.0
Task Strength 1.0
Limits Strength 1.0
☑ Specify Tolerance
Tolerance 1.0
☐ Translate Root
Weight 1.0

When you apply any IK chain to a hierarchy, Modo creates an IK Solver item and parents it to the root of the hierarchy (by default, **FullBodyIK**). This has the settings that control the IK for the entire chain (including all child items of the root).

IK/FK Blend: Controls to what degree the chain is affected by the IK solver. This attenuates between the Fully Automatic IK solution at **100**% toward fully manual keyframe positioning (FK) at **0%**. You need to keyframe items in their driven state prior to blending toward FK; otherwise, the item interpolates toward the last keyframed position.

Solve Continuous: Specifies whether the IK solving starts from the chain's rest pose (**Off**) or from the previous IK solution (**On**). The differences are subtle, but the latter can be effective for things like tails or tentacles.

Steps: Indicates the maximum number of times the IK solver evaluates while trying to satisfy all the tasks on a chain and reach a final solution. If the IK solver finds a satisfactory solution before it reaches the maximum number of steps, then it stops early.

Precision: Controls how accurately the solver tries to match the tasks on the chain. Higher values decrease damping but require more calculations and may reduce performance depending on the complexity of the chain. Lower values increase damping and can result in less accurate solving.

Task Strength: Specifies the overall strength of the tasks assigned to this chain and determines how much influence the tasks have on the IK solving.

Limits Strength: Controls the overall strength of the limits on the chain. When joints approach their **Min** or **Max** rotation angles, they become stiffer and less likely to rotate further. Lower values for this setting result in softer damping of the rotation; higher values make the rotation of the joints decrease more abruptly. For very simple chains, this value needs to be raised before the limits come into effect.

Specify Tolerance: Uses the **Tolerance** value while solving (when enabled). When disabled, Modo calculates the tolerance internally based on the size and complexity of the chain and its tasks.

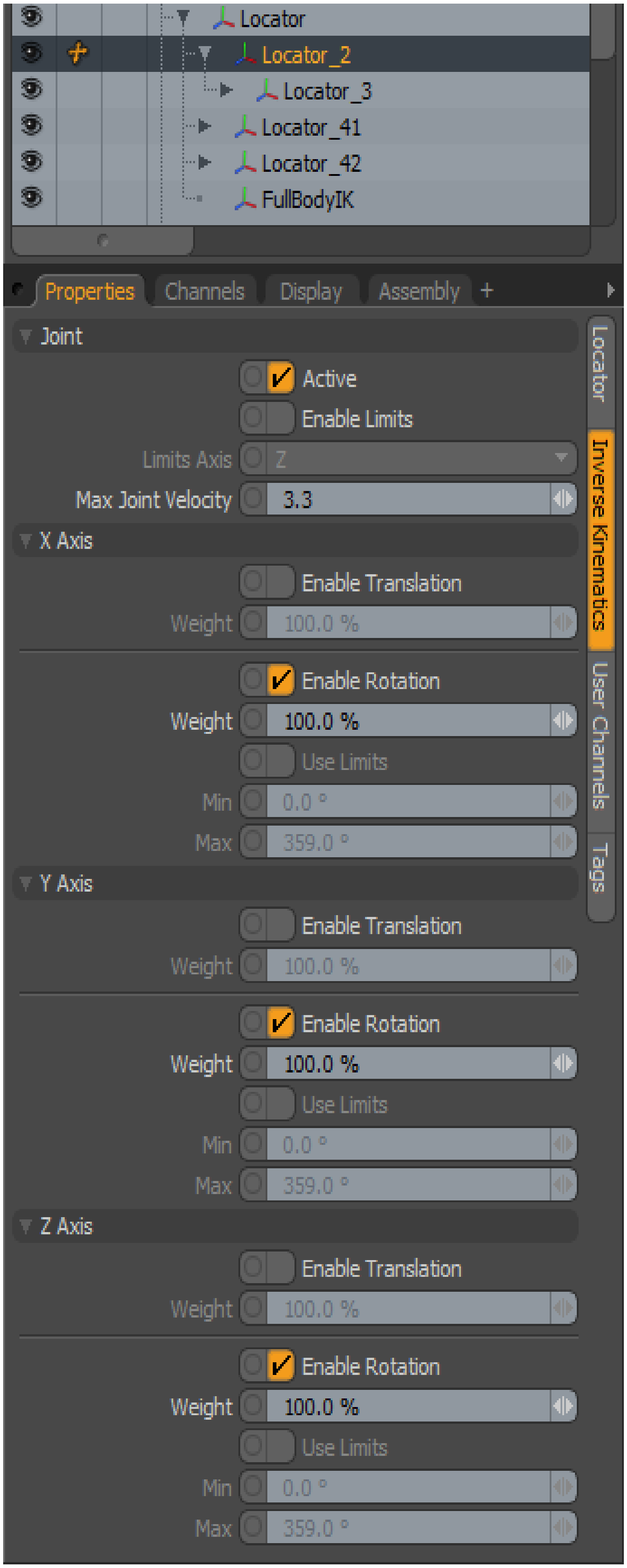
Tolerance: Determines (when enabled) how close a chain needs to match its tasks before the solver considers the solution to be complete. Lower values cause the joints to match their tasks more closely but require more steps to calculate.

Translate Root: Specifies (when enabled) whether the chain root item is allowed to move when solving the IK.

Weight: Controls how freely the chain root moves when you enable **Translate Root**. Zero (**0**) prevents it moving, but **1.0** allows it to move freely. Raise this setting above 1 to increase the likelihood of the root moving even more.

Joint Settings

Joint—


Locator
Locator_2
Locator_3
Locator_41
Locator_42
FullBodyIK
Properties
Channels
Display
Assembly
+
Locator
Inverse Kinematics
User Channels
Tags

Joint

☑ Active

☐ Enable Limits

Limits Axis: Z

Max Joint Velocity: 3.3

X Axis

☐ Enable Translation

Weight: 100.0 %

☑ Enable Rotation

Weight: 100.0 %

☐ Use Limits

Min: 0.0 °

Max: 359.0 °

Y Axis

☐ Enable Translation

Weight: 100.0 %

☑ Enable Rotation

Weight: 100.0 %

☐ Use Limits

Min: 0.0 °

Max: 359.0 °

Z Axis

☐ Enable Translation

Weight: 100.0 %

☑ Enable Rotation

Weight: 100.0 %

☐ Use Limits

Min: 0.0 °

Max: 359.0 °

Active: Specifies (when enabled) whether the joint is affected by the IK solving. Disable this option to have it ignore IK.

Enable Limits: Specifies whether limits should be active on the joint in the IK system. You can have limits on a joint to restrict its rotation on each axis.

Limits Axis: Specifies the axis of the joint for any set limits. Use when you set limits on a joint. Normally, this is the axis that is pointing towards the joint's first child joint, which is usually the axis running along the IK segment.

Max Joint Velocity: Adjusts the amount of rotation a joint can make for each iteration of the IK solver. With larger values, the joint can rotate more. For very simple chains (for example, a two-segment arm or leg), increase this setting because there are fewer joints to spread the rotations across. On more complex chains, use this to reduce jitter that may occur when goals or tasks are moved a long way out of reach.

X/Y/Z Axis—

Each of the three axes provide identical options for you to enable or disable translations and rotations and limit the amount of movement.

Enable Translation: Allows (when enabled) the joint's child to move on the specified axis of the current joint while trying to reach its goal (This is not applicable to the chain root.)

Weight: Controls the amount of joint translation: **0%** disables movement and **100%** allows full movement. You can increase the value above 100% to make movement more likely — especially when several joints have **Translation** enabled.

Enable Rotations: Toggles rotation on the specified axis.

Weight: Adjusts the freedom of rotation around the specified axis. At **0%** the joint does not rotate; at **100%** it can rotate fully. Raising this value above 100% makes this axis more likely to rotate.

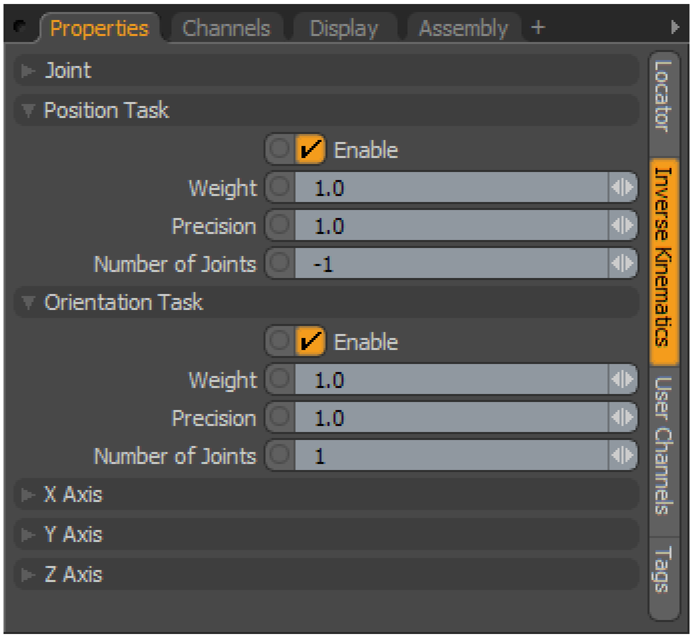
Use Limits: Forces (when enabled) the IK to only rotate within the specified values and to produce a hard stop when it reaches the **Min** or **Max** value.

Min/**Max**: Determine the range that a joint is allowed to rotate within. Modo only uses these limits when you enable **Use Limits**.

Task Options

When you add Goals to a Joint, Modo adds **Task** options to the target Joint for you to adjust how the joint changes to reach toward the goal. Tasks are related to goals because they are the instructions to the joint to retain a position or rotation (or both). For a **Position Task**, the IK solver attempts to keep the chain item at the World location the item had when you added the task. For a **Rotation Task**, the IK solver attempts to keep the chain item at the same World orientation that it had when you added the task.

Position Task—



Position Task

☑ Enable

Weight 1.0

Precision 1.0

Number of Joints -1

Orientation Task

☑ Enable

Weight 1.0

Precision 1.0

Number of Joints 1

Enable: Controls whether the task is currently active or not.

Weight: Specifies the strength of the task to determine how much effect it has on the chain. When there are multiple tasks on a chain, you can use this to make specific tasks have more or less influence on the solution.

Precision: Controls how closely the solver tries to match the joint to its task.

Number of Joints: Specifies the number of joints towards the root that this task affects. Set to **-1** to affect all joints.

Orientation Task—

Enable: Controls whether the task is currently active or not.

Weight: Specifies the strength of the task to determine how much effect it has on the chain. When there are multiple tasks on a chain, you can use this to make specific tasks have more or less influence on the solution.

Precision: Controls how closely the solver tries to match the joint to its task.

Number of Joints: Specifies the number of joints towards the root that this task affects. Set to **-1** to affect all joints.

Working with Weight Maps

Learn how Weight Maps define their area of influence upon a mesh.

Weight Maps and Weight Containers

These Weight Maps are incredibly useful and tremendously powerful, but the drawback of a traditional embedded Weight Map is that it's part of the mesh and when the mesh isn't available, then the Weight Map isn't either.

A weight container is the same thing as a traditional Weight Map in that it stores weight values for vertices, but it is separate from the mesh and is an item of its own. This is useful when rigging when you may want to abstract the weight values from the geometry. In this way, you can create a rig and attach it to the geometry. Then you can apply weight to the model and store the values in the container. The model is instantly affected by the rig. Additionally, because the weight container is its own item for storing values, it can store weights for more than a single item, which further streamlines the rigging process.

Creating a Weight Container

You can easily add weight containers to a scene by using the **Add Items > Deformers > Weight Container** command. Before you can create any weights, you need to link them to the target mesh(es). There are several commands that make these connections in the **Setup** and **Animation** interface tabs in the toolbox in the **Deformers** sub-tab.

Add Weight Container

Adds a weight container to the **Items** list. If you select any items before choosing this command, Modo automatically adds them to the container. Additionally, if you have selected vertices, Modo automatically assigns a weight value of **1** to them.

Add Item

Adds any selected items to any selected weight container. Before clicking this button, select any number of items and a weight container. Additionally, if you select any vertices, Modo automatically assigns a weight value of **1** to them.

Remove Item

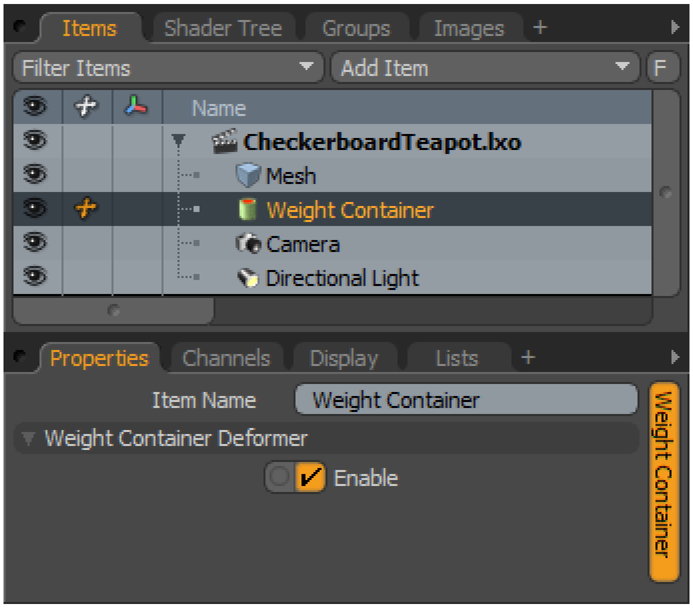
Removes any selected items from any selected weight container. Before clicking this button, select any number of items and a weight container.

Add Points

Adds any selected vertices to any selected weight container and automatically assigns them a value of **1**.

Remove Points

Removes any selected vertices from any selected weight container. Before clicking this button, select any number of vertices and a weight container.


Items
Shader Tree
Groups
Images
+
Filter Items
Add Item
F
Name
CheckerboardTeapot.lxo
Mesh
Weight Container
Camera
Directional Light
Properties
Channels
Display
Lists
+
Weight Container Deformer
Weight Container

Item Name: Weight Container

☑ Enable

Item Name: Displays the current item's name. Change the name by clicking the current name and typing the new name.

Weight Container Deformer—

Enable: Determines the effect that the container item has. When disabled, the layer has no influence on the scene. However, Modo saves disabled layers with the scene and its values are persistent across Modo sessions.

Weighting Tools

With Modo's weighting tools you can quickly and interactively adjust vertex Weight Maps used for animation — including those created automatically using Modo's **Bind** command. You can adjust the Weight Map values by using Modo's familiar brush-based workflow and, combined with a number of useful commands, you have a powerful way to optimize and correct deformations related to weights. To see the results of the tools, you need to select the target weight (done automatically when you select a joint) and make sure that the 3D viewport is set to either the **Vertex Map** viewport style or has **Show Weight Maps** enabled to display the Weight Map as an overlay to the viewport.

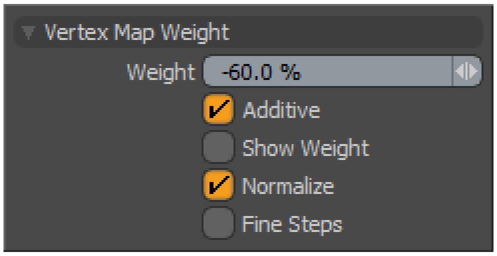
Using the Tools

The weighting tools are in the **Setup** and **Animation** interface tabs in the **Weighting** sub-tab. To use the Vertex Map tools, you need to select the target Mesh Item layer first in the Items (Scenes) List tab. You also need to create or select the appropriate Weight Map in the **Vertex Map** list viewport. See the Working with Vertex Maps topic for information on creating Vertex Maps.

> **Tip:** The performance speed for editing the **Weighting Tools** only updates the cached elements that have changed when editing the **Adjust Weights** option. When working on a selection of a sub-set of points, such as an arm or finger on a larger character, the performance gains are significant. But if the entire mesh is changing, such as updating the root joint which controls all of the mesh points, the interaction may be slower.

Adjust Weights

With the Adjust Weights tool, you can adjust the weight values interactively in the viewport. Drag a handle to the left or right to increase or decrease the overall value, You can specify whether to add to the existing values, display the numeric values of each vertex, or normalize the target map. Any adjustments that you make are applied to the entire map, or you can isolate the changes to specific sections with a component selection.



Vertex Map Weight

Weight -60.0 %

☑ Additive

☐ Show Weight

☑ Normalize

☐ Fine Steps

Weight: Determines the weight value to apply to the target Vertex Map.

Additive: When enabled, adds the weight values to existing map values instead of replacing them.

Show Weight: When enabled, shows the current weight value for the vertex in the 3D viewport at each vertex position for the target item.

Normalize: When enabled, takes into account the weight values of other Weight Maps present for the target vertices and adjusts the values as needed so that the combined weight values never exceed **1.0** (100%).

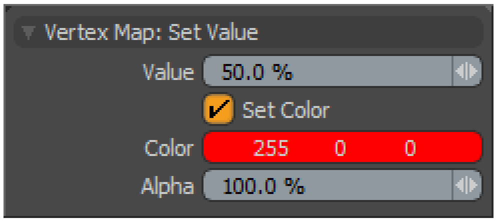
Fine Steps: When enabled, applies weight values with more precision to provide precise decimal values to the target.

Isolate Weights

Clicking the **Isolate Weights** button eliminates all weight values outside of a selected area and, therefore, isolates the selected area. This is useful to eliminate crosstalk between neighboring surfaces with distance-based Weight Maps. Select the area that is to remain and then click **Isolate Weights** to remove all other weight values from the map.

Paint

The Paint Weights tool paints Weight Map values in a similar manner to painting texture maps. After selecting the target Weight Map and specifying a **Value** to paint, right-click in the 3D viewport to specify a brush size. Drag over the surface to apply the weight value. Press **Ctrl** to remove the weight amount when painting. You can also paint vertex color maps with this tool. You only need to select the appropriate map type before painting with this tool.



Vertex Map: Set Value

Value 50.0 %

☑ Set Color

Color 255 0 0

Alpha 100.0 %

Paint Value: Determines the strength of the painting action. Higher values saturate the canvas more quickly than lower values. Set to **100%** to quickly overlay an area or to **5%** or less to make subtle adjustments.

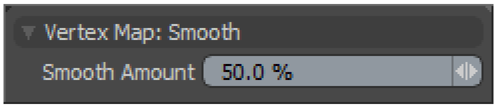
Set Color: Use to be able to define a color value and apply it to the target map. When applying values to a vertex color map, this is enabled automatically.

Color: Defines a color with RGB values to apply to the target vertex color map.

Alpha: Determines the opacity of the color applied to the vertex color map. Use **100%** for an opaque color; use less than **5%** for a transparent color.

Smooth

The Smooth tool blurs (or softens) Weight Map values to smooth out their values across the surface they are applied to. After selecting the target Weight Map, define a **Smooth Amount** to apply. Right-click in the 3D viewport to set a brush size and then drag over the surface to smooth the weight values. Press **Ctrl** as you drag to sharpen (increase the contrast in) neighboring weight values.



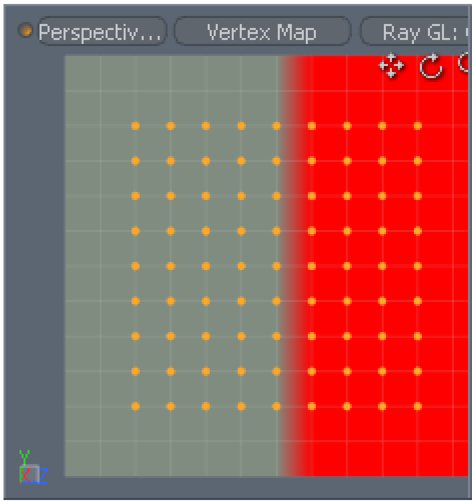
Vertex Map: Smooth

Smooth Amount 50.0 %

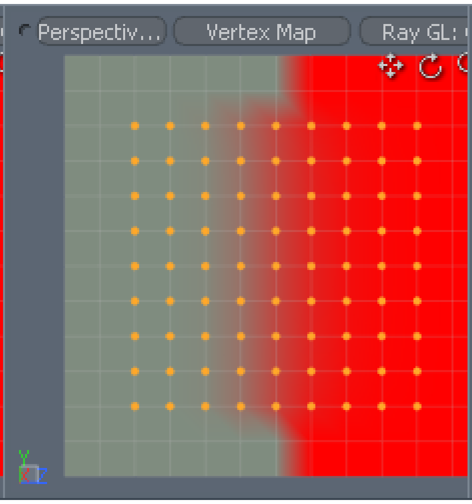
Smooth Amount: Controls the strength of the smoothing applied to the Vertex Map. Higher values (> 50%) produce increasingly softer results; lower values (< 50%) produce less noticeable softening.

Smooth Selected

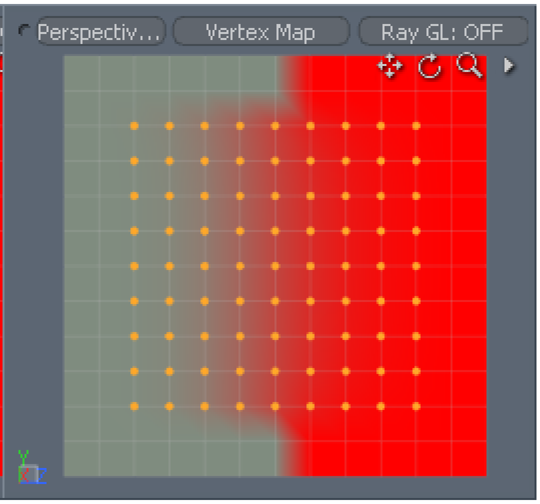
Softens the Vertex Map values of any selected geometry component automatically. If you apply this command multiple times, Modo continues to smooth the values. This is similar to a blur filter applied to an image and also works well when applied with any of the availableUsing Falloffs.



Initial Vertex Selection



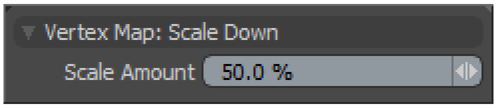
Smooth Selected Applied Once



Smooth Selected Applied a Second Time

Scale Down

The Scale Down tool interactively adjusts the values in a map toward **0** (zero). Pressing the **Ctrl** key scales up the values. The overall effect is similar to dodging and burning in an image editor.



Vertex Map: Scale Down

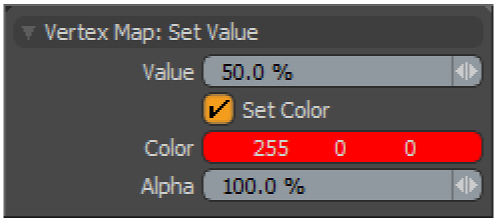
Scale Amount 50.0 %

Scale Amount: Determines the strength of the scaling applied to the values.
Erase—

Removes any values in the target map that the brush touches. This doesn't just lower the value to 0%, but it completely removes the value for each affected vertice. There are no user-adjustable values for this tool.

Set Value

The Set Value tool is a combination paint tool and erase tool. It affects any vertex it touches 100% with no fading or attenuation. Pressing **Ctrl** key as you use it erases values.


Vertex Map: Set Value
Value 50.0 %
☑ Set Color
Color 255 0 0
Alpha 100.0 %

Value: Determines the weight applied when editing Weight Map values.

Set Color: Use to be able to define a color to apply to the map when painting onto vertex color maps. This option is enabled automatically.

Color: Defines a color in the target vertex color map with RGB values.

Alpha: Defines a transparency value used in the target RGBA vertex color map. Use **100%** for an opaque color; use less than **5%** for a transparent color.

Working with Morph Containers

A Morph Container is an item that allows you to store morph information separate from a mesh. It works the same way as a Weight Container, but instead of storing Weight Map information, it stores Morph Map information. Morph containers can also handle more than one morph map at a time.

Normally, when working with a rig using a Morph Influence, you have to add the required meshes to your scene and set them up with the same Morph Map, then set the Morph Influence to use that Morph Map. Morph Containers allow for morph relationships to be created without any meshes in the scene, making rigging more modular and portable.

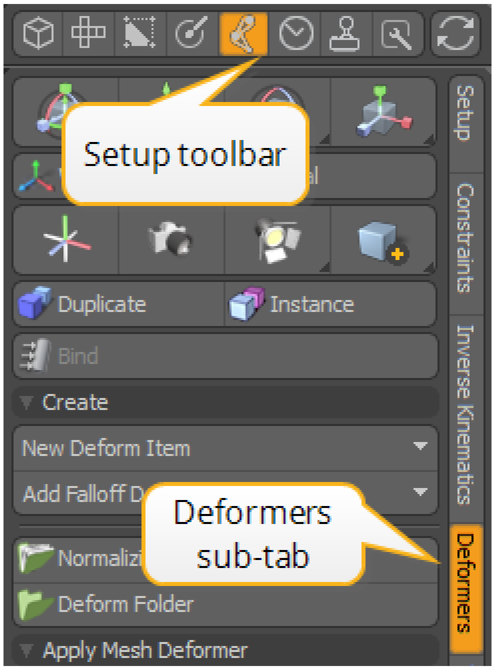
Note: For more information on Weight Containers, see Working with Weight Maps.

Note: For more information on Morph Maps, see Morph Maps.

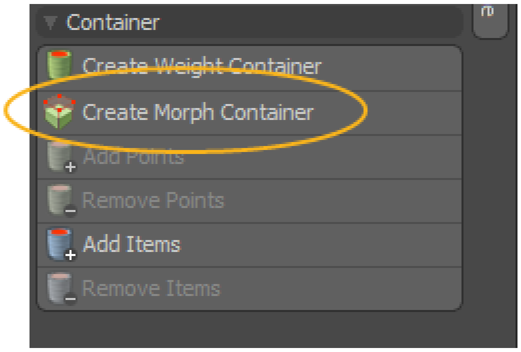
Creating a Morph Container

To create a Morph Container:

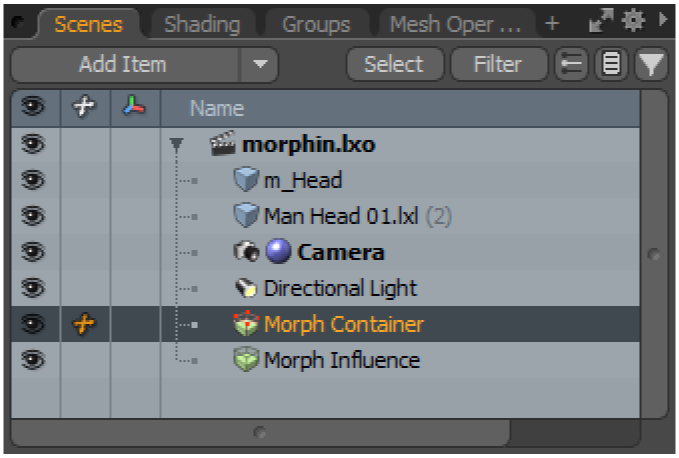
1. Click the **Setup** toolbar button, then click the **Deformers** sub-tab.


Setup toolbar
Setup
Constraints
Inverse Kinematics
Deformers
Duplicate
Instance
Bind
Create
New Deform Item
Add Falloff D
Deformers sub-tab
Normalizi
Deform Folder
Apply Mesh Deformer

2. Click the **Create Morph Container** button.


Container
Create Weight Container
Create Morph Container
Add Points
Remove Points
Add Items
Remove Items

A Morph Container item is added to the Item List on the right.


Scenes
Shading
Groups
Mesh Oper ...
Add Item
Select
Filter
Name
morphin.lxo
m_Head
Man Head 01.lxl (2)
Camera
Directional Light
Morph Container
Morph Influence

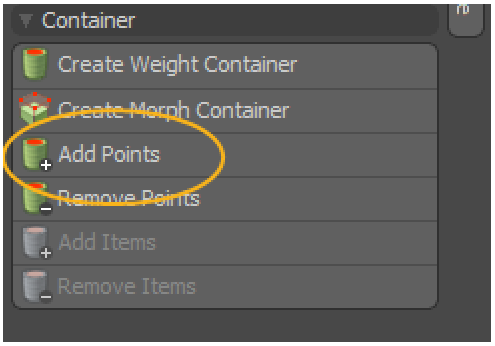
You can also add a Morph Container in the Item List: click **Add Item**, then in the **Add Item** dialog, under **Items** > **Deformers**, double-click Morph Container.

Adding Points and Items

You can add points and items to your Morph Container in the **Setup** toolbar.

To add points to a Morph Container:

1. In **Vertices** selection mode, select the required vertices on your mesh.
2. With the vertices selected, click the Morph Container item in the Item List.
3. In the **Setup** toolbar, click **Add Points**.

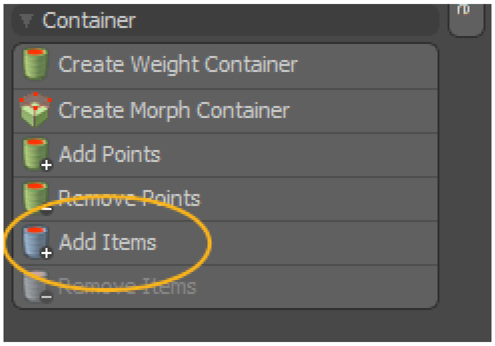

Container
Create Weight Container
Create Morph Container
Add Points
Remove Points
Add Items
Remove Items

The selected points are added to the Morph Container.

To remove points from a Morph Container, select the vertices in the 3D viewport, then in the **Setup** toolbar, click **Remove Points**.

To add items:

1. Select the new item and the Morph Container in the Item List.
2. In the **Setup** toolbar, click **Add Items**.


Container
Create Weight Container
Create Morph Container
Add Points
Remove Points
Add Items
Remove Items

The selected item is added to the Morph Container.

To remove items from a Morph Container, select the item, then in the **Setup** toolbar, click **Remove Items**.

Manipulating the Morph Container

When you're done with setting up the Morph Container, you can edit the vertices added to it. To manipulate the morph container:

1. Select the Morph Container in the Item List.
2. In **Vertices** selection mode, select the vertices in the 3D viewport and transform them as needed, for instance, move them.

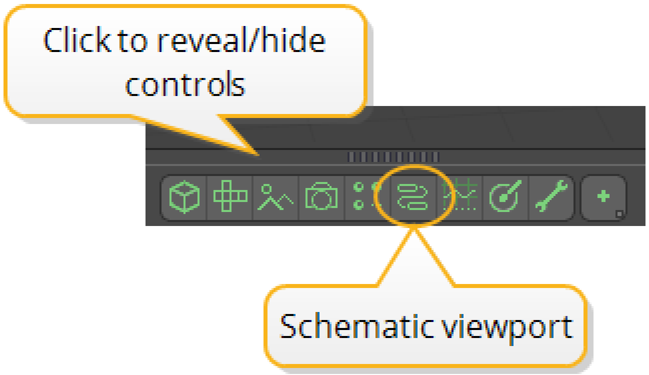
   The transform is applied to the vertices within the Morph Container, but when you de-select it and select the mesh instead, the vertices return to their original positions.

Using the Morph Container with a Morph Influence

To use a Morph Influence to drive the Morph Maps in the Morph Container, add a Morph Influence and connect it to your Morph Container. The easiest way to do this is in the Schematic viewport.

To open the Schematic viewport:

- In the **Modo** layout, click the thin gray line below the 3D viewport, and click the Schematic viewport 𝄞 button.



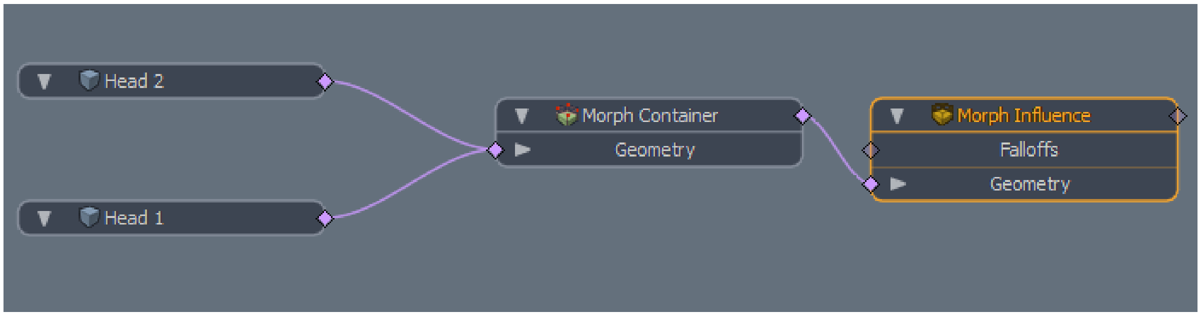
>

MadCap:conditions="Default.NoTranslate">OR

- Switch to the **Setup** layout from the menu bar by clicking **Layout** > **Layouts** > **Setup**.

1. In the Schematic, click **Add...**, and under **Deformers** > **Influences**, double-click **Morph Influence**.

   A Morph Influence is added to the scene.
2. Connect the **Morph Container** node to the **Morph Influence** node's Geometry input.


Head 2
Head 1
Morph Container
Geometry
Morph Influence
Falloffs
Geometry

3. Select the Morph Influence node to open its properties in the lower-right panel.

   Edit the **Strength** value to adjust the effect of the Morph Influence.

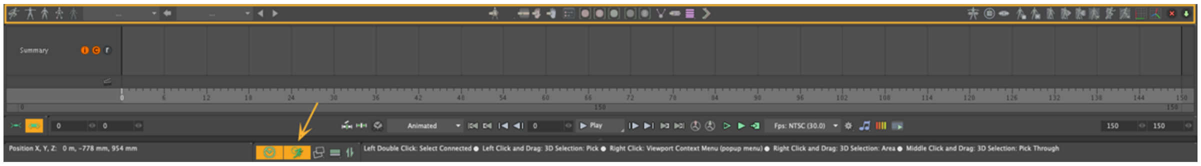
   For more information on working with the Morph Influence, see Morph Influence.

Utilizing the Auto Character System (ACS)

The **Auto Character System (ACS)** is now bundled with Modo, allowing rigging and animation features to work more seamlessly.

ACS Tools in Modo

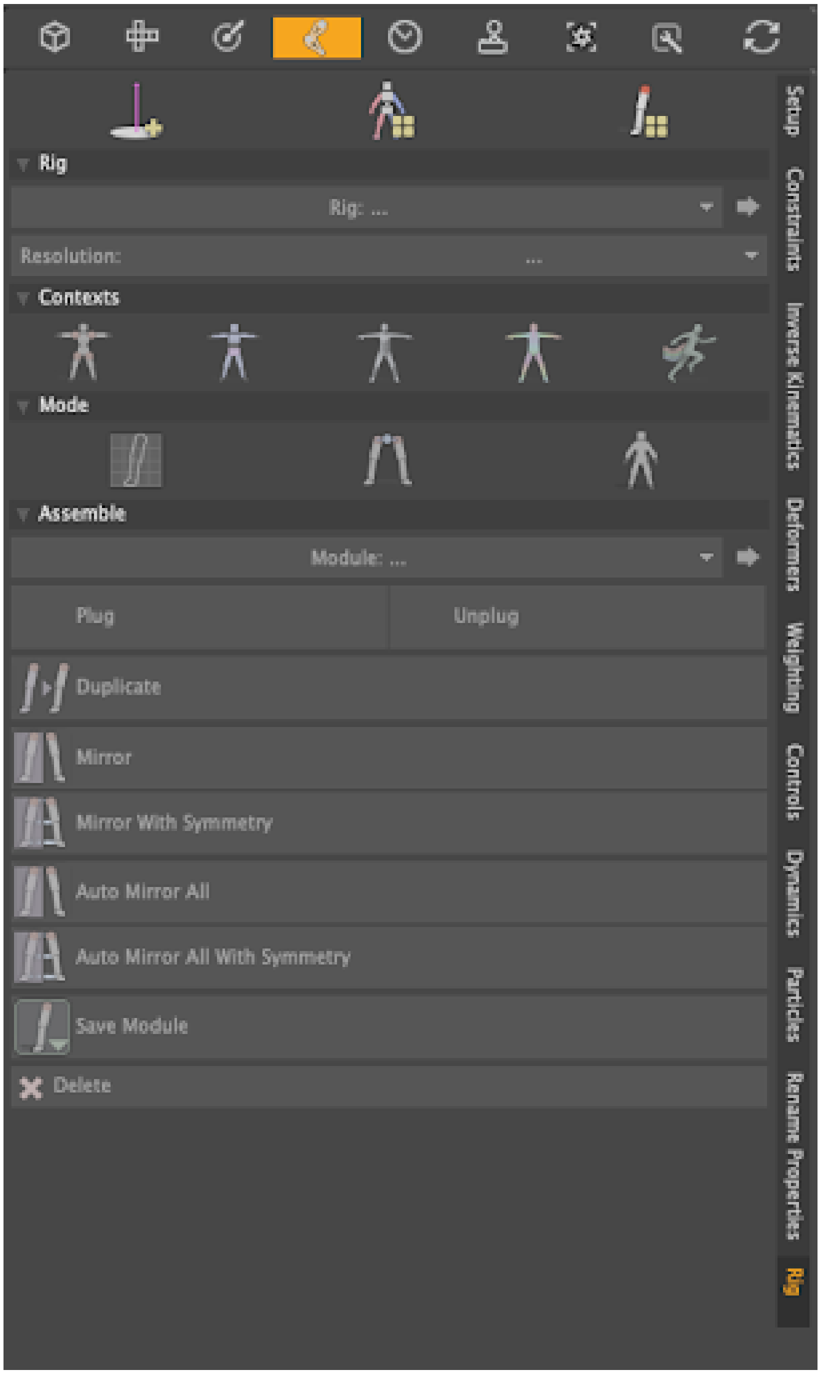
- An ACS **toolbar** is available in the Timeline in the Modo layout. See Working with the Timeline.


Summary
Animated
Play
Fps: NTSC (30.0)
150
150

Position X, Y, Z: 0 m, -778 mm, 954 mm

Left Double Click: Select Connected ● Left Click and Drag: 3D Selection: Pick ● Right Click: Viewport Context Menu (popup menu) ● Right Click and Drag: 3D Selection: Area ● Middle Click and Drag: 3D Selection: Pick Through

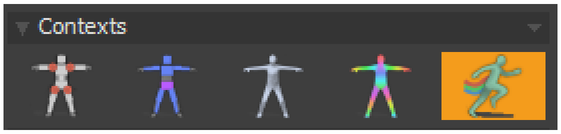
- The **Rig vertical tab** contains ACS tools.
  You can find this by going to the **Setup** tab, and selecting **Rig**.


Rig
Rig: ...
Resolution:
...
Contexts
Mode
Assemble
Module: ...
Plug
Unplug
Duplicate
Mirror
Mirror With Symmetry
Auto Mirror All
Auto Mirror All With Symmetry
Save Module
Delete
Setup
Constraints
Inverse Kinematics
Deformers
Weighting
Controls
Dynamics
Particles
Rename Properties
Rig

ACS Basics

Once you have a rig in the scene and have opened the **Rig** tab, ACS options become enabled.

The **Contexts** section of the tab allows you to set what you would like to edit. Each of these also has various sub-contexts to choose from.



Contexts		
Edit		Use this context to assemble a rig from modules, edit the existing rig, or develop custom modules. This is also where you can save your rig.
Fit		Use this context to fit the guide to character mesh(es). When done, be sure to use the Apply button to apply fit guide to the rig.
Mesh		Use this context to manage meshes that are meant to be part of the rig as well as set up and edit mesh resolutions.
Form		Use this context to bind meshes to the rig and edit weight maps that define mesh resolutions.
Animate		Use this context for animating rigged characters.

Learn more about ACS

The Auto Character System (ACS) Website and Manual Page includes extensive videos showing how to use ACS.

To find these links through Modo:

1. Go to **Help** on the top bar.
2. Select **Kit Help** > **Auto Character System Help**.

Tip: The **Rig menu** also includes links to ACS Samples and other features. You can find this by going to: **Rig** > **Folders** > **Open Samples Folder**.

Shading and Lighting

Shading and Lighting in Modo are necessary functions. Lighting plays an important role in illuminating the scene and setting the mood. Shading defines the look of the modeled surfaces as either shiny or dull, reflective or transparent and rendering calculates the combination of the two to produce the final image.

The following topics provide information on applying materials, using the Shader Tree and working with layers. Learn about setting up the environment, lighting, and volumes, as well as baking workflows.

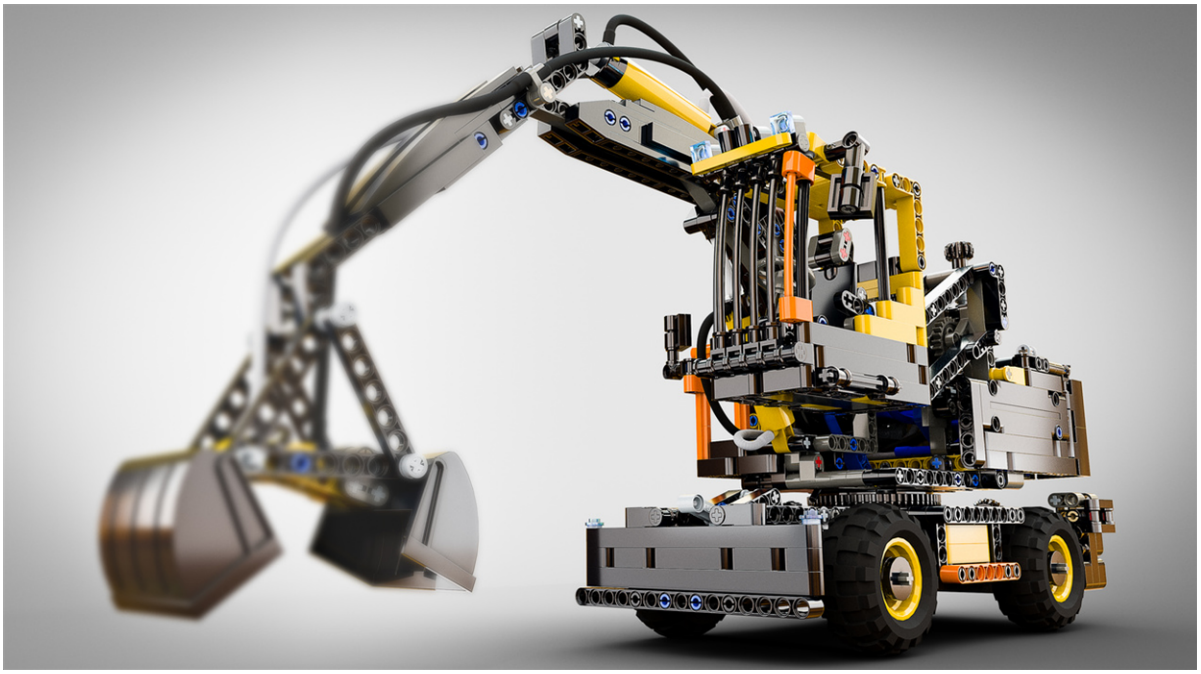


Image courtesy of Madeira3d.

Defining Shaders

Surfacing is the process of applying shading attributes to the 3D geometry in the scene. The various shading attributes make, for example, a surface red and shiny or dull and pitted, or any of the millions of variations a surface can posses. Surfacing in Modo uses a **layer** and **mask** based metaphor to assign these attributes to a surface.

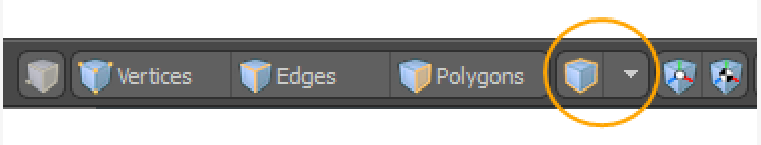
Masks determine the area where the surfacing properties are applied. Within these masks the attributes are defined. This is similar conceptually to how silk-screening works for the garment industry. Take for instance the graphic on a t-shirt, each color on the shirt requires a new screen (mask) and when the next color (layer) is applied it supersedes the previous color where they overlap. Of course transparent ink could be used to allow the other colors to show through - this is the same in Modo too, changing a layer's transparency allows the lower layers to show through.

In order to create the mask that confines the surfacing to just the target area, a Polygon **Tag** needs to be defined. Polygon tags are simply user-defined labels associating any number of polygons together as a group. The tag definitions are stored in the geometry itself and can be easily recalled by Modo for a variety of purposes.

Modo does have several tag types but the main tag used for surfacing is the **Material** tag (**Parts** and **Selection Set** tags are the other types and offer additional functionality for layering materials for different effects, but that's getting ahead of ourselves). For **Material** tags, simply defining the tag prompts Modo to create the mask for the surfacing. When tagging geometry for surfacing in this way, the geometry itself determines the shape of the mask - when the model moves, so does the mask (with all surfacing going along for the ride as well).

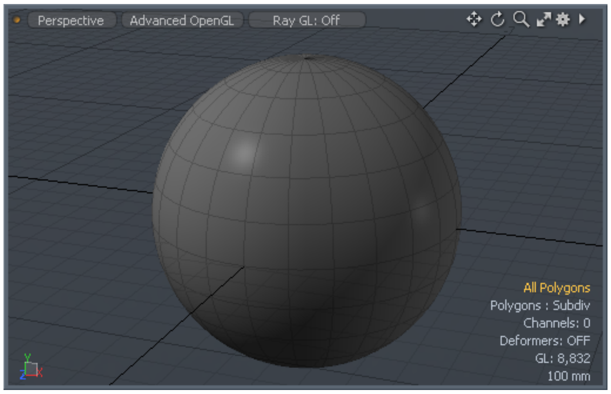
So the important summary here is that Modo surfacing uses a simple mask and layered metaphor for surfacing and tags define the masks that limit that surfacing to the intended areas of the geometry. There are several steps to apply a tag to a surface.

Tip: To quickly select **Materials**, **Parts**, **Selection Groups**, or **Selection Sets** click the **Select Polygon Tag** icon below the menu bar. Make your selection from the dropdown menu.

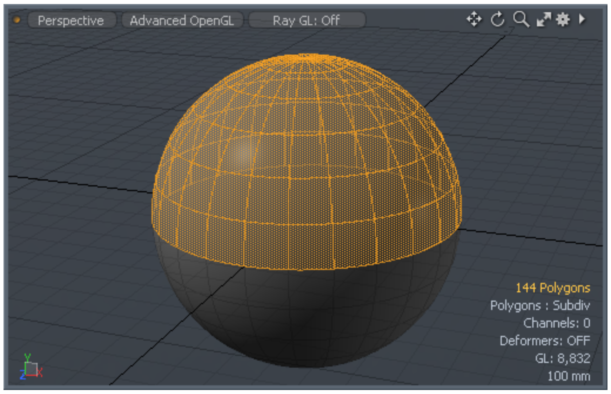


Assigning Material Tags

1. To assign a tag, you first need some geometry in the scene. For example, create a default 1m sphere.


Perspective
Advanced OpenGL
Ray GL: Off
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: OFF
GL: 8,832
100 mm

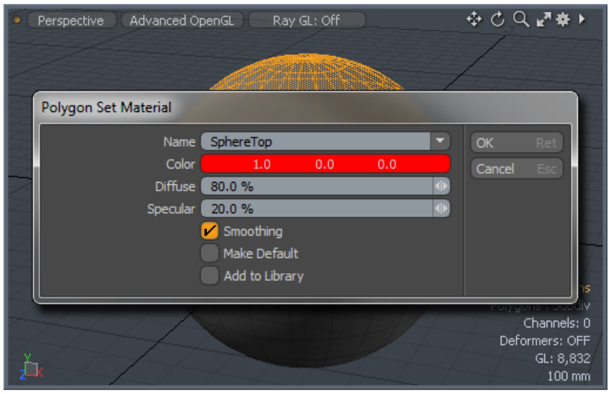
2. In **Polygons** selection mode, select the target polygons you wish to apply the material to. Here, the upper half of the sphere has been selected. For information on making selections, see Selecting Items.


Perspective
Advanced OpenGL
Ray GL: Off
144 Polygons
Polygons : Subdiv
Channels: 0
Deformers: OFF
GL: 8,832
100 mm

3. With the selection active, choose the command **Texture** > **Assign Material Group** or press the **M** keyboard shortcut to open the **Polygon Set Material** dialog box. Here you define the **Name** of the tag as well as set some basic surfacing attributes.

   In the example, a red color has been used and the surface is named **Sphere Top**.

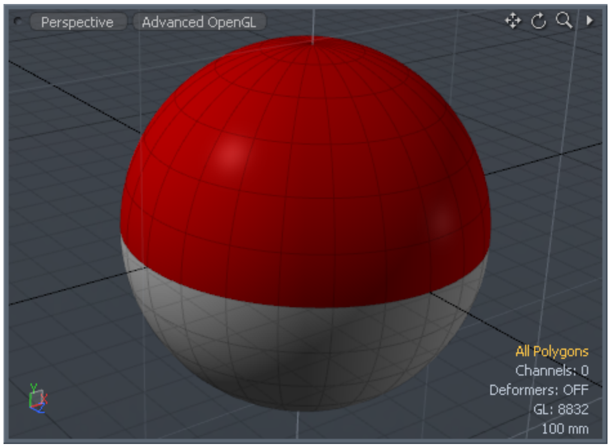
Note: You must change the name from **Default**, or the whole mesh will receive the material.


Perspective
Advanced OpenGL
Ray GL: Off
OK Ret
Cancel Esc
Channels: 0
Deformers: OFF
GL: 8,832
100 mm

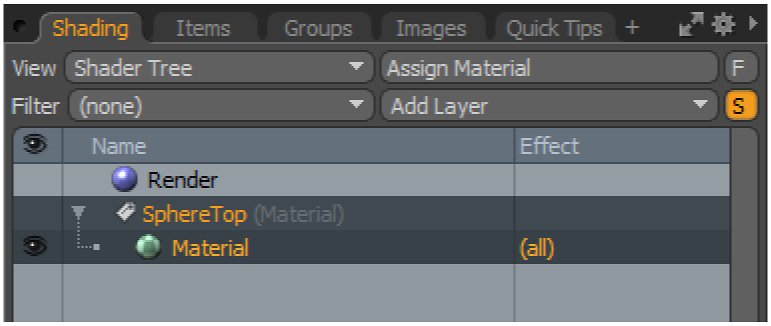
Polygon Set Material

Name: SphereTop
Color: 1.0 0.0 0.0
Diffuse: 80.0 %
Specular: 20.0 %
☑ Smoothing
☐ Make Default
☐ Add to Library

4. Click **OK** or press **Enter**. This assigns the tag to the selected polygons and defines a material in the Shader Tree.

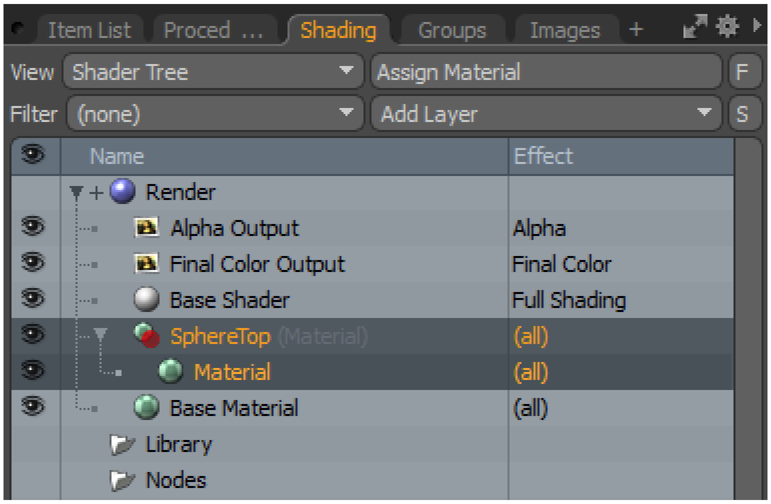

Perspective
Advanced OpenGL
All Polygons
Channels: 0
Deformers: OFF
GL: 8832
100 mm

Click the **Shading** tab to see the Shader Tree and select **Material** view mode. You can see that a tag 🏷 is added to the **Materials** panel showing the name that was defined, and a new material item ● has been added to it. If you can't see the material item, click the arrow widget ▸ next to the named tag to make it visible. Now, any changes made to this new material item are limited to only polygons in the scene with the tag **Sphere Top**.


Shading
Items
Groups
Images
Quick Tips
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Name
Effect
Render
SphereTop (Material)
Material
(all)

Selecting the **Material** item layer in the **Materials** panel displays its associated attributes in the **Properties** panel for further editing.

In **Shader Tree** view mode, you can see the material with the other layers. The material is held in a Material Group with the name that you assigned to the material.


Item List
Proced ...
Shading
Groups
Images
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
SphereTop (Material)
(all)
Material
(all)
Base Material
(all)
Library
Nodes

With the material group highlighted, select the **Texture Layers** tab in the properties and you can see that the **Polygon Tag** has been set to the name of the material. This ensures that only the polygons that are tagged with that name will receive the material.

The properties of the **Material** layer can be changed to adjust the color, reflection, and so on.

You can continue recursively adding additional material tags to define all the various surfaces in a scene in the same way, just remember that a polygon can only have a single **Material** tag at a time, so if you accidentally assign a second material, it overwrites the previous value. To fix this, select the offending polygon(s) and reassign the previous tag. If you wish to apply complex layering to materials, this requires additional alternative tag types (such as applying a label onto a bottle of wine). You can define these groupings through the addition of Part tags and Selection Sets.

Tip: In complex scenes, you should consider a strategy for assigning material tags to the various surfaces. Descriptive names are often better than cryptic ones. Consider the case of a car model. One naming scheme might title materials based on their type, such as **Glass**, **Rubber**, **Leather**, and **Chrome**. Another scheme might use the item itself, such as **Windshield**, **Tire**, **Seat**, and **Bumper**. Both examples are much better than **Upper Front**, **Inside**, **Lower Left**, and so on, because when you hand off a file or return to it at a later date, the descriptive names still make sense.

Procedural Material Tags

Material Tags have a procedural version that can be used in the Mesh Operations List. The Material Tag is located under **Procedural** > **Polygon**. The Material Tag has a **Selection** input to identify the polygons that are to receive the material.

Note: For information on the selection types, see Procedural Selection.

You need to set the **Material Name** in the Material Tag properties to associate the tag with a material in the Shader Tree. Unlike material tagging in direct modeling (as described above), a material is not automatically added to the Shader Tree. This means that applying materials with procedural tags is a two-stage process:

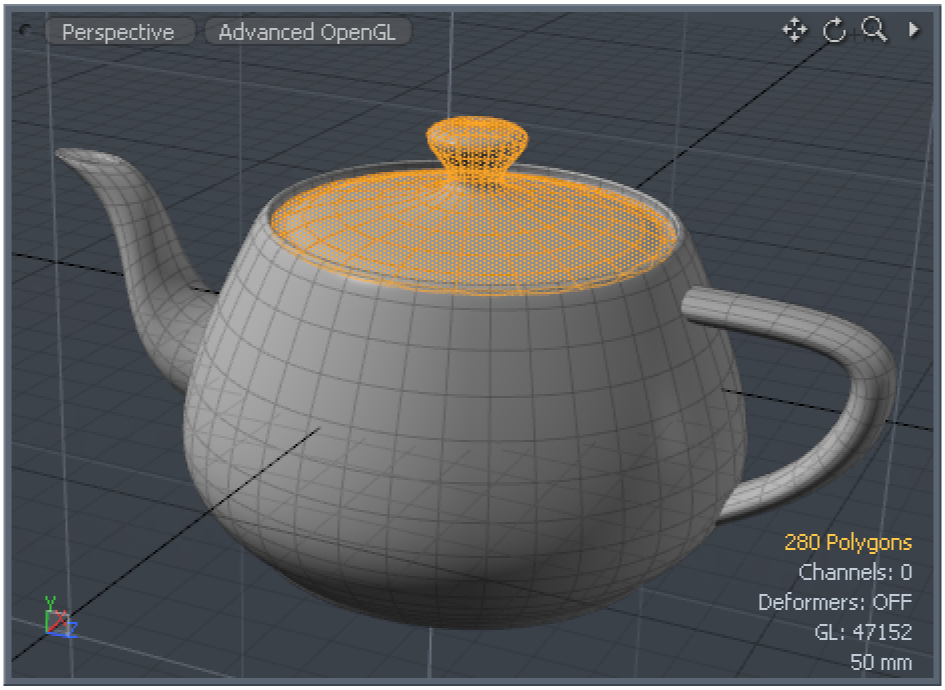
1. Creating a material tag containing a polygon selection, and assign a **Material Name**.
2. Add a material group with a material to the Shader Tree, and set the group's **Polygon Tag** to match the procedural tag's **Material Name**.

For a step-by-step example of how to use a procedural Material Tag, see Procedural Modeling Example.

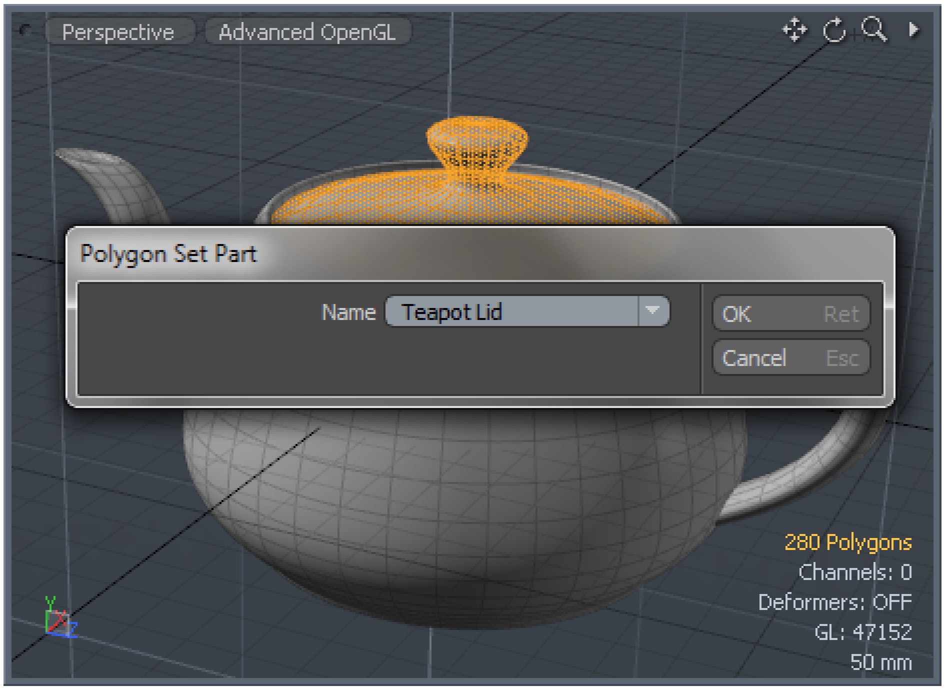
Part Tags

Parts work in a similar fashion to **Material** tags, offering an additional means of defining polygon groupings. Parts can be a useful way to define additional surfacing layers in the Shader Tree, especially in cases where the surface layer spans across two or more **Material** tag boundaries. They can also be useful when exporting geometry to other applications. Parts can be easily defined in the following manner.

In **Polygons** mode, select the geometry you wish to assign the part tag to, in this case the lid of the teapot.

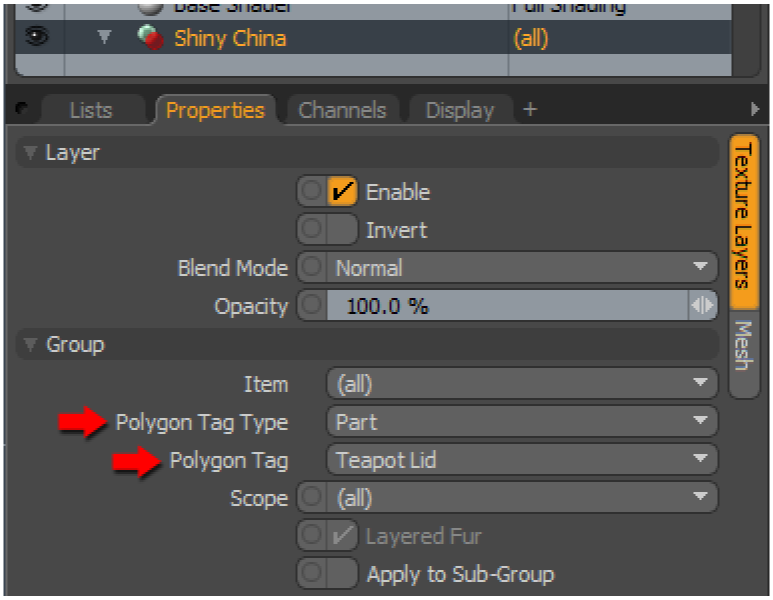

Perspective
Advanced OpenGL
280 Polygons
Channels: 0
Deformers: OFF
GL: 47152
50 mm

Choose the command **Geometry** > **Polygon** > **Set Part** to open the **Polygon Set Part** dialog box. You can assign a new name or use the dropdown context menu to choose an existing part tag from the list. Click **OK** or press the **Return** (or **Enter**) key on the keyboard to assign the **Part** tag to the selection.


Perspective
Advanced OpenGL
Polygon Set Part
OK Ret
Cancel Esc
280 Polygons
Channels: 0
Deformers: OFF
GL: 47152
50 mm

Name: Teapot Lid

Once applied, **Part** tags can be recalled in two ways. Firstly, by creating a Shader Tree **Material Group** item within its properties panel, you can define the **Polygon Tag Type** as a **Part**, and in the **Polygon Tag** dropdown menu, select the a part you defined in the previous step. When defined in this way, the effect of any Shader Tree items added to this group is confined to the polygons grouped by the tag, masking the layers in the same manner as a **Material** tag.



Shiny China (all)

Layer

☑ Enable

☐ Invert

Blend Mode Normal

Opacity 100.0 %

Group

Item (all)

Polygon Tag Type Part

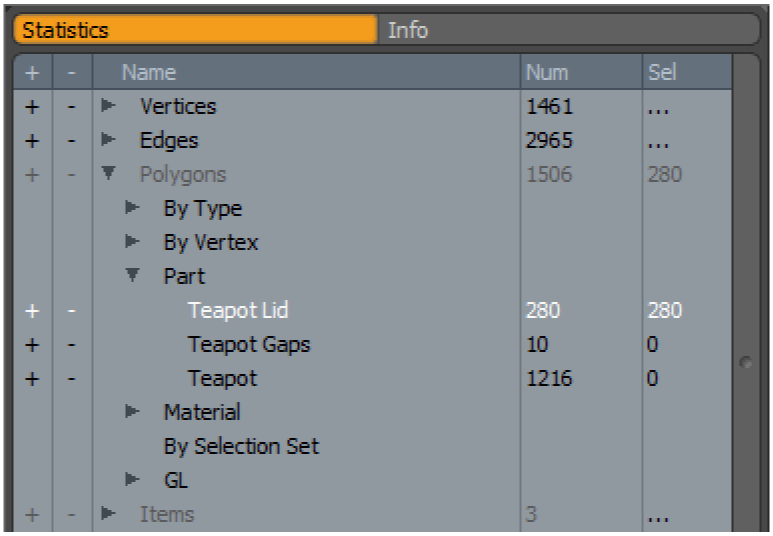
Polygon Tag Teapot Lid

Scope (all)

☑ Layered Fur

☐ Apply to Sub-Group

The second way to recall a **Part** is to use the Statistics viewport panel under the **Lists** tab. Within the viewport, under the **Polygon** > **Part** section, you can recall part selections by using the **+/-** buttons in the left column of the panel. In the example, clicking on the **+** symbol has activated the **Teapot Lid** selection. Clicking the **-** button deselects any polygons contained in that tag.



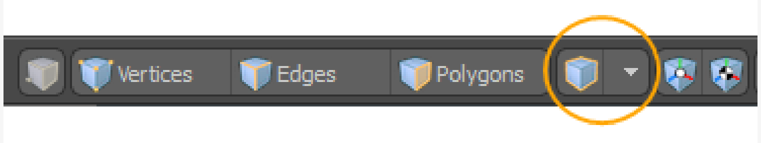
+	-	Name	Num	Sel
+	-	Vertices	1461	...
+	-	Edges	2965	...
+	-	Polygons	1506	280
		By Type		
		By Vertex		
		Part		
+	-	Teapot Lid	280	280
+	-	Teapot Gaps	10	0
+	-	Teapot	1216	0
		Material		
		By Selection Set		
		GL		
+	-	Items	3	...

Tip: You can add tags to curve elements in the same way that tags are added to polygons. Select the Curve itself while in **Polygons** mode and apply the tags as demonstrated above. This can be useful when defining alternate or additional guides when styling a Fur Material layer.

Polygon Set Material/Part/Smoothing

The **Polygon Set Material** function assigns a material tag to the current polygon selection (or, if nothing is selected, all polygons in the active item layers). A Material Group item is created in the **Shader Tree** with the same name and proper tag assignment, and a single child Material is inserted. Once defined, any modifications made to properties of the child material are limited to only those polygons that hold that specific tag. Please see Defining Shaders for more information on specifying and applying tags and the Shader Tree page for adding and modifying additional layers to the **Shader Tree**.

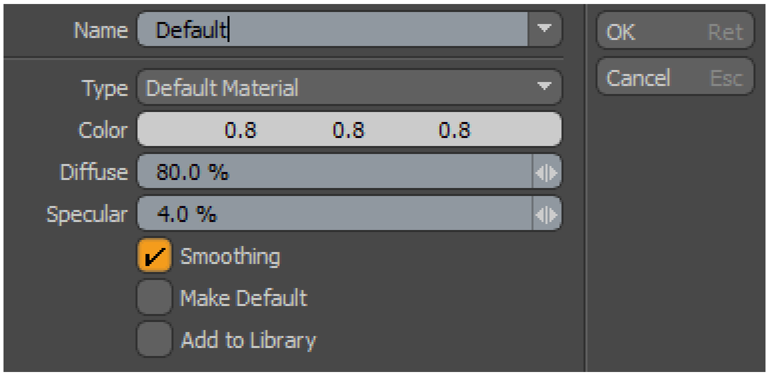
Tip: To quickly select **Materials**, **Parts**, **Selection Groups**, or **Selection Sets** click the **Select Polygon Tag** icon below the menu bar. Make your selection from the drop-down menu.



You can apply a material tag in several ways.

- Click **Geometry** > **Polygon** > **Assign Material Group**.
- Click the **Assign Material** button in the **Shading** viewport.
- Click the **M** keyboard shortcut to assign material tags to a polygon.

When invoked, a dialog opens.


Name Default

Type Default Material

Color 0.8 0.8 0.8

Diffuse 80.0 %

Specular 4.0 %

☑ Smoothing

☐ Make Default

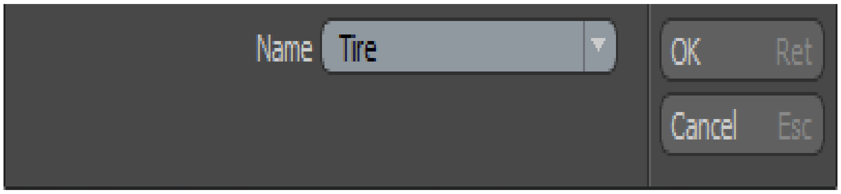
☐ Add to Library

Polygon Set Material

Option	Description
Name	The **Name** function allows you to define a new name for the **Material Tag** by typing into the name input field or by clicking the dropdown widget to the right of the field and selecting from a list of tags defined previously within the scene.
Type	Select the material type. The following options are available: • **Default Material** - The default configuration. • **Unreal Material** - The color is translated to Unreal base color. • **Unity Material** - The color is translated to Unity base albedo. **Note:** Modo remembers the material type selected, and automatically sets **Type** to that material the next time you open the dialog.
Color	The **Color** option allows you to specify the **Diffuse Color** value for the material item assigned to the material group.
Diffuse	The **Diffuse** option allows you to specify the **Diffuse Amount** value for the material item assigned to the material group.
Specular	The **Specular** option allows you to specify the **Specular Amount** value for the material item assigned to the material group.
Smoothing	Use this field to toggle the **Smoothing** percentage value in the material item assigned to the material group, producing either smoothed or faceted geometry shading.
Make Default	When **Make Default** is enabled, all subsequent polygons that are created default to that **Material Tag** name, inheriting the material setting automatically instead of the receiving the setting defined by the default **Base Material**.
Add to Library	When **Add to Library** is enabled, any new tags assigned are automatically created as **Library** items (nothing happens when assigning an existing tag). The **Library** function works like the instancing functionality of the Shader Tree, providing a simple way to assign material settings to multiple **Material Group** items while only modifying a single Material or texture. For information on using the Shader Tree library, please see Shader Tree.

Polygon Set Part

Parts work in a similar fashion to **Material** tags, offering an additional means of defining polygon groupings. Parts can be a useful way to define additional surfacing layers in the Shader Tree, especially in cases where the surface layer spans across two or more material tag boundaries. Parts can also be useful when exporting geometry to other applications. You can assign **Part** tags from the menu bar command **Geometry** > **Polygon** > **Set Part**. When invoked, a dialog opens.

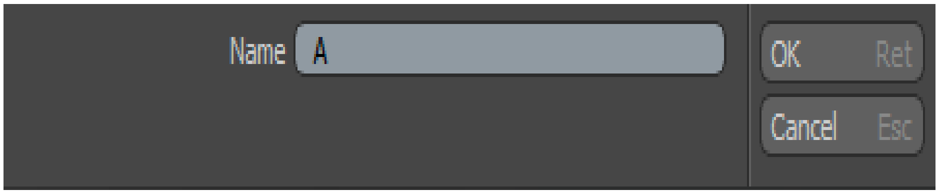


Option	Description
Name	The **Name** function allows you to define a new name for the **Part Tag** by typing into the name input field or by clicking the dropdown widget to the right of the field and selecting from a list of tags defined previously within the scene.

Polygon Set Smoothing Group

The Modo Material item allows you to define the smoothing **Angle** and **Amount** for a surface, determining if adjacent polygons are shaded smooth or creased (producing a faceted surface). This option is set based on their difference of angle, which in some cases may not provide you with enough control. This is especially true in the case of lower resolution game models where you might prefer to set smoothing based on groups. The borders between these groups determine where creases appear. The interior of the smoothing group is still based on the material smoothing settings or vertex normal map as Modo smoothing groups only add creases, they do not remove them.

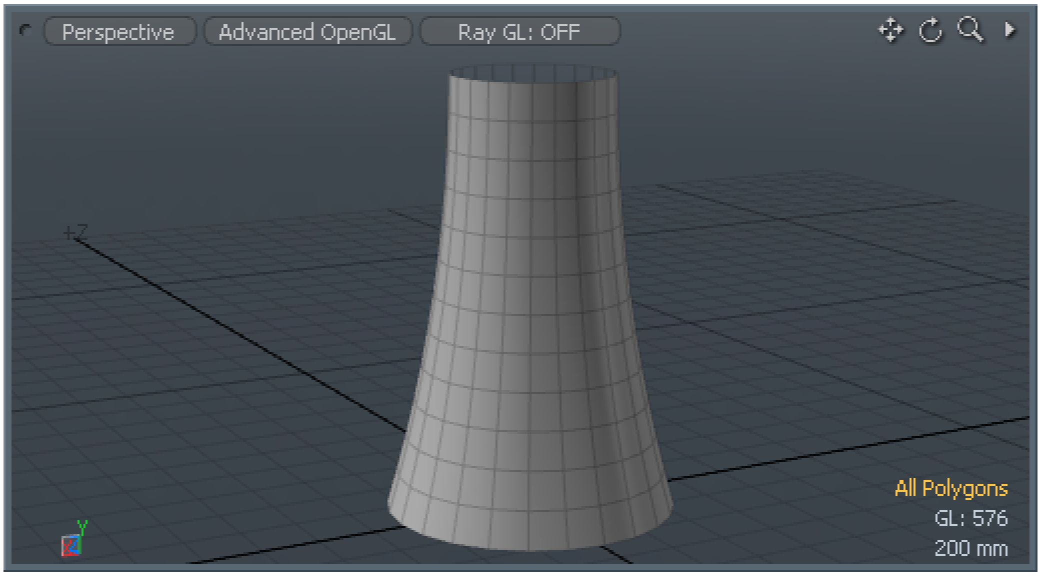
Modo assigns smoothing groups by applying a specific type of polygon tag with an alphanumeric character name. You can access the command to tag a polygon as such in the menu bar under **Geometry** > **Polygon** > **Set Smoothing Group**. Invoking the command opens the **Polygon Tag** dialog.



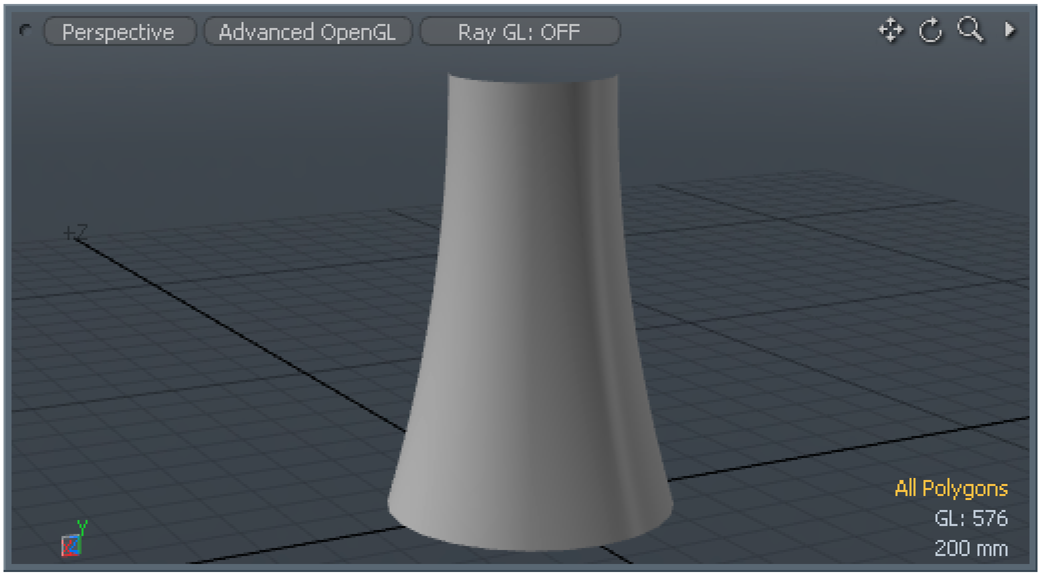
When defining a smoothing group, you must assign alphanumeric characters (A-B, 0-9) representing **sets**. Smoothing group tags are not treated as **names**, instead each tag is considered a collection of characters or

set. If two adjacent group tags contain the same character then they are smoothed. They are only creased if adjacent smoothing group tags don't contain any of the same characters.

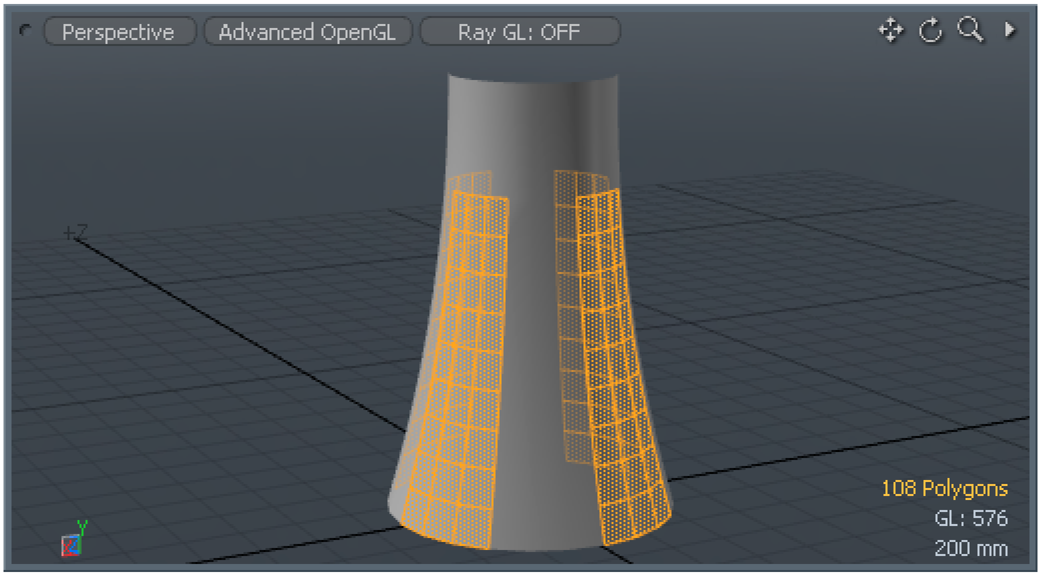
Here is a piece of cylindrical geometry that requires creased edges.


Perspective
Advanced OpenGL
Ray GL: OFF
+Z
Y
All Polygons
GL: 576
200 mm

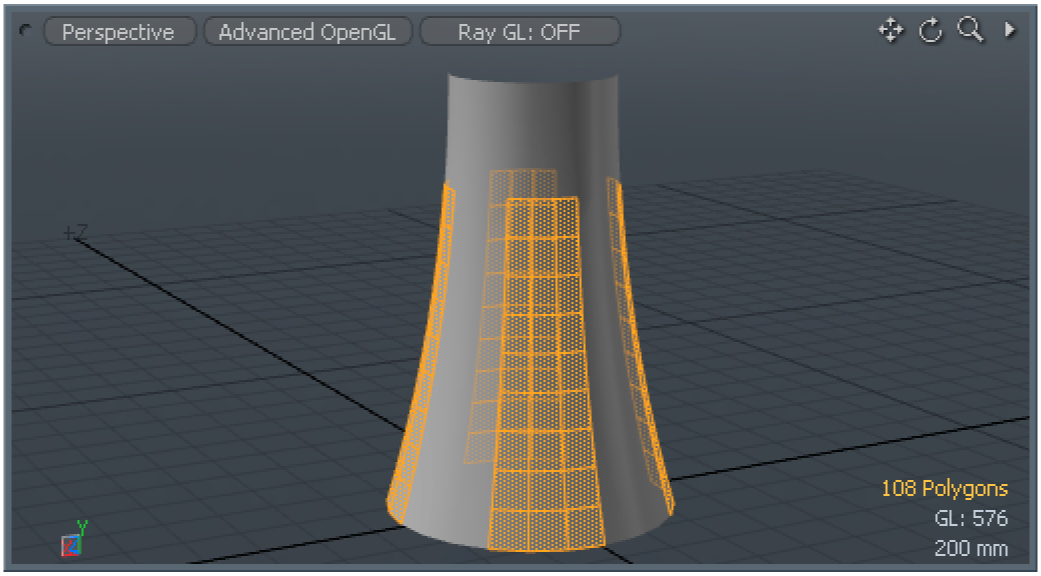
Turning off the edges (numeric pad **/** keyboard shortcut) allows you to better see the smoothing.


Perspective
Advanced OpenGL
Ray GL: OFF
+Z
Y
All Polygons
GL: 576
200 mm

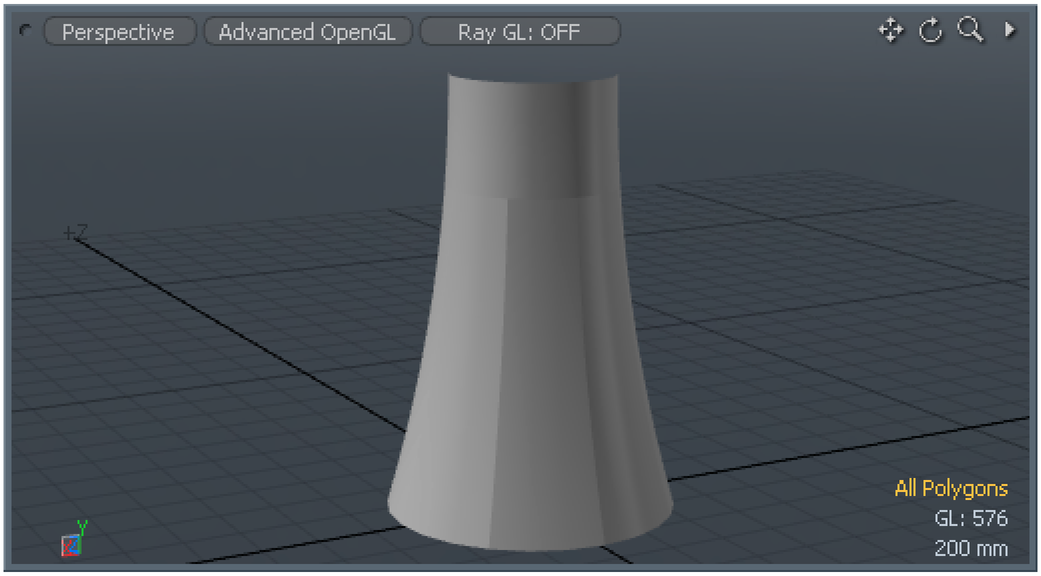
Sections of geometry are selected and tagged with a smoothing group name **A**.


Perspective
Advanced OpenGL
Ray GL: OFF
+Z
Y
108 Polygons
GL: 576
200 mm

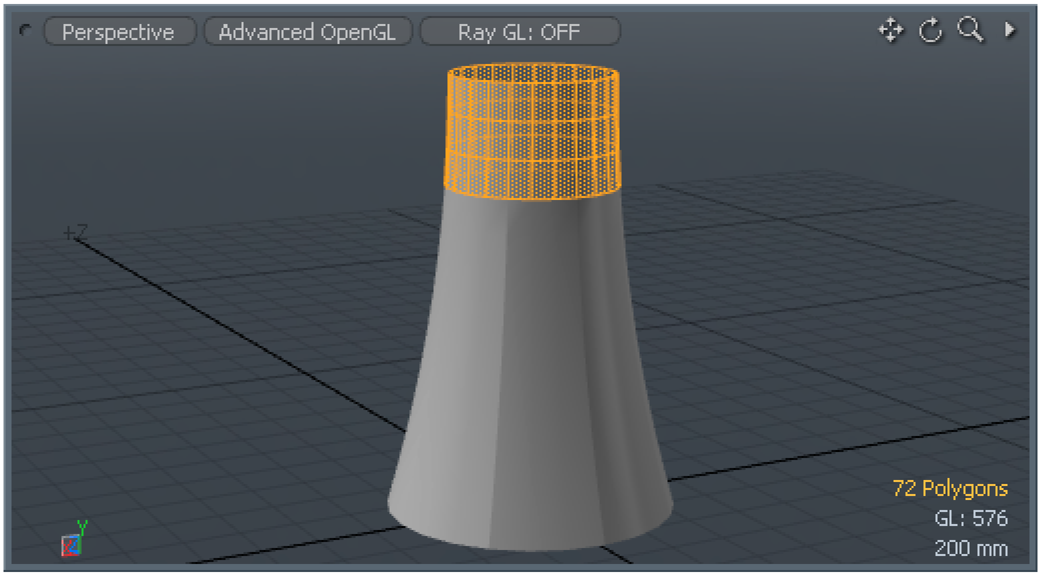
Alternating sections are selected and tagged **B**.


Perspective
Advanced OpenGL
Ray GL: OFF
+Z
Y
108 Polygons
GL: 576
200 mm

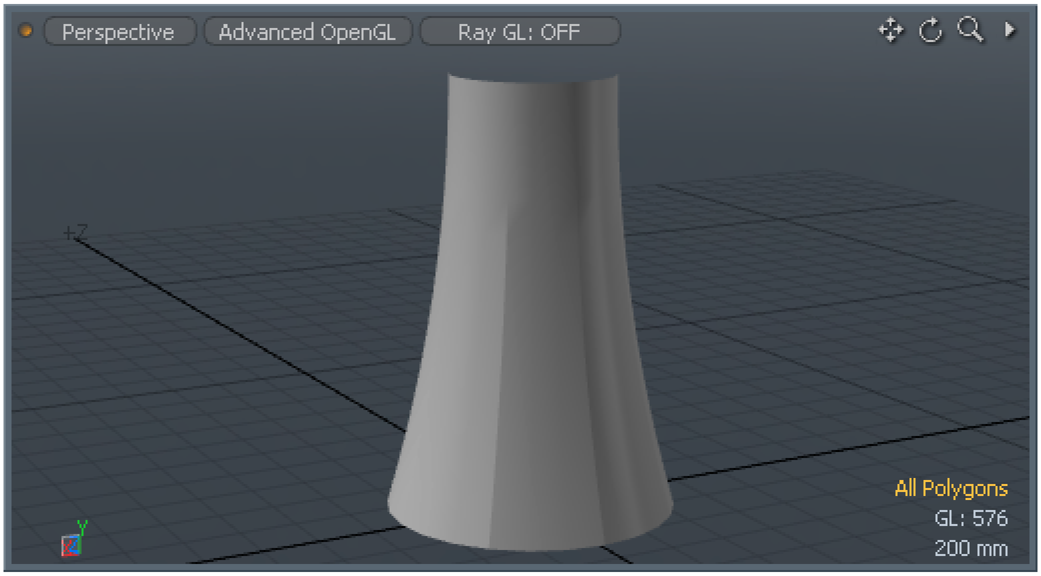
The borders between adjacent sections are now creased.


Perspective
Advanced OpenGL
Ray GL: OFF
+Z
All Polygons
GL: 576
200 mm

The top section is now selected and assigned the smoothing group name **AB**.


Perspective
Advanced OpenGL
Ray GL: OFF
+Z
72 Polygons
GL: 576
200 mm

Since the top section belongs to both the **A** and **B** sets, it is smoothed by both.



Tip: Creasing can now be assigned much more easily by using the Split Vertex Normals command.

Shader Tree

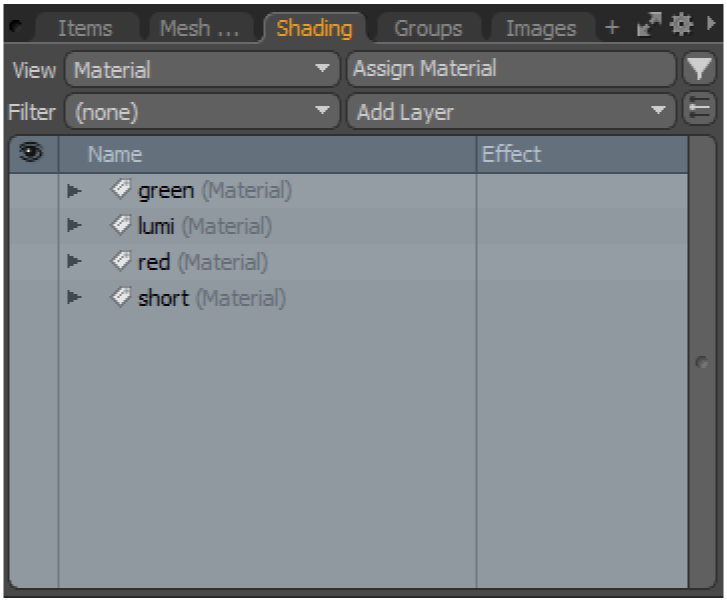
Modo employs a unique global shading model for surfacing, allowing attributes to apply to any number of layers or items within a scene or, even the entire scene very easily. The basics of shading any surface begin with applying a polygon tag (see Defining Shaders). This determines where the surfacing attributes are applied during render time. Once those tags are applied, the actual shading itself is controlled by the layers of the **Shader Tree**.

These layers determine if a surface is shiny or dull, red, or green, transparent or opaque, or whatever surfacing attribute combinations you may want to apply. Found in the **Shading** viewport, the **Shader Tree** has two distinct modes for operation. The **Material** mode, which simplifies assigning surfaces and applying surfacing, and the more advanced **Shader Tree** mode gives full control to the shading and rendering of the scene. If you have never worked with Modo before, it may be helpful to make sure you are comfortable with the viewport UI, by referring to the Shader Tree viewport topic.

Shader Tree Views

Before covering how to work with the **Shader Tree**, it should first be explained how the two view modes of the **Shading** viewport differ. In the top-left of the panel, there is a **View** option; clicking the button offers two choices: **Materials** and **Shader Tree**. They are similar in many respects, but their differences make surfacing tasks easier for certain workflows.

Material View



The **Material** view of the tree is meant to only displays the **Material** tags 🏷 of selected item layers. By only viewing the tags of the selected items, the number of layers visible at any time is reduced, providing a more streamlined workflow. Also, the tags are displayed alphabetically in this mode, making it easier to find a specific layer.

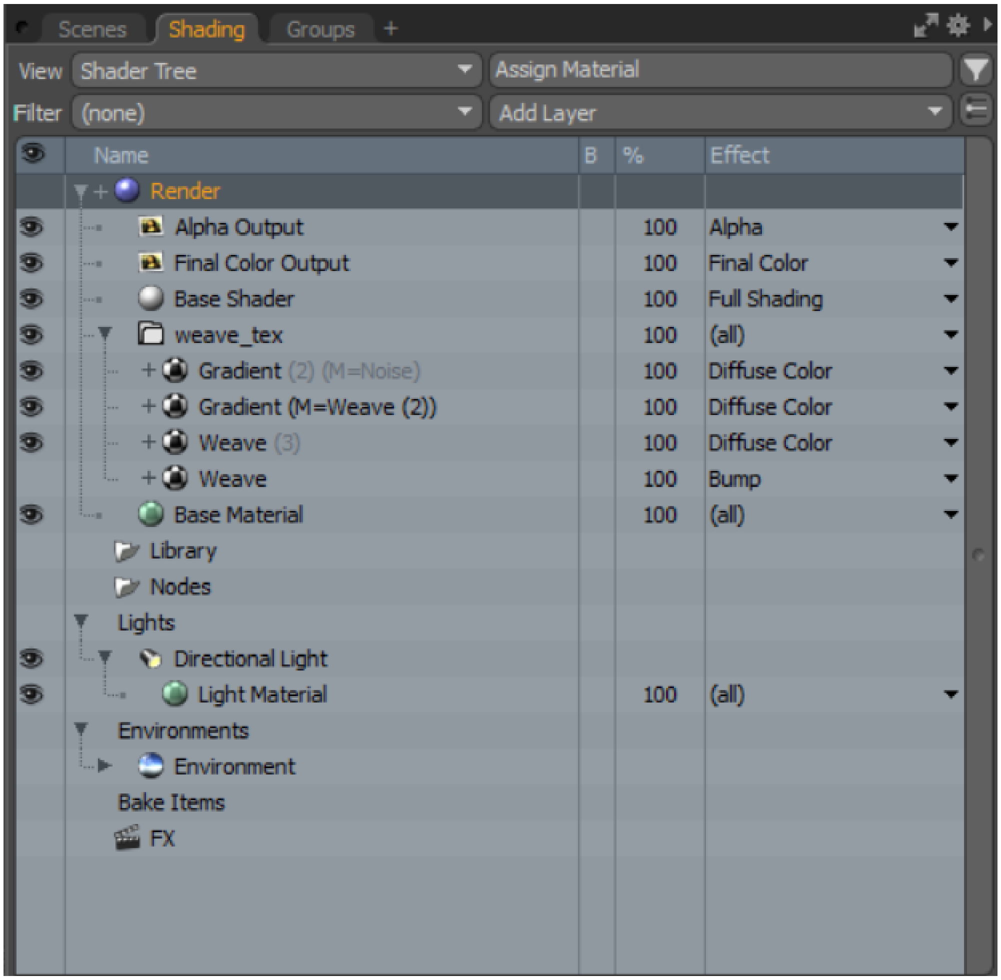
In **Material** view, each individual tag operates as a layer, using hierarchy to define how surfacing is applied. Toggle the preceding arrow ▸, to reveal the associated material item. This is the same item as in **Shader Tree** mode and represents multiple aspects of the final rendered surface.

You can apply new tags to any target geometry by first selecting the surface in **Polygons** selection mode and pressing the **Assign Material** button in the viewport, or by pressing the **M** keyboard shortcut. This opens a dialog, allowing you to customize the tag name. Clicking **OK** then adds the resulting tag as a layer to the tree. Additional layers can also be added to individual tags to control other shading aspects of the surface by clicking the **Add Layer** button and choosing from any of the numerous options.

Note: For more information on working with materials, see Material.

To provide the simplified workflow of the **Materials** mode, certain layers aren't visible in the tree. Environments can't be edited directly, nor can any shading or render output edits be made. These are done while in the **Shader Tree** view.

Shader Tree View



Name	B	%	Effect
Render			
Alpha Output		100	Alpha
Final Color Output		100	Final Color
Base Shader		100	Full Shading
weave_tex		100	(all)
Gradient (2) (M=Noise)		100	Diffuse Color
Gradient (M=Weave (2))		100	Diffuse Color
Weave (3)		100	Diffuse Color
Weave		100	Bump
Base Material		100	(all)
Library			
Nodes			
Lights			
Directional Light			
Light Material		100	(all)
Environments			
Environment			
Bake Items			
FX			

The **Shader Tree** view displays the entire tree as it's evaluated for rendering. In the **Shader Tree**, tags aren't displayed directly, but instead are replaced by the Material Group mask layers. Hierarchy plays a role here as well, where surface shading is constrained by the masks defined by the material groups.

Note: To hide unselected background elements, click the Show Only Selected Items button next to **Add Layer**.

Understanding the Shader Tree

The **Shader Tree** is so named because it is conceptually similar to a tree where the Render Item can be considered the trunk, with each individual material group item as the branches. Each branch can contain other material groups nested inside for different layering effects. Eventually, all these layers (nested or otherwise) feed into the Render Item. This information is then fed into any Render Outputs, producing the final rendered image when you activate any of the render commands.

To produce the final result, layers are processed in a bottom to top fashion. Thus, anyone familiar with Photoshop's layer's palette might be familiar with how Modo's **Shader Tree** works - as any opaque layer above another always supersedes the underlying layers, obscuring them. These underlying layers can only be made visible by either changing the transparency of the upper layer, changing the blending mode, or by applying a mask to the layer.

Adding and Modifying Shader Tree Item Layers

While the **Shader Tree** does control the shading and rendering of any scene, the main functionality of the tree itself is for adding, removing, organizing, and otherwise managing the many **Shader Tree** item layers. When adding new items to the tree, the various item layers can be added by clicking the **Add Layer** option of the full viewport window and selecting an item from the context menu. New items are always added to the top of the list in the **Shader Tree** directly below the **Base Shader** item. In cases where a Material Group Item is selected, new layers are added within that sub-group, directly below the Group Item itself. For information regarding which types of layers can be added, see Shader Tree Item Layers.

Any of the editable properties related to item layers appear within the Properties Panel when the layer is selected in the **Shader Tree**. Selected layers are indicated by the dark bar highlighting the layer name. Multiple non-consecutive layers can be selected by holding down the **Ctrl** key on the keyboard and clicking subsequent layers in the list. Consecutive layers can be selected automatically by holding down the **Shift** key on the keyboard, and clicking on a new layer - any layers between the current and active selections are selected as well. Multiple layer selections of identical item types allow for gang editing of property values, eliminating the need to individually adjust item properties.

For organizational purposes, you may wish to rename layers, which can be accomplished by selecting the target layer and choosing the menu bar command **Item** > **Rename**. This opens a dialog where you can enter the new name.

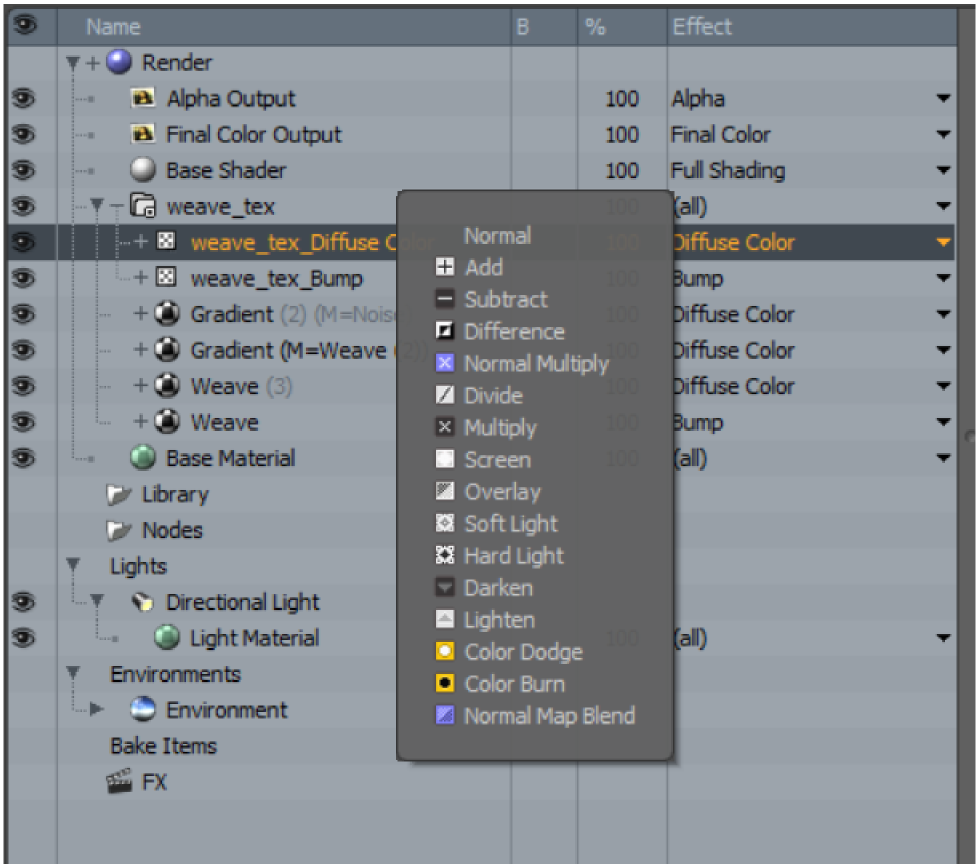
Tip: You can import PBR textures to the Shader Tree using the PBR loader. See PBR Textures for more information.

Tip: Be aware, if multiple items are selected, including anything in the **Item List**, all selected items receive the new name.

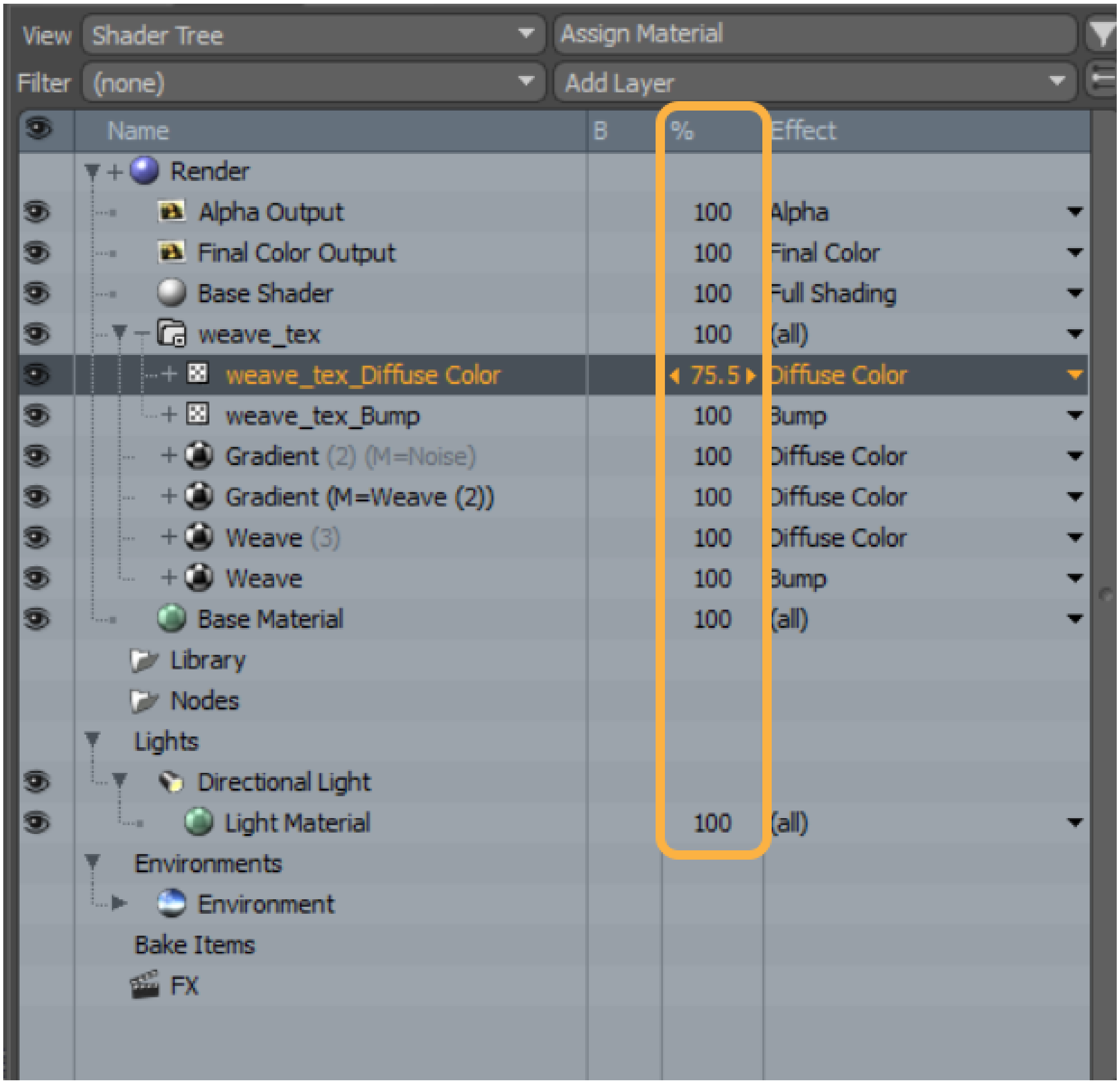
To avoid this from happening, it's better to use the **Rename** option from the context menu when right-clicking on the layer name itself.

From the shader tree, you can also change the **Blend Mode** and **Opacity** of layers.

Blend Modes are assigned by clicking in the **B** column and selecting a blend mode from the context menu.


Name
B
%
Effect
Render
Alpha Output
Final Color Output
Base Shader
weave_tex
weave_tex_Diffuse Color
weave_tex_Bump
Gradient (2) (M=Noise
Gradient (M=Weave (2))
Weave (3)
Weave
Base Material
Library
Nodes
Lights
Directional Light
Light Material
Environments
Environment
Bake Items
FX
100
100
100
Alpha
Final Color
Full Shading
(all)
Diffuse Color
Bump
Diffuse Color
Diffuse Color
Diffuse Color
Bump
(all)
(all)
Normal
Add
Subtract
Difference
Normal Multiply
Divide
Multiply
Screen
Overlay
Soft Light
Hard Light
Darken
Lighten
Color Dodge
Color Burn
Normal Map Blend

Opacity is increased or deceased by click-dragging the number in the **%** column.



View: Shader Tree
Filter: (none)

Name	B	%	Effect
Render			
Alpha Output		100	Alpha
Final Color Output		100	Final Color
Base Shader		100	Full Shading
weave_tex		100	(all)
weave_tex_Diffuse Color		75.5	Diffuse Color
weave_tex_Bump		100	Bump
Gradient (2) (M=Noise)		100	Diffuse Color
Gradient (M=Weave (2))		100	Diffuse Color
Weave (3)		100	Diffuse Color
Weave		100	Bump
Base Material		100	(all)
Library			
Nodes			
Lights			
Directional Light			
Light Material		100	(all)
Environments			
Environment			
Bake Items			
FX			

Finally, you can also rename a layer inline. Simply double-click directly on the layer name and then type a new name inline, pressing the **Enter** key when done to accept the change.

Any layer within the **Shader Tree** can be repositioned by clicking and dragging the layer, and moving it to a new position. The orange line that appears when dragged indicates the position the item takes when the mouse button is released. When nesting layers within a material group, the layer inherits the mask, if any, specified by the Material Group Item. Layer positions are an important concept of the **Shader Tree**, as they control the order in which items are evaluated at render time. Higher layers in the tree supersede lower layers.

Layers can be easily removed by selecting the target layer in the **Shader Tree** and selecting the menu command **Texture** > **Delete Shader Tree Layer**. Also, right-clicking the layer and selecting **Delete** from the context menu or by pressing the **delete** key on the keyboard also allows you to delete layers.

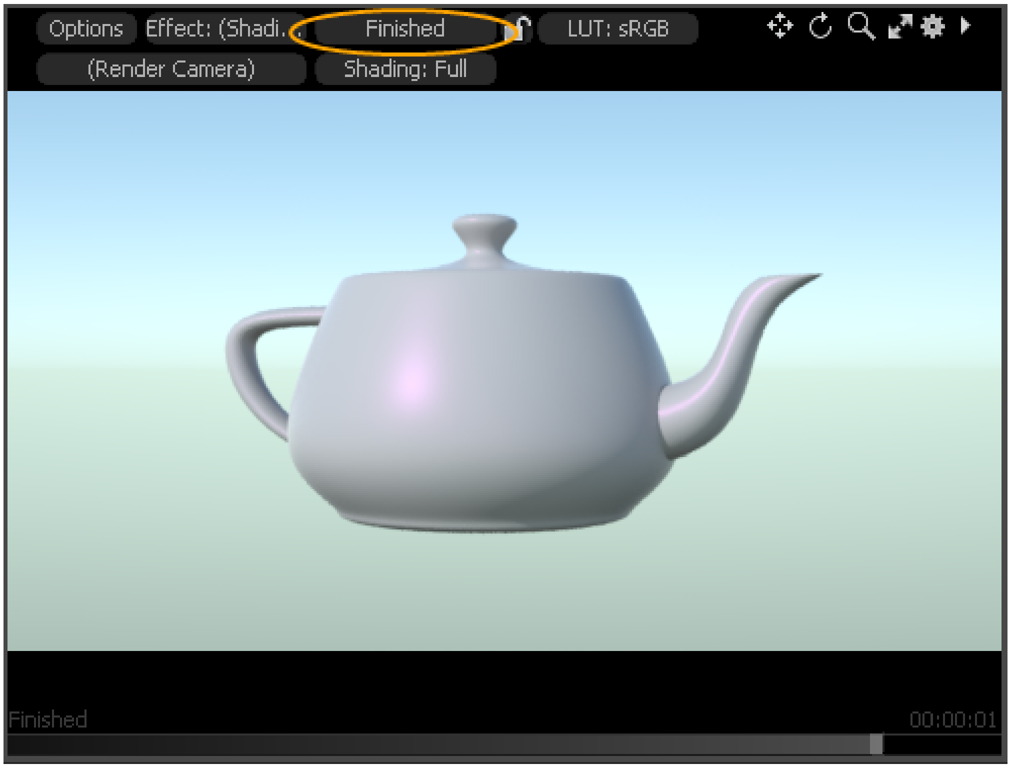
Tip: When assigning the layer order, remember the tree works bottom up. When setting a layers **Effect** to **Layer Mask**, it only masks the texture layer directly above it. Also, gradients that use other layers for **Input Parameters** need to be below the gradient itself, so the layer is calculated first, to feed into the gradient. If a layer's effect is dependent on another, and it isn't working, sometimes changing the layer order can solve the issue.

Adding Environment Presets to the Shader Tree

You can quickly select an Environment Preset from the Preset Browser and drag it into the **Environment** folder on the Shader Tree. A contextual menu appears, allowing you to either replace or add your selected Environment Preset. This is useful when experimenting with different backgrounds using different colors, gradients, and image maps.

To apply a environment preset:

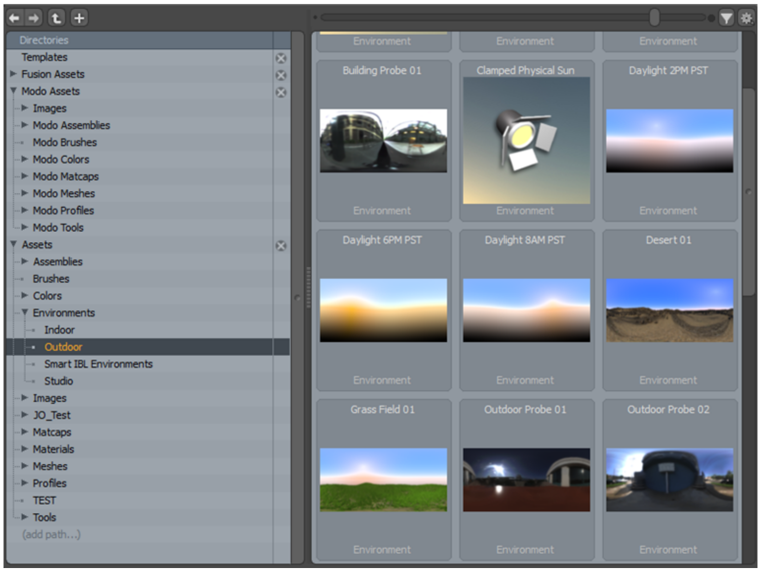
1. Load your mesh using the **Model** layout and click **Render** > **Open Preview**.
2. Click the Shader Tree button to render your scene.



3. On the top of the Toolbar, on the left panel, click **Preset**.

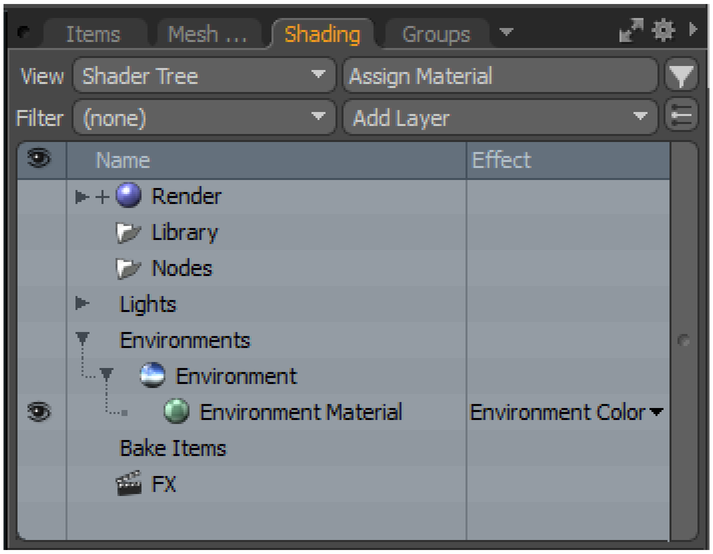
   The **Preset Browser** opens in a window.

4. On the left panel, expand **Assets** > **Environments**.


Directories
Templates
Fusion Assets
Modo Assets
Images
Modo Assemblies
Modo Brushes
Modo Colors
Modo Matcaps
Modo Meshes
Modo Profiles
Modo Tools
Assets
Assemblies
Brushes
Colors
Environments
Indoor
Outdoor
Smart IBL Environments
Studio
Images
JO_Test
Matcaps
Materials
Meshes
Profiles
TEST
Tools
(add path...)
Building Probe 01
Clamped Physical Sun
Daylight 2PM PST
Daylight 6PM PST
Daylight 8AM PST
Desert 01
Grass Field 01
Outdoor Probe 01
Outdoor Probe 02
Environment

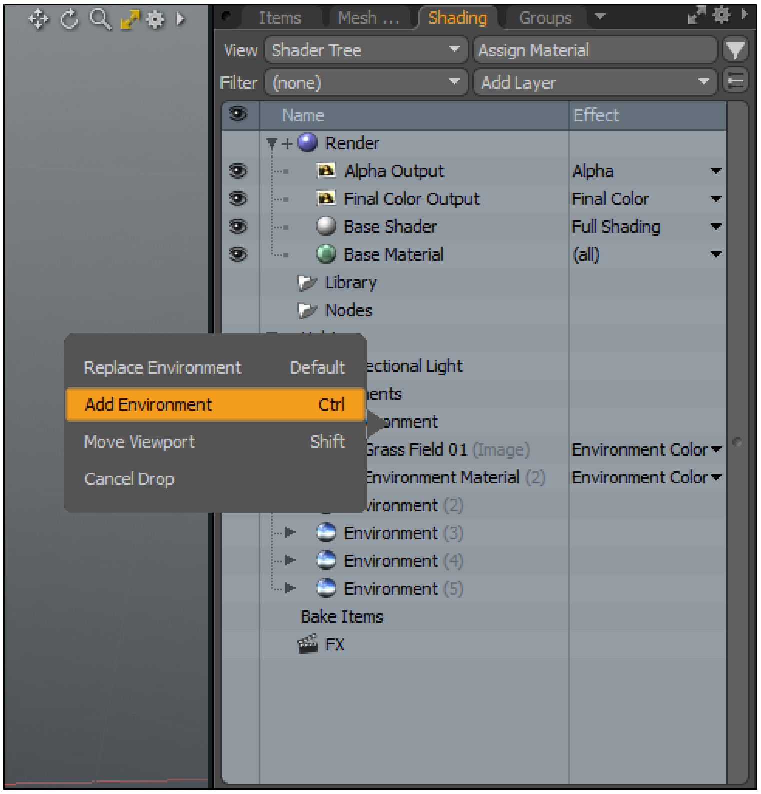
5. Expand any one of the sub-categories.
6. On the right panel, open the **Shading** tab to view the **Shader Tree**.
7. Click the arrow beside **Environments** and then **Environment** to expand the items in the list.

   The default **Environment Material** item displays in your list.


Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Library
Nodes
Lights
Environments
Environment
Environment Material
Environment Color
Bake Items
FX

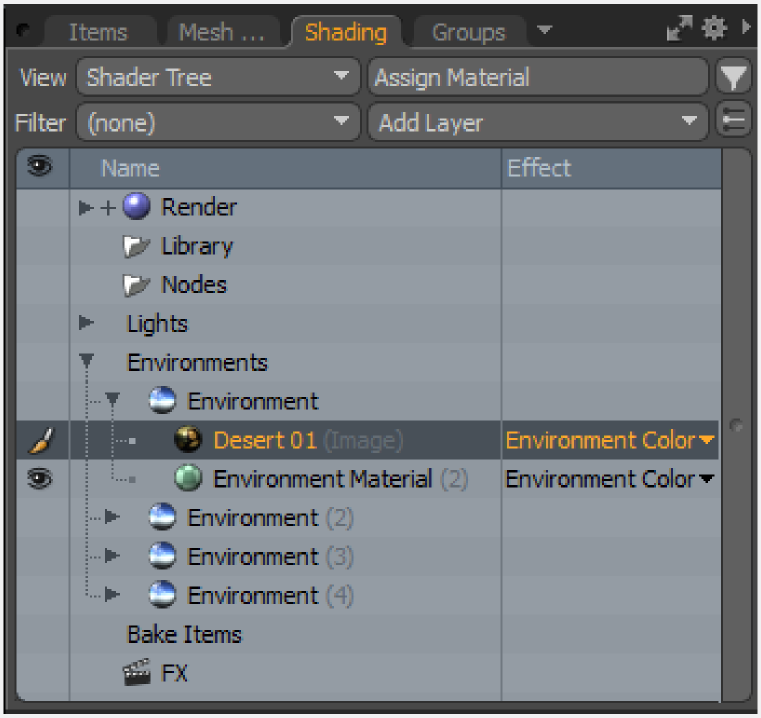
8. In the **Preset Browser**, **Ctrl/Cmd** + click and drag one of the environments onto the **Environments** item and hold until the contextual menu displays.

Tip: If you have more than one Environment item, you can place your selected Environment Preset anywhere in your list under **Environments** by dragging it into place.


Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Base Material
(all)
Library
Nodes
ectional Light
ents
onment
Grass Field 01 (Image)
Environment Color
Environment Material (2)
Environment Color
vironment (2)
Environment (3)
Environment (4)
Environment (5)
Bake Items
FX
Replace Environment
Default
Add Environment
Ctrl
Move Viewport
Shift
Cancel Drop

9. Select one of the following:

Option	Description	Shortcut Key
Replace Environment	Replaces the existing Environments set up with the selected Environment Preset.	n/a
Add Environment	Keeps the current environment set up and adds a new **Environment(#)** item under the parent **Environments** item. This allows you to select the new Environment item and change the **Properties** to determine which ray types should be visible and to apply other properties.	**Ctrl**

	 Items Mesh ... Shading Groups View Shader Tree Assign Material Filter (none) Add Layer Name Effect Render Library Nodes Lights Environments Environment Desert 01 (Image) Environment Color Environment Material (2) Environment Color Environment (2) Environment (3) Environment (4) Bake Items FX **Note:** If the Preview Viewport is open at that time, the environment is updated in the viewport.	
Move Viewport	Replaces the existing Environments set up with the selected Environment Preset.	**Shift**
Cancel Drop	Cancels your action.	n/a

Hierarchies

Hierarchies are quite important to the **Shader Tree**, as this is how associations are made to material groups, which generate the masks necessary to limit surfacing to individual sections of geometry. Any item that is a child of a material group inherits the Group Item's mask settings. Hierarchies are denoted by the small indentation in subsequent item layers. Material Group Items that contain child layers are preceded by a small arrow ▸ . Clicking the arrow toggles the hide/reveal action of the nested layers. Successive nested Material Group Items are further indented, indicating their association to the parent item. The parent/child associations are created by dropping a layer onto another closed group, or between two layers of an open group. When adding new layers, selecting the parent Material Group Item before selecting the new item layer type nests the new layer inside the group.

Finally, if a texture layer or procedural (image-based or otherwise) is dropped onto another texture layer, the dropped layer automatically becomes a layer mask to the parent. Masked layers are preceded by a small **+** icon. Click the icon to toggle the view of the item's mask, which is now also indented to show its relationship to the parent item.

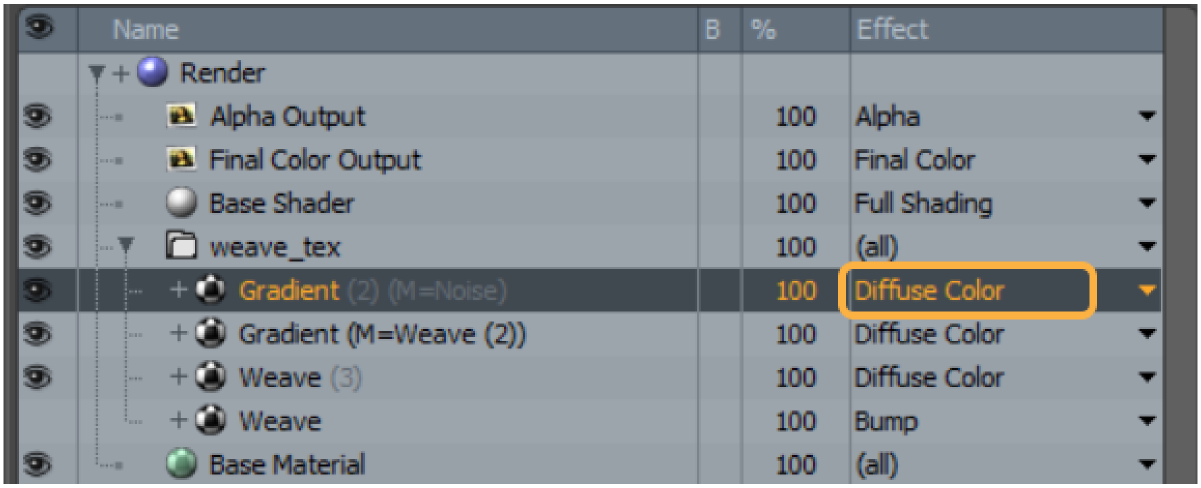
Tip: The key to a smooth workflow is organization when it comes to the **Shader Tree**. If left unattended, materials and groups could quickly stack up, leaving a long unwieldy list that is difficult to navigate. Multiple material groups can be further grouped by multi-selecting several layers, using the **Shift** or **Ctrl** modifier keys. Once selected, press **Ctrl**+**G** on the keyboard to move the highlighted layers into another group. When used this way, the groups properties are set to **(all).** When set as such, the group merely provides a means to organize multiple material groups.

The resulting **Group Item** can be renamed either by right-clicking the title and choosing **Rename** from the context menu, or by double-clicking for inline editing. Grouping items layers can greatly simplify the visual appearance of the **Shader Tree** list, making it much easier to navigate. You can quickly expand and collapse all nested layers within a hierarchy by pressing the **Shift** key when toggling the arrow ▸ icon.

Specifying Layer Effects

By default, any new texture layer added to the **Shader Tree** modulates the **Diffuse Color** channel, meaning the layer affects the apparent visual color of the surface. Should you wish to affect the surface differently, such as specifying a layer as a bump or displacement map, this is specified in the **Effect** column of the **Shader Tree**. A layer can be set to modulate a specific effect, by way of a right-clicking the effect name itself in the effect column, and selecting a new effect type from the pop-up context menu.

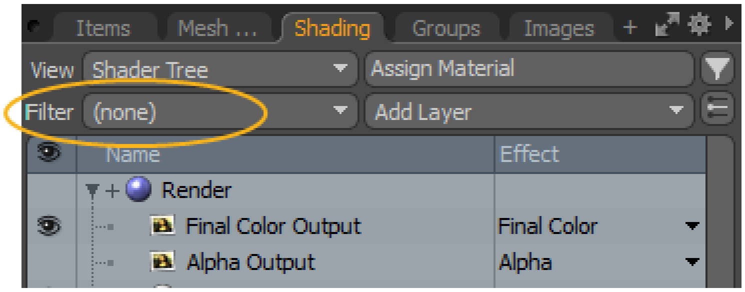
Whenever a texture layer is created, such as an **Image Map** or a procedural texture, an associated **Texture Locator** is added as well to the Items (Scenes) List. Texture locators hold position and projection settings specific to the texture layer, and define how it is applied to the surface, while also providing a means to animate the texture using basic transforms, applied with keyframes to the locator itself.



Name	B	%	Effect
Render			
Alpha Output		100	Alpha
Final Color Output		100	Final Color
Base Shader		100	Full Shading
weave_tex		100	(all)
Gradient (2) (M=Noise)		100	Diffuse Color
Gradient (M=Weave (2))		100	Diffuse Color
Weave (3)		100	Diffuse Color
Weave		100	Bump
Base Material		100	(all)

Filtering the Item List View

At the top of the **Shading** viewport, under the **View** option, is the **Filter** control. This determines how layers are displayed within the viewport. The **Filter** control always defaults to the **(none)** value, which displays all items within the **Shader Tree**. Alternate options available in the menu can be selected by clicking the small triangle widget to the right of the bar. When any of the other options are selected, they temporarily toggle the visibility of the items in the list, without removing or disabling them, making it easier to locate a specific layer, or to make selections for gang editing multiple items easier.



View: Shader Tree

Filter: (none)

Name	Effect
Render	
Final Color Output	Final Color
Alpha Output	Alpha

The **Filter** options, in addition to **(none)**, include:

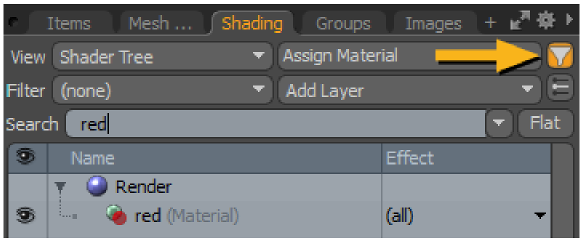
- **Material** - displays only **Material Items** in the tree.
- **Texture** - displays only **Texture Items** , in the tree.
- **Group** - displays only **Material Group Items** in the tree.
- **Shader** - displays only **Shader Items** in the tree.
- **Render Output** - displays only **Render Output Items** in the tree.
- **Fur Materials** - displays only **Fur Material Items** in the tree.
- **Filter on Selection** - when this option is enabled, filtering only displays **Shader Tree** layers applied to currently-selected items.

Searching Layers

Next to the **Assign Material** button is the Find button that opens the **Find** functionality, allowing you to search for layers by name. Clicking the button opens a search bar. Typing some text into the input field isolates only the **Shader Tree** layers that contain the same string. There are two modes for the **Search** function that are designated from the arrow drop down: **Simple Search** and **Pattern Matching**.

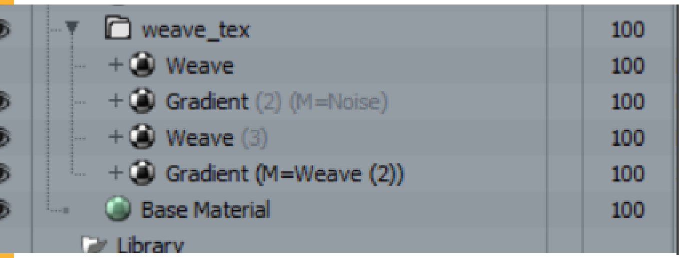
Simple Search locates layers based on matching the exact input (including spaces); so typing "1 2" locates "Apple1 2" and "1 2Banana", but not "Apple12" or "2Banana".

Pattern Match allows for a more robust type of search, allowing for symbolic characters. For more information, see Shader Tree UI. Enabling the **Match Case** option further constrains the search to also matching the upper and lower casing of the alphabet characters. The **Flat** option toggles the indentation of the layers in the panel, making it easier to locate specific layers.

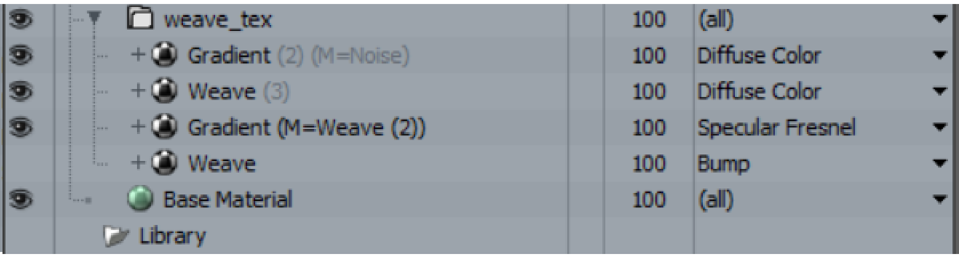

Items
Mesh ...
Shading
Groups
Images
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Search
red
Flat
Name
Effect
Render
red (Material)
(all)

Ordering Layers

Layers can be organized within shaders, so that layers with the same effect are positioned next to each other. The orders of the layers will be the same as the order of controls in the material properties form.

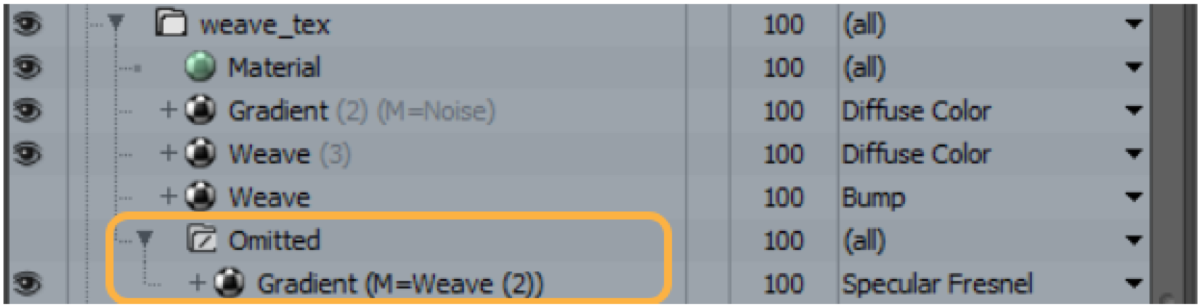

weave_tex 100
Weave 100
Gradient (2) (M=Noise) 100
Weave (3) 100
Gradient (M=Weave (2)) 100
Base Material 100
Library

Before ordering


weave_tex 100 (all)
Gradient (2) (M=Noise) 100 Diffuse Color
Weave (3) 100 Diffuse Color
Gradient (M=Weave (2)) 100 Specular Fresnel
Weave 100 Bump
Base Material 100 (all)
Library

After ordering

Layers that are not used by a particular Shading Model or placed in an **Omitted** group.



👁	weave_tex	100	(all)
👁	Material	100	(all)
👁	Gradient (2) (M=Noise)	100	Diffuse Color
👁	Weave (3)	100	Diffuse Color
	Weave	100	Bump
	Omitted	100	(all)
👁	Gradient (M=Weave (2))	100	Specular Fresnel

Using the Library

The **Shader Tree** has a **Library** function that serves to store surfacing definitions that can then be applied to Material Group Items elsewhere in the tree. When the **Library** material is updated, the changes are propagated to all the references in the tree. This can be useful to reduce the number of total layers required in the **Shader Tree** as well as simplify workflows where identical materials are used multiple times for different surfaces.

Use of the **Library** is easy; you can drag and drop preset materials to the **Library** section of the **Shading** viewport (custom materials can be created as well, as long as they exist under a single Material Group Item). To apply any of the library items to a surface, click the topmost Material Group Item in the **Library**, then drag and drop this into the target location in the **Shader Tree**, most likely inside another material group, masking it. This drops a named reference item ▶ into the group that represents the shading of the library item. Any updates or additions made to the material group in the **Library** are propagated to all the associated references throughout the **Shader Tree**. Like standard **Shader Tree** layers, placement of the layer within the tree determines shading, but library references cannot be masked or modified directly. To convert a library reference to standard **Shader Tree** layers, right-click on the layer and apply the **De-Reference** command.

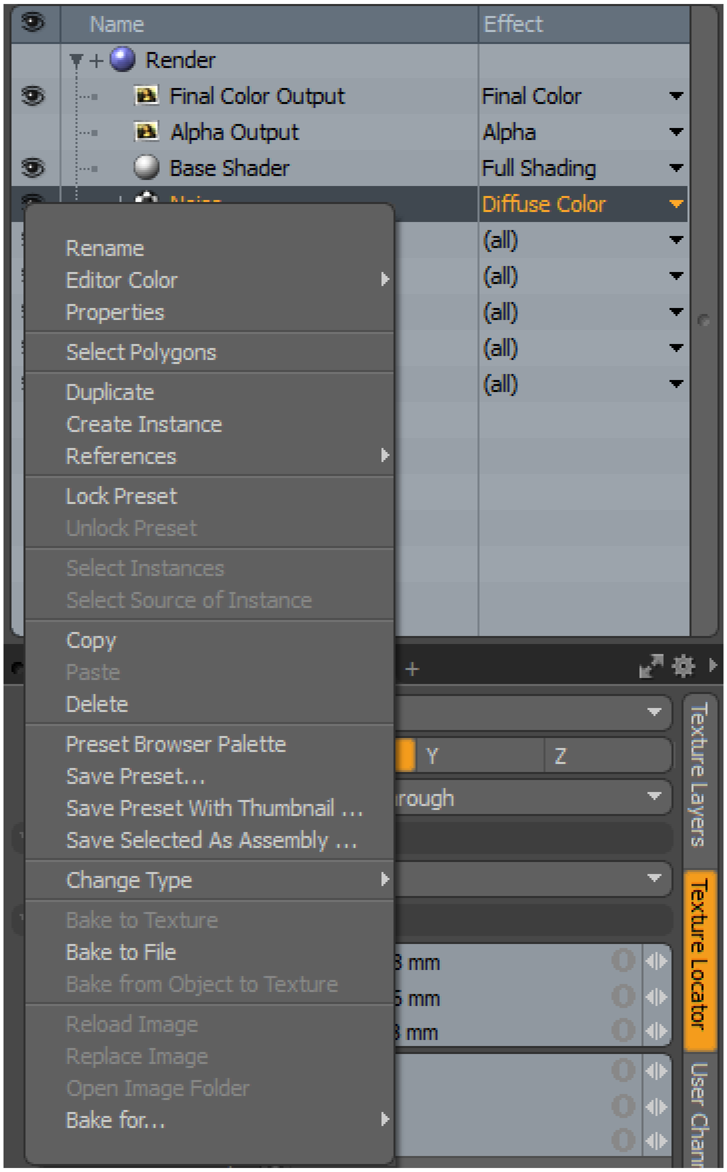
Tip: When creating **Library** materials, **Shader Tree** material group layers (and their hierarchy) can also be dragged and dropped into the **Library**, however, any masks generated by the material group itself is retained in the library item and, therefore, needs to be disabled before its creation or the resulting surface doesn't properly evaluate.

Toggling Layers

Sometimes you may wish to toggle the effect a layer has on a scene without removing the item from the list. This can be accomplished by clicking over the eye icon 👁 within the visibility column on the left side of the viewport. Toggling a layer disables/enables its affect on shading in the scene, but the item itself retains all settings and remains part of the **Shader Tree** list when scenes are saved or closed. The ability to toggle layers can be extremely helpful in a variety of scenarios, including creating various shading passes that you want to enable and disable when making Render Passes, applying a variety of textural options when rendering out conceptual product stills, and experimenting with alternate shading options while exploring looks for a scene.

Contextual Menu

Additional options for working with the item layers are available by right-clicking on an item layer name and choosing an option for the pop-up contextual menu.


Name
Effect
Render
Final Color Output
Final Color
Alpha Output
Alpha
Base Shader
Full Shading
Diffuse Color
(all)
(all)
(all)
(all)
(all)
Rename
Editor Color
Properties
Select Polygons
Duplicate
Create Instance
References
Lock Preset
Unlock Preset
Select Instances
Select Source of Instance
Copy
Paste
Delete
Preset Browser Palette
Save Preset...
Save Preset With Thumbnail ...
Save Selected As Assembly ...
Change Type
Bake to Texture
Bake to File
Bake from Object to Texture
Reload Image
Replace Image
Open Image Folder
Bake for...
Y
Z
rough
mm
mm
mm
Texture Layers
Texture Locator
User Chann

The following options are available in the contextual menu:

- **Rename** - allows you to assign a new name to the specified layer.

- **Editor Color** - 12 color options are provided for colorizing the **Shader Tree** layer itself; meant to be used as a visual organizational device. Choosing any of the colors from the supplied menu applies the selected color as the layer background.
- **Properties** - opens the **Properties** viewport with the selected layers corresponding attribute settings.
- **Select Polygons** - makes all the polygons in the scene related to the current mask selected, switching the Modo interface into **Polygons** selection mode if not already active.
- **Duplicate** - creates a copy of a layer with identical settings. Changes in the original item are not reflected in the duplicate layer, nor are modifications to the duplicate reflected in its original layer.
- **Create Instance** - creates a clone of a layer that references the original source. Changes to one layer automatically propagate to all instances. Instanced layers are denoted by their italicized name in the **Shader Tree**. This is especially useful when separate surfaces with identical properties reside within different masks. When instances are used, changing one item's attributes automatically clones the changes on the instance. Changes to the instance itself act as local overrides, modifying the attribute directly, while changes are not applied back to the source of the instance. Further changes to the source are propagated, except to those changes made individually on each instance, on a channel-by-channel basis.
- **References** - opens a menu with access to the controls for working with references.
- **Lock Preset**/**Unlock Preset** - when the **Lock Preset** command is applied to any **Library** material, this prevents any accidental changes from being applied to the material group. The target layer needs to be unlocked for further editing. The **Lock Preset**/**Unlock Preset** commands have no effect on standard **Shader Tree** layers.
- **Select Instances** - if a layer has any instances of it in the **Shader Tree**, the **Select Instances** command selects all the associated instances of the target automatically.
- **Select Source of Instance** - selects the parent source items of any instance. Instances are denoted by their italicized name in the **Shader Tree**.
- **Copy**/**Paste**/**Delete** - entire layers, as well as the settings of individual layers themselves, can be copied, pasted, and/or deleted using these commands. When pasting values from one layer to another, the layers must be of the same type (for example, material item to material item).
- **Preset Browsers Palette** - opens a **Preset Browser** palette.
- **Save Preset** - a **Preset** stores settings so they can be applied at a later time. In the case of materials, a preset stores **Shader Tree** layers with an associated icon for visual browsing in the **Preset Browser**. A material **Preset** can be as simple as a single layer, or it can contain any number of layers with complex blending. For multi-layer **Presets**, these must all be contained under a single material group 🗀 item. Right-clicking on the target layer and selecting the **Save Preset** option from the contextual menu presents you with an OS dialog, prompting where to save the **Preset** itself.
- **Save Preset with Thumbnail** - works in the same way as the **Save Preset** command, above, but once the preset is saved, a second dialog opens, allowing you to specify any saved image that is then used as the browser icon.
- **Save Selected as Assembly** - saves the selected layer as an assembly preset.

- **Change Type** - offers the ability to convert a layer of one type, such as an image map, to another type such as a procedural. Associated **Texture Locators** retain their settings, if applicable, but layers themselves lose their settings.
- **Bake to Texture** - offers the ability to generate a texture map image of a layer. Baking textures from inside the **Shader Tree** is useful for baking a single **Effect**, such as Color or Diffuse, into an image, whereas the **Bake** command (from the **Render** menu) bakes all light and texture contributors into a final shaded image. It is reasonable to think of the **Shader Tree**-level of baking as similar to "flattening" layers in Photoshop.

  Only layers of the same **Effect** setting, underneath the texture layer that is triggering the bake, contribute to the baked layer. Once "baked", this texture can remain in the **Shader Tree** as a pre-computed result of all contributors, which results in a savings of render time and provides a real-time view of the effect. The **Bake to Texture** command requires a UV map on the baked surface, as well as a blank image map to bake into. For more on baking textures, see the Baking topic.
- **Bake to File** - functionally identical to the **Bake to Texture** command, above, but writes the resulting texture as a file to disc, to a location specified by you. When saving to a 16-bit or greater format, the **Floating Point** option must be enabled.
- **Bake From Object to Texture** - offers the ability to use the UV map of a low polygon count model to bake a texture map from a higher polygon count model. Maps can be used for any of the various effects, including diffuse color, displacement, and normal. Like **Bake to Texture**, a UV map and a blank image map to bake into are required . Both surfaces must have different material names and cannot share the same UV map. When calculating a displacement map, the **distance setting** from the dialog should be used as the **Displacement Distance** setting in the material dialog box when applied. For more on baking textures, see the Baking topic.
- **Reload Image** - reloads the image, as saved to disk. This is useful for when edits made in an external app don't trigger Modo to reload automatically, or when you wish to undo a series of internal edits all at once.
- **Replace Image** - opens a dialog allowing you to select an alternative bitmap image that replaces the selected image, while retaining its existing settings.
- **Open Image Folder** - opens the saved file's folder location on the internal hard disc, using the system's default file manager.

Shader Tree Item Layers

Every new scene in Modo, by default, provides a few basic **Shader Tree** layers to get you started. The **Material** item ●, created when you assign a material tag, is sufficient for many surfacing tasks. However, for reproducing more-complex surfaces, additional shader items eventually need to be added, for example when applying a bitmap image to produce the label on a can of soda.

The following items may be added to any scene from within the **Shader Tree** viewport, by clicking the **Add Layer** option of the full viewport window, and selecting an item from the pop-up context menu. For information regarding adding and working with **Shader Tree** item layers, please refer to the Shader Tree topic.

Shader Tree Items	
Folder	A **Folder** item acts solely as an organizational device for the **Shader Tree**.
Material Group	A **Material Group** item can be thought of as a container that holds any of the various **Shader Tree** items, such as Shaders, Materials, Textures, and even Render Outputs.
Material	A **Material** item is the default, general purpose **Shader Tree** layer that controls all the various standard shading attributes for a surface, determining its final rendered look.

Custom Materials	
Cel Edges	The **Cel Edges** material attempts to mimic the hand-drawn look of the outlines of Cel cartoons, creating colored strokes that follow surface contours and edges.
Cel Material	The **Cel Material** attempts to mimic the flat, posterized shading of Cel cartoons, creating flat, colored bands that follow surface contours and edges.
Hair Material	The **Hair Material** is designed to replace the default BRDF shading calculations of the standard material item, when applied to fur or hair, to simulate the anisotropic quality of hair.
Halftone Material	The **Halftone Material** stylizes the target surface, replacing the smooth shading with a black user-defined pattern overlaid on the diffuse color, simulating the look of a low-fidelity image printed in a newspaper or magazine.
Iridescence Material	The **Iridescent Material** is a general term for colors which change, based on the viewing angle, and this effect simulates that seen in nature, such as in pearls, butterfly wings, and

Custom Materials	
	peacock feathers.
Skin Material	The **Skin Material** is designed to completely replace the default BRDF shading calculations of the standard material item, when applied to a surface, to simulate skin (made up of multiple layers and materials, each absorbing, scattering, and reflecting light differently), creating a very subtle but distinct look.
Thin Film Material	The **Thin Film Material** works by simulating interference effects on the wavelengths of incoming light, refracting them between two virtual thin layers of film, like one would see with thin layers of oil over a puddle of water.

Enhance: Modo Textures	
Enhance:modo Textures	Enhance:Modo textures are a collection of 147 advanced procedural layers, many with additional parameters, that further extend the collection.

Image Map Textures	
Image Map	The **Image Map** item could be considered the most important texture layer, as it allows bitmap images to be applied to surfaces in Modo.
Projection Shader	The **Projection Shader** simplifies the workflow of using projected bitmap images. When applied, it replaces all surface shading with a photographic image.

Processing Textures	
Constant	The **Constant** texture simply generates a constant value or color

Processing Textures	
	layer.
Gradient	The **Gradient** texture layer provides a method for creating parametric materials. Put simply, gradients remap colors or value-based input parameters to produce an output based on the values in the gradient itself.
Gradient Fill Image	The **Gradient Fill Image** layer is a hybrid texture, as it is procedurally-generated gradient but it is based on a dynamically created **bitmap** image.
Occlusion	The **Occlusion** texture calculates the shading of surfaces, based on how occluded it is, and any receding or recessed areas are shaded darker (or lighter).
Process	The **Process** layer offer some basic image adjustment options that affect any texture layer directly below it in the **Shader Tree**, including image map, procedural and gradient texture layers.
Variation Texture	The **Variations Texture** layer is a means to apply color or value variations to multiple particle objects, surfaces, or items.
Vertex Map Texture	The **Vertex Map Texture** allows the assignment of certain Vertex Maps for use in the **Shader Tree**, specifically vertex RGBA color maps and vertex Weight Maps.

Special Textures	
Fur Material	The **Fur Material** texture layer covers surfaces with a range of fur-related textures (fuzzy, frazzled, fluffy, fibrous - all manner of fur options).
MatCap Shader	The **MatCap Shader** texture creates a surface material and lighting environment simply by painting an object, so that it looks like how you want the surface to appear.
Material Presets	The **Material Preset** layer references a material preset from the Preset browser.

Special Textures	
Shader	The **Shader** texture is a "base shader" item that reads the information from the entire **Shader Tree** and computes the effects of the illumination on all the surfaces in the scene, making the **Shader** the most important layer in the tree.
Surface Generator	The **Surface Generator** texture works in tandem with a Replicator Item to offer a greater degree of control over the random placement and density of replicated objects.

Textures	
Cellular	The **Cellular** texture is a procedural texture that is mathematically created at render-time to create a cellular pattern.
Checker	The **Checker** texture is a procedural texture that is mathematically created at render-time to create a checkerboard pattern with two colors or values.
Dots Procedural	The **Dots** texture creates a grid-like pattern of dots across the surface of the mesh.
Gabor Noise	The **Gabor Noise** texture is a procedural texture that is mathematically created at render-time to generate patterns that can easily be stretched and rotated, like brush strokes.
Grid Procedural	The **Grid** texture is a procedural texture that is mathematically created at render-time to create a grid pattern between two value or color settings: the Line Value/Color and the Filler Value/Color.
Noise Procedural	The **Noise** texture is a procedural texture that is mathematically created at render-time to modulate between the two colors/values, based on the particular algorithm and user settings.
Poisson Cells	The **Poisson Cellular** texture is a procedural texture that is mathematically created at render-time to modulate between the **Cell** and **Filler** colors/values, based on the type and user settings.

Textures	
Ripples Procedural	The **Ripple** texture is a procedural texture that is mathematically created at render-time to simulate raindrops on a puddle, creating a series of concentric circles that originate within a given area, and continue outward toward infinity.
RT Curvature	The **RT Curvature** shader is a method of shading surface discontinuity. This type of shading is most useful in enhancing surfacing details, both modeled and that which is added by displacement. The shading can be enhanced further when used to modulate other types of shading effects.
Weave Procedural	The **Weave** texture is a procedural texture that is mathematically created at render-time to simulate woven yarn in a basic basket-weave pattern. (A series of parallel rows, combined with columns that thread over-and-under in alternating rows to form the basic pattern.)
Wood Procedural	The **Wood** texture is a procedural texture that is mathematically created at render-time to simulate wood grain.

Texture Locators	
Texture Locator	The **Texture Locator** shader is automatically generated and added to the items list and linked by the texture layers **Locator** property whenever a texture layer is created.

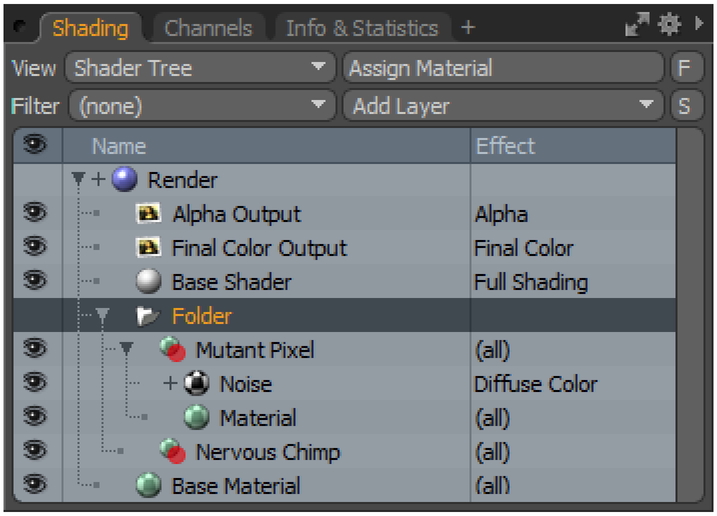
Folder

A **Folder** item acts solely as an organizational device for the **Shader Tree**. Any number of layers can be placed into a folder. You can add a folder to the tree using the **Add Layers** button, and selecting the **Special** > **Folder** option from the pop-up menu.

By default the **Folder** item is added to the top-most level under the nearest Material Group Item layer (based on what's currently selected in the tree). Once the **Folder** has been added, any number of layers can be selected, then dragged and dropped onto the folder, thus, adding them to it. The arrow icon preceding the

Folder can be toggled to tuck away the content of the folder, and the **Folder** itself can be renamed. The **Folder** item has no associated attributes or properties.

For more on working with **Shader Tree** layers, please refer to the Shader Tree topic.



Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Folder	
Mutant Pixel	(all)
Noise	Diffuse Color
Material	(all)
Nervous Chimp	(all)
Base Material	(all)

Material Group

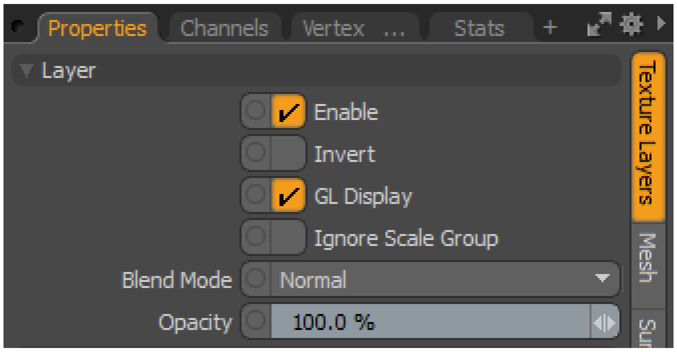
Material groups are containers that hold any of the various Shader Tree items, such as shaders, materials, textures, and even render outputs. These containers can be useful for simply organizing the Shader Tree, but are incredibly powerful when used to control how surfaces are applied to polygons at render time. Referred to in the **Add Layer** function of the Shader Tree as a **Group**, in this topic, the term used is Material Group to avoid confusion with the Groups Viewport.

Surfacing in Modo is separated from the geometry. Individually, Shader Tree item layers affect an entire scene, unless you limit them to specific areas using a mask. The masks themselves are procedurally created, and defined by Polygon Tags. Tags are simply bits of information tied to a polygon that associate it with a particular grouping. The most common tag is a **Material** tag. When applied to any polygon selection using the **M** shortcut or **Texture** > **Assign Material Group...** command, it automatically creates an appropriately named **Material Group** layer in the Shader Tree, with the proper tag specification and a **Material** sub-item. Once created, changes made to the various layers under the **Material Group** only affect the polygons with that tag. Since polygons can only have one type of material tag at a time, additional layer masks can be applied using Selection Sets, polygon Parts, and item masks.

These masks that the material groups define are based on the settings under the **Group** header. They can be used individually, or in combination, and while the functions may appear simple, they are an efficient means of surfacing a scene. Understanding what each does is an important step in understanding the Shader Tree itself and realizing the power that it provides.

Material Group Properties

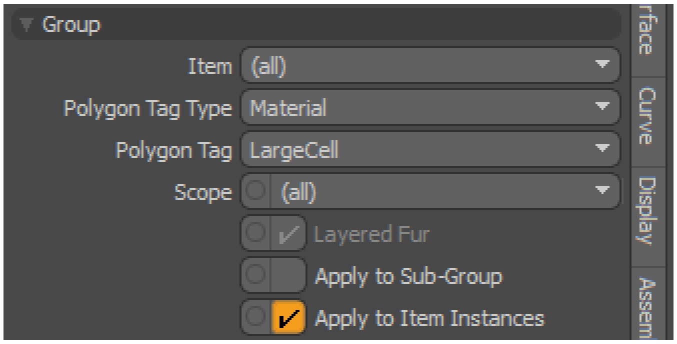
Layer Options



☑ Enable
☐ Invert
☑ GL Display
☐ Ignore Scale Group
Blend Mode Normal
Opacity 100.0 %

Enable	Toggles the effect of the group (and its contained layers) on and off, duplicating the functionality of toggling visibility in the **Shader Tree**. When un-checked (disabled), the layer has no effect on the shading of the scene. However, disabled groups are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the group, producing a negative effect.
Blend Mode	Affects blending between different groups, allowing you to stack several groups for different effects. For more on blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current group. Reducing this value increasingly reveals lower layers in the **Shader Tree**, if present, or dims the effect of the layer itself on the surface.

Group Options



Item (all)
Polygon Tag Type Material
Polygon Tag LargeCell
Scope (all)
☑ Layered Fur
☐ Apply to Sub-Group
☑ Apply to Item Instances

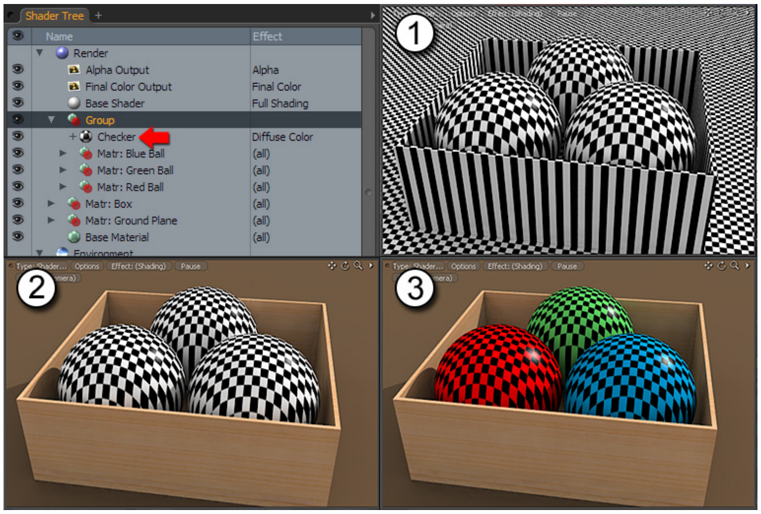
Item	Limits the application of the layers within a material group to a specific Mesh Item (these are the Mesh Item layers of the Items (Scenes) List). When defined, these are considered an **Item Mask**. The **(all)** setting applies the materials to the entire scene. This setting can also be used in combination with a polygon tag, further limiting the surface.
Polygon Tag Type	Specifies the type of polygon tag. Several types are possible and each limits or masks the application of the shading items contained within the group to only the polygons tagged as such: • **Material** - The most common polygon tag type. Material groups with this type of tag are automatically created when using the **M** keyboard shortcut. Polygons can only have one **Material** tag at a time. • **Part** - Similar to a selection set but, like a material tag, a polygon can only belong to one **Part** group at a time. • **Selection Set** - Useful for recalling previous polygon selections. However, when specified here as a tag type, the selection set masks the material groups layers to the associated polygons of the specified selection set. One polygon can belong to many different selection sets.
Polygon Tag	When any type of tag is created, you are presented with a dialog box requesting a name. All tags of the defined type contained in the scene appear here, where you can use this dropdown to select the particular named tag you wish to use.
Scope	The effect of the material group's layers can be further limited to only **Fur** materials, polygonal **Surface**, or **All**. As a fur material typically derives its surfacing from the underlying polygons (technically, from the surface's material item layer), duplicating a material group and setting the scope to **Fur** allows you to specify surfacing attributes exclusive to the fur layers. This is how it's possible to have green grass growing from a brown dirt surface.

Layered Fur	When enabled, you can have multiple Fur item layers with different settings on a single surface.
Apply to Sub-Group	Using undefined Material Groups (when no tags are specified) can be a convenient way to organize **Shader Tree** layers. With the addition of the **Apply to Sub-Group** option, they can also be an effective way to apply a setting to a number of material groups all at once.
Apply to Item Instances	When enabled, the Material Group is applied to all instances of the original item.

Apply to Sub-group Example

In the example below, there are three colored spheres, a box, and a ground plane. In the **Shader Tree**, you can see that the three colored spheres are grouped together into an undefined **Material Group**. Directly below the group parent, a checkerboard texture has been added (next to the red arrow). Since it is above all the other layers in the tree, and the group above it is undefined, it affects the entire scene, as illustrated by example one.

By simply enabling the **Apply to Sub Groups** option to the top **Material Group** layer, the checkerboard effect is limited to those material groups that are descendants of the group, as illustrated by example two. By setting the checkerboard pattern to **Normal Multiply** as the blending mode, you can quickly see how powerful the **Apply to Sub-Group** option can be for quickly and easily applying a single texture layer to multiple surfaces, as illustrated in example three.

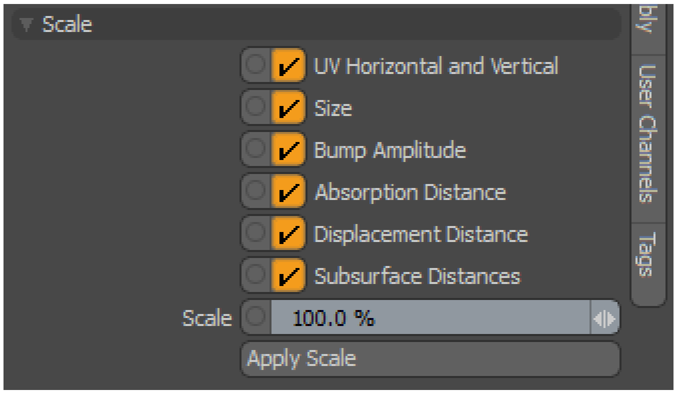


Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Group	
Checker	Diffuse Color
Matr: Blue Ball	(all)
Matr: Green Ball	(all)
Matr: Red Ball	(all)
Matr: Box	(all)
Matr: Ground Plane	(all)
Base Material	(all)
Environment	

Tip: As item layers and materials add up in the **Shader Tree**, it can quickly become a long list that gets confusing and tedious to edit. While there are many functions that speed up and simplify the task of locating certain layers (such as the material selection mode), good **Shader Tree** organization is important.

You can select multiple related layers and press the keyboard shortcut **Ctrl**+**G**, which automatically groups together the selected items. In this case the material group creates no mask for shading, but is simply used as an organizational tool. Renaming the layer further helps in identifying it in later sessions.

Scale Options



Manually scaling individual texture layers in the Shader Tree may be difficult. The separate layers might not contain the same relative scale or tiling, which means you have to figure out what the relative scaling is. These parameters allow you to scale all layers within a material relative to one another in the Shader Tree.

The items you can scale here are:

- **UV Horizontal and Vertical** - Scales the Horizontal and Vertical Wrap values when the layer's **Projection Type** is set to **UV Map**.
- **Size** - Scales the size when the layer's **Projection Type** is not set to **UV Map**.
- **Bump Amplitude** - Modifies the amount of bump on the surface.
- **Absorption Distance** - Colored transparent surfaces don't often exhibit even coloration. Thin areas appear colorless while thicker areas are tinted with color. **Absorption Distance** determines how far a ray must travel to get 100% of the Transparency Color.
- **Displacement Distance** - Determines the maximum range that vertices can be created, or moved away, from the original mesh surface during micro-polygon tessellation.
- **Subsurface Distance** - Determines how far light must travel through the surface before exiting in order to be fully tinted by the subsurface color.

Note: For more details on each of these properties, see Material.

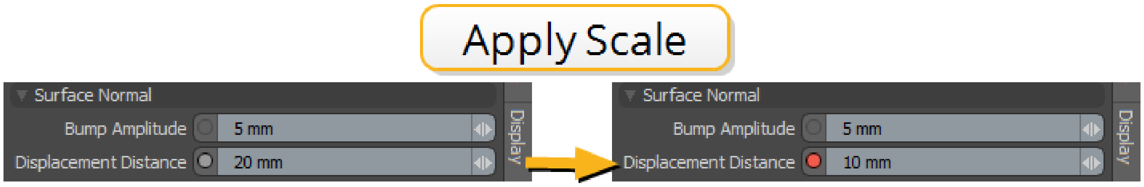
Enable the layers you want to scale and specify the **Scale** amount. Your render is updated with the new **Scale** value. By default, all layers are scaled to 100%.

This example shows the gray snake skin leather AxF texture scaled:

Scale at 100%

Scale at 50%

To bake the scale value into the selected attributes of the material, click **Apply Scale**. This modifies the values on the material. For example, scaling **Displacement Distance** to 50% halves the material's **Displacement Distance** value.



Surface Normal
Bump Amplitude 5 mm
Displacement Distance 20 mm

Surface Normal
Bump Amplitude 5 mm
Displacement Distance 10 mm

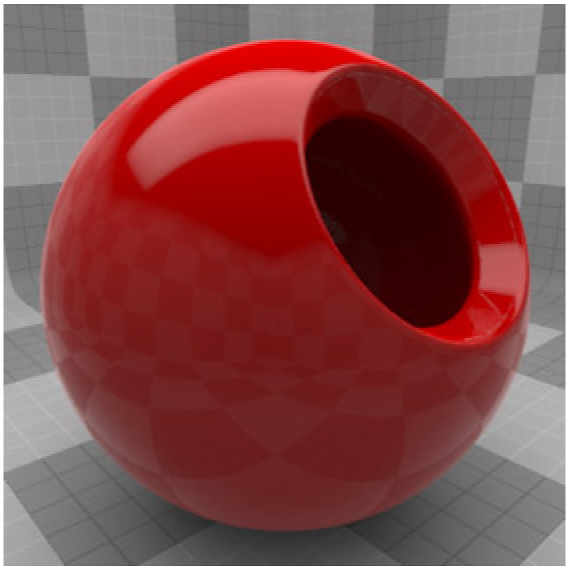
Note: You can still scale individual texture layers manually, independent of the Group's **Scale** options. To do this, select the texture layer in the Shader Tree, and in the **Layer** properties, enable **Ignore Scale Group**.

Material

The **Material** item is the default general purpose Shader Tree layer that controls all the standard shading attributes for a surface's final rendered look. The Material item contains channels to control **Diffuse**, **Specular**, **Reflections**, and **Transparency**, among other surfacing attributes.

By default, every new scene contains a single material layer called the **Base Material** that effectively shades the entire scene until material tags are defined, limiting materials and shading to specific surfaces.

For more information about assigning material tags, see Polygon Set Material/Part/Smoothing.

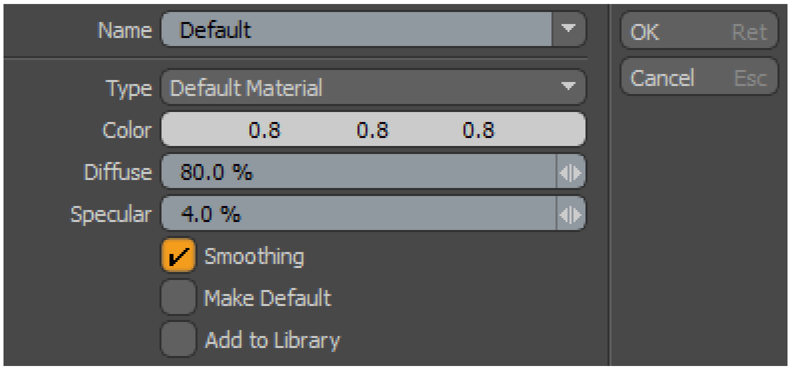


Applying a Material

The following procedure applies a material to a selected mesh item, creating a material tag.

1. Select a mesh item and press **M**.

   The **Polygon Set Material** dialog displays.



Name Default

Type Default Material

Color 0.8 0.8 0.8

Diffuse 80.0 %

Specular 4.0 %

☑ Smoothing

☐ Make Default

☐ Add to Library

2. In **Name**, type a name for the material.
3. Click the **Type** dropdown menu and select one of the following options:
   - **Default Material** - Set to use the default Material properties.
   - **Unreal Material** - Set to use the Unreal Material properties.
   - **Unity Material** - Set to use the Unity Material properties.

Note: Modo remembers the material type selected, and automatically sets **Type** to that material the next time you open the dialog.

4. Click the **Color** field to change the color.

Tip: You can also use the Eyedropper tool to pick any color on your screen. See Using Standard Tool Controls for more information.

5. Set the **Diffuse Amount** and Specular Amount percentages.
6. Enable any of the following options:
   - **Smoothing** - Displays the texture in more detail in the 3D viewport. 100% Smoothing is enabled by default.
   - **Make Default** - Sets as the default material for all new materials.
   - **Add to Library** - Saves your material to the Shader tree under the **<Name>** layer. For example, if you named your material My Blue, a new **My Blue** > **Material** layer is created.

Tip: You can save your custom material for use in other projects. In the Shader tree, right-click on the material item, and select **Save Preset...**. For more information, see Preset Browser.

7. Click **OK**.

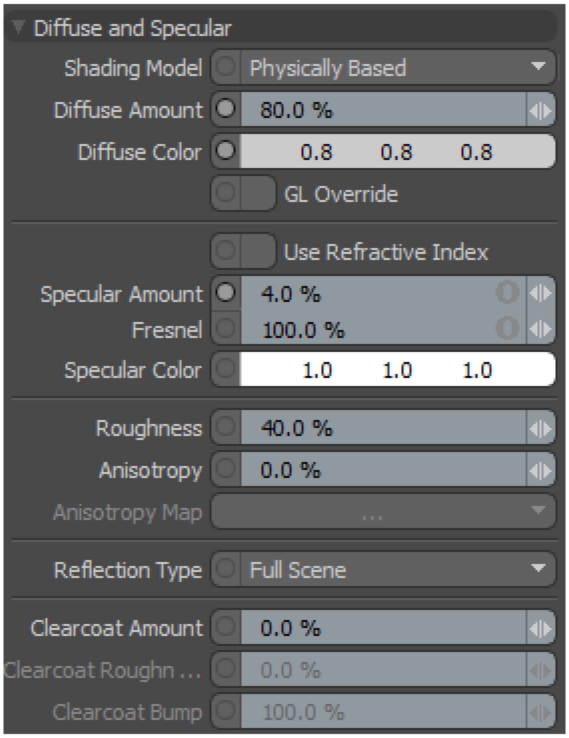
Material Properties

The Material layer's property form is divided into three sub-tabs:

- Material Ref (Reflective) - all material attributes that impact how much light is reflected from the surface, such as **Diffuse**, **Specular**, and **Mirror** reflected light.
- Material Trans (Transmissive) - controls how light is transmitted through a material such as **Transparency** and **Subsurface Scattering**.
- Material Rays - Ray Tracing - used when controlling rays for clipping and **Volumetric** rendering.

In addition to these sub-tabs, nearly all material properties can be controlled through a texture bitmap or procedurally by creating a new texture layer and setting the desired **Effect** you wish to control. For information more on adding layers and setting their **Effect**, please see Shader Tree.

Material Ref - Diffuse and Specular



Diffuse and Specular

Shading Model: Physically Based

Diffuse Amount: 80.0 %

Diffuse Color: 0.8 0.8 0.8

☐ GL Override

☐ Use Refractive Index

Specular Amount: 4.0 %

Fresnel: 100.0 %

Specular Color: 1.0 1.0 1.0

Roughness: 40.0 %

Anisotropy: 0.0 %

Anisotropy Map: ...

Reflection Type: Full Scene

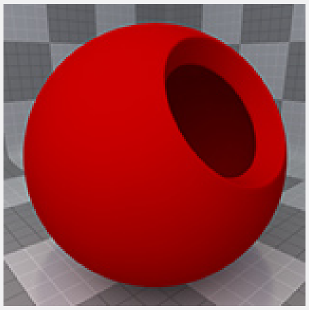
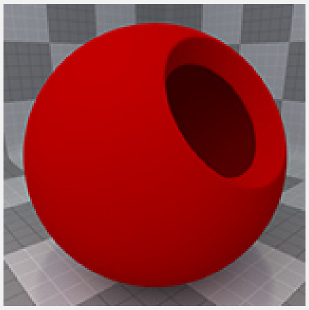
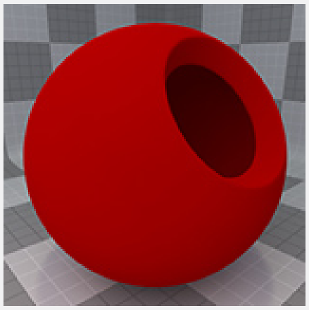
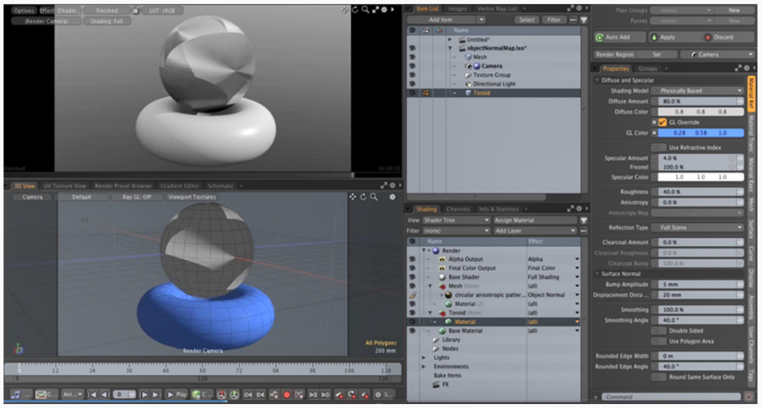
Clearcoat Amount: 0.0 %

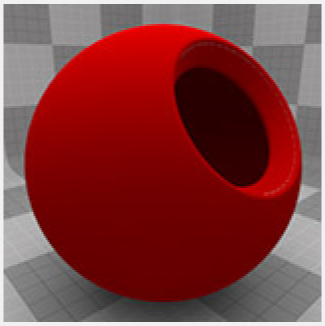
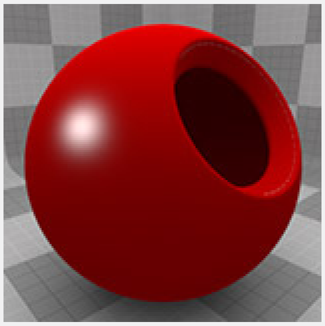
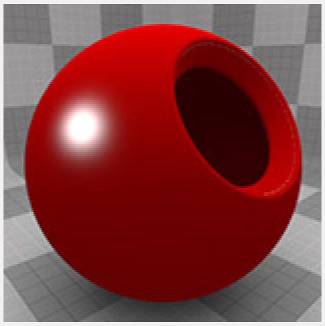
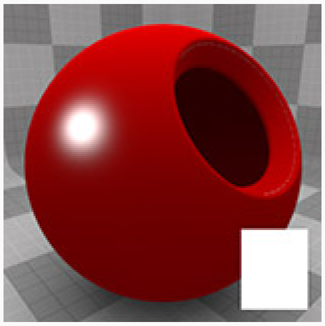
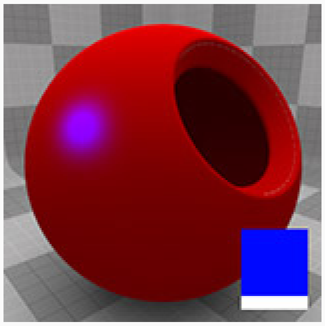
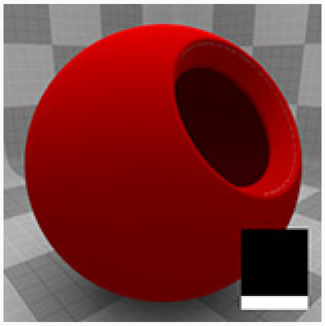
Clearcoat Roughn ...: 0.0 %

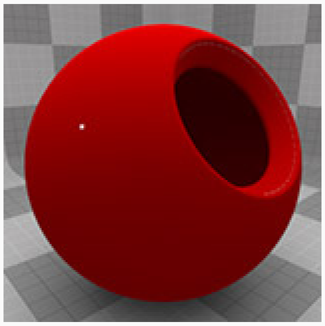
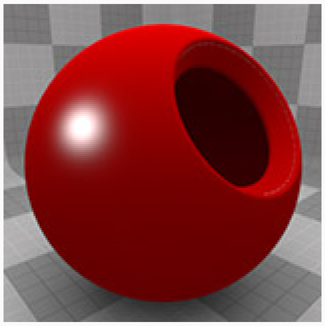
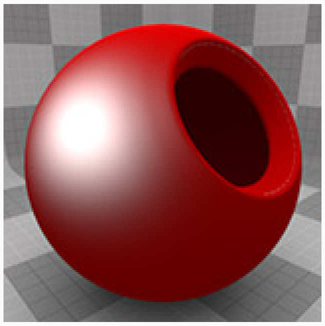
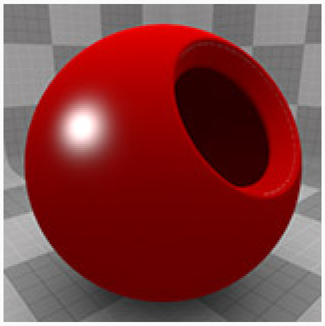
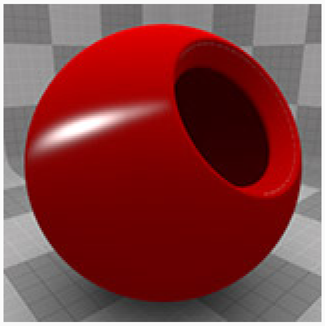
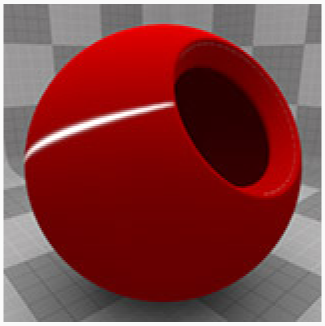
Clearcoat Bump: 100.0 %

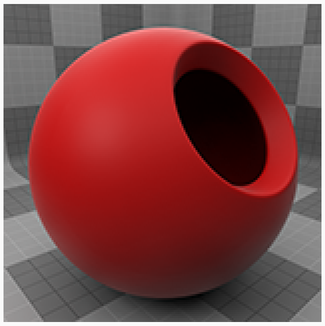
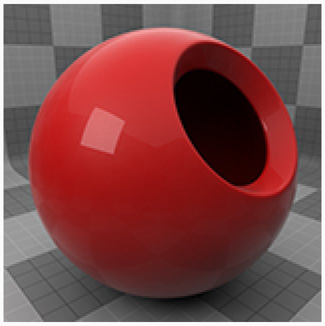
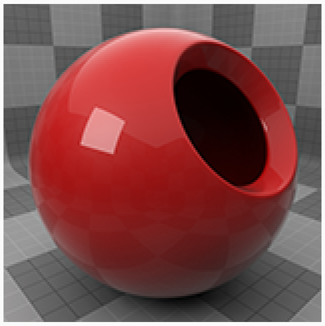
Shading Model	Controls how light and surfacing attributes are evaluated for final rendering. These shading models are also know as bidirectional reflectance distribution functions, or BRDFs. Each shading model subtly changes how specular, Fresnel, reflection, and illumination falloff are computed during rendering: • **Physically Based** - exhibits the most faithful highlight shape and Fresnel falloff shading and is based on the Generalized Trowbridge-Reitz BRDF model, popularized by a number of famous movie studios because of its ability to produce realistic surface shading. • **Traditional** - allows you to match surfaces to those rendered using Modo's legacy BRDF, based on the on the Phong-Blinn model. • **Energy Conserving** - automatically controls the diffuse/specular/reflection mix to produce realistic shading results, but for most cases it is superseded by the Physically Based model for its superior results. This is another legacy shading model used when the **Conserve Energy** option was enabled. It is based on the Ashikhmin-Shirley BRDF. This option is available for some materials. For example, see Conserve Energy.

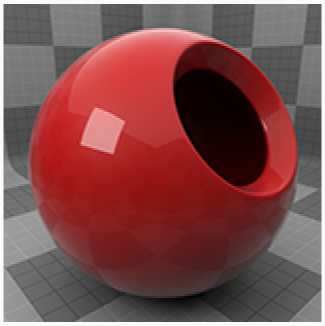
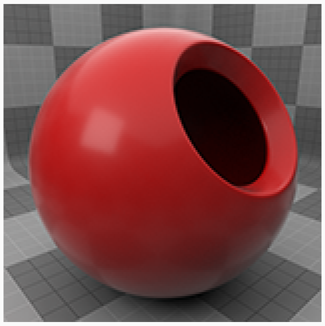
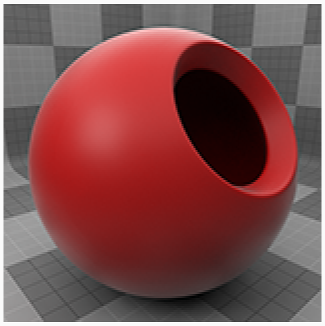
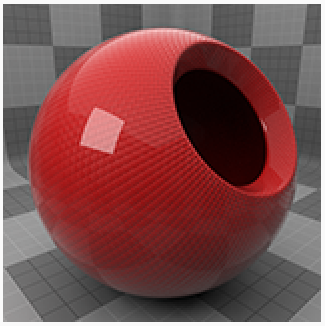
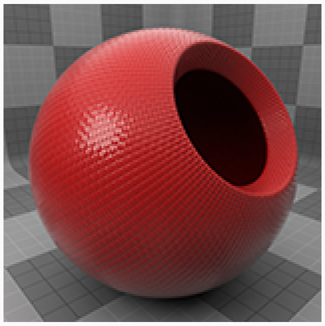
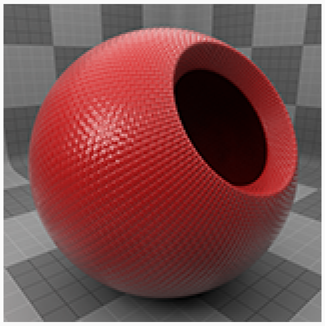
	• **Principled** - allows you to use physically principled BRDF properties.
Diffuse Amount	In the real world, all surfaces absorb certain wavelengths of light while reflecting others. The light that is reflected is what we perceive as the color of the object. This is known as diffuse light. The **Diffuse Amount** attribute acts as a multiplier to the **Diffuse Color** setting determining how much of that light is reflected from, or absorbed by, the surface. A setting of 100% returns the same value defined as the **Diffuse Color** value when lit by a pure white light at 100% intensity. A **Diffuse Amount** of 50% appears half as bright under the same light as it would only returning half of the light bounced from the surface. All surfaces, no matter how shiny absorb at least a small amount of light, so the default value is defined as 80%.
Diffuse Color	This setting defines the color of the surface when lit by a pure white light source, multiplied by the **Diffuse Amount**. This attribute is effectively the color of your object, representing the light that bounces from the surface. Diffuse colors can also be driven beyond traditional color space into high dynamic range values by applying **Diffuse Amounts** over 100%, but they will not be physically plausible at that point.
Diffuse Roughness	Extremely matte surfaces, such as stone, concrete, or plaster, reflect light very differently than smooth or glossy surfaces, such as glass or plastic. This is because, on a microscopic level, the matte surface is actually quite rough, spreading the light outwards making the surface appear more flat. **Diffuse Roughness** aims to simulate this microscopic roughness effect by modifying the standard shading model, producing a more believable matte surface. A value of 0% disables the effect, while values above zero increasingly produce a rougher, more matte surface. Since **Diffuse Roughness** simulates a lack of Specular, it is best applied when the **Specular Amount** is 0% (see image examples below). **Note:** The **Diffuse Roughness** attribute is only enabled for the **Traditional** and **Energy Conserving** legacy shading models.

	 Diffuse Roughness 0%  Diffuse Roughness 50%  Diffuse Roughness 100%
GL Override	When enabled, uses the defined **GL Color** values to change the color of specified polygons displayed in the Camera View without affecting the rendering in the Preview Viewport. This is useful when working on complex models containing a number of parts, which you want to apply different colors to. 
Use Refractive Index	When enabled, use the defined **Refractive Index** value to set the specular/reflectivity amount for a surface. The main benefit being that the **Refractive Index** is a commonly measured value posted in several online databases used to define the physical accuracy of surfaces.
Specular Amount	In the real world specular highlights are the reflections of very bright light sources on a surface. How these reflections spread over a surface tells you a lot about that surface, whether it is shiny or dull, smooth or rough. Modo simulates this reflection with these five settings: • Specular Amount • **Fresnel** • Specular Color • Roughness

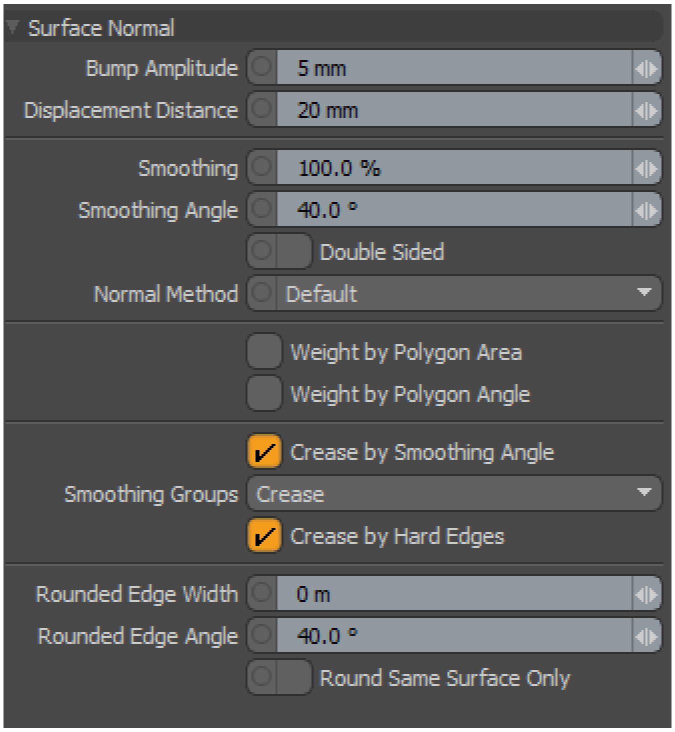
	- Anisotropy Generated only from direct light sources, the **Specular Amount** is multiplied with the **Specular Color** to create the overall rendered result. Increasing the value results in a more intense highlight while decreasing the value mutes the effect. When this value is 0%, the **Specular Color**, **Roughness**, and **Anisotropy** controls are disabled and specularity is not calculated.
	 Specular Amount 0%  Specular Amount 50%  Specular Amount 100%
Fresnel	At glancing angles (any surface that is near perpendicular to the direction the camera is facing) many types of surfaces increase their reflectance values. The **Fresnel** setting increases the specular amount at these glancing angles to produce a physically plausible result.
Specular Color	Defines an RGB value that tints the specular highlight.
	  
Roughness	Surfaces can be very different from each other, such as a rubber ball, a plastic cup, and a terracotta brick. These are all seemingly smooth surfaces, but on a microscopic scale their surfaces are very different. The microscopic surface variations has a direct effect on the way the light reflects off of each surface. The **Roughness** setting simulates these microscopic differences making surfaces

	appear more dull or rough, or shiny and smooth. It also affects both the Specular highlight as well as ray traced and environmental reflections (reflection roughness only occurs when the Material option is enabled). Increasing this value widens the reflection, creating a much broader specular highlight. When used with blurry reflections, a very high value yields a very distorted reflection.
	 Roughness 0%  Roughness 50%  Roughness 100%
Anisotropy	Millions of microscopic scratches on a surface distort highlights and reflections over their surface parallel to the directions of the scratches. The **Anisotropy** value affects the highlights, simulating this effect. Anisotropy is controlled by a UV map or further controlled by a texture maps when the layers effect is set to the **Anisotropy Direction**. It can be caused by many man-made processes, such as machining metal and also appears in textiles where shiny threads spread the highlight over a surface.
	 Anisotropy 0%  Anisotropy 50%  Anisotropy 100%
Anisotropy Map	The UV map dropdown allows you to choose which map determines the anisotropic direction. The **U** direction (horizontal) defines the direction of the virtual scratches for positive **Anisotropy** values.
Reflection Type	Allows you to choose between **Full Scene** reflection evaluation or **Environment**

	Only. This setting effectively switches between ray traced reflections and an environment map technique. The following options are available: • **Full Scene** - More accurate at the expense of greater render times. • **Environment Only** - Utilizes only the settings in the Environment Item.
Clearcoat Amount	Allows you to easily add an additional un-blurred and un-tinted reflective layer to a surface, simplifying the simulation of multiple layers of paint, which can be especially useful for automotive surfaces. Many metallic colors spread a wide colored highlight over the surface and yet, still exhibit a smooth reflective layer as a result of an additional transparent layer called a Clearcoat. Setting **Clearcoat Amount** to 0% effectively turns the function off, increasing values add the reflective effect. We recommend that, in most cases, you'll just want to leave this setting at 100%.
	 Clearcoat Amount 0%  Clearcoat Amount 50%  Clearcoat Amount 100%
Clearcoat Roughness	Blurs the reflection shading added by Clearcoat, working the same as the normal **Roughness** setting, but specific to just the Clearcoat Amount for independent control. **Note:** The **Clearcoat Amount** must be above 0% to see any roughness changes.

	 Roughness Amount 0%  Roughness Amount 50%  Roughness Amount 100%
Clearcoat Bump	Defines the degree the Clearcoat shading ignores for any bump mapping that has been applied to the surface.
	 Clearcoat Bump 0%  Clearcoat Bump 50%  Clearcoat Bump 100%

Material Ref - Surface Normal



Surface Normal

Bump Amplitude 5 mm

Displacement Distance 20 mm

Smoothing 100.0 %

Smoothing Angle 40.0 °

☐ Double Sided

Normal Method Default

☐ Weight by Polygon Area

☐ Weight by Polygon Angle

☑ Crease by Smoothing Angle

Smoothing Groups Crease

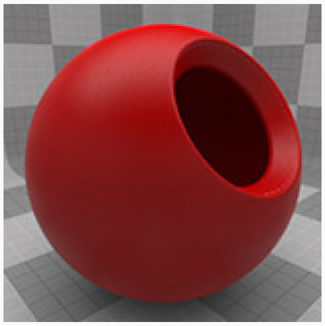
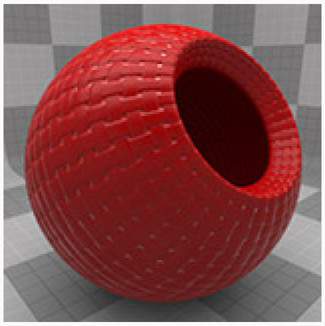
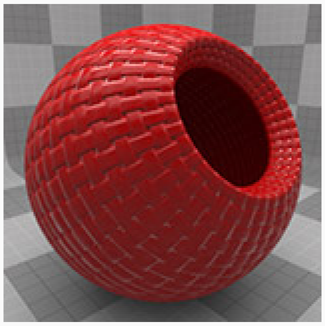
☑ Crease by Hard Edges

Rounded Edge Width 0 m

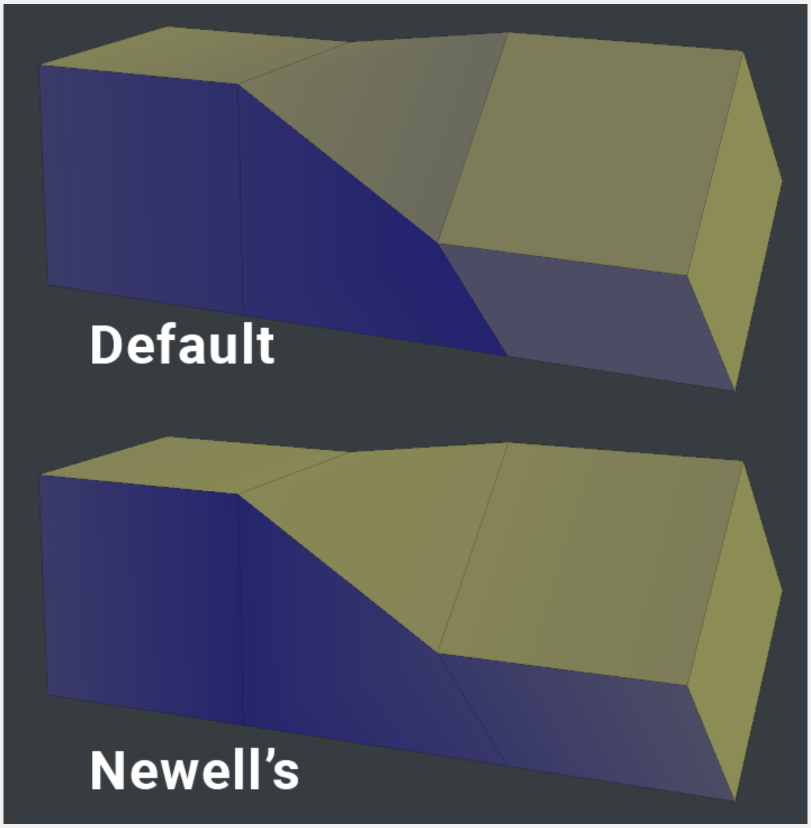
Rounded Edge Angle 40.0 °

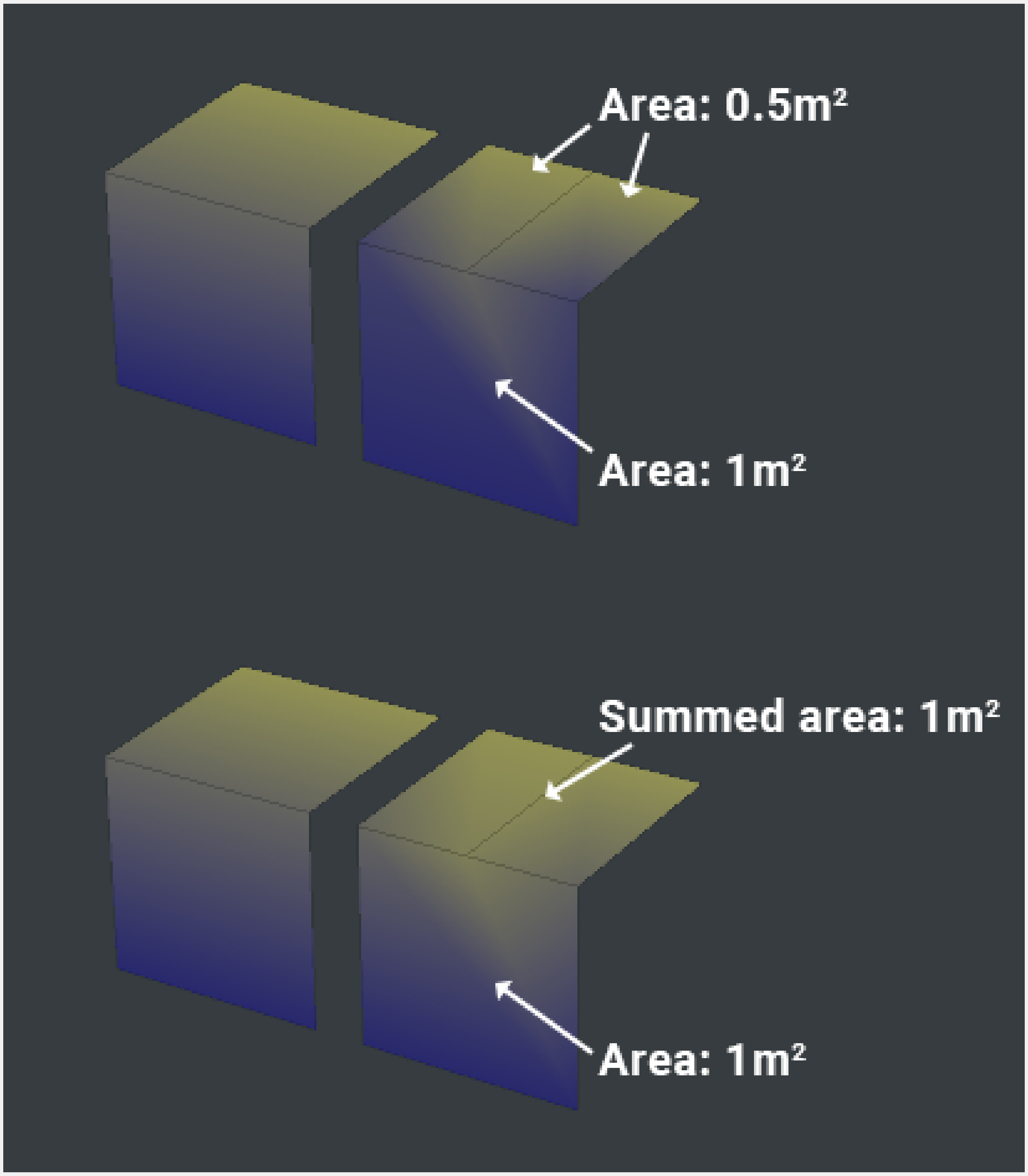
☐ Round Same Surface Only

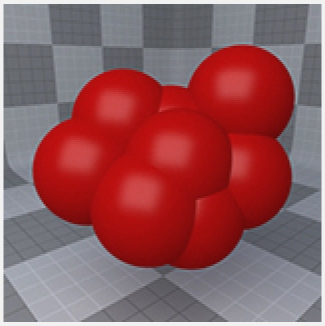
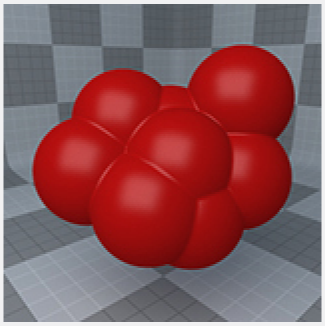
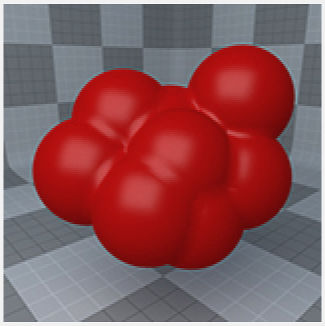
Bump Amplitude	Bump maps are a means of creating high frequency details on a surface that would be difficult, if not impossible, to model. Bump mapping itself is a technique that modifies a ray's direction when shading a surface, giving the illusion that a surface has more detail than it actually does. **Bump Amplitude** allows a single control per material for modifying the amount of bump on a surface, defined as a measurement. The size defined is relative to the surfaces actual size. The actual bump details come from the texture layer(s) driving the effect with the **Bump Amplitude** setting acting as a multiplier to create the final result. Please see Shader Tree for information on adding layers and setting their effects.

	 Bump Amplitude 0 mm  Bump Amplitude 50 mm  Bump Amplitude 100 mm
Displacement Distance	Similar to Bump maps, Displacement is a means to add detail to a surface that would be difficult, if not impossible to model otherwise. The maps themselves can be generated in Modo using the sculpting toolset, or in an external application such as ZBrush™ or Mudbox™; even procedural textures are useful as Displacement sources. The **Displacement Distance** setting determines the maximum range that vertices can be created, or moved away, from the original mesh surface during micro-polygon tessellation. With a value of 2 cm, no vertex can be displaced more than 2 cm away from the original surface. Depending on the mesh's scale, this value may need to be increased significantly. When textures are applied to the displacement channel, should the combined texture value become greater than 100%, the resulting surface appears to be clipped. The maximum displacement distance is different from other material parameters because the renderer needs to know it way in advance, before any textures are evaluated. At the start of rendering, all surfaces are just empty bounding boxes. The vertices and polygons of a particular surface are only created in the geometry cache when a ray pierces the bounding box of the surface. For displaced surfaces, these bounding boxes must be padded with the maximum displacement distance in order to avoid holes, since displacement can cause geometry that would normally appear in a later bucket to show up in an earlier one. For this reason, the actual distance a vertex moves can never exceed the **Displacement Distance**, therefore texture values outside the range of -100% to 100% are always clipped.
Smoothing	A shading technique giving the illusion that geometry has more detail than it actually does. Without smoothing, geometric faces can appear faceted or flat. The **Smoothing**

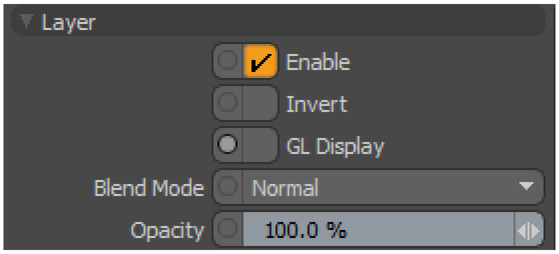
	amount is set as a percentage, rather than a simple toggle, so that you have the flexibility to fade in the smoothing amount, or even drive it above 100% or below 0% to create interesting effects. **Tip:** At values over 100%, the smoothing effect results in pillowing around each polygon.
Smoothing Angle	Determines which faces should receive smoothing. The **Smoothing Angle** sets a maximum angle tolerance between two polygons based on their normals. If the angle between the polygons is greater than the smoothing angle, no smoothing occurs. This is useful so that hard edges, such as cubes, are not smoothed across. The default value of 40° is a good general case value.
Double Sided	When enabled, shading occurs from both sides of a polygon regardless of the normal direction. By default, this control is disabled so that polygons that do not face the camera seem invisible, as they only have one side. **Note:** In some cases **Double Sided** polygons can cause render errors with ray tracing of shadows and other effects. The **Double Sided** flag should be used sparingly, and never to correct poorly modeled meshes.
Normal Method	Specifies how to generate the geometric face normals. • **Default** -Derive the normal from the first three vertices of the polygon. • **Newells** - Calculates a nicer geometric normal. More expensive to compute but can look better for non-planar polygons.

	
Weight by Polygon Area	Changes the influence polygons have on the normals of the vertices. When enabled, the Sum Coplanar Polygon Areas option is displayed.
Sum Coplanar Polygon Areas	Adds up the effective areas of contiguous coplanar polygons. Effectively pretending that connected polygons which are more are less flat are one large polygon when calculating their area.

	
Weight by Polygon Angle	Similar to how Weight by Polygon Area works, which gives more influence over vertex normals to larger polygons, this option gives more influence to polygons which form a larger angle at that vertex.
Crease by Smoothing Angle	Toggles creasing the shading of the mesh by the material's smoothing angle. Disabling this option has a similar effect to setting the material angle to 180 degrees.

Smoothing Groups	Selects how to deal with smoothing groups. • **Off** - Ignores Smoothing Groups entirely. • **Crease** - Smoothing is split between polygons which don't share any smoothing groups. This is the default. • **Force** - Smoothing is split between polygons which don't share any smoothing groups and smoothing is blended between polygons with matching smoothing groups, overriding any creasing that may come from the smoothing angle.
Crease by Hard Edges	Toggles splitting the smoothing of the mesh by any hard edges. Disabling it ignores hard edges entirely.
Rounded Edge Width	Adjusts the rendered result of surface normals at a polygon's edge to blend them with adjoining polygon normals, giving the impression of a small rounded edge between the two intersecting faces. The widths of rounded edges can be controlled by image maps, procedural textures, or gradients. The value of a texture using this effect, acts as a multiplier on the edge width specified in the material properties. You can define the width to determine the amount, or roundness, but keep in mind it is merely a shading trick and won't round the edges of the actual geometry, nor change the object's silhouette in the rendered image. Generally, it is best to set the width to just a few pixels in the final rendered image.
	 Edge Width 0 mm  Edge Width 2 mm  Edge Width 4 mm
Rounded Edge Angle	Controls which edges should appear rounded based on how sharp they are. Specifically, an edge will appear rounded if the surface normals of the adjoining polygons vary by more than the **Rounded Edge Angle** setting. The default value is 40 degrees. Lowering the setting to zero degrees causes all edges to appear rounded.
Round Same Surface Only	When enabled, edge rounding is constrained to the surface within the same item layer. When disabled, rounded edges are applied to the surface regardless of item layer.

Material Trans - Layer



☑ Enable

☐ Invert

☐ GL Display

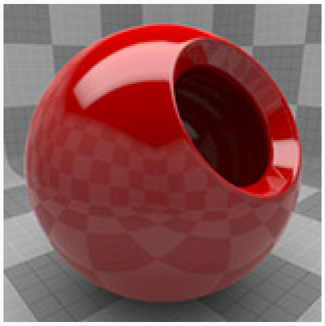
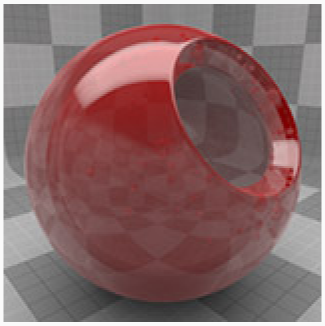
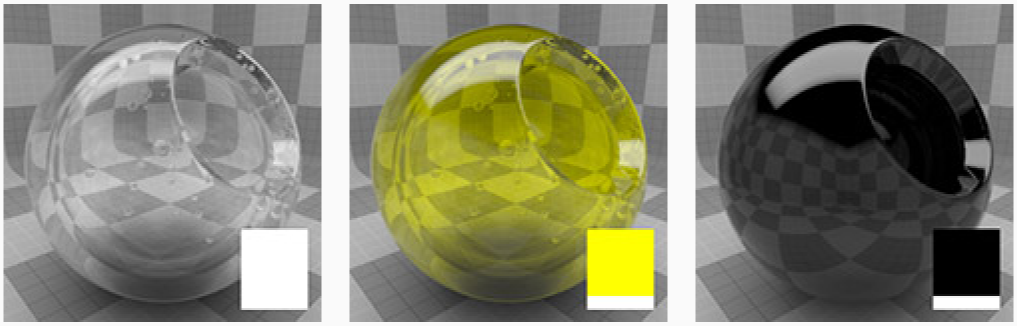
Blend Mode: Normal

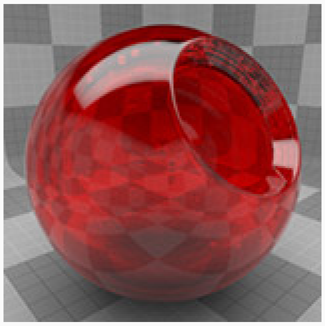
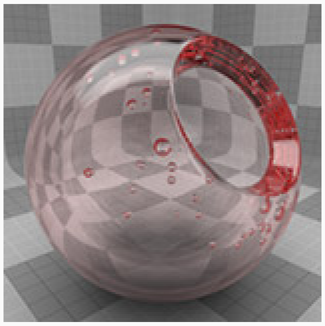
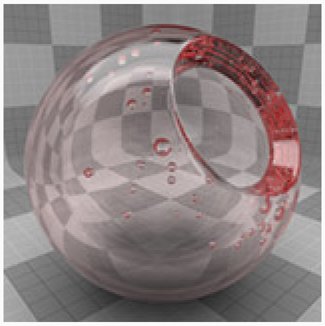
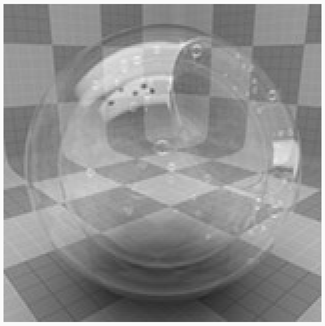
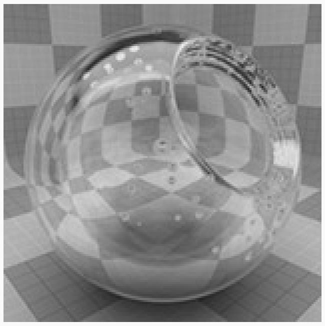
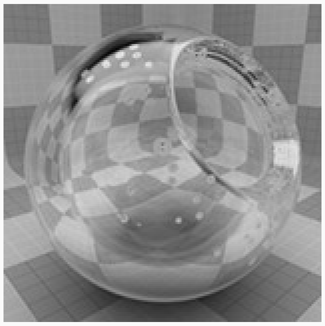
Opacity: 100.0 %

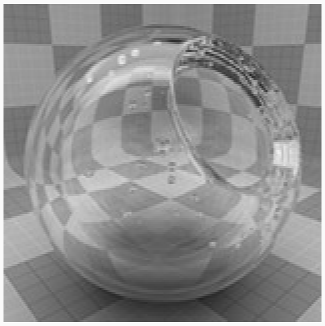
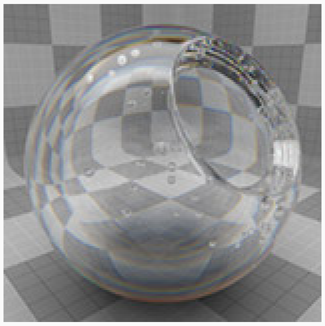
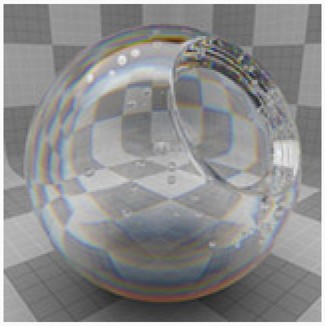
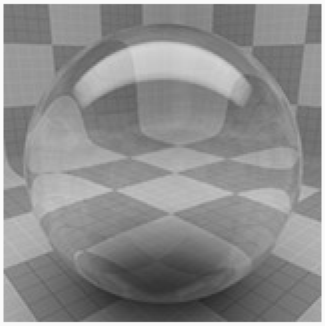
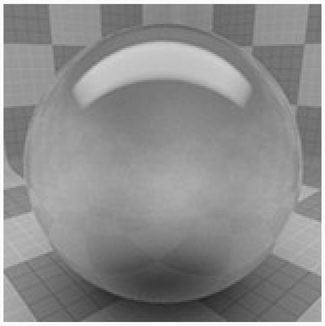
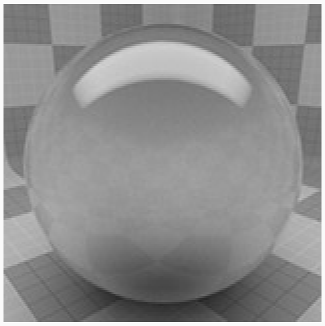
Enable	Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the layer producing a negative effect.
GL Display	Toggles the display of textures in the Camera view to improve readability and performance when working on complex scenes. When this option is disabled, textures are displayed in the Preview Viewport but not in the Camera view. For more information, see Camera Items.
Blend Mode	Affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the Shader Tree if present, or dims the effect of the layer itself on the surface.

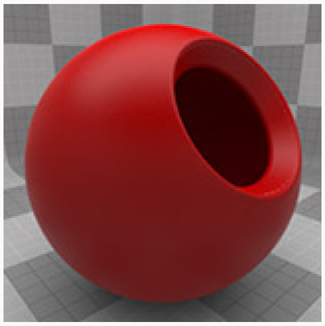
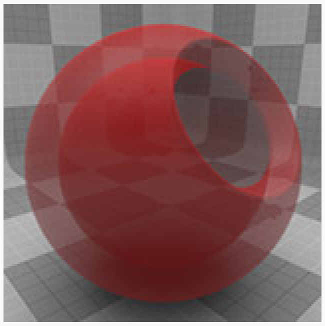
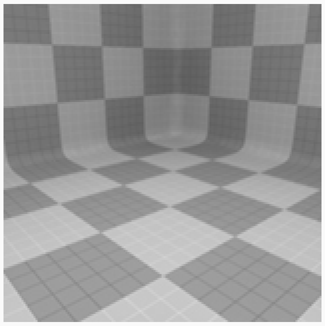
Material Trans - Transparency

Transparency Amount	Controls how transparent or opaque a surface is. 0% is completely opaque and disables the setting altogether, settings above 0% gradually increase how transparent the surface is ramping toward 100%, which is completely transparent.

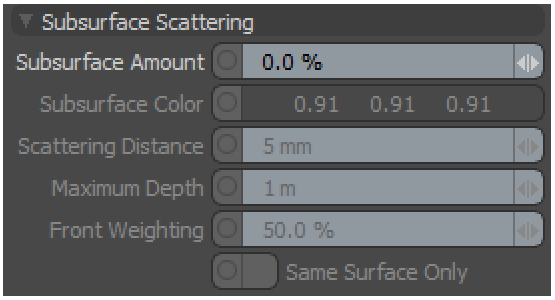
	 Trans. Amount 0%  Trans. Amount 50% Trans. Amount 100%
Transparency Color	The tint color of a ray as it penetrates the transparent surface. Since many transparent materials, such as glass, should have no diffuse amount whatsoever, the transparent color is used for tinting the material. At the default, pure white, a material that is 100% transparent becomes completely invisible unless there are reflections and/or specular highlights. By changing the transparency color, the material becomes visible as Modo simply tints whatever the camera sees behind the surface.
	
Absorption Distance	Colored transparent surfaces don't often exhibit even coloration. Thin areas appear colorless while thicker areas are tinted with color. The **Absorption Distance** controls this effect, determining how far a ray must travel to get 100% of the **Transparency Color**. The falloff is determined by Beer's Law, just as in the real world. There are two **Absorption Distance** modes: • Setting the **Absorption Distance** to values greater than 0 determines falloff using Beer's Law. • Setting the **Absorption Distance** to exactly 0 forces transmitted light to instantly takes on the transparency color rather than following Beer's Law.

	Note: The **Absorption Distance** = 0 mode should only be used on objects with no thickness, such as celluloid film or sunglasses lenses.
	 Absorption Dist. 100 mm  Absorption Dist. 300 mm  Absorption Dist. 500 mm
Refractive Index	Determines how far a light ray is bent when it travels through a material and, when **Use Refractive Index** is enabled, the index determines the amount of surface specularity and reflectivity. The higher the value the more distorted the refractive/reflective items become. For example, a value of 1.0 is equivalent to light traveling through a vacuum and does not bend the rays at all. Increasing the value to 1.333 approximates the amount of distortion that occurs when light travels through water. **Note:** This setting requires that the surface has some amount of transparency active or that the **Use Refractive Index** option is enabled.
	 Refractive Index 1.0  Refractive Index 2.0  Refractive Index 3.0
Dispersion	**Dispersion** is the difference of **Refractive Index** between various wavelengths of lights. Pure white light in the real world contains all colors of the spectrum as

	it passes through solid objects, and the refraction of different wavelengths varies creating a rainbow effect. You've likely witnessed this when shining a bright light through a prism. This effect in gemstones is usually referred to as 'fire'. In Modo, **Dispersion** is calculated as the difference between a high refractive index (at the violet end of the spectrum) and low refractive index (at the red end of the spectrum). A typical dispersion value for glass would be around .02 whereas diamond would be .044. Negative dispersion values invert to the usual curve.
	 Dispersion 0  Dispersion 0.05  Dispersion 0.1
Refraction Roughness	Some transparent surfaces are better described as translucent or cloudy. Light passes through the surface, but is diffused in such a way as to make opposing items blurry or even completely indistinguishable. Frosted glass exhibits this effect, items close to the surface are clear, but as they recede from view, edges and details soften increasingly. The **Refraction Roughness** calculates this effect by firing a number of jittered rays for each refraction ray and averages the result. A value of 0% disables the effect, whereas values above 0% increase the blurriness of items viewed through the surface.
	 Ref. Roughness 0%  Ref. Roughness 10%  Ref. Roughness 20%

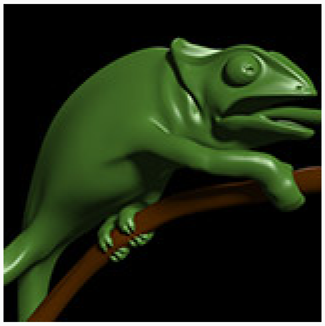
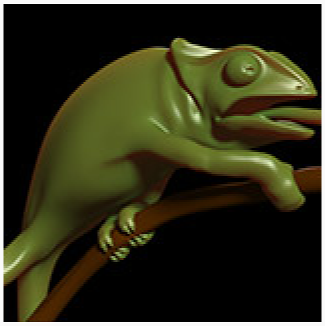
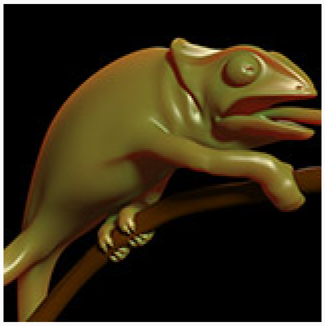
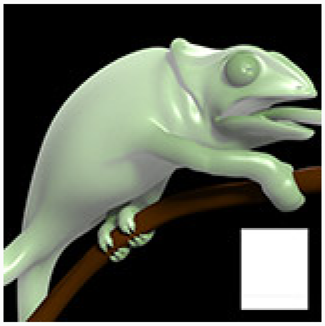
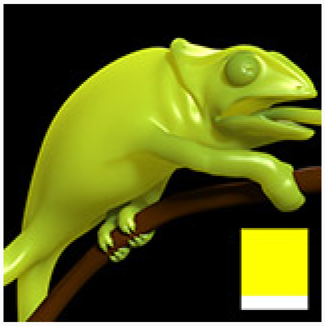
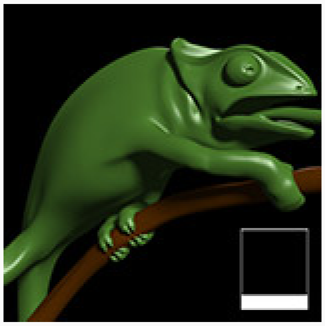
Dissolve Amount	Allows you to fade an object completely from view, with a setting of 100% making the surface completely invisible. The Dissolve Amount is different from transparency, as it is a convenient way to fade all aspects of a surface simultaneously.
	 Dissolve Amount 0%  Dissolve Amount 50%  Dissolve Amount 100%

Material Trans - Subsurface Scattering



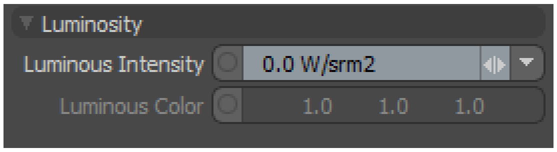
Subsurface Scattering
Subsurface Amount 0.0 %
Subsurface Color 0.91 0.91 0.91
Scattering Distance 5 mm
Maximum Depth 1 m
Front Weighting 50.0 %
☐ Same Surface Only

Subsurface Amount	Subsurface scattering contributes to the diffuse shading of materials. The **Subsurface Amount** is a percentage value that determines the contribution of diffuse shading made by subsurface scattering. It is a good idea to keep this in mind in order to balance the percentages between diffuse color and subsurface amount depending on which effect you prefer for final shading. **Tip:** Combining high a Diffuse Amount and high Subsurface Amount can produce unnatural shading results.

	 Subsurface Amount 0%  Subsurface Amount 50%  Subsurface Amount 100%
Subsurface Color	The color value that is used to tint the light as it is attenuated through the material. This setting and the **Scattering Distance** work together to determine how far the light must travel to be tinted.
	  
Scattering Distance	Determines how far light must travel through the surface prior to exiting in order to be fully tinted by the subsurface color. If the light travels beyond the scattering distance it is further attenuated according to the original subsurface color. There are two scattering modes: • Setting the **Scattering Distance** to values greater than 0 produces a diffusion-style effect. • Setting the **Scattering Distance** to exactly 0 produces a simple translucency, like diffuse but with an opposite surface normal. **Note:** The **Scattering Distance** = 0 mode should only be used on objects with no thickness, such as leaves, paper, or curtains.
Maximum Depth	Provides is a fast way to approximate an opaque solid core within the subsurface layer (such as a bone inside a finger) by clipping any scattering samples that reach

	beyond Maximum Depth value specified. This may lead to a slightly darker scattering because of the clipped samples exclusion from the solution, but an increase in the number of total samples can help to reduce or eliminate this effect. A setting of zero disabled the option.
Front Weighting	Sets a bias for the SSS rays. At 50%, SSS rays scatter evenly forward and backward though a surface. Higher values bias the scattered rays toward the light source that emitted them, creating a more waxy/flat look to SSS. Values lower than 50% bias the scattered rays away from the light source, darkening the overall SSS effect.
Same Surface Only	When enabled, Modo's subsurface scattering works by limiting the SSS sample cache values to affect the shading to a single surface. This prevents certain artifacts that occur with very thin surfaces or intersections with non-SSS surfaces. This optimization can stop adjoining SSS surfaces from sharing values, resulting in a slight dark border between them. If this occurs, disable **Same Surface Only.**

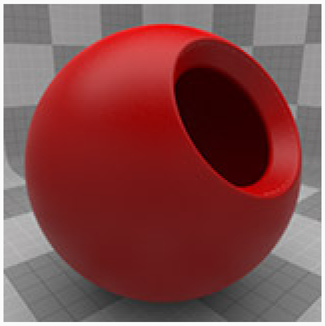
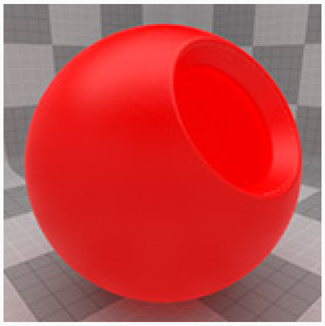
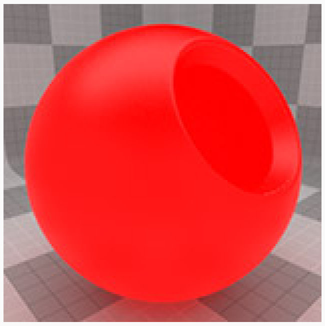
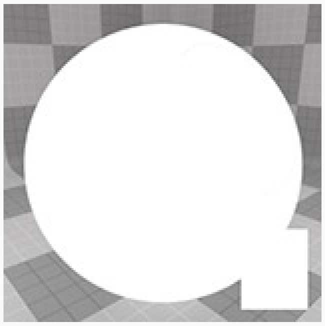
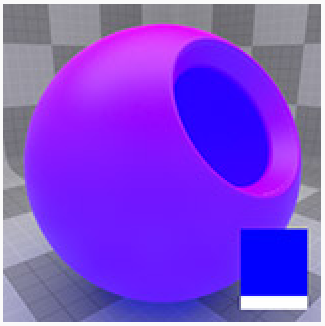
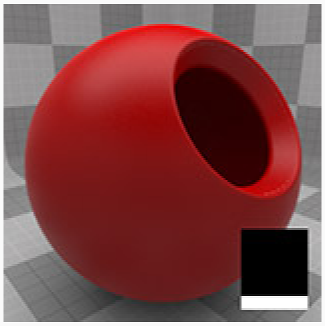
Material Trans - Luminosity


Luminosity

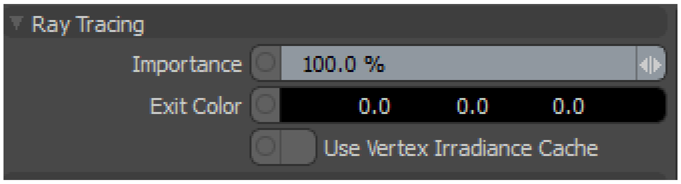
Luminous Intensity 0.0 W/srm2

Luminous Color 1.0 1.0 1.0

Luminous Intensity	A multiplier that drives the brightness of the **Luminous Color** as it is emitted from the surface. This is different from the **Diffuse Amount** in that diffuse requires some amount of light to hit the surface, whereas luminous comes from the surface itself. It is accurate to think of luminous surfaces as light sources - particularly when using **Global Illumination**. The **Luminous Intensity** is multiplied to the **Luminous Color**. By default, the value is zero which results in the **Luminous Color** control being disabled.

	 Luminous Intensity 0  Luminous Intensity 0.5  Luminous Intensity 1
Luminous Color	This setting allows you to specify a color value for light emitted from a surface as specified by the **Luminous Intensity** setting.
	  

Material Rays - Ray Tracing



Ray Tracing

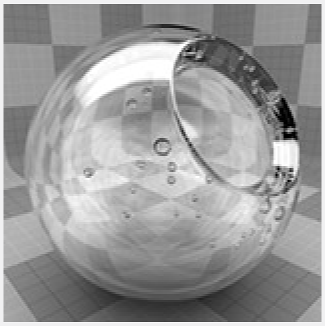
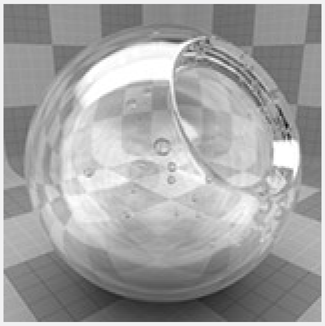
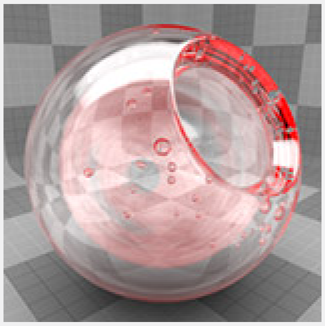
Importance 100.0 %

Exit Color 0.0 0.0 0.0

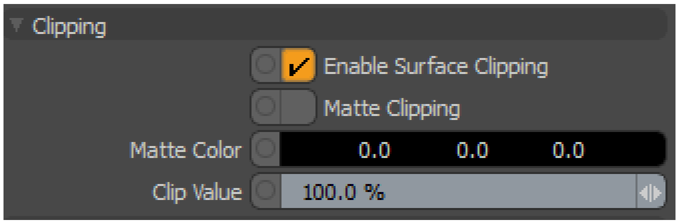
Use Vertex Irradiance Cache

Importance	Allows you to increase or decrease the sample counts for individual materials. Setting the **Importance** value influences the **Reflection Samples**, **Refraction Samples**, **Spectral Samples**, and **Subsurface Samples** Render Settings values. For more information, see Render Items. **Importance** is an estimate of how much a particular shading evaluation contributes to the final color of a pixel. For surfaces directly viewed by the camera, it's basically what fraction of a pixel the shading evaluation represents, which depends on how many

	antialiasing samples are being shaded together as part of a single evaluation versus the total number of antialiasing samples in the pixel. If there are 8 AA samples in a pixel and they are all shaded together as a group (which can happen if they all belong to the same surface and the Shading Rate is large), that single shading evaluation entirely determines the pixel color and thus has an importance of 100%. At the other extreme, if each antialiasing sample is shaded separately (which can happen if the Shading Rate is small or zero), then each of the 8 shading evaluations have an importance of 1/8 or 12.5%. This importance value is used to modulate the number of rays used to compute various phenomena, such as soft shadows, Monte Carlo indirect illumination, blurry reflections, and similar. At 100% importance, the full number of rays specified are used. For example, if there are 64 blurry reflection rays specified in a material's properties, then 64 rays are fired during the shading evaluation, but in a shading evaluation with 12.5% importance, only eight reflection rays (12.5% of 64) are fired. The **Importance** setting imposes a limit on importance when shading surfaces are directly viewed by the camera. This can cause more rays to be used in some cases, reducing artifacts that can be caused by Clamp Colors, but can also possibly increase render time.
Exit Color	Originally, when a series of refraction rays hit the maximum refraction depth, Modo returned black as the final value, so scenes with low refraction depth values exhibited lots of dark areas. **Exit Color** allows you to specify the color Modo returns once it has reached its ray threshold. The color is still attenuated with the initial ray bounces (colors, absorption, and so on), however, this setting can reduce a lot of the dark areas present with lower refraction depth values. In the first example below, there is a good amount of black revealed because of the low refraction depth. In the second image, the exit color is changed to a medium gray, simulating the background color, which results in a more accurate representation. Finally, the last image changes the exit color to red making the effects of the **Exit Color** setting obvious.

	 Exit Color (Default)  Exit Color Gray  Exit Color Red
Use Vertex Irradiance Cache	You can enable the **Irradiance Cache** to increase the performance of Irradiance Cache global illumination rendering for certain kinds of geometry, like characters and fur. This is because cached values are easily referenced, speeding up the evaluation of subsurface scattering, fur, and G.I. in general. When enabled on a surface, Irradiance Cache values are calculated at each vertex position (at the subdivision level) and interpolated in-between producing the shaded result. **Note:** The cache size is directly related to the subdivision vertex density, more polygons equals more IC values.
LPE Label	You can provide a label name here, which appears as the label for any Light Path Expression output you generate from this item. See Rendering with mPath.

Material Rays - Clipping



Clipping

☑ Enable Surface Clipping

☐ Matte Clipping

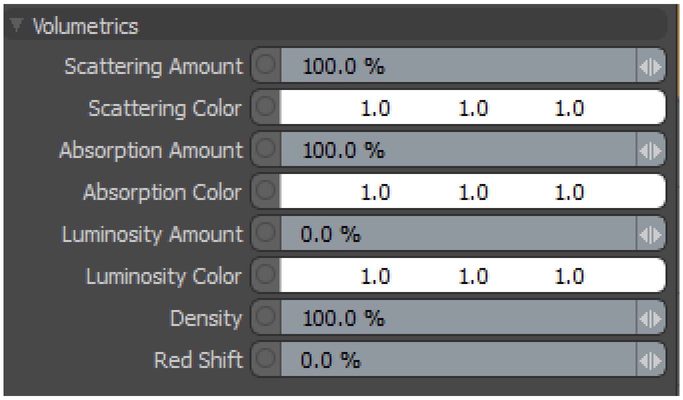
Matte Color 0.0 0.0 0.0

Clip Value 100.0 %

Enable Surface Clipping	Enabled by default, surface clipping refers to the removal of part of the scene as defined by the **Clipping Plane** settings of the Camera Items. You can define a distance where all (or selected) surfaces are clipped, removing them from the rendered image while leaving the

	scene geometry intact. In order for **Surface Clipping** to work, it must first be enabled in the Camera item, under the **Clipping** attributes. You can define the distance value as well, determining the distance away from the camera where the scene clips the rendered image. You can disable the clipping effect on a per-material basis by disabling this option. **Tip:** A similar effect can also be obtained through the use of render booleans that are not camera view dependent.
Matte Clipping	Provides a means to colorize interior areas of clipped geometry producing a filled area of color, the color being determined by the **Matte Color** value.
Matte Color	Determines the color of clipped interior areas when **Clipping** is enabled in both the **Material** clipping attributes and in the Camera item.
Clip Value	Defines a distance where all (or selected) surfaces are clipped, removing them from the rendered image while leaving the scene geometry intact.

Material Rays - Volumetrics



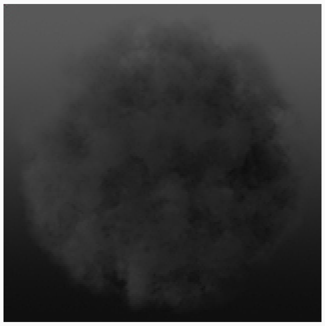
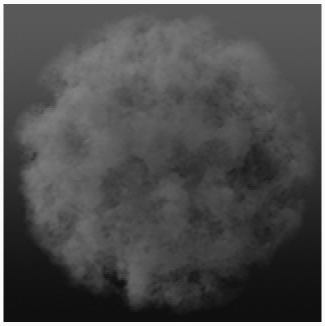
Volumetrics
Scattering Amount 100.0 %
Scattering Color 1.0 1.0 1.0
Absorption Amount 100.0 %
Absorption Color 1.0 1.0 1.0
Luminosity Amount 0.0 %
Luminosity Color 1.0 1.0 1.0
Density 100.0 %
Red Shift 0.0 %

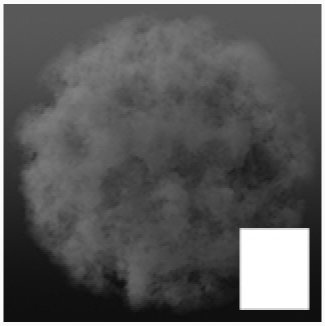
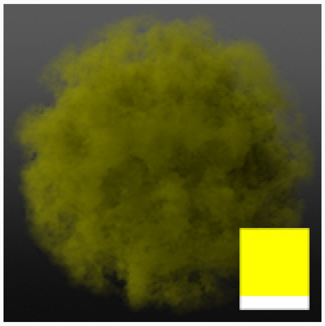
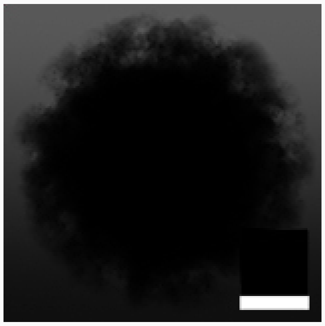
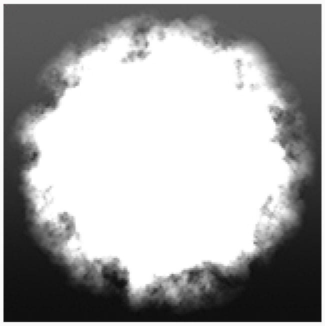
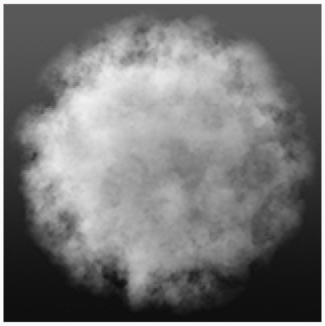
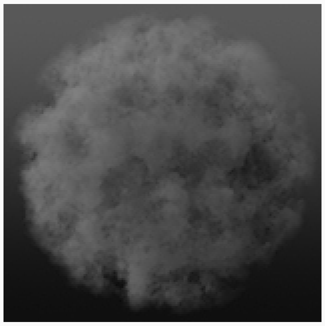
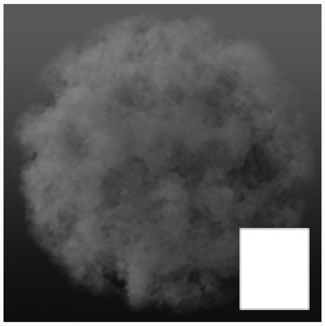
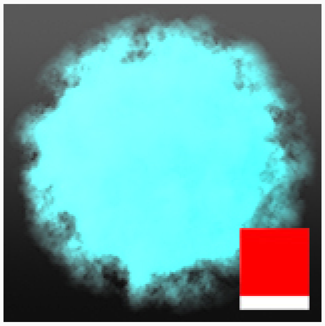
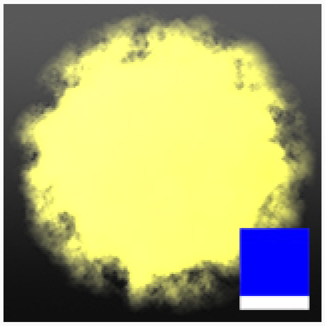
When light shines on certain matter suspended in the air, such as water vapor, smoke, dust, or even pollution, the particles become visible. This is known in computer graphics as a volumetric effect. Two things then happen to the light: part of it is bounced back in all directions (called scattering) and part of it is absorbed (called absorption).

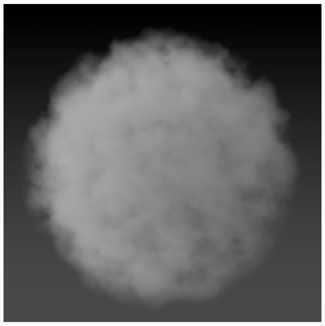
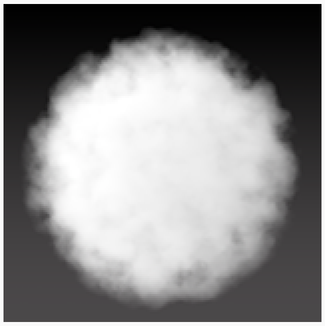
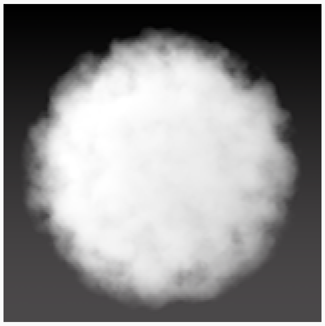
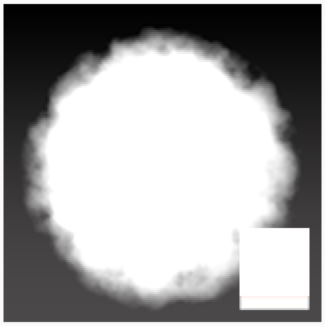
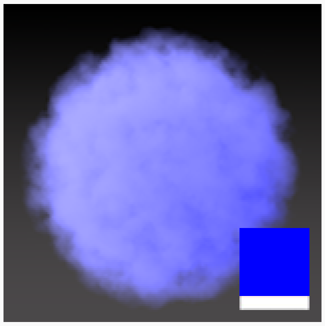
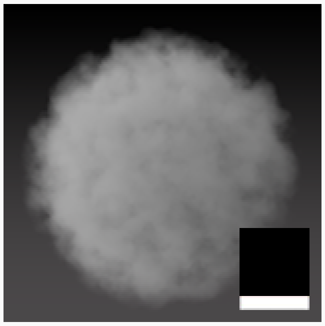
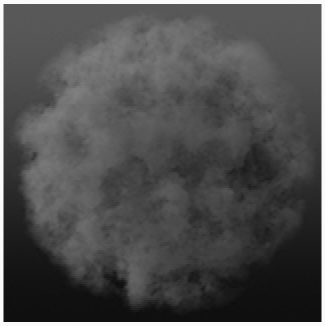
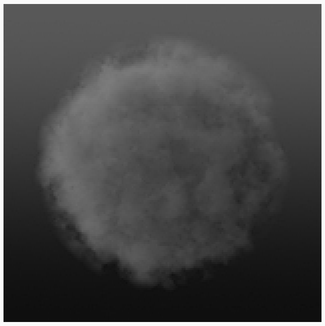
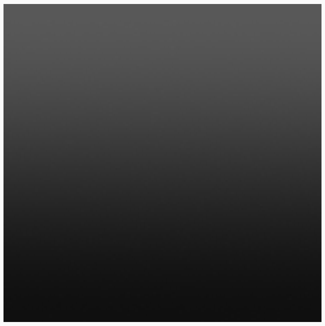
The **Density** parameter is the particle density of the volume, so the higher the density the more apparent particles there are. When the **Density** is increased, the scattering and the absorption also increase, so your volume becomes both more opaque and more luminous. However, when **Density** is increased, the volume becomes more opaque and the light scattered inside is more attenuated, so the net result in terms of overall luminosity is hard to predict. The final result depends on the balance of scattering versus absorption versus density.

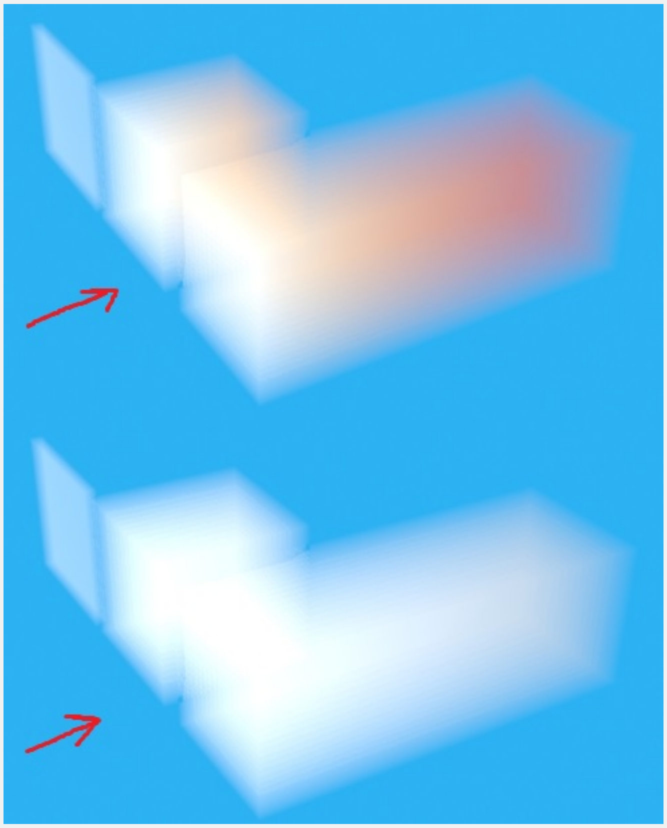
In simple terms one can say that scattering is similar to diffuse shading and absorption is inversely similar to luminosity (more light reflected back makes the volume appear luminous). This is however more subtle than that since the overall luminosity also depends on how much absorption happens inside the volume.

The following settings only apply to the volume item and otherwise have no affect on the surface. Likewise, the other surfacing attributes, such as **Diffuse Color** and **Reflection Amount** have no affect on a volumetric item. The volume item also contains similar options that act as multiplies of the values below, allowing you some additional tweaking options.

Scattering Amount	Scattering defines the amount of light that is reflected from the volume. This can also be described as how luminous the volume is, where a cloud is puffy and white, it is because it reflects back a lot of the light it receives (also absorbing very little). Conversely, a dense smoke cloud from an oil fire reflects back very little light, looking very dark and ominous. A value of 0% reflects back no light received from the source, up toward 100% (the default value) where the volume reflects back all of the light it receives.
	 Scattering Amt. 0%  Scattering Amt. 50%  Scattering Amt. 100%
Scattering Color	Represents the apparent color of the volume; the light that is reflected from the particulate matter.

	  
Absorption Amount	Determines how much light is absorbed into the particular matter. If low amounts of light are absorbed, then more light is reflected back producing a brighter, more luminous volume. Where more light is absorbed, the volume appears darker, more powdery, and dust like.
	 Absorption Amt. 0.01%  Absorption Amt. 50%  Absorption Amt. 100%
Absorption Color	Determines the color that is absorbed into the volume (which produces an inverted hue to what is applied).
	  
Luminosity Amount	When applied to a volume, gradually produces an overall consistent coloration to the volume as if it were a source of illumination. A value of 100% colors the entire volume the **Luminosity Color** value, flattening out the form, and eliminating any details.

	Note: The Luminosity level does not contribute any light to **Global Illumination** in a scene.
	 Luminous Amount 0%  Luminous Amount 50%  Luminous Amount 100%
Luminosity Color	Determines the coloration of the **Luminosity Amount** added to the volume.
	  
Density	Determines the thickness of the volume, controlling the apparent amount of particular matter that is visible.
	 Density 100%  Density 50%  Density 0%
Red Shift	When light goes into or through volumetrics, for example fog or smoke, light towards the

	red end of the spectrum makes it through a lot easier than other colors, so the non-red parts of the light gets filtered out the deeper into a volume the light goes. Rather than make light tint automatically, like it does in nature, the **Red Shift** channel lets you control how much this happens.  The top example has a 200% Red Shift applied and the bottom example has no Red Shift applied. The arrows in this image indicate the light direction.

Custom Materials

This topic lists the custom materials available in Modo. You can find these materials in the **Shader Tree**, by clicking **Add Layer** > **Custom Materials**. To learn more about the usage and properties of a particular material, click the relevant topic title below.

AxF Material

AxF (Appearance Exchange Format) is a file format for exchanging digital material appearance developed by X-Rite. Modo supports the import of a subset of AxF materials and shader variants, allowing you to get full quality materials on your surfaces when using the AxF material.

Note: AxF materials are supported by the mPath and Default renderers.

Handbag with AxF materials from the X-Rite sample library rendered using the mPath renderer.

Supported File Format Versions

Modo supports AxF versions 1.0, 1.1, 1.4, and 1.5 of the SVBRDF material type.

Supported Texture Effects

- Base Color
- Normal
- Specular Color
- Specular Lobe

- Anisotropy Rotation
- Fresnel
- Clearcoat IOR
- Clearcoat Normal
- Alpha (Transparency)
- Height (Displacement)

Supported Shading Variant

- Lambert Diffuse Shading
- GGX Specular Shading

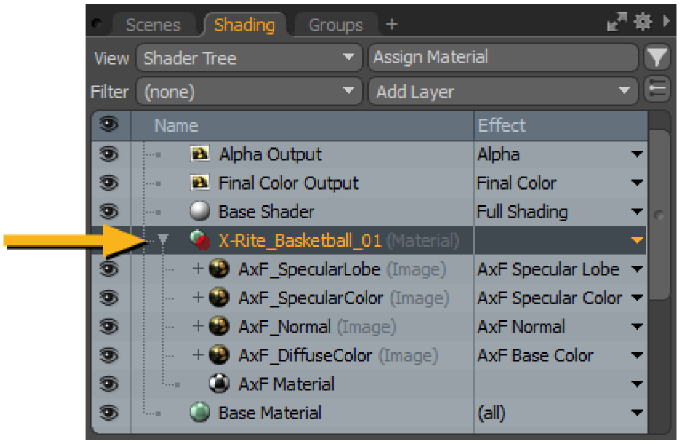
Importing AxF Materials

To import an AxF material, you can drag and drop it into Modo from your file browser.

Alternatively:

1. In the menu bar, click **File** > **Import**.
2. In the file browser, navigate to your file and click **Open**.

   The material is added to the Shader Tree.



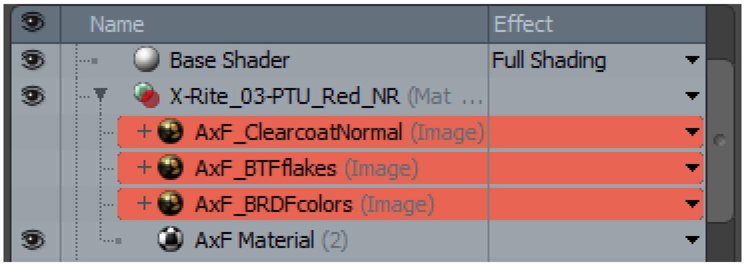
View: Shader Tree
Assign Material
Filter: (none)
Add Layer

Name	Effect
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
X-Rite_Basketball_01 (Material)	
AxF_SpecularLobe (Image)	AxF Specular Lobe
AxF_SpecularColor (Image)	AxF Specular Color
AxF_Normal (Image)	AxF Normal
AxF_DiffuseColor (Image)	AxF Base Color
AxF Material	
Base Material	(all)

Expand the material by clicking the arrow in front of it to reveal its components.

> **Note:** You may need to move the components out of the material folder to make the effect visible on your mesh.

Modo supports versions 1.0, 1.1, 1.4, and 1.5 of AxF materials. Other versions may contain components that Modo doesn't recognize. If a component is not supported by Modo, it is marked with red in the Shader Tree and its effect does not influence the material.



Name	Effect
Base Shader	Full Shading
X-Rite_03-PTU_Red_NR (Mat ...	
AxF_ClearcoatNormal (Image)	
AxF_BTFflakes (Image)	
AxF_BRDFcolors (Image)	
AxF Material (2)	

AxF Real World Size

The real world scale of AxF textures are supported on import of the material, and stored to the **Size X** and **Y** fields in the **Texture Locator** tab of the AxF texture. You can override these sizes by entering your own values in millimeters and meters into **Size X** and **Y**.

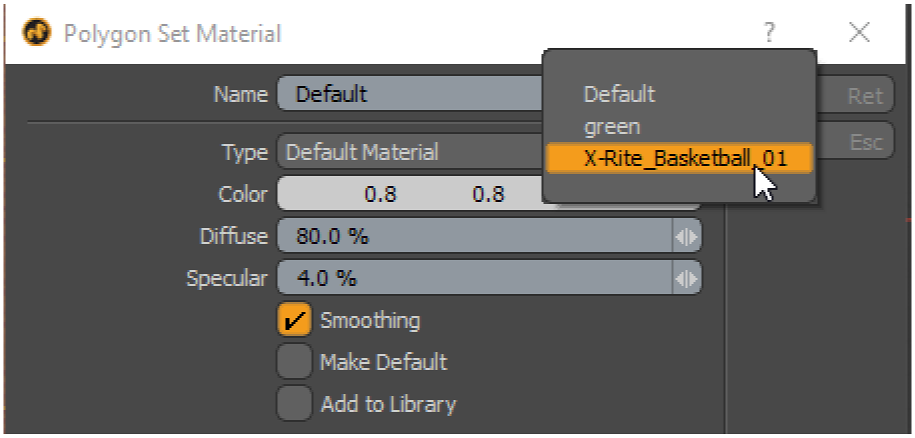
If these options are disabled, you need to assign a real world size to your UV maps. See Adding Real World Size to UV Maps.

Assigning AxF Materials

To assign an imported AxF material to your item or part:

1. Select the item, or use **Polygons** selection mode to select the required polygons.
2. Press the **M** key.
3. Alternatively, click **Assign Material** in the Shader Tree.

   This opens the **Polygon Set Material** dialog.
4. Select the material you imported in the **Name** dropdown.



Name Default

Type Default Material

Color 0.8 0.8

Diffuse 80.0 %

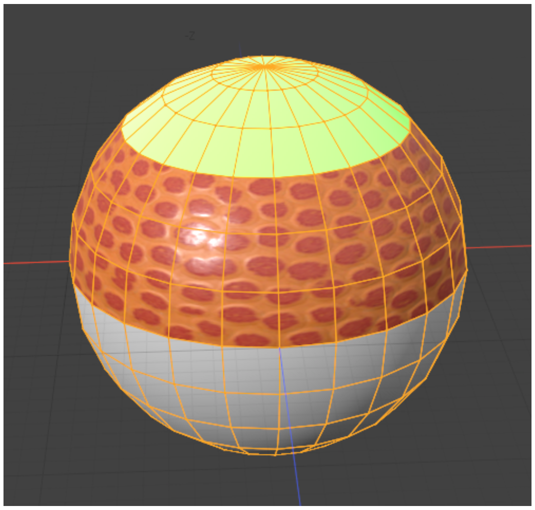
Specular 4.0 %

☑ Smoothing

☐ Make Default

☐ Add to Library

The material is assigned to your selection.

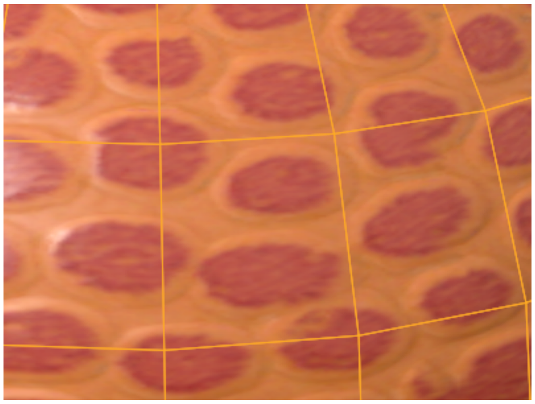


Note: For more information on assigning materials, see Assigning Material Tags.

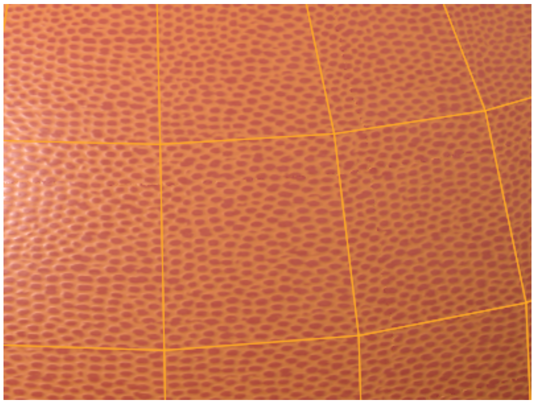
You can achieve different texture effects using the **Horizontal Wrap** and **Vertical Wrap** settings in the **Texture Locator** properties:

1. Expand the AxF material group by clicking the arrow in front of it.
2. Select all materials within the group.
3. In the **Properties** panel, click the **Texture Locator** tab and edit the **Horizontal Wrap** and **Vertical Wrap** values.

   The texture is updated with the new settings.



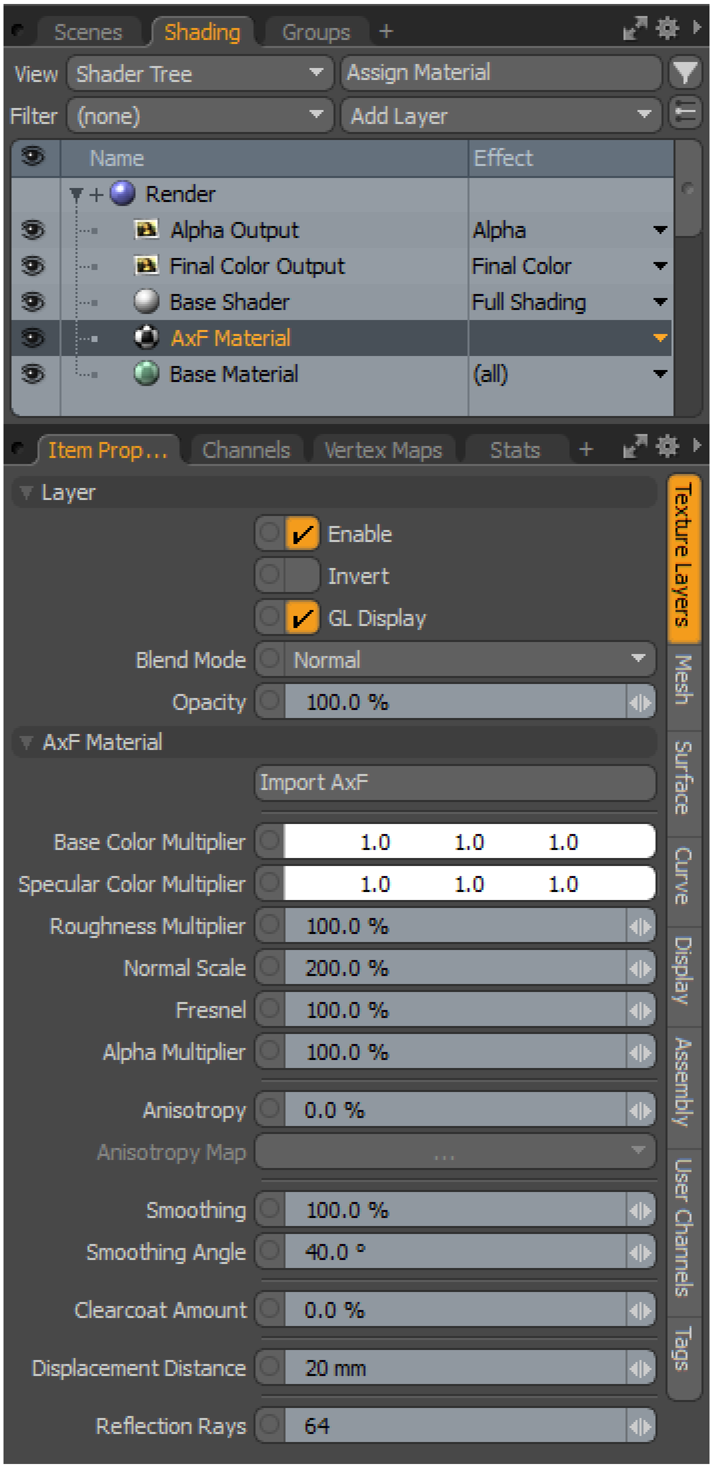
Horizontal/Vertical Wrap of 1.0



Horizontal/Vertical Wrap of 10.0

Creating an AxF Material

To create an AxF material, in the **Shader Tree**, click **Add Layer** > **Custom Materials** > **AxF Material**. The material is added to the Shader Tree and the layer's **Properties** panel displays.


Scenes
Shading
Groups
+
Assign Material
Add Layer
Item Prop...
Channels
Vertex Maps
Stats
+
Texture Layers
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags
Import AxF

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
AxF Material	
Base Material	(all)

Layer

☑ Enable
☐ Invert
☑ GL Display

Blend Mode: Normal
Opacity: 100.0 %

AxF Material

Base Color Multiplier: 1.0 1.0 1.0
Specular Color Multiplier: 1.0 1.0 1.0
Roughness Multiplier: 100.0 %
Normal Scale: 200.0 %
Fresnel: 100.0 %
Alpha Multiplier: 100.0 %
Anisotropy: 0.0 %
Anisotropy Map: ...
Smoothing: 100.0 %
Smoothing Angle: 40.0 °
Clearcoat Amount: 0.0 %
Displacement Distance: 20 mm
Reflection Rays: 64

Applying an AxF file to an AxF Material

Once you've added an AxF material, you can import AxF files and apply them to the material in the **Properties** tab.

1. Select the AxF Material in the Shader Tree and go to the **Properties** tab in the lower right panel.

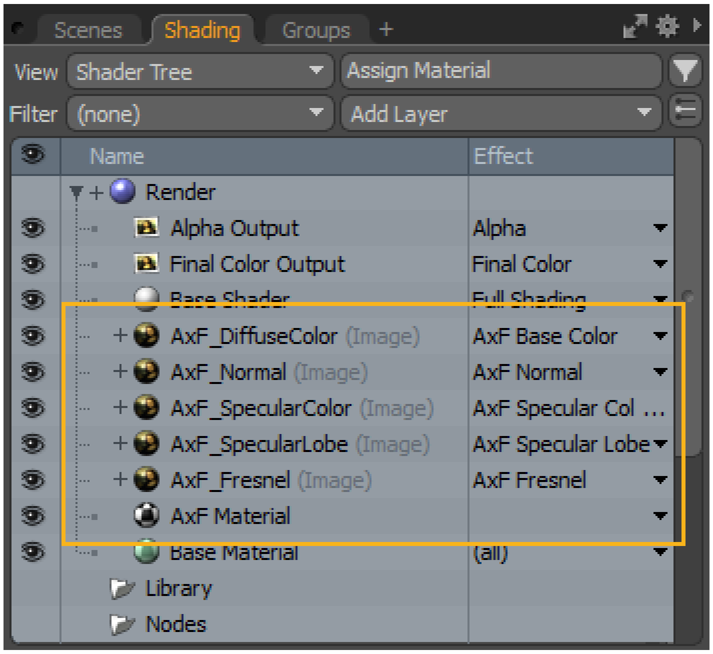
   The **Properties** tab is populated with the material's properties.

2. Click the **Import AxF** button.

   This opens a file browser.

3. Navigate to the **.axf** file you want to import, then click **Open**.

   The file is loaded into Modo and applied to the selected AxF Material.



Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
AxF_DiffuseColor (Image)	AxF Base Color
AxF_Normal (Image)	AxF Normal
AxF_SpecularColor (Image)	AxF Specular Col …
AxF_SpecularLobe (Image)	AxF Specular Lobe
AxF_Fresnel (Image)	AxF Fresnel
AxF Material	
Base Material	(all)
Library	
Nodes	

AxF Material Properties

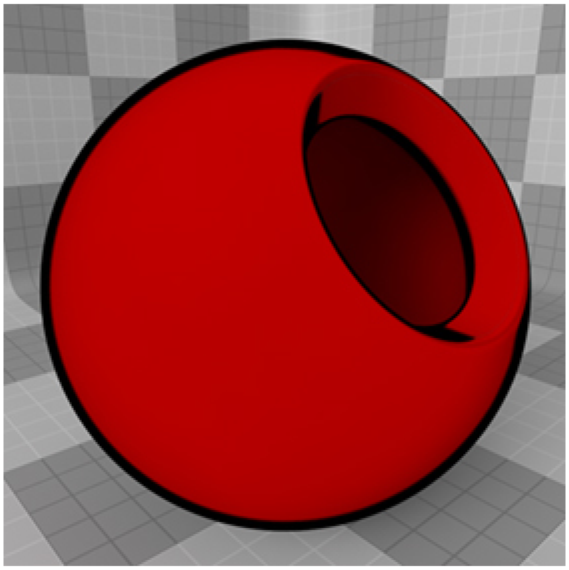
Enable	Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the layer producing a negative effect.

GL Display	Toggles the display of textures in the Camera view to improve readability and performance when working on complex scenes. When this option is disabled, textures are displayed in the Preview Viewport but not in the Camera view. For more information, see Camera Items.
Blend Mode	Affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the Shader Tree if present, or dims the effect of the layer itself on the surface. **Note:** This property does not function, because AxF Opacity is not yet supported by Modo. Currently, a value above 50% is opaque, and a value of 50% or below makes the layer invisible.
Import AxF	Click this button to load an AxF file and apply it to the selected AxF material. For more information, see Applying an AxF file to an AxF Material.
Base Color Multiplier	Defines an RGB value for the material's base color that is applied to the surface. When **Base Color** textures are applied, this parameter acts as a multiplier to the final blended texture values.
Specular Color Multiplier	Defines an RGB value that tints the specular highlight. When **Specular Color** textures are applied, this parameter acts as a multiplier to the final blended texture values.
Roughness Multiplier	Surfaces can be very different from each other, such as a rubber ball, a plastic cup, or a terracotta brick. These are all seemingly smooth surfaces, but on a microscopic scale their surfaces are very different. The microscopic surface variations have a direct effect on the way the light reflects off of each surface. The **Roughness** setting simulates these microscopic differences making surfaces appear more dull or rough, or shiny and smooth. It also affects both the specular highlight as well as ray traced and environmental reflections. When **Roughness** textures are applied, this parameter acts as a multiplier to the final blended texture values.
Normal Scale	Changes the normal vector on the surface affected by the AxF material to

	accentuate or mask surface distortions generated by the AxF normal texture effect.
Fresnel	At glancing angles (any surface that is near-perpendicular to the direction the camera is facing) many types of surfaces increase their reflectance values. The **Fresnel** setting increases the specular amount at these glancing angles to produce a physically plausible result.
Alpha Multiplier	Specifies the surface transparency. When **Alpha** textures are applied, this parameter acts as a multiplier to the final blended texture values.
Anisotropy	Millions of microscopic scratches on a surface distort highlights and reflections over their surface parallel to the directions of the scratches. The **Anisotropy** value affects the highlights, simulating this effect. Anisotropy is controlled by a UV map or further controlled by texture maps when the layer's effect is set to **Anisotropy Direction**. It can be caused by many man-made processes, such as machining metal and also appears in textiles where shiny threads spread the highlight over a surface.
Anisotropy Map	The UV map dropdown allows you to choose which map determines the anisotropic direction. The U direction (horizontal) defines the direction of the virtual scratches for positive **Anisotropy** values.
Smoothing	A shading technique giving the illusion that geometry has more detail than it actually does. Without smoothing, geometric faces can appear faceted or flat. The Smoothing amount is set as a percentage so that you have the flexibility to fade in the smoothing amount, or even drive it above 100% or below 0% to create interesting effects. **Note:** At values over 100%, the smoothing effect results in pillowing around each polygon.
Smoothing Angle	Determines which faces should receive smoothing. It sets a maximum angle tolerance between two polygons based on their normals. If the angle between the polygons is greater than the smoothing angle, no smoothing occurs. This is useful so that hard edges, such as cubes, are not smoothed across. The default value of 40° is a good general case value.
Reflection Rays	This property refers to baking with **F9** or **Preview** render. Specifies the number of rays used for sampling reflection. A higher number of rays produces better rendering quality, but increases render time.

Cel Edges

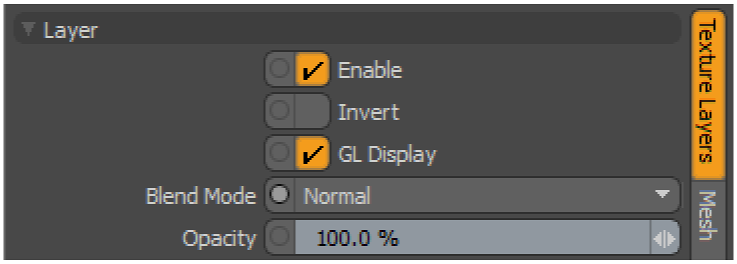
In the early days of animation, all art was hand drawn on transparent acetate called a **Cel** and painted with flat solid colors. This produced a unique look that is still highly desirable even today. The **Cel Edges** material attempts to mimic the hand drawn look of the outlines of these cartoons, creating colored strokes that follow surface contours and edges.



When combined with a Cel Material item, a low-fidelity cartoon-like look can be achieved. You can add the material to any surface using the **Add Layer** function of the **Shader Tree**, found in the menu under **Custom Materials** > **Cel Edges Material**.

Note: For information regarding adding and working with Shader Tree Items Layers, please see Shader Tree.

Layer Properties



Layer

☑ Enable

☐ Invert

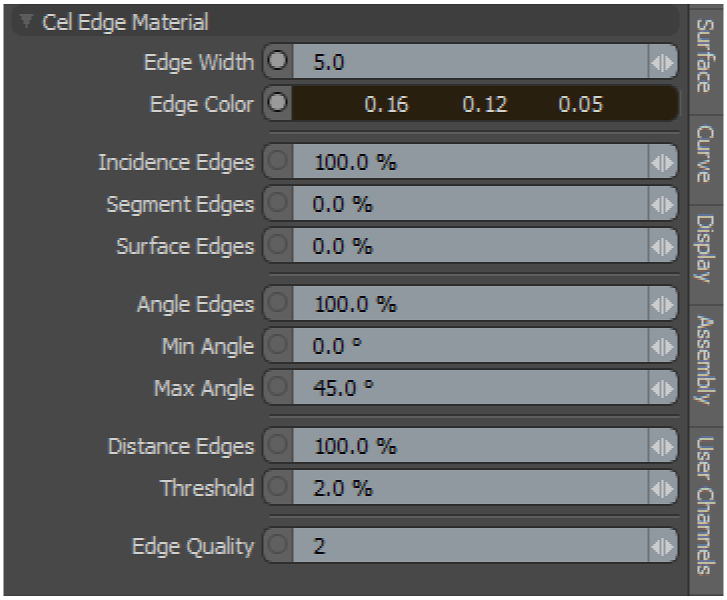
☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
GL Display	Enables or disables the drawing of the material in the 3D viewport **Default** mode. Disabling this can improve readability and performance when working on complex scenes. **Note:** Some materials, like Cel Edges, are not displayed in the 3D viewport.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Cel Edge Material Properties



Cel Edge Material

Edge Width 5.0

Edge Color 0.16 0.12 0.05

Incidence Edges 100.0 %

Segment Edges 0.0 %

Surface Edges 0.0 %

Angle Edges 100.0 %

Min Angle 0.0 °

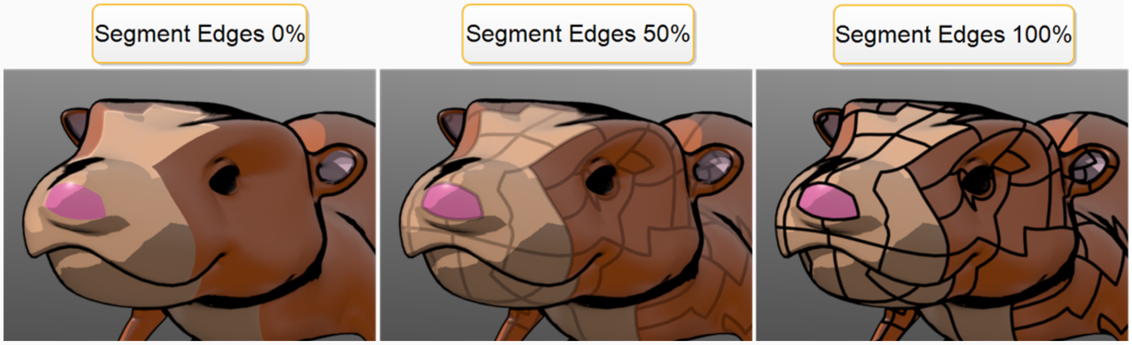
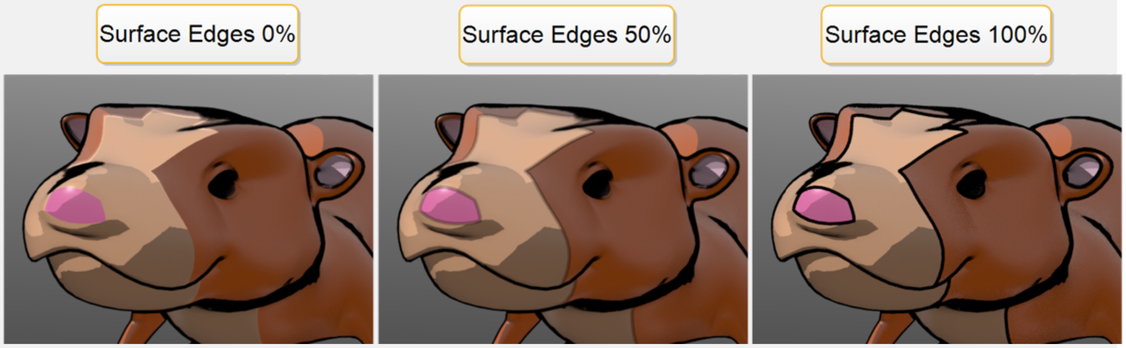
Max Angle 45.0 °

Distance Edges 100.0 %

Threshold 2.0 %

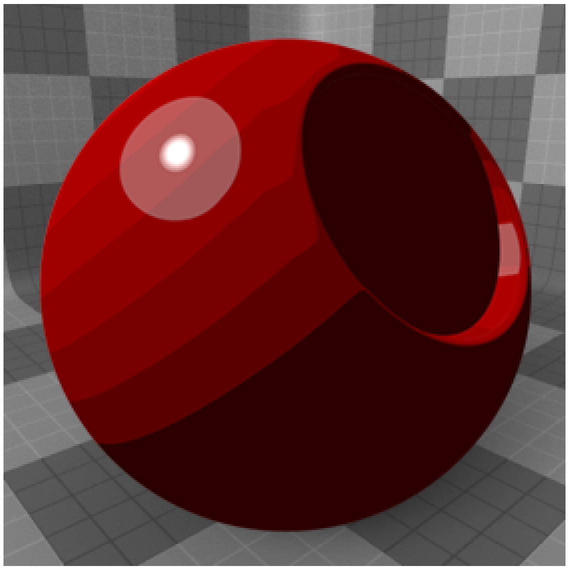
Edge Quality 2

Edge Width	Defined as a pixel value, this control determines the overall width of the edge stroke. **Tip:** You should adjust this value when changing frame resolution because a two pixel contour appears half as thick when the render resolution is doubled.
Edge Color	This value determines the color of the actual stroke.
Incidence Edges	This value determines the amount that the **Incidence Edges** contribute to the final Cel edge. Incidence edges are those that appear at the steepest glancing angles, perpendicular to the camera. This is useful for obtaining a more organic thick and thin stroke, as the incidence angle changes depending on the surfaces position relative to the camera.
Segment Edges	This value specifies the percentage of cel edges to overlay with nearby surface segments. The following images show **Segment Edges** set to **0%**, **50%**, and **100%** respectively.

	
Surface Edges	This value specifies the percentage of cel edges to overlay with nearby material groups. The following images show **Surface Edges** set to **0%**, **50%**, and **100%** respectively. 
Angle Edges	This value determines the amount that interior contour edges, within the **Min** and **Max Angle** values, contribute to the final Cel Edge.
Min/Max Angle	These two values determine the angle spread between opposing polygonal surfaces where a stroke is rendered. Typically, only an adjustment to the **Max** value is necessary. Lowering the value produces more contour edges and raising it produces progressively less.
Distance Edges	This function detects which interior areas of a surface are considered edges when surface contours overlap. Edges are shaded when two adjacent sampling points are greater than the given threshold.
Theshold	This value is used to determine the distance between adjacent sampling points for **Distance Edges**. Increasing it produces fewer interior edges for a surface, while decreasing it produces more edges.
Edge Quality	This value determines the overall quality of the Cel edges. Increasing this value produces cleaner edges, as more edge determining rays are fired, at the expense of longer render times.

Cel Material

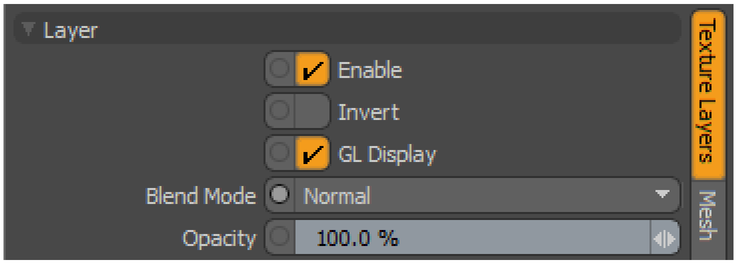
In the early days of animation, all art was hand drawn on transparent acetate called a **Cel** and painted with flat solid colors. This produced a unique look that is still highly desirable even today. The **Cel Material** attempts to mimic the flat posterized shading of these cartoons, creating flat colored bands that follow surface contours and edges. This effect produces a unique, stylized surface that looks more two-dimensional.



When combined with the Cel Edges material item, a low-fidelity cartoon-like look can be achieved. You can add the material to any surface using the **Add Layer** function of the **Shader Tree**, found in the menu under **Custom Materials** > **Cel Material**.

Note: For information regarding adding and working with Shader Tree item layers, please see Shader Tree.

Layer Properties


Layer

☑ Enable

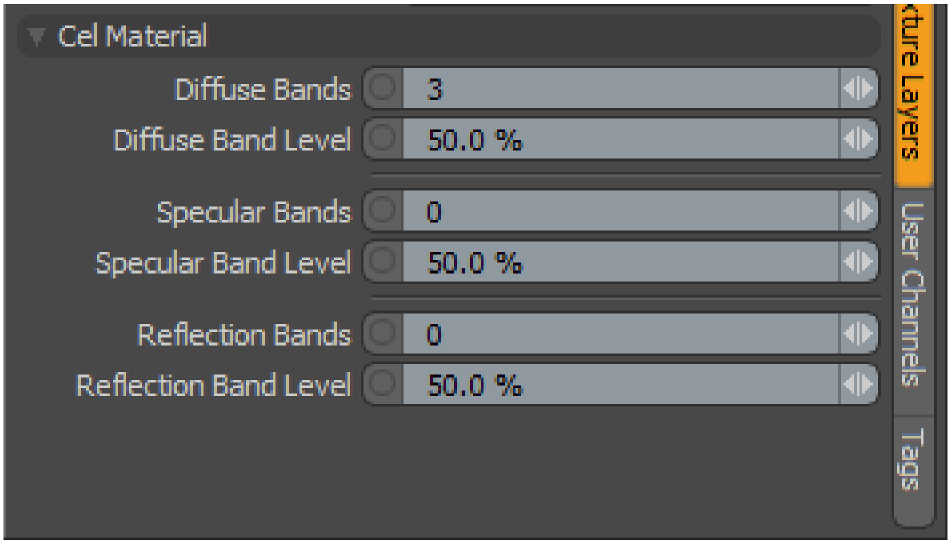
☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
GL Display	Enables or disables the drawing of the material in the 3D viewport **Default** mode. Disabling this can improve readability and performance when working on complex scenes. **Note:** Some materials, like Cel Edges, are not displayed in the 3D viewport.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.



Cel Material

Diffuse Bands 3

Diffuse Band Level 50.0 %

Specular Bands 0

Specular Band Level 50.0 %

Reflection Bands 0

Reflection Band Level 50.0 %

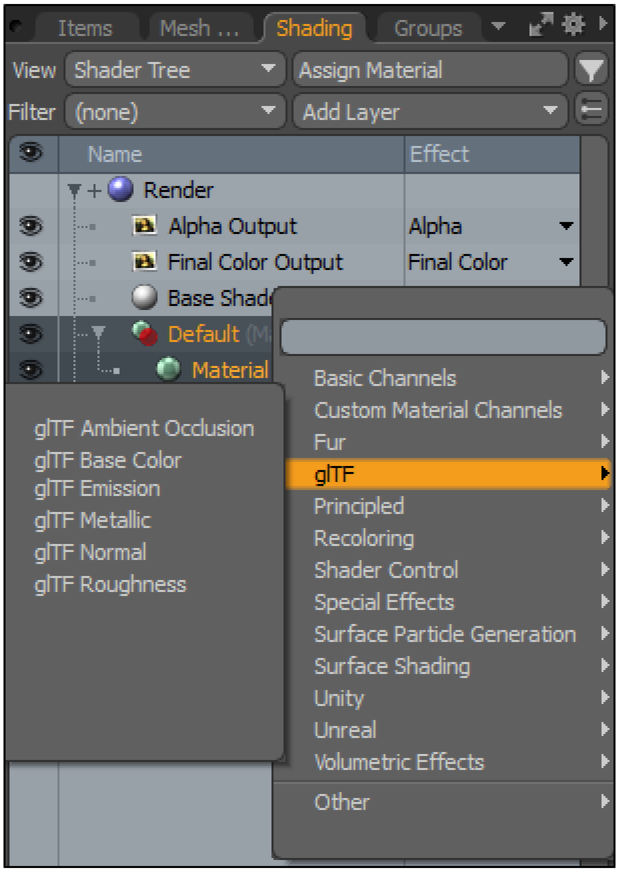
Cel Material	
Diffuse Bands	Determines the number of bands, or stepped transitions, representing the diffuse shading across the surface. A value of **0** disables the Cel shading effect for diffuse.
Diffuse Band Level	This is an offset value, making it possible to adjust the overall brightness of the **Diffuse Bands**.
Specular Bands	Determines the number of bands, or stepped transitions, representing the specular shading across the surface. A value of **0** disables the Cel shading effect for specular.
Specular Band Level	This is an offset value, making it possible to adjust the overall brightness of the **Specular Bands**.
Reflection Bands	Determines the number of bands, or stepped transitions, representing the reflection shading across the surface. A value of **0** disables the Cel shading effect for reflections.
Reflection Band Level	This is an offset value, making it possible to adjust the overall brightness of the **Reflection Bands**.

glTF Shader

The **glTF** (GL Transmission Format) is a file format for 3D scenes, models, materials, textures, images, articulated and skinned animation using key frame animations of transformed nodes information stored in hierarchy using the JSON standard and binary data aligned to the GL memory specifications for minimal real time run processing.

Modo supports glTF shader system directly in Modo Shader tree, the Advanced 3D (OpenGL) Viewport, the Preview Viewport, and in the glTF export plug-in. Results can be seen in the **Advanced** 3D (OpenGL) Viewport and the Preview Viewport. Results with the Modo glTF shader system closely relate to external real time glTF previewers.

Tip: Import textures for your glTF material using the PBR loader. See PBR Textures for more information.


Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shad
Default (M
Material
Basic Channels
Custom Material Channels
Fur
glTF
Principled
Recoloring
Shader Control
Special Effects
Surface Particle Generation
Surface Shading
Unity
Unreal
Volumetric Effects
Other
glTF Ambient Occlusion
glTF Base Color
glTF Emission
glTF Metallic
glTF Normal
glTF Roughness

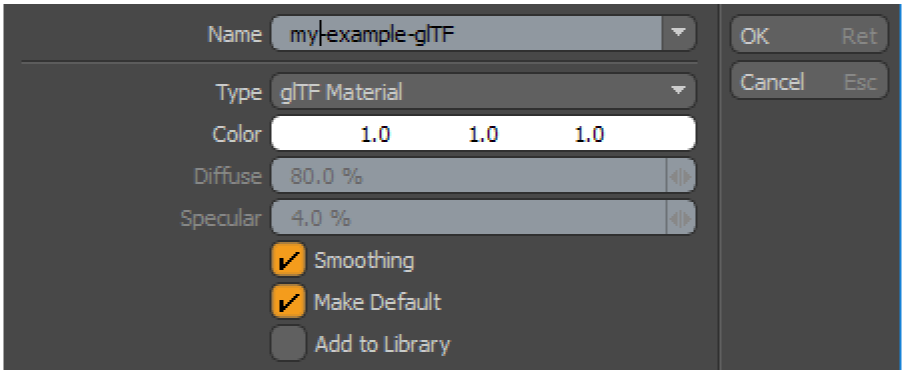
Adding a glTF Material

To add a glTF material and set it as your default material:

1. Activate **Polygons** selection mode, make a selection, and press **M**.

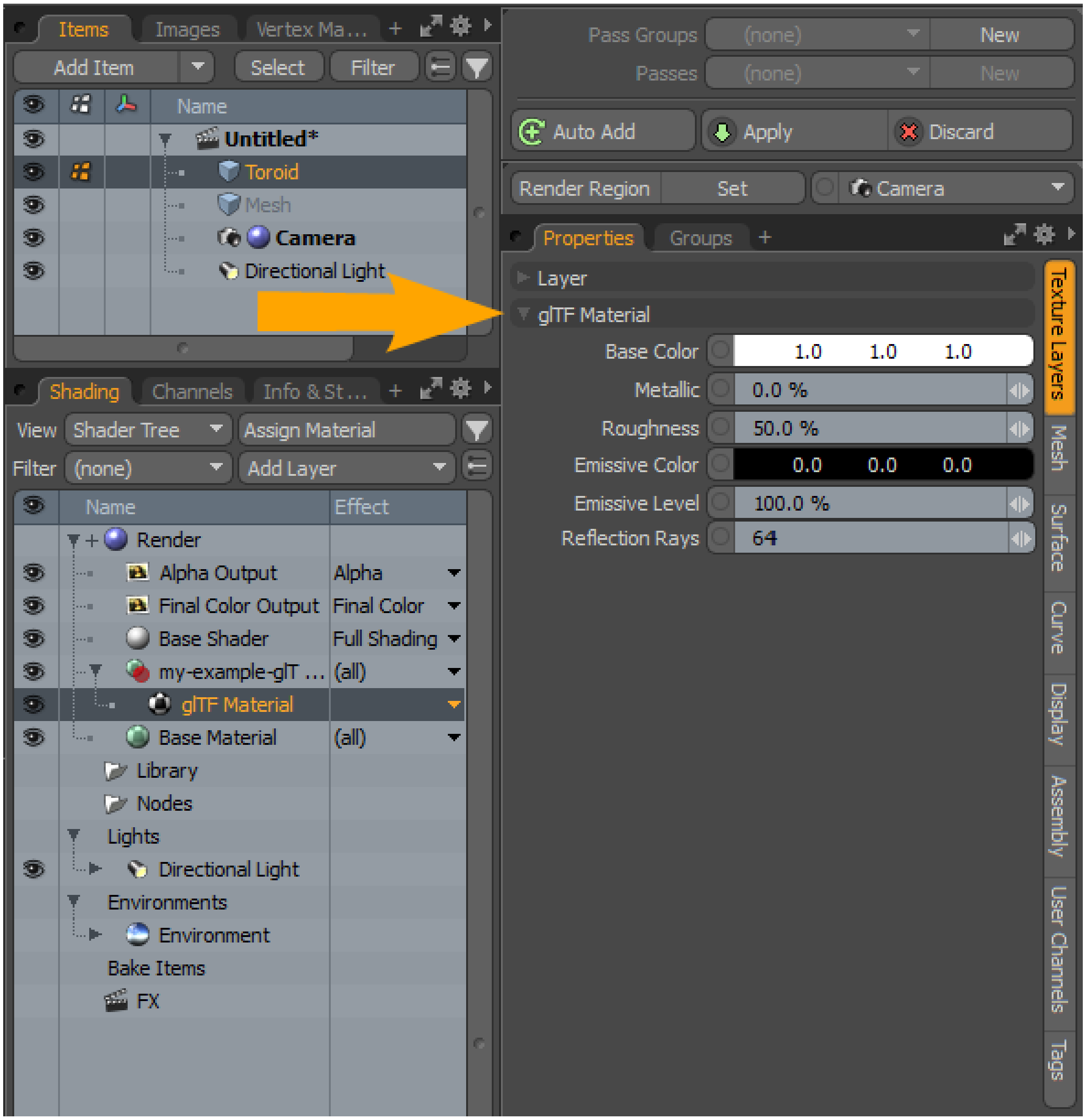
   The **Polygon Set Material** dialog displays.

2. Set the **Type** to **glTF Material** and enable **Make Default**.


OK Ret
Cancel Esc

Name: my-example-glTF
Type: glTF Material
Color: 1.0 1.0 1.0
Diffuse: 80.0 %
Specular: 4.0 %
☑ Smoothing
☑ Make Default
☐ Add to Library

3. Open the **Shader** tab on the right panel, and edit the properties in the **Properties** > **Texture Layers** tab. For more information, see glTF Material Properties.


Items
Images
Vertex Ma...
Add Item
Select
Filter
Name
Untitled*
Toroid
Mesh
Camera
Directional Light
Shading
Channels
Info & St...
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
my-example-glT ...
(all)
glTF Material
Base Material
(all)
Library
Nodes
Lights
Directional Light
Environments
Environment
Bake Items
FX
Pass Groups
(none)
New
Passes
(none)
New
Auto Add
Apply
Discard
Render Region
Set
Camera
Properties
Groups
Layer
glTF Material
Texture Layers
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

Base Color: 1.0 1.0 1.0
Metallic: 0.0 %
Roughness: 50.0 %
Emissive Color: 0.0 0.0 0.0
Emissive Level: 100.0 %
Reflection Rays: 64

Note: For information about exporting glTF files, see Importing and Exporting File Formats. For information about glTF preferences, see GLTF V2 I/O

Changing a glTF Shader Type

You can quickly change the glTF shader type.

Note: For information about adding and working with **Shader Tree** item layers, see Shader Tree.

To change a glTF shader type:

1. Open the **Shading** tab on the right panel.
2. Select the parent **Material** item in the list.
3. From the **Effect** dropdown menu, select **glTF**, and select a shader type.

glTF Shader Image Effect Types

Before applying any of the glTF Shader image effect types, review the following rules.

Note: Anything outside these rules generates a warning.

- **Swizzling** must be on for above image map effects.
- **glTF Metallic** swizzling channel must be **Blue**.
- **glTF Roughness** swizzling channel must be **Green**.
- **glTF Ambient Occlusion** swizzling channel must be **Red**.
- **Transparency Amount** swizzling channel must be **Alpha**.
- **glTF Metallic**, **glTF Roughness**, and **glTF Ambient Occlusion** must have the same image clip linked in the same material group.
- **glTF Base Color** and **Transparency Amount** must have the same image clip linked in the same material group.

The following **glTF** Shader image effect types are available:

glTF Image Effect Type	**Description**
glTF Ambient Occlusion	Occlusion is a simple and fast way to simulate global illumination without the overhead of calculating any scene lighting, as it only takes into account surfaces and how they adjoin to each other. An occlusion map is used to indicate areas of indirect lighting Ambient Occlusion describes how cracks, crevices, corners, and other small, semi-enclosed spaces tend to be darker than surrounding surfaces under real-world lighting conditions. Their insides are occluded, so ambient light can't get in. **glTF Ambient Occlusion** adds realism to views rendered and can be helpful for bringing out details when a scene isn't supposed to have a direct light source.
glTF Base Color	The base color has two different interpretations depending on the value of metalness. When the material is a metal, the base color is the specific measured reflectance value at normal incidence (F0). For a non-metal the base color represents the reflected diffuse color of the material.
glTF Emission	Uses an emissive map to control the color and intensity of the light emitted by the material.
glTF Metallic	Specifies the percentage of the material's electric conductivity. 0% means no conductivity (its dielectric or isolator, for example rubber, paint or wood material), 100% means full electric conductivity (material behaves like a metal).

	Note: Image maps with Swizzling effects **glTF Metallic** and **glTF Roughness** in a single material group must have the same Image Map linked in them. If you have created a material group containing **glTf Metalic** and **glTF Roughness** the following settings are required: - glTF Metallic channel must be "Blue". - glTF Roughness channel must be "Green".
glTF Normal	Uses a tangent space normal map. Geometry appears less detailed.
glTF Roughness	Specifies the surface roughness of the material applied to surfaces to appear more dull or rough, or shiny and smooth. **Note:** Image maps with Swizzling effects **glTF Metallic** and **glTF Roughness** in a single material group must have the same Image Map linked in them. If you have created a material group containing **glTF Metalic** and **glTF Roughness** the following settings are required: - glTF Metallic channel must be "Blue". - glTF Roughness channel must be "Green".

Exporting Skinning Animation

Meshes with applied joints with transformation animation and weight vertex maps can be exported. Each deformer group under the mesh are evaluated as separate glTF skins linking to its own **Weight** and **Joints** vertex maps contained in the skinned mesh data.

Note: Make sure you freeze any transformations you've made to your skinned mesh before exporting into glTF. Unfrozen skinned mesh transformations are ignored during import into any application that supports glTF according to the glTF v2 specifications.

- **Weight map** - A vertex map of weights maximally containing influences of four joints for each vertex individually for glTF v2 specification.
- **Joint map** - Contains indices of up to four nodes per vertex that have weight influence over vertices.

For each glTF skin there should be only one root (armature) joint node and all other joint nodes should be children in the hierarchy of that node. Each skin is defined by the **inverseBindMatrices** property (which points to an accessor with IBM data), used to bring coordinates being skinned into the same space as each joint, and a joints array property that lists the nodes indices used as joints to animate the skin.

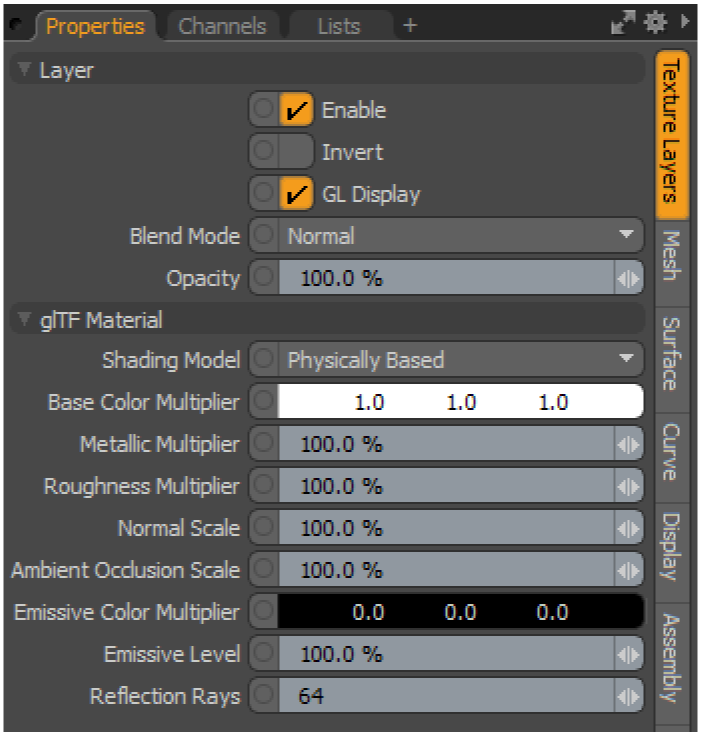
The **Save Skinned Meshes in World Space** preference option should be used for skinned animation in glTF v2 for compatibility with external previewers (for example, Babylon.js). Enabling **Save Skinned Meshes in World Space** in glTF v2 I/O preferences applies a mesh transformation matrix to mesh vertices and moves them to bind position in world space. World space mesh vertices are transformed by inverse bind matrices of joints and their vertex weight influence correctly.

For more information about preferences, see GLTF V2 I/O.

Exporting Cameras

Cameras from Modo are exported fixed as perspective cameras with transformation properties and placeholder properties with Z set to the value 0.01 and Y set to the value of 0.8726.

glTF Material Properties



Layer

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %

glTF Material

Shading Model: Physically Based
Base Color Multiplier: 1.0 1.0 1.0
Metallic Multiplier: 100.0 %
Roughness Multiplier: 100.0 %
Normal Scale: 100.0 %
Ambient Occlusion Scale: 100.0 %
Emissive Color Multiplier: 0.0 0.0 0.0
Emissive Level: 100.0 %
Reflection Rays: 64

- **Shading Model** - Controls how light and surfacing attributes are evaluated for a final rendering.
  - **Physically Based** - Uses highlight shape and Fresnel falloff shading. This is based on the Generalized Trowbridge-Reitz BRDF model, popularized by a number of famous movie studios because of its ability to produce realistic surface shading.
  - **Unlit** - Defines the **Base Color Multiplier** used for surface shading without being affected by any type of lighting (such as Directional or Environments).

> **Note:** The Material Transparency Amount property can also rendered when enabling the Unlit shading model.

- **Base Color Multiplier** - Defines an RGB value for the material's base color that is applied to the surface. When **Base Color** textures are applied, this parameter acts as a multiplier to the final blended texture values.
- **Metallic Multiplier** - Metallic colors typically spread a wide colored highlight over the surface. Specify the metal quality of the material by changing the percentage value. A value of **0** specifies a non-metallic surface, while a value of **1** results in a metal surface. When **Metallic** textures are applied, this parameter acts as a multiplier to the final blended texture values.

- **Roughness Multiplier** - Surfaces can be very different from each other, such as a rubber ball, a plastic cup, and a terracotta brick. These are all seemingly smooth surfaces, but on a microscopic scale their surfaces are very different. The microscopic surface variations has a direct effect on the way the light reflects off of each surface.

  The **Roughness** setting simulates these microscopic differences making surfaces appear more dull or rough, or shiny and smooth. It also affects both the specular highlight as well as ray traced and environmental reflections. When **Roughness** textures are applied, this parameter acts as a multiplier to the final blended texture values.
- **Normal Scale** - Scales for the normal map in the Main Maps.
- **Ambient Occlusion Scale** - Controls the amount of occlusion applied. A value of 0.0 means no occlusion. A value of 1.0 means full occlusion. This value is ignored if the corresponding texture is not specified. This value is linear.
- **Emission Color Multiplier** - Specifies which part of the material should glow due to emitting light.
- **Emission Level** - Specifies the level of emission.
- **Detailed Normal Scale** - Scales for the normal map in the Secondary Maps.
- **Height Scale** - Scales for the Height Map.
- **Fresnel Reflectance 0** - Determines the reflection of light.
- **Reflection Rays** - This property refers to baking with **F9** or **Preview** render. Specifies the number of rays used for sampling reflection. A higher number of rays produces better rendering quality, but increases render time.

Hair Material

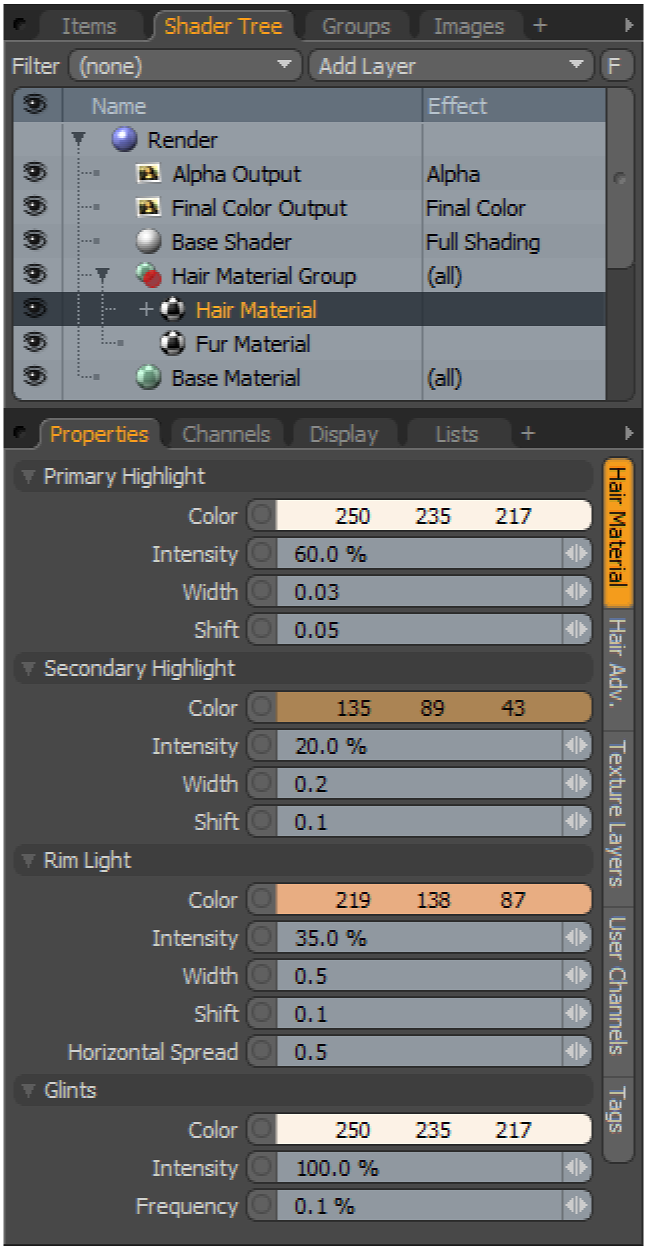
The **Hair Material** is designed to replace the default BRDF shading calculations of the standard Material item when applied to fur. Fur, or hair as the case may be, is anisotropic (directionally dependent) and requires a different type of shading model to look correct. On a microscopic level, the surface of hair is quite complex and this affects how light reflects off it and scatters through it, producing a unique look that is difficult to simulate without a dedicated hair shader.

Modo ships with a hair shading system using a special model for quickly approximating multiple scattering layers, which is way too expensive to compute otherwise with brute force. The model is called **Dual Scattering Approximation**, and is based on this technical paper. The basic idea is that multiple scattering is essentially broken into two parts: global scattering (lighting of the overall volume) and local scattering (the shading on individual fibers that are visible).

The **Hair Material** item is added to the **Shader Tree** using the **Add Layer** function, **Add Layer** > **Custom Materials** > **Hair Material**.

Note: For more information on adding and manipulating Shader Tree layers, please see Shader Tree.

For the best results, the placement of the layer is important, as the **Hair Shader** itself must reside in the same group mask as the **Fur Material** item it is shading. Once a **Hair Material** is added, its settings supersede those of any **Material** item, rendering the material unnecessary. Once positioned, selecting the **Hair Shader** item reveals its attributes in the **Properties** viewport, where its settings may be adjusted to fine tune the look of the rendered hair.


Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer
F
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Hair Material Group
(all)
Hair Material
Fur Material
Base Material
(all)
Properties
Channels
Display
Lists
+
Hair Material
Hair Adv.
Texture Layers
User Channels
Tags

Primary Highlight
Color 250 235 217
Intensity 60.0 %
Width 0.03
Shift 0.05

Secondary Highlight
Color 135 89 43
Intensity 20.0 %
Width 0.2
Shift 0.1

Rim Light
Color 219 138 87
Intensity 35.0 %
Width 0.5
Shift 0.1
Horizontal Spread 0.5

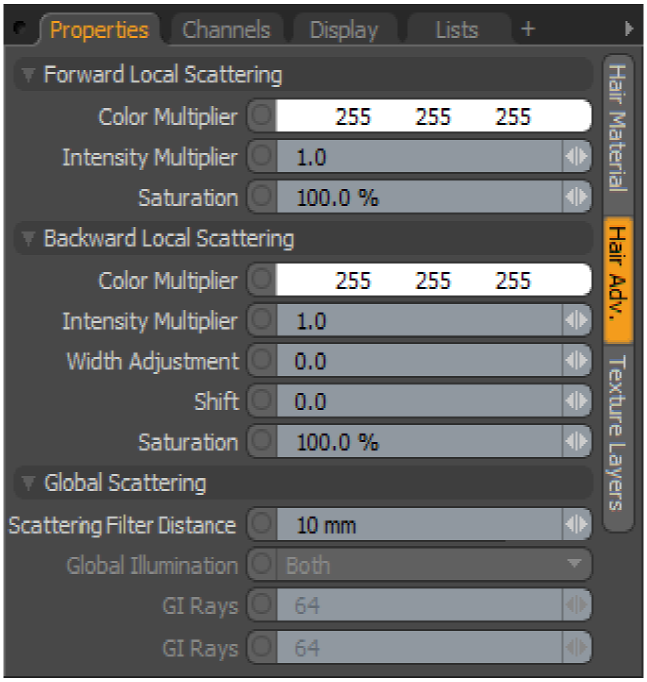
Glints
Color 250 235 217
Intensity 100.0 %
Frequency 0.1 %

Option	Description
Primary Highlight - Represents the main light colored highlight that is reflected off the surface of the hair. The more apparent this highlight is, the more sheen the hair has.	
Primary Color	R, G, and B values that determine the color of the **Primary Highlight** shading.
Primary Intensity	The **Intensity** option acts as a multiplier controlling the strength of the **Primary Highlight**.

Option	Description
Primary Width	This option controls the size of the highlight along the hairs length.
Primary Shift	This option controls the position of the highlight along the hairs length, basically simulating the tiny shingle-like cuticles of the hair's surface that modify the angle of the reflected highlight.
Secondary Highlight - Represents light that is refracted into the hair strand, hitting the back surface and reflecting back towards the front. Realistically, the magnitude of the longitudinal shift for the **Secondary Highlight** should be larger than for the **Primary Highlight** because the light path is more disrupted with additional refractions and reflections.	
Secondary Color	R, G, and B values that determine the color of the **Secondary Highlight** shading.
Secondary Intensity	The **Intensity** option acts as a multiplier controlling the strength of the **Secondary Highlight**.
Secondary Width	This option controls the size of the highlight along the hairs length.
Secondary Shift	This option controls the position of the highlight along the hairs length, basically simulating the tiny shingle-like cuticles of the hair's surface that modify the angle of the reflected highlight.
Rim Light - Represents light that is transmitted through the hair similar to the effects of subsurface scattering.	
Rim Color	R, G, and B values that determine the color of the **Rim Highlight** shading.
Rim Intensity	The **Intensity** option acts as a multiplier controlling the strength of the **Rim Highlight**.
Rim Width	This option controls the size of the highlight along the hairs length.
Rim Shift	This option controls the position of the highlight along the hairs length, basically simulating the tiny shingle-like cuticles of the hair's surface that modify the angle of the reflected highlight.
Rim Spread	Controls how broad or tight the **Rim Light** highlight is, similar to a specular highlight in a sense.
Glints - Representative of tiny caustics refracted through the hair, basically this is how hair looks shimmery.	

Option	Description
Glints Color	R, G, and B values that determine the color of the **Glints** shading.
Glints Intensity	The **Intensity** option acts as a multiplier controlling the strength of the **Glints**.
Glints Frequency	Determines how often a glint is shaded, controlling the overall density of **Glints**.

Hair Shader Advanced



Forward Local Scattering
Color Multiplier 255 255 255
Intensity Multiplier 1.0
Saturation 100.0 %
Backward Local Scattering
Color Multiplier 255 255 255
Intensity Multiplier 1.0
Width Adjustment 0.0
Shift 0.0
Saturation 100.0 %
Global Scattering
Scattering Filter Distance 10 mm
Global Illumination Both
GI Rays 64
GI Rays 64

Option	Description
Forward Local Scattering - This controls how multiple fibers in a small area scatter light forward, or away from the source of light. It contributes to the appearance of rim lighting, but it is wider, darker, and the color is more saturated than direct rim lighting. This is because the light is scattering multiple times, making it spread out more.	
Forward Color Multiplier	This setting allows you to adjust the color of the **Forward Local Scattering** by multiplying the color values. When each color channel is set to 1.0 (when the color is white), the **Forward Local Scattering** color is only dependent on the colors in the

Option	Description
	hair material settings overall.
Forward Intensity Multiplier	This allows you to make the **Forward Local Scattering** brighter or darker, by multiplying the value.
Forward Saturation	This allows you to control the color intensity of the **Forward Local Scattering**, making it more or less colorful.
Backward Local Scattering - The backward scattering adjustments represent areas where the hair is front lit, but the specific hair strand is in shadow.	
Backward Color Multiplier	This setting allows you to adjust the color of the **Backward Local Scattering** by multiplying the color values. When each color channel is set to 1.0 (when the color is white), the **Forward Local Scattering** color is only dependent on the colors in the hair material settings overall.
Backward Intensity Multiplier	This allows you to make the **Backward Local Scattering** brighter or darker, by multiplying the value.
Backward Width Adjustment	This option controls the size of the highlight along the hairs length, relative to the initial width, adjusting it wider or thinner.
Backward Shift	This option controls the position of the highlight along the hairs length, relative to the initial **Shift** value, moving the color of the backward scattering along the length of the hair strand.
Backward Saturation	This allows you to control the color intensity of the **Backward Local Scattering**, making it more or less colorful.
Global Scattering - controls the way in which light transmits, scatters, and spreads through a whole volume of hair. It is brightest and least spread out closest to the light, and gets dimmer, more spread out, and more colorful as the light transmits through more strands of hair. It affects both forward and backward multiple scattering.	
Scattering Filter Distance	The **Hair Material** uses advanced calculations to approximate the multi-scattering from light in dense fiber clusters. This estimation could occasionally produce sharp discontinuities which would appear as noise that no amount of samples would clear up. **Scattering Filter Distance** adds a random offset to the origin of the ray used to

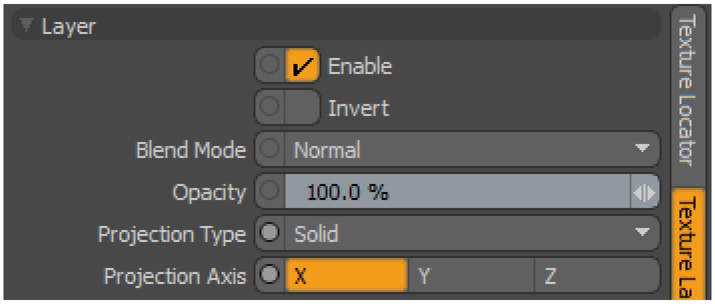
Option	Description
	estimate the scattering. This effectively converts the sharp discontinuities into actual noise that goes away with more antialiasing samples. **Note:** Large filtering distances can also add illumination bias, shifting the overall brightness of the hair, but when applied sensibly it works very well.
Global Illumination	When illuminating hair (or fur) using **Global Illumination**, you can adjust the **Global Illumination** settings of the hair shader to control how the GI rays are used: • **None** - the color of the hair is only affected by direct and ambient lighting. It also doesn't bounce light onto other surfaces. • **Receive Indirect Lighting** - light bounces off other surfaces onto the hair, causing the hair to be lit indirectly. This is done with Monte-Carlo global illumination, by casting rays in every direction, instead of just towards a light. • **Cast Indirect Lighting** - the hair contributes indirect lighting onto other surfaces. In other words, indirect rays being fired from another surface cause the hair shading to be evaluated if they hit the hair surface, instead of just appearing black. • **Both** - receive and cast indirect lighting.
GI Rays	This controls how many rays are used for calculating indirect lighting, either in the **Receive** or **Both** modes. More rays produce smoother results at the expense of longer render times.

Texture Layers

Since the Hair Material uses a custom shading model to render the resulting Fur, it is not compatible with blending or masking techniques allowed by other layers. It should also be noted that even when using **Texture Layers** to drive individual **Hair Shader** channels, the **Hair Material** channel values still affect the final result. This is because texturing only affects the single scattering of the hair material, but the multiple scattering calculations rely on the channels settings.

Layer Properties

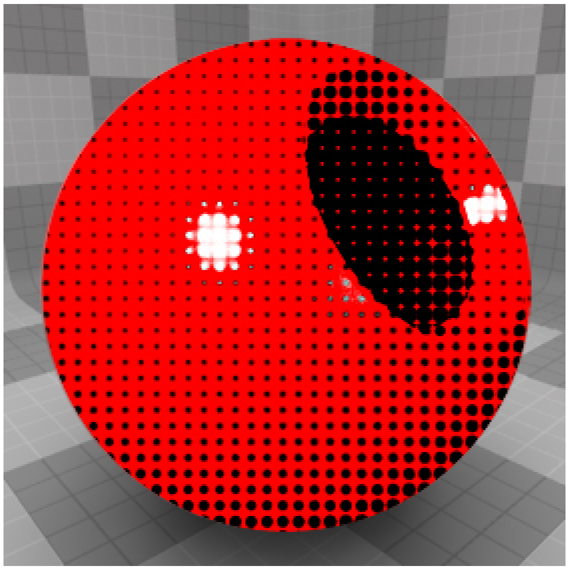
>



Layer	
Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Projection Type	Defines how a material us applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Halftone Material

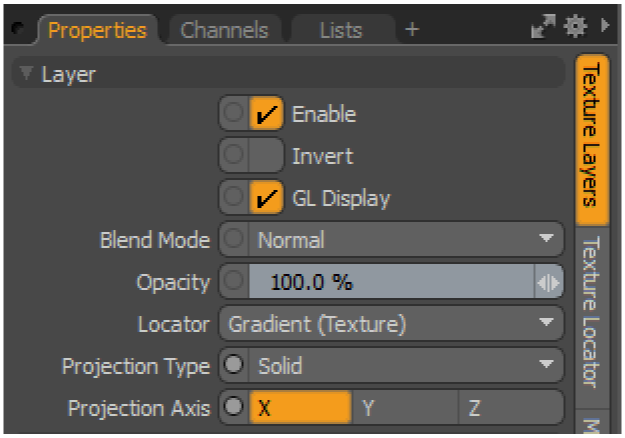
The **Halftone Material** stylizes the target surface replacing the smooth shading with a black user-defined pattern laid over its diffuse color, simulating the look of a low-fidelity image printed in a newspaper or magazine.



You can add the material to any surface using the **Add Layer** function of the Shader Tree, found in the menu under **Custom Materials** > **Halftone Material**. The direction and contouring of the lines can be dependent on the surface's UV map, defined in the Texture Locator item, or by enabling the **Screen Coordinates** option, the lines simply follow along the user-defined **Angle** value relative to the frame.

Note: For information regarding adding and working with Shader Tree item layers, please see Shader Tree.

Layer Properties



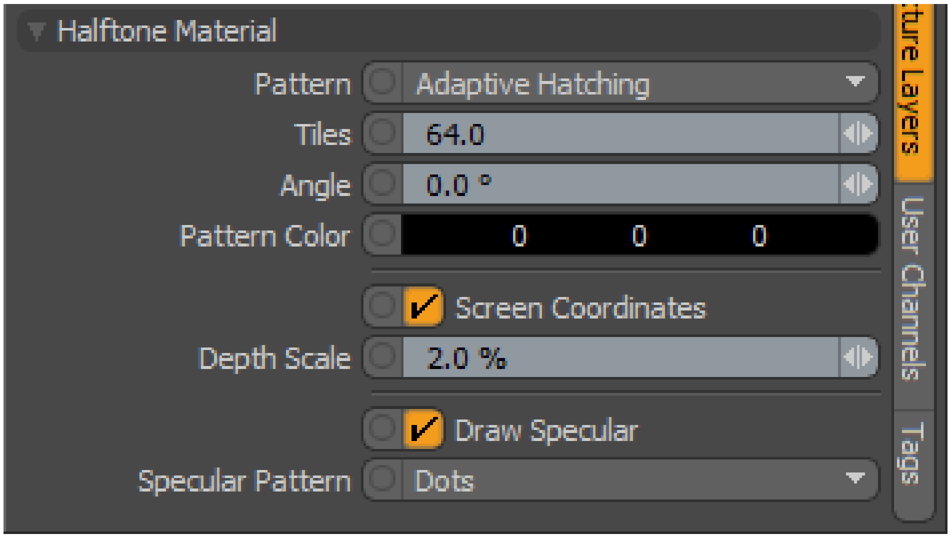
Layer
☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of

	toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Halftone Material Properties

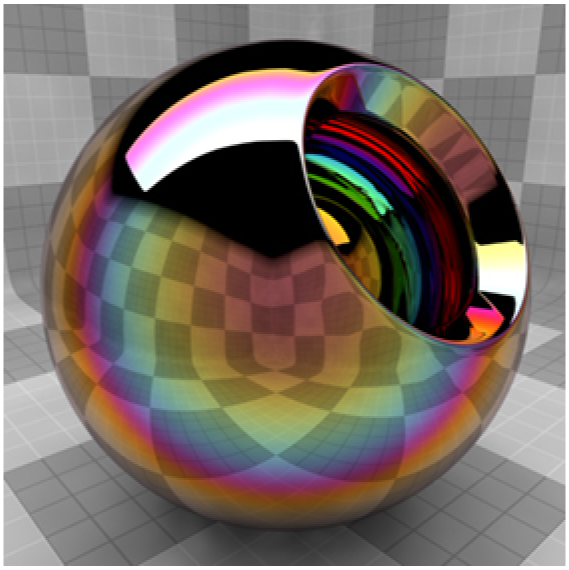

Halftone Material
Pattern: Adaptive Hatching
Tiles: 64.0
Angle: 0.0 °
Pattern Color: 0 0 0
☑ Screen Coordinates
Depth Scale: 2.0 %
☑ Draw Specular
Specular Pattern: Dots

Halftone Material	
Pattern	Determines the design motif used to create the halftone pattern across the surface: • **Dots** - applies a series of variable sized circles that simulating surface shading. • **Lines** - applies a series of lines of variable width simulating surface shading. • **Crosshatch** - applies a series of perpendicular lines of variable width simulating surface shading. • **Adaptive Hatching** - simulates the technique of woodcut shading where the lines can't get very thin. At the lightest shading, it draws a dashed line which becomes more solid and thicker as it gets darker. A perpendicular cross hatch is added for the darkest tones.
Tiles	Determines the size or spacing of the halftone **Pattern**. Larger values result in finer pattern spacing, while smaller values produce a larger more apparent pattern.
Angle	Determines the angle of the effect, relative to the final rendered frame. An **Angle** of **0** degrees produces an ordered grid or horizontal flow to the pattern, and a **45** degree angle produces an effect similar to a traditional black and white newsprint. **90** degrees produces a vertical flow to the pattern.
Pattern Color	Determines the color of the halftone pattern added to the surfaces diffuse color.
Screen Coordinates	When enabled, the **Pattern** is drawn relative the frame itself, based on a user defined **Angle** value. When disabled, the UV map defined in the associated Texture Locator determines the direction and angle of the halftone pattern, giving you ultimate control of halftone pattern results.
Depth Scale	When **Screen Coordinates** is enabled, the **Depth Scale** value can be applied to simulate the **Pattern** UV map contouring without actually using a UV map. This is done by scaling the **Tiles** values as the surface recedes into the distance. The **Depth Scale** amount controls how much the depth affects the **Tiles** scaling.
Draw Specular	When enabled, any specular highlights created by direct light sources in the scene are added as an additional white pattern overlaid on the surface.
Specular Pattern	Determines the design motif used to create the specular overlay halftone pattern: • **Dots** - applies a series of variable sized circles simulating surface shading. • **Lines** - applies a series of lines of variable width simulating surface shading. • **Crosshatch** - applies a series of perpendicular lines of variable width simulating surface shading.

	- **Adaptive Hatching** - simulates the technique of woodcut shading where the lines can't get very thin. At the lightest shading, it draws a dashed line which becomes more solid and thicker as it gets darker. A perpendicular cross hatch is added for the darkest tones.

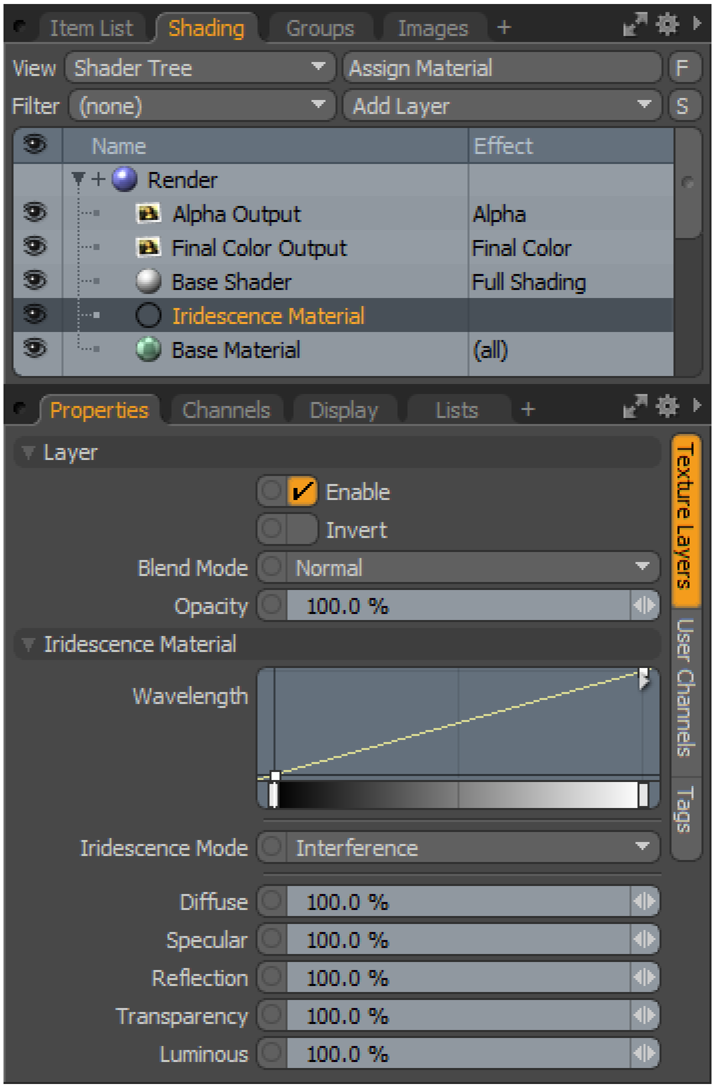
Iridescence Material

Iridescence is a general term for colors which change with viewing angle. The effect is typically due either to interference or diffraction. Diffraction is the splitting of light into component wavelengths. Interference is caused by the strengthening and weakening of different wavelengths, and often results in a characteristic spectrum of **Newton's Colors** or **Newton's Rings**. Both are seen in nature such as in pearls, butterfly wings, oily puddles, and peacock feathers.



Usage

The **Iridescence Material** gives you powerful control over viewing-angle-dependent effects of either type. Adding keyframes to the gradient control allows you to set the color at the desired viewing angle, from straight-on to perpendicular glancing. Applying a texture to **Iridescence Colors** modulates the coloring, producing interesting results. When the **Iridescence Material** is selected in the **Shader Tree**, the attributes appear in the **Properties** panel.


Item List
Shading
Groups
Images
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Iridescence Material
Base Material
(all)
Properties
Channels
Display
Lists
Layer
Texture Layers
User Channels
Tags
Iridescence Material
Wavelength

☑ Enable

☐ Invert

Blend Mode Normal

Opacity 100.0 %

Iridescence Mode Interference

Diffuse 100.0 %

Specular 100.0 %

Reflection 100.0 %

Transparency 100.0 %

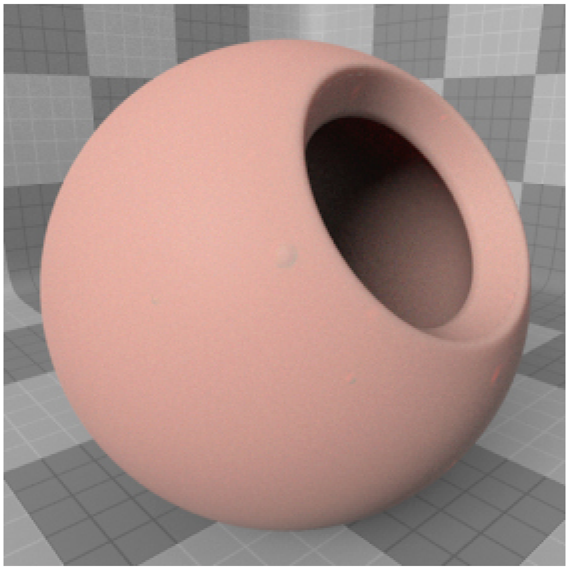
Luminous 100.0 %

Option	Description
Layer	
Enable	Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the layer producing a negative effect.
Blend Mode	Affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, please see Layer Blend Modes.

Option	Description
Opacity	Changes the transparency of the current layer. Reducing this values increasingly reveals lower layers in the **Shader Tree** if present, or dims the effect of the layer itself on the surface.
Iridescence Material	
Wavelength	In **Diffraction** mode, the vertical keyframe position corresponds to the color in the visible spectrum, which repeats as it goes higher. In **Interference** mode, the keyframe heights correspond the optical path difference giving rise to the interference. It may appear repetitive, but does not actually repeat.
Iridescence Mode	Allows you to choose between two different modes, **Diffraction** and **Interference**, each producing their own unique effect.
Surface Attributes	These controls, **Diffuse**, **Specular**, and so on, adjust the strength of the coloration applied to the named **Material** property. This allows you to selectively control which attributes of the surface receive the iridescent coloration and to what degree it is applied.

Skin Material

The **Skin Material** is designed to completely replace the default BRDF shading calculations of the standard Material item when applied to a surface. Human skin is a rather complex organ, made up of multiple layers and materials, each absorbing, scattering, and reflecting light differently creating a very subtle but distinct look that we as humans are intimately familiar with. In order to simulate such a complex surface a special **Material** shader was required. The material is a made up of mostly familiar options, that combine to simulate the subtle nuance that is skin.



Usage

The **Skin Material** item is added to the **Shader Tree** using the **Add Layer** > **Custom Materials** > **Skin Material** function.

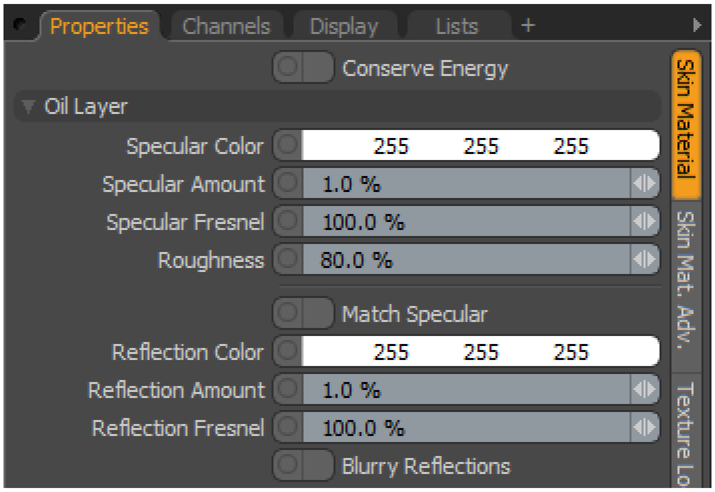
Note: For more information on adding and manipulating Shader Tree layers, please see Shader Tree.

For the best results, the **Skin Material** should be placed above the **Material Item** it is replacing (feel free to disable or remove the **Material Item**). Once positioned, selecting the **Skin Material** item reveals its attributes in the **Properties** viewport where its settings may be adjusted to fine tune the look of the rendered skin. Skin is not a consistent color across its entire surface, so you may want to add texture layers, set to the appropriate effect to represent the natural variations inherent in skin, to produce the most realistic results. There are a number of special layer effect types specific to the **Skin Material** under **Custom Material Channels** > **Skin Material Channels** that correspond to the settings of the **Skin Material** itself.

Note: Since the **Skin Material** completely replaces the **Material Item**'s settings, in order to obtain the best results for **Bump Maps** and **Displacement Maps**, you may need to adjust the respective attributes directly in the **Skin Shader** properties panel.

Skin Material Properties

Oil Layer



○ Conserve Energy

Oil Layer

Specular Color 255 255 255

Specular Amount 1.0 %

Specular Fresnel 100.0 %

Roughness 80.0 %

○ Match Specular

Reflection Color 255 255 255

Reflection Amount 1.0 %

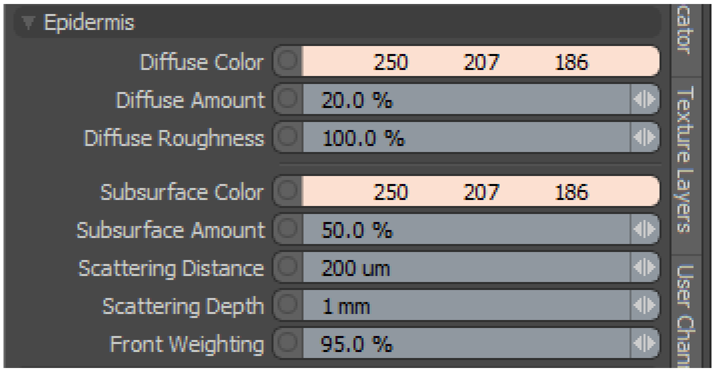
Reflection Fresnel 100.0 %

○ Blurry Reflections

Conserve Energy	When enabled, the material maintains a physically accurate blend of **Diffuse** and **Specular** light reflecting over the skins surface. It also accurately spreads the energy of each contribution to the oil and dermal layers so that the overall contribution is never more than 100%.
Oil Layer - The uppermost layer of skin excretes an oil that produces a shiny, specular reflection on its surface. Depending on the amount on the skin, it becomes less or more reflective. Skin that is flushed, say from exercise or exertion, is shinier than clean skin or skin with powder make-up on, so the **Amount** and **Roughness** should be adjusted depending on the desired look.	
Specular Color	Determines the color of the highlight created from reflections of direct light sources.
Specular Amount	Determines the strength of the specular highlight.
Specular Fresnel	At glancing angles (those perpendicular to the camera), reflectance values of skin increase. The **Specular Fresnel** setting realistically increases the specular amount at these glancing angles producing a physically correct surface.
Roughness	This control determines the size of the spread for both **Specular** and **Blurry Reflections**. A higher **Roughness** value produces these rougher, more spread out highlights, while lower values produce tighter more focused highlights.

	For example, a clay pot reflects light very differently than a porcelain cup. Though both surfaces appear to be smooth, on a microscopic level the clay pot is much rougher than the porcelain and this affects how the light is reflected, spreading the reflection across the surface.
Match Specular	When enabled, the **Reflection** options are automatically adjusted to match the equivalent specular settings.
Reflection Color	Determines the tinting or colorization of **Reflection** on the surface.
Reflection Amount	Determines the amount of the environment reflections that appear on the surface, like with the specular setting, oilier surfaces appear more reflective.
Reflection Fresnel	At glancing angles (those perpendicular to the camera), reflectance values of skin increase. The **Reflection Fresnel** setting realistically increases the reflection amount at these glancing angles producing a physically correct surface.
Blurry Reflections	When enabled, reflections take in to account the **Roughness** option and spread reflections across the surface, such as would happen with real skin.

Epidermis



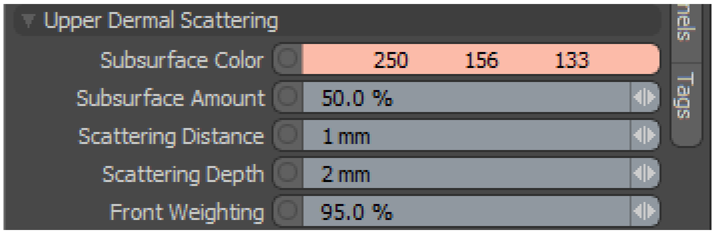
Epidermis
Diffuse Color 250 207 186
Diffuse Amount 20.0 %
Diffuse Roughness 100.0 %
Subsurface Color 250 207 186
Subsurface Amount 50.0 %
Scattering Distance 200 um
Scattering Depth 1 mm
Front Weighting 95.0 %

Epidermis - The epidermis is the outer layer of effectively dead skin cells. This layer is quite thin and on a microscopic level, looks very much like tiny overlapping scales. This layer is mainly responsible for the color of the skin, depending on the actual amounts of melanin present in the cells.

Diffuse Color	Determines the **Diffuse Color** of the skins surface. This is the apparent visual color of the skin's outer surface that reflects back when illuminated.
Diffuse Amount	Determines the amount of light reflected back from the skins surface, the rest of the

	light is absorbed for subsurface scattering (SSS).
Diffuse Roughness	The **Diffuse Roughness** control represents a lack of specular on a Epidermal surface, reading the **Diffuse** shading outwards across the surface, simulating the tiny microscopic variations of the individual skin cells.
Subsurface Color	Determines the color of the light that is absorbed, scattered and the reflected back out. The epidermis is fairly translucent and the subsurface color is basically the same color as the **Diffuse Color** for human skin.
Subsurface Amount	Determines the amount of scattered light that is reflected back out. The subsurface scattering is combined with the **Diffuse Color** to produce the final epidermal layer.
Scattering Distance	Determines the distance that light scatters through the surface before reflecting back out. Deeper rays attenuate toward the SSS color, reaching the full SSS color at the **Scattering Distance** value.
Maximum Depth	Determines the maximum depth to which a ray can contribute to the final solution, in essence clipping the SSS to approximate a solid core. A value of **0** disables this option.
Front Waiting	The **Front Weighting** option biases the SSS ray forward or backward. Settings between **0** - **49** bias the SSS rays backwards, **51** - **100** bias the rays forward, and a setting of **50** distributes the rays equally in both directions.

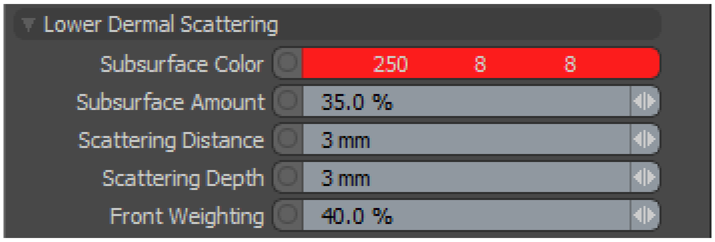
Upper Dermal Scattering


Upper Dermal Scattering
Subsurface Color 250 156 133
Subsurface Amount 50.0 %
Scattering Distance 1 mm
Scattering Depth 2 mm
Front Weighting 95.0 %

Upper Dermal Scattering - The upper dermal layer contains tissue for cushioning the epidermis, and also houses the nerve endings, hair follicles, sweat glands, and blood vessels. Because of these constructs, the light scattered from here is differently colored than the epidermal layer.	
Subsurface Amount	Determines the amount of scattered light that is reflected back out. **Upper Dermal Scattering** is combined with the Epidermal layer solution to produce the final

	shaded result.
Scattering Distance	Determines the distance that light scatters through the surface before reflecting back out. Deeper rays attenuate toward the SSS color, reaching the full SSS color at the **Scattering Distance** value.
Maximum Depth	Determines the maximum depth to which a ray can contribute to the final solution, in essence clipping the SSS to approximate a solid core. A value of **0** disables this option.
Front Weighting	The **Front Weighting** option biases the SSS ray forward or backward. Settings between **0** - **49** bias the SSS rays backwards, **51** - **100** bias the rays forward, and a setting of **50** distributes the rays equally in both directions.

Lower Dermal Scattering



Lower Dermal Scattering

Subsurface Color 250 8 8

Subsurface Amount 35.0 %

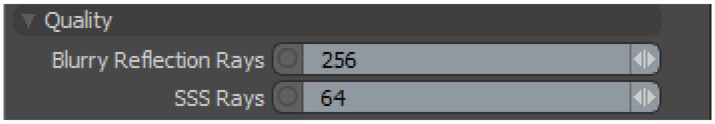
Scattering Distance 3 mm

Scattering Depth 3 mm

Front Weighting 40.0 %

Lower Dermal Scattering - The lower dermal area contains more blood vessels and typically is the layer that contributes the most red because of this.	
Subsurface Amount	Determines the amount of scattered light that is reflected back out. **Lower Dermal Scattering** is combined with the Upper Dermal and Epidermal layer solutions to produce the final shaded result.
Scattering Distance	Determines the distance that light scatters through the surface before reflecting back out. Deeper rays attenuate toward the SSS color, reaching the full SSS color at the **Scattering Distance** value.
Maximum Depth	Determines the maximum depth to which a ray can contribute to the final solution, in essence clipping the SSS to approximate a solid core. A value of **0** disables this option.
Front Weighting	The **Front Weighting** option biases the SSS ray forward or backward. Settings between **0** - **49** bias the SSS rays backwards, **51** - **100** bias the rays forward, and a setting of **50** distributes the rays equally in both directions.

Quality

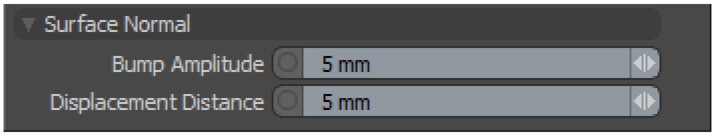


Blurry Reflection Rays: 256

SSS Rays: 64

Quality	
Blurry Reflection Rays	Defines the number of rays used to calculate **Blurry Reflections** when that option is enabled. Higher ray counts produce smoother, more realistic results, but also result in longer calculation times when rendering
SSS Rays	Defines the number of rays used to calculate the SSS results. Higher ray counts produce more accurate SSS results, but also take longer to calculate.

Surface Normal



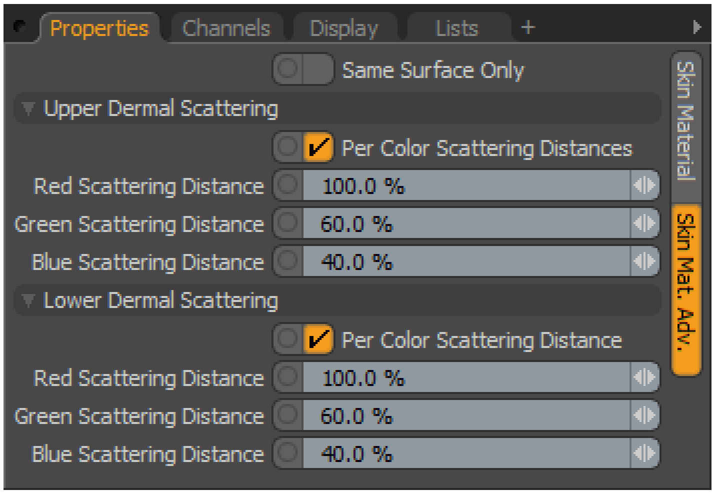
Bump Amplitude: 5 mm

Displacement Distance: 5 mm

Surface Normal	
Bump Amplitude	Determines the maximum strength of any bump map applied to the skin surface.
Displacement Distance	Determines the maximum displacement distance when any type of displacement texture layer is used.

Skin Material Advanced

Since the **Skin Material** uses a custom shading model to render the surface, it is not compatible with blending or masking techniques allowed by other layers. Basically this means that any surface with the skin shader applied can **only** be a skin surface.



Same Surface Only

Upper Dermal Scattering

☑ Per Color Scattering Distances

Red Scattering Distance 100.0 %

Green Scattering Distance 60.0 %

Blue Scattering Distance 40.0 %

Lower Dermal Scattering

☑ Per Color Scattering Distance

Red Scattering Distance 100.0 %

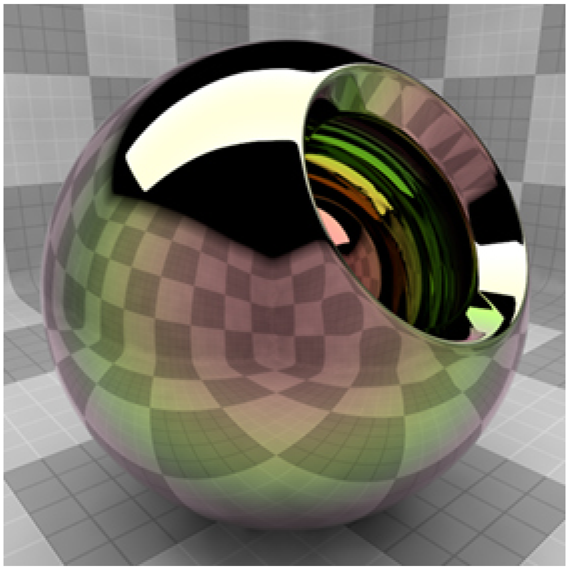
Green Scattering Distance 60.0 %

Blue Scattering Distance 40.0 %

Option	Description
Upper Dermal Scattering	
In real skin, each wavelength of light penetrates the skin to a different depth, which attenuates the colors slightly as rays travel further into the surface. For the most accurate results you can enable and adjust these settings, but they also increase render times as individual rays need to be fired for each wavelength calculation.	
Per Color Scattering Distances	When enabled, this option calculates per color scattering for SSS of the **Upper Dermal** layer.
Red/Green/Blue	These controls allow you to adjust how each wavelength attenuates through the surface. Red, being the highest wavelength, should penetrate further than the lower wavelengths of green and then blue. Values of **100** travel the maximum **Scattering Distance** specified.
Lower Dermal Scattering	
Per Color Scattering Distances	When enabled, this option calculates per color scattering for SSS of the **Lower Dermal** layer.
Red/Green/Blue	These controls allow you to adjust how each wavelength attenuates through the surface. Red, being the highest wavelength, should penetrate further than the lower wavelengths of green and then blue. Values of **100** travel the maximum **Scattering Distance** specified.

Thin Film Material

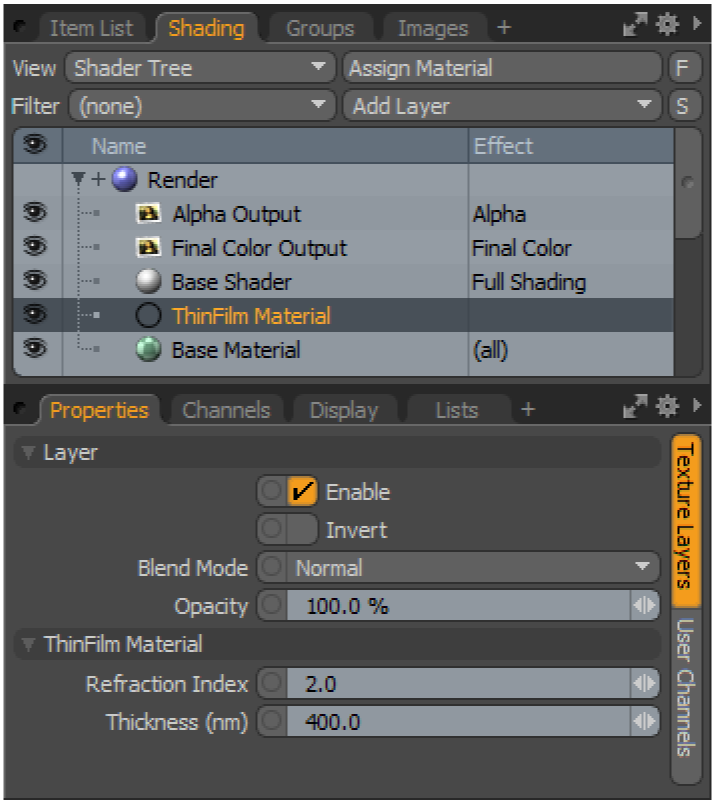
The **Thin Film** material works by simulating interference effects on the wavelengths of incoming light, refracting them between two virtual thin layers of film, like you would see with thin layers of oil over a puddle of water. Once added, adjust the **Thickness** and **Refraction Index** to recolor any specular and mirror reflections. You can also apply a texture to the **Thickness** to simulate the oil-on-water look, or texture the **Refraction Index** to achieve the required look.



Note: For physical correctness, the shader needs to look at the **Refraction Index** of the underlying material. If it is less than the **Refraction Index** of the **Thin Film** material, a **90°** phase shift is applied, making the colors change abruptly.

Usage

The **Thin Film** item is added to the Shader Tree using the **Add Layer** > **Custom Materials** > **Thin Film** function. For information regarding adding and working with Shader Tree Items Layers, please see Shader Tree. Once added, you can adjust the **Refraction Index** and **Thickness** values. You can also add additional texture layers and select the Effect type **Custom Material Channels** > **Thin Film Material Channels** to modulate those settings. When the **Thin Film** item is selected, the attributes appears in the **Properties** panel.



View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
ThinFilm Material	
Base Material	(all)

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

ThinFilm Material

Refraction Index: 2.0

Thickness (nm): 400.0

Option	Description
Layer	
Enable	Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the layer producing a negative effect.
Blend Mode	Affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, please see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. Reducing this values increasingly reveals lower layers in the **Shader Tree** if present, or dims the effect of the layer itself on the surface.
Thin Film Material	
Refraction Index	The **Refraction Index** value determines how much the ray of light is bent, which in

Option	Description
	terms of the **Thin Film Material** controls the coloration of the resulting light ray.
Thickness (nm)	This value determines the difference in thickness between the two virtual film layers. Modulating this value with a texture layer provides the most realistic results (possibly with the Swirl procedural texture set to **Bubble** as the noise type).

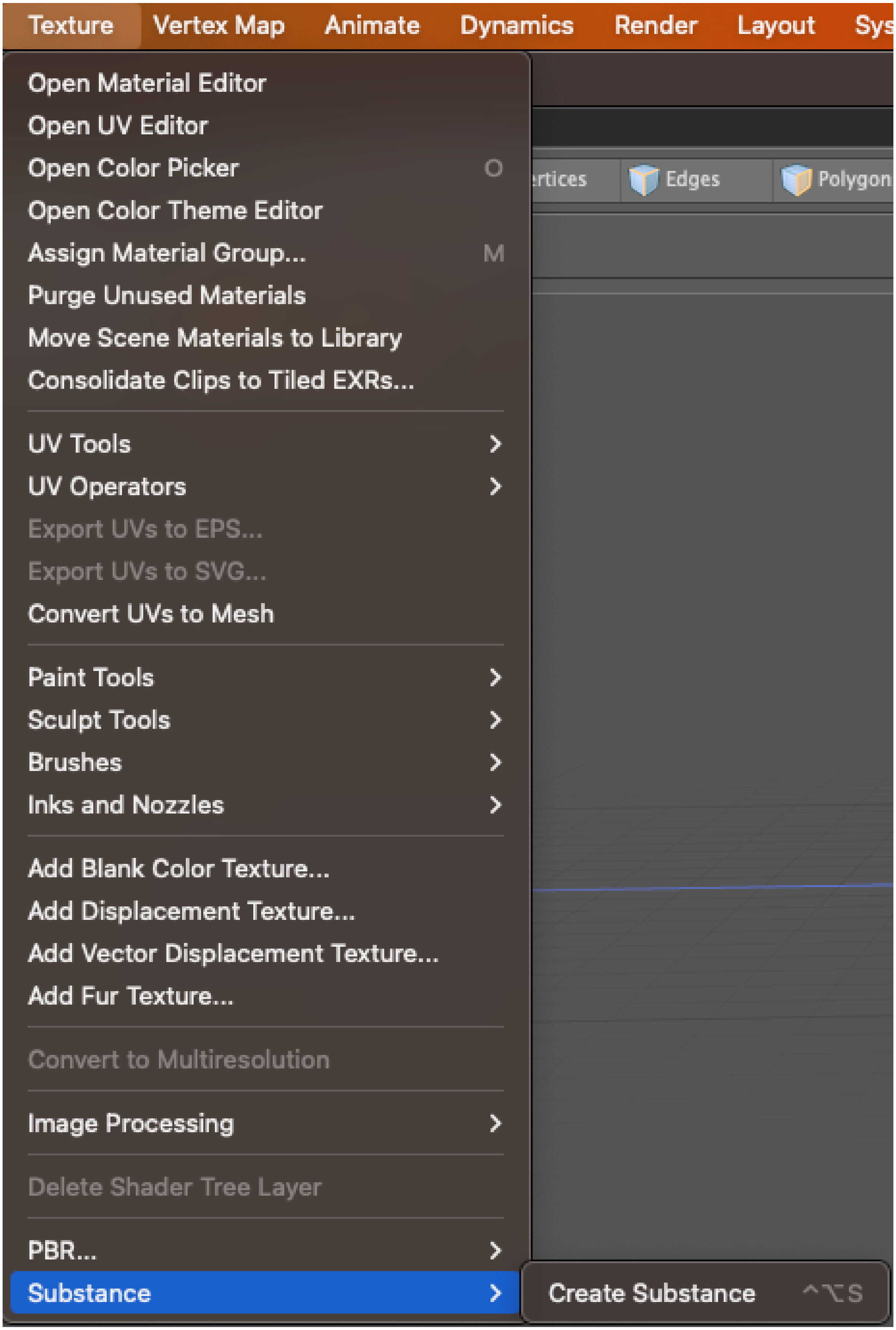
Substance Material

Substance support is built-in to Modo, enabling you to use Substance Materials directly in Modo.

Adding a Substance Material

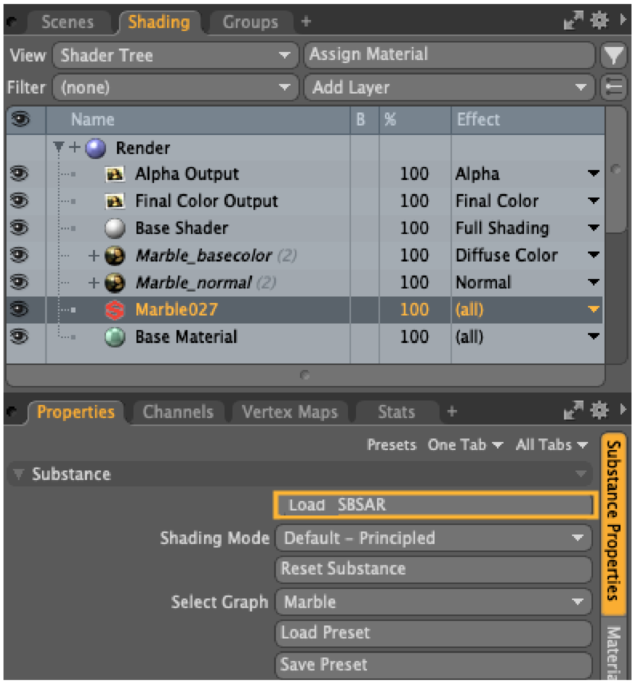
To add a Substance Material to the Shader Tree:

1. Open the **Texture** menu.
2. Select **Substance** > **Create Substance**.


Texture
Vertex Map
Animate
Dynamics
Render
Layout
Sys
Open Material Editor
Open UV Editor
Open Color Picker O
Open Color Theme Editor
Assign Material Group... M
Purge Unused Materials
Move Scene Materials to Library
Consolidate Clips to Tiled EXRs...
UV Tools
UV Operators
Export UVs to EPS...
Export UVs to SVG...
Convert UVs to Mesh
Paint Tools
Sculpt Tools
Brushes
Inks and Nozzles
Add Blank Color Texture...
Add Displacement Texture...
Add Vector Displacement Texture...
Add Fur Texture...
Convert to Multiresolution
Image Processing
Delete Shader Tree Layer
PBR...
Substance
Create Substance ^⌥S
rtices
Edges
Polygon

Adjusting a Substance Material

- Select the Substance Material you've created in the Shader Tree (*Marble027* in the example below).
- In the **Properties** panel, select **Load SBSAR**.


Scenes
Shading
Groups
Assign Material
Add Layer
Properties
Channels
Vertex Maps
Stats
Presets
One Tab
All Tabs
Substance
Load SBSAR
Reset Substance
Load Preset
Save Preset
Substance Properties
Materia

View: Shader Tree

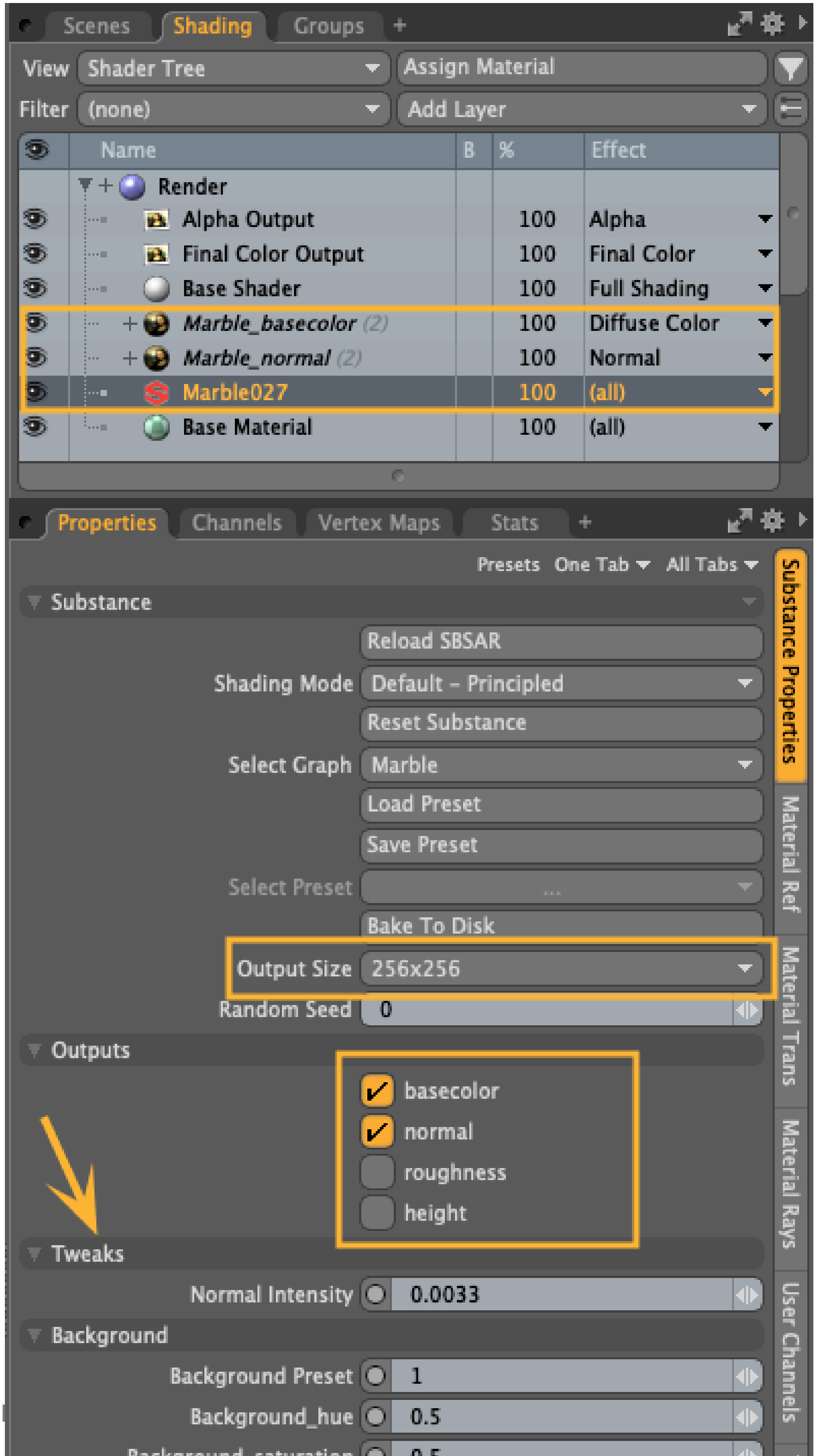
Filter: (none)

Name	B	%	Effect
Render			
Alpha Output		100	Alpha
Final Color Output		100	Final Color
Base Shader		100	Full Shading
Marble_basecolor (2)		100	Diffuse Color
Marble_normal (2)		100	Normal
Marble027		100	(all)
Base Material		100	(all)

Shading Mode: Default – Principled

Select Graph: Marble

- In the **Properties** panel, based on the specific .sbsar file loaded, various options will be available under **Tweaks**, allowing you to modify the appearance of the procedural texture.
- To create **Outputs**, select the effect checkboxes required (such as *basecolor* and *normal* in this example). This will create image maps and add them to the Shader Tree with the correct effect (*Marble_basecolor* and *Marble_normal* here). You can adjust the size of these in the **Output Size** drop down.


Scenes
Shading
Groups
Assign Material
Add Layer
Properties
Channels
Vertex Maps
Stats
Presets
One Tab
All Tabs
Substance
Reload SBSAR
Reset Substance
Load Preset
Save Preset
Bake To Disk
Outputs
Tweaks
Background
Substance Properties
Material Ref
Material Trans
Material Rays
User Channels

View: Shader Tree

Filter: (none)

Name	B	%	Effect
Render			
Alpha Output		100	Alpha
Final Color Output		100	Final Color
Base Shader		100	Full Shading
Marble_basecolor *(2)*		100	Diffuse Color
Marble_normal *(2)*		100	Normal
Marble027		100	(all)
Base Material		100	(all)

Shading Mode: Default – Principled

Select Graph: Marble

Select Preset: ...

Output Size: 256x256

Random Seed: 0

☑ basecolor
☑ normal
☐ roughness
☐ height

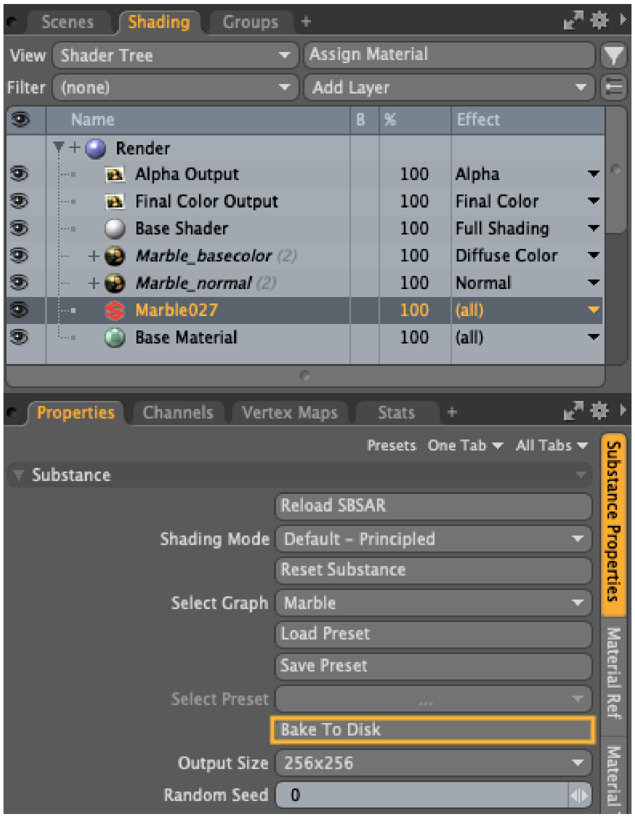
Normal Intensity: 0.0033

Background Preset: 1

Background_hue: 0.5

Bake to Disk

Use the **Bake to Disk** option to export the created image maps to disk for use in Modo, or to export for other software.


Scenes
Shading
Groups
+
Assign Material
Add Layer
Properties
Channels
Vertex Maps
Stats
+
Presets
One Tab
All Tabs
Substance
Reload SBSAR
Reset Substance
Load Preset
Save Preset
Bake To Disk
Substance Properties
Material Ref
Material

View: Shader Tree
Filter: (none)

Name	B	%	Effect
Render			
Alpha Output		100	Alpha
Final Color Output		100	Final Color
Base Shader		100	Full Shading
Marble_basecolor (2)		100	Diffuse Color
Marble_normal (2)		100	Normal
Marble027		100	(all)
Base Material		100	(all)

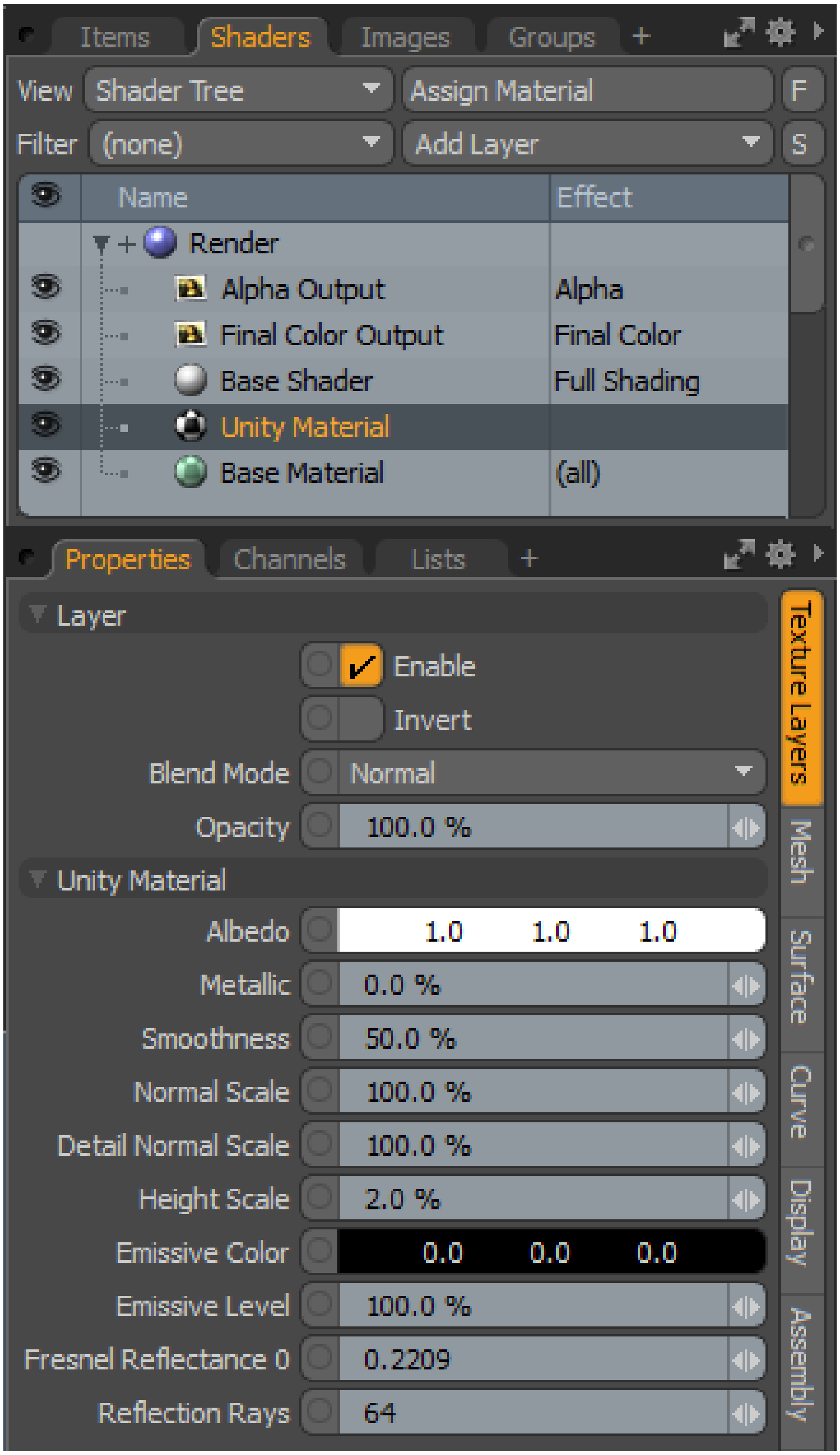
Shading Mode: Default - Principled
Select Graph: Marble
Select Preset: ...
Output Size: 256x256
Random Seed: 0

Unity Material

Unity Material is a new PBR (physically based rendering) material, which introduces the standard material of Unity 5 into Modo, as a custom material associated with dedicated shaders for offline render, preview, and advanced viewports.

To create a Unity Material, in the **Shader Tree**, click **Add Layer** > **Custom Materials** > **Unity Material**. The material is added to the **Shader Tree** and the layer's **Properties** panel displays.

Tip: Import textures for your Unity material using the PBR loader. See PBR Textures for more information.


Items
Shaders
Images
Groups
+
Assign Material
F
Add Layer
S
Properties
Channels
Lists
+
Texture Layers
Mesh
Surface
Curve
Display
Assembly

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Unity Material	
Base Material	(all)

Layer

Enable	☑
Invert	☐
Blend Mode	Normal
Opacity	100.0 %

Unity Material

Albedo	1.0 1.0 1.0
Metallic	0.0 %
Smoothness	50.0 %
Normal Scale	100.0 %
Detail Normal Scale	100.0 %
Height Scale	2.0 %
Emissive Color	0.0 0.0 0.0
Emissive Level	100.0 %
Fresnel Reflectance 0	0.2209
Reflection Rays	64

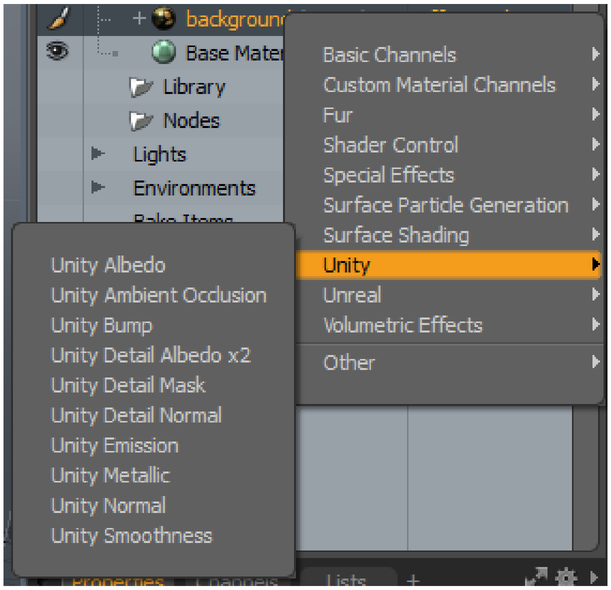
Layer	
Enable	Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the layer producing a negative effect.
Blend Mode	Affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, please see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the **Shader Tree** if present, or dims the effect of the layer itself on the surface.
Unity Material	
Albedo	This value specifies the base color of the surface.
Metallic	This option specifies how metallic your surface is. A value of **0** specifies a non-metallic surface, while a value of **1** results in a metal surface.
Smoothness	This value specifies how smooth the surface is. A surface with a smoothness of **0** reflects light in a uniform way, whereas rough surfaces reflect light at different angles.
Normal Scale	Scales for the normal map in the Main Maps of Unity 5.
Detail Normal Scale	Scales for the normal map in the Secondary Maps of Unity 5.
Height Scale	Scales for the Height Map in Unity 5.
Emissive Color	Specifies which part of the material should glow due to emitting light.
Emissive Level	Specifies the level of emission.
Fresnel Reflectance 0	Determines the reflection of light.
Reflection Rays	This property refers to baking with **F9** or **Preview** render. Specifies the number of rays used for sampling reflection. A higher number of rays produces better rendering quality, but increases render time.

Textures for Unity Material

Modo supports texturing for Unity Material. To add a texture, follow these steps:

1. In the **Shader Tree**, click **Add Layer** > **Image Map** > **(load image)**.

   A file browser opens.
2. Navigate to the file you'd like to use and click **Open**.
3. Right-click the **Effect** column of the texture layer in the **Shader Tree**.
4. Select **Unity** and choose from the available effects.


background
Base Mater
Library
Nodes
Lights
Environments
Bake Items
Basic Channels
Custom Material Channels
Fur
Shader Control
Special Effects
Surface Particle Generation
Surface Shading
Unity
Unreal
Volumetric Effects
Other
Unity Albedo
Unity Ambient Occlusion
Unity Bump
Unity Detail Albedo x2
Unity Detail Mask
Unity Detail Normal
Unity Emission
Unity Metallic
Unity Normal
Unity Smoothness
Properties
Channels
Lists

The following texture effects are available:

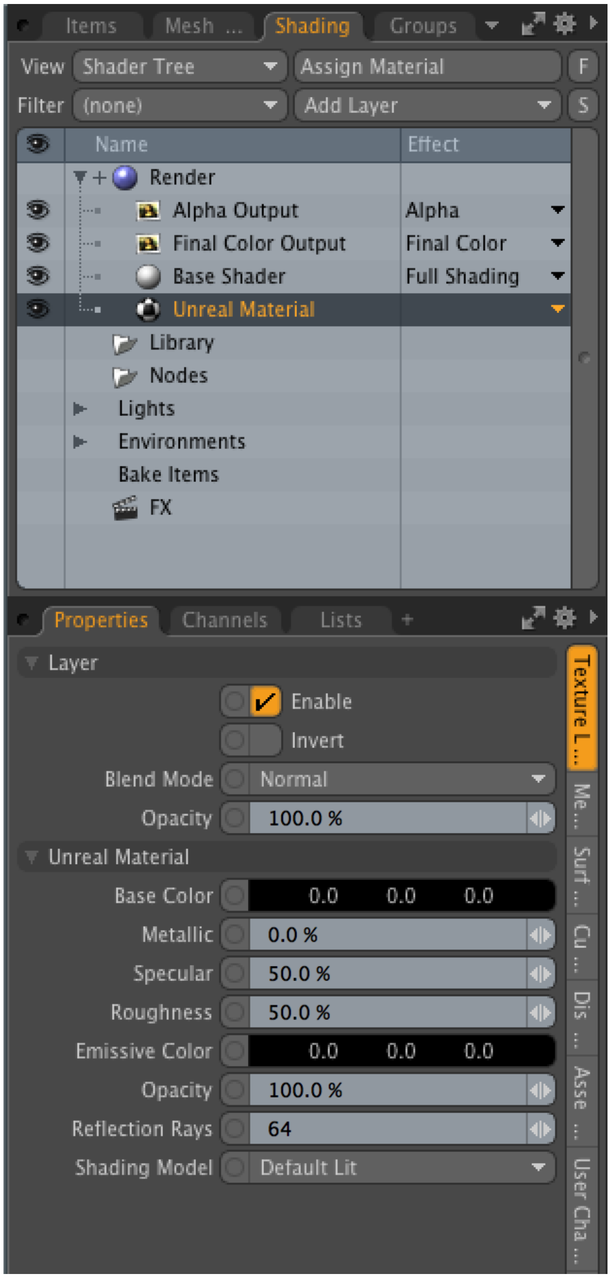
- Albedo
- Ambient Occlusion
- Bump (This can be achieved by using a height map in Unity. In Modo, the effect is directly specified.)
- Detail Albedo x2 (The Detail Albedo x2 in the Secondary Maps section of Unity 5.)
- Detail Mask
- Detail Normal (The normal map in the Secondary Maps section of Unity 5.)
- Emission
- Metallic
- Normal
- Smoothness

Unreal Material

Unreal Material is a new PBR (physically based rendering) material, which introduces the standard material of Unreal Engine 4 into Modo, as a custom material associated with dedicated shaders for offline render, preview, and advanced viewports.

To create an Unreal Material, in the **Shader Tree**, click **Add Layer** > **Custom Materials** > **Unreal Material**. The material is added to the **Shader Tree** and the layer's **Properties** panel displays.

Tip: Import textures for your Unreal material using the PBR loader. See PBR Textures for more information.


Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Name
Effect
Properties
Channels
Lists
+
Texture L...
Me...
Surf...
Cu...
Dis...
Asse...
User Cha...

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Unreal Material	
Library	
Nodes	
Lights	
Environments	
Bake Items	
FX	

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Unreal Material

Base Color: 0.0 0.0 0.0

Metallic: 0.0 %

Specular: 50.0 %

Roughness: 50.0 %

Emissive Color: 0.0 0.0 0.0

Opacity: 100.0 %

Reflection Rays: 64

Shading Model: Default Lit

Layer	
Enable	Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the layer producing a negative effect.

Layer	
Blend Mode	Affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the **Shader Tree** if present, or dims the effect of the layer itself on the surface.
Unreal Material	
Base Color	Specifies the color of the material. Specify **R**, **G**, and **B** color values.
Metallic	This option specifies how metal-like your surface is. A value of **0** specifies a non-metallic surface, while a value of **1** results in a metal surface.
Specular	Scales the amount of specularity on non-metallic surfaces. It has no effect on metallic surfaces.
Roughness	Specifies the roughness of the material.
Emissive Color	Specifies which part of the material should glow due to emitting light.
Opacity	Specifies the transparency of the material.
Reflection Rays	This property refers to baking with **F9** or **Preview** render. Specifies the number of rays used for sampling reflection. A higher number of rays produces better rendering quality, but increases render time.
Shading Model	This controls how your material reflects light. This option determines how Modo creates the final result of the material with lighting. The following options are available: • **Default Lit** - This includes the most commonly used features, such as direct lighting, normal mapping, and reflections. • **Clear Coat** - This option is for applying and customizing clear coat texture effects, such as clear coat amount and clear coat roughness.

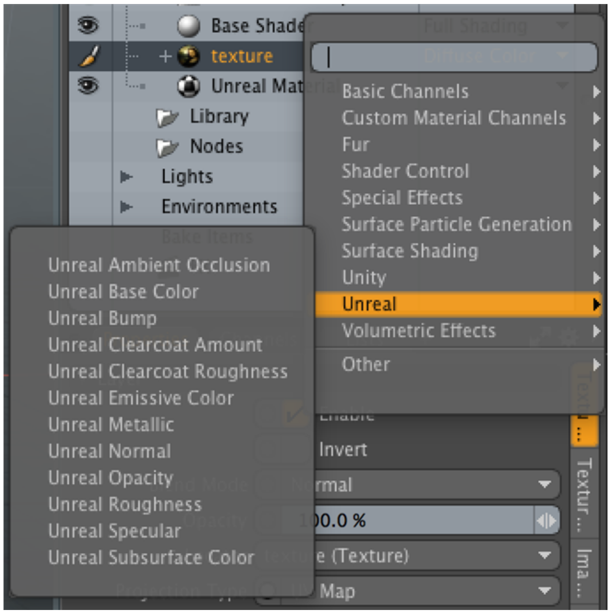
Textures for Unreal Material

Modo supports texturing for Unreal Material. To add a texture, follow these steps:

1. In the **Shader Tree**, click **Add Layer** > **Image Map** > **(load image)**.

   A file browser opens.

2. Navigate to the file you'd like to use and click **Open**.
3. Right-click the **Effect** column of the texture layer in the **Shader Tree**.
4. Select **Unreal** and choose from the available effects.


Base Shade
texture
Unreal Mat
Library
Nodes
Lights
Environments
Basic Channels
Custom Material Channels
Fur
Shader Control
Special Effects
Surface Particle Generation
Surface Shading
Unity
Unreal
Volumetric Effects
Other
Unreal Ambient Occlusion
Unreal Base Color
Unreal Bump
Unreal Clearcoat Amount
Unreal Clearcoat Roughness
Unreal Emissive Color
Unreal Metallic
Unreal Normal
Unreal Opacity
Unreal Roughness
Unreal Specular
Unreal Subsurface Color
Invert
rmal
00.0 %
e (Texture)
Map
Textur ...
Ima ...

The following texture effects are available:

- Ambient Occlusion
- Base Color
- Bump (This can be achieved by BumpOffset in Unreal Engine. In Modo, the effect is directly specified.)
- Clearcoat Amount
- Clearcoat Roughness
- Emissive Color
- Metallic
- Normal
- Opacity
- Roughness
- Specular
- Subsurface Color

Enhance:modo Textures

Enhance:modo Textures are a collection of 147 advanced procedurals layers, many with additional parameters that further extend the collection. The complete shader set allows the creation of an infinite number of different textures, patterns, and surfaces. The textures are all procedurally based, so they have no fixed resolution, providing a near infinite level of detail, even when applied with a UV map. In addition, all of the texture parameters can be animated so that many different special effects are possible.

You can add the following items to any scene from within the Shader Tree viewport, by left-clicking the **Add Layer** button in the full viewport window, and then selecting an item from the dropdown. For more information regarding working with Shader Tree items and layers and **Waveforms**, see Shader Tree and Waveforms, respectively.

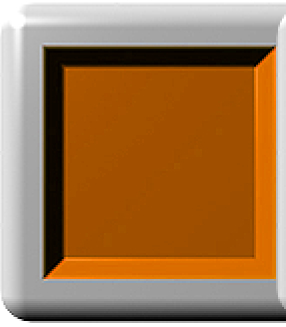
Display

Counter 1

Counter 2

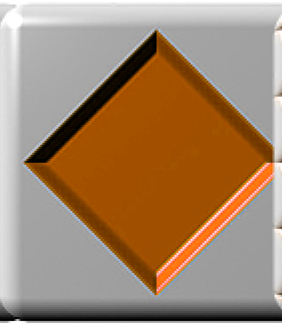
UVLEDs

Geometric

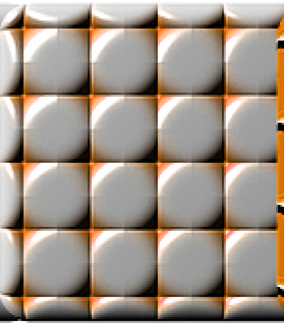


Box

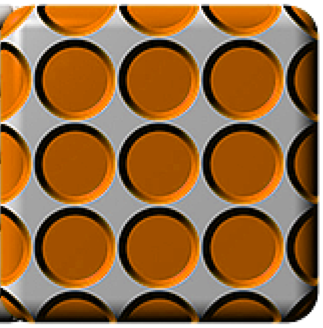
Circular



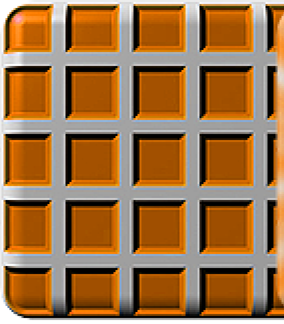
Corners



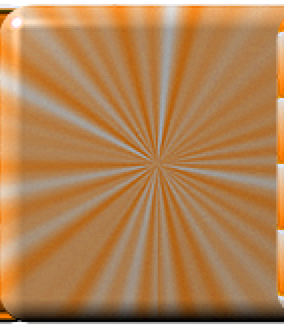
Cubic



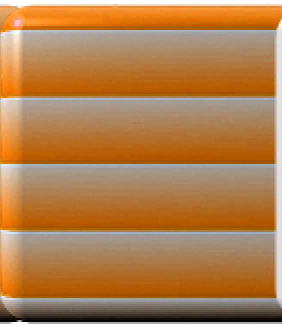
Dimples



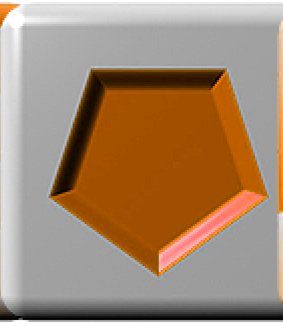
Grid



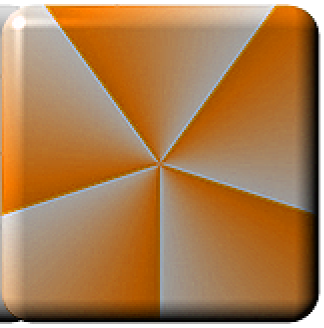
Iris



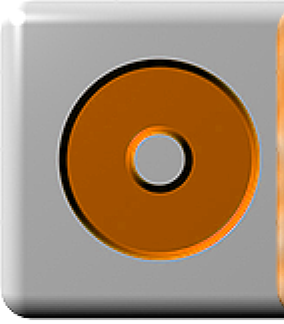
Linear



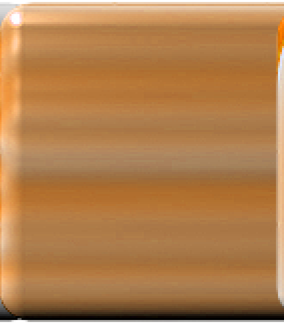
Polygon



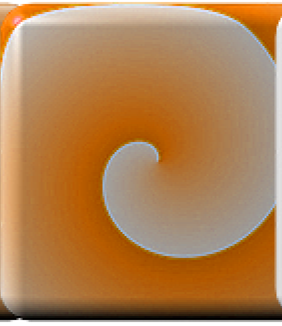
Radial



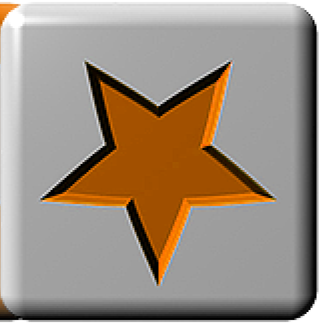
Ring



RndLinear (Random Linear)

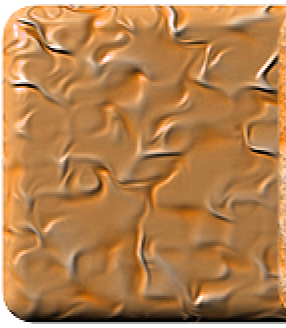


Spiral

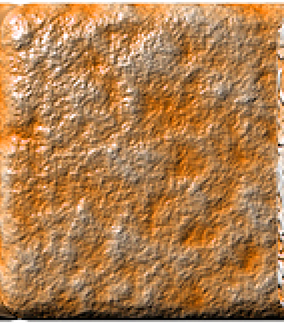


Star

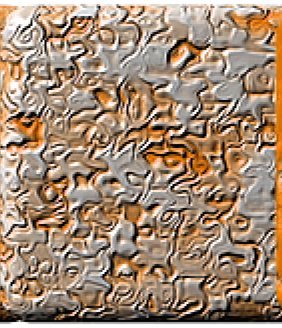
Noise



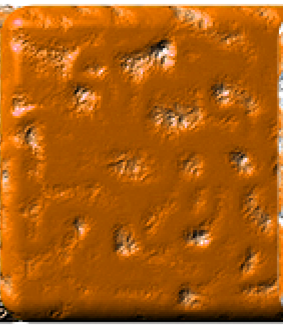
Agate



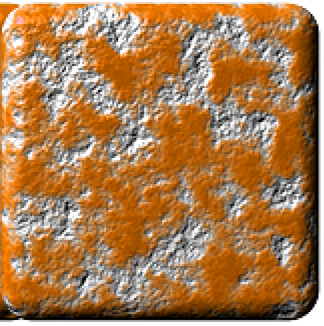
Bozo



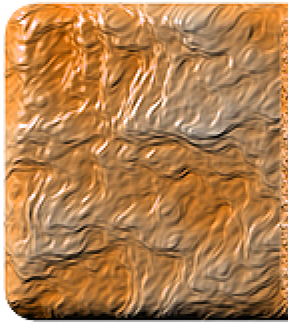
Cruddy

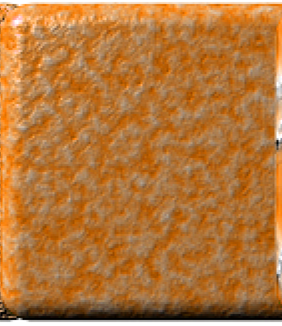


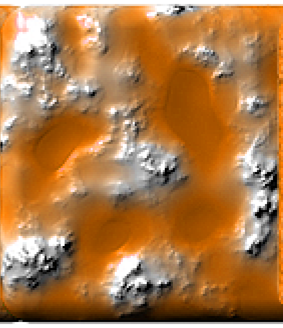
Dented



Etched







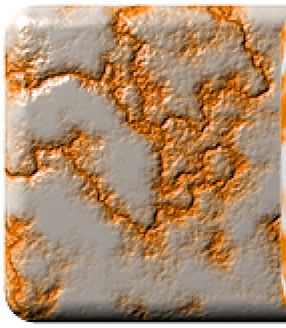
Flow Bozo

FBM

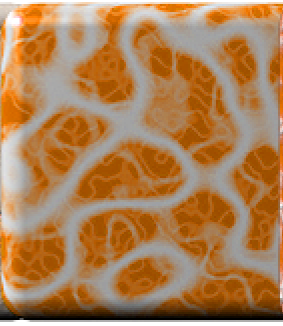
Granite

Hybrid

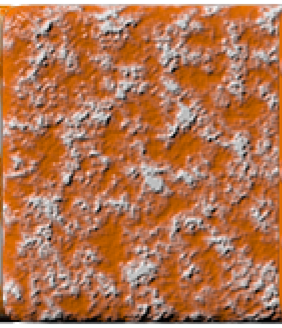
Lump



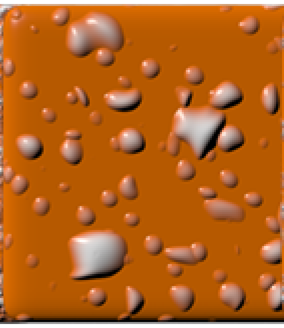
Marble Noise



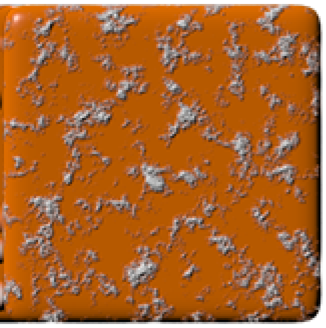
Marble Vein



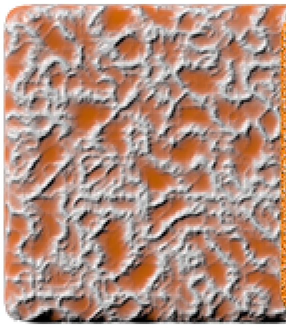
Multi-Fractal



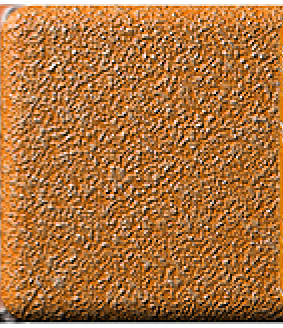
Pebbles



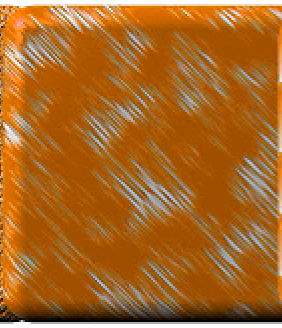
Puffy Clouds



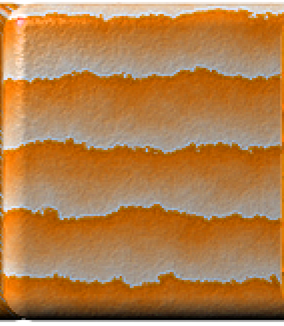
Ridged



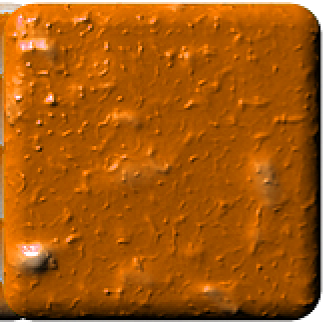
Scar



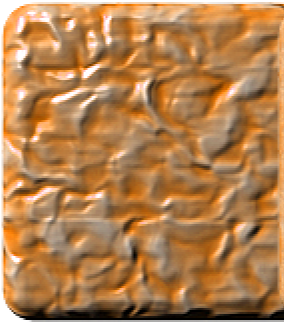
Scruffed



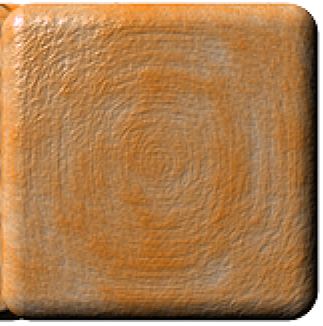
Strata



Stucco

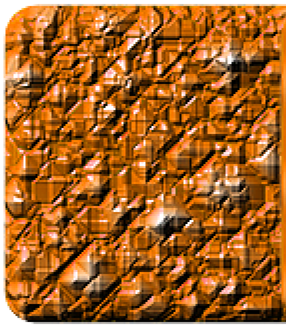


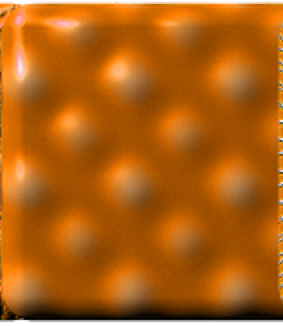
Vector Bozo

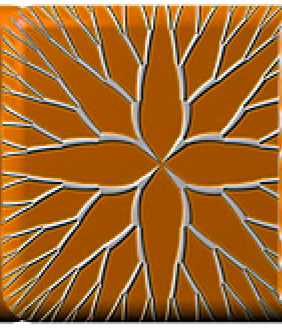


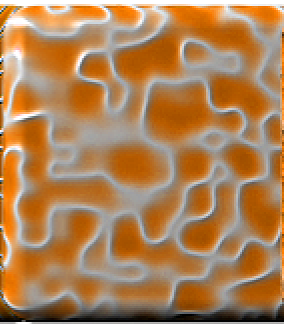
Wrapped fBm

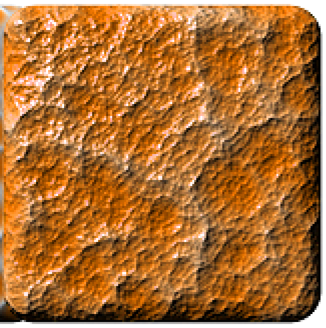
Organic











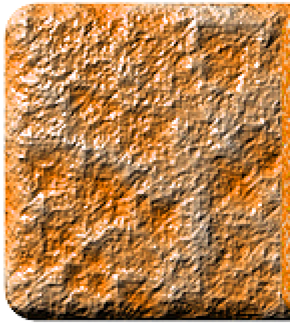
Art Deco

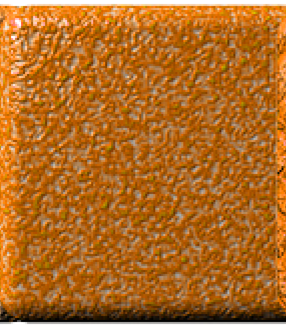
Blister

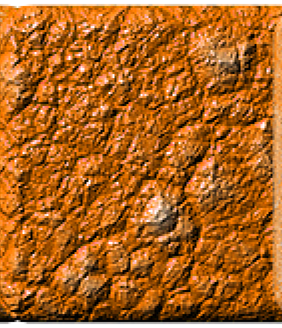
Branches

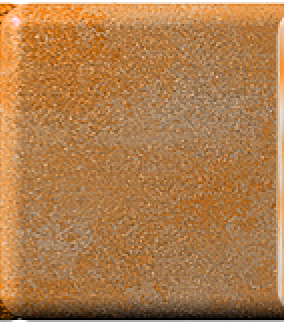
Caustic

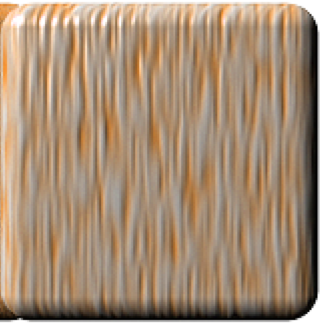
Cellular











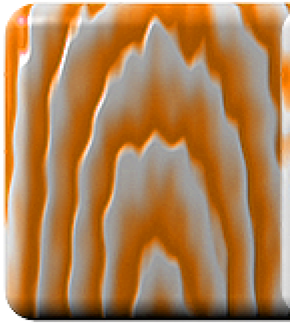
Cheesy

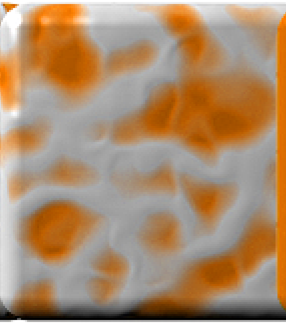
Concrete

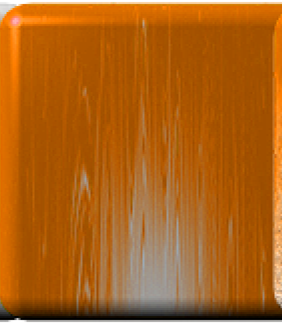
Crackle

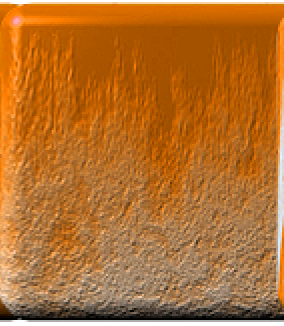
Dirt

Disturbed









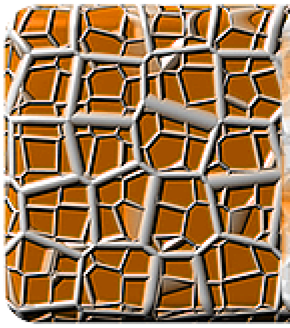
Easy Wood

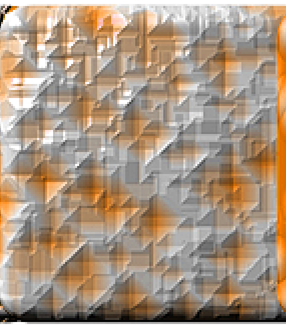
Electric

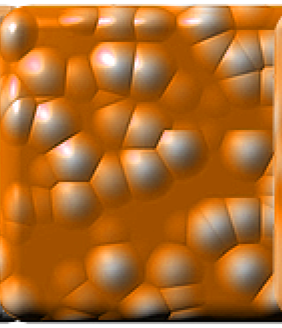
Fire

Fire Wall

Hard Wood







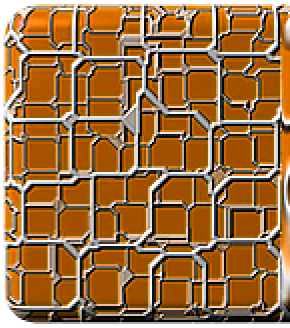
Membrane

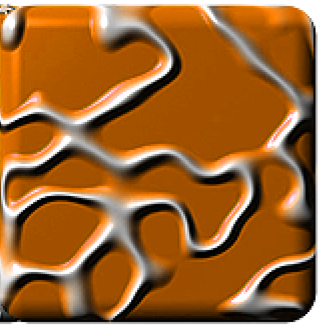
Minky

Scatter

Sin Blob

Veins

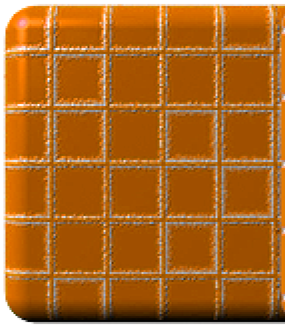


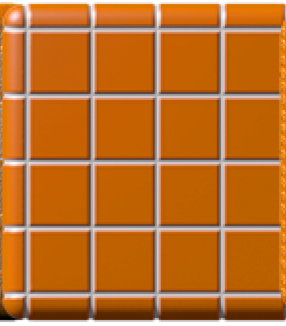


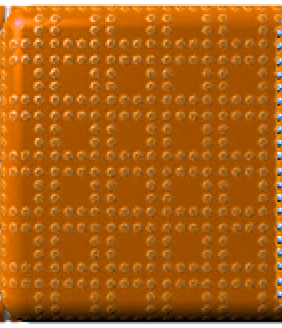
Wires

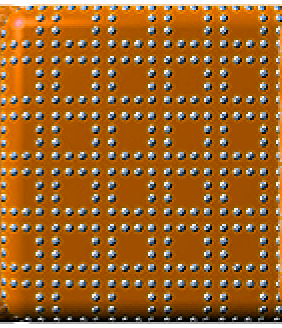
Worm Vein

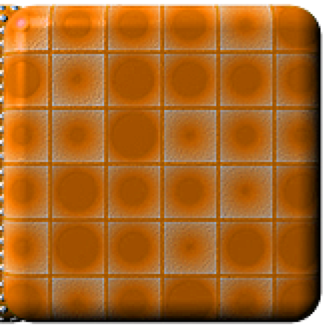
Panels

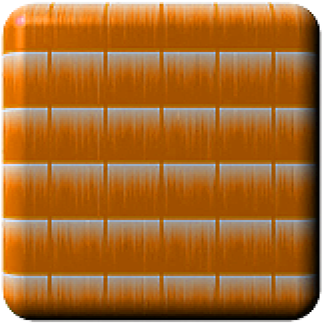

Peel


Plates

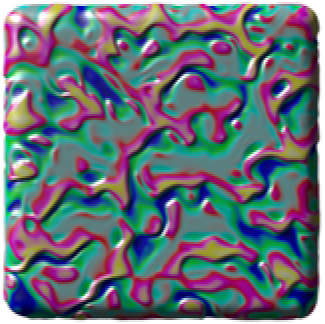

Rivet Rust

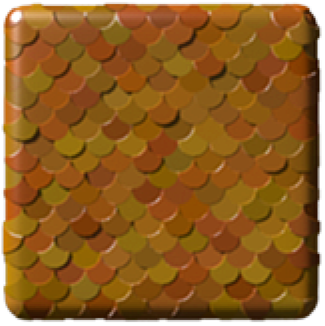

Rivets


Rust

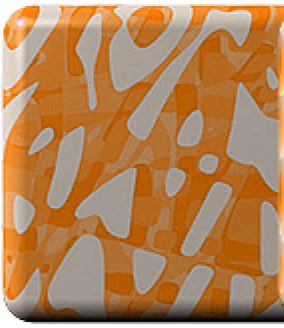

Smear

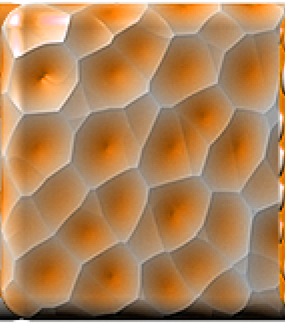
Process

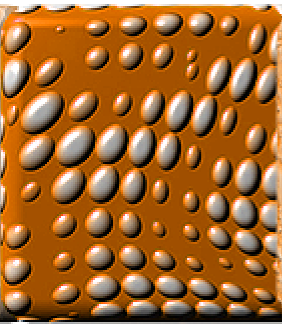

Easy Gradient

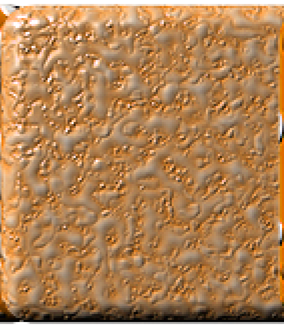

Regional HSV

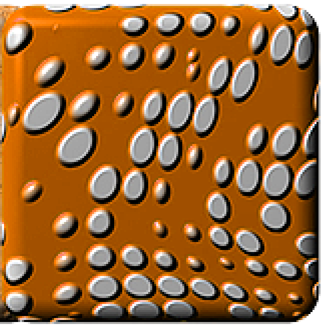
Skins

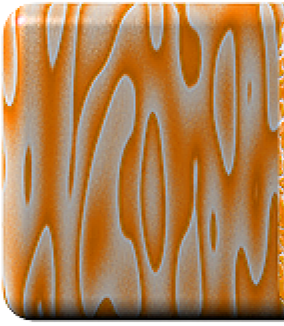

Camo

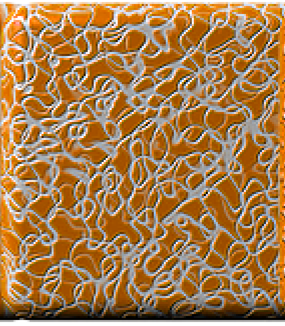

Crumpled

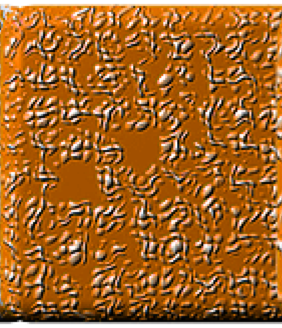

Dino Skin

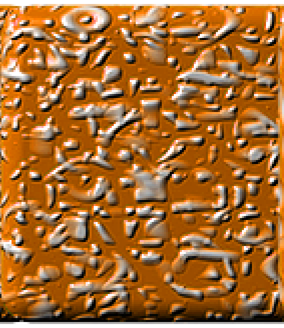

Disease

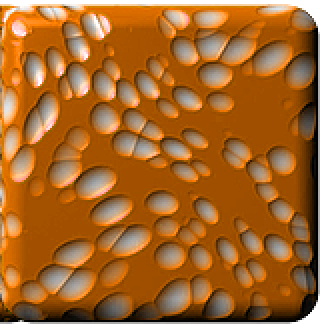

Frog Skin

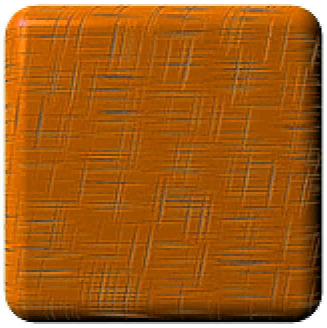

Grainy Wood


Leather

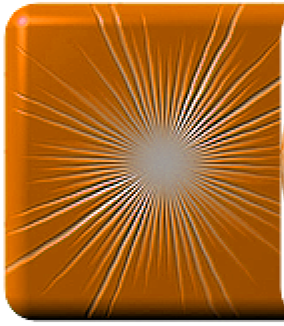

Monster

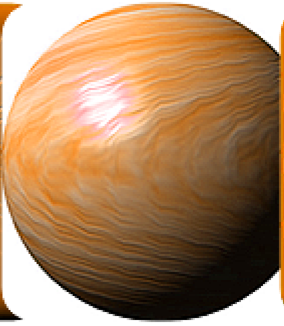

Pastella

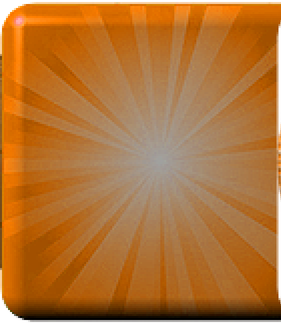

Peened

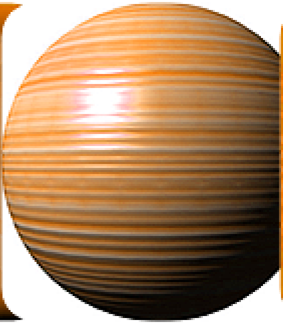

Scratches

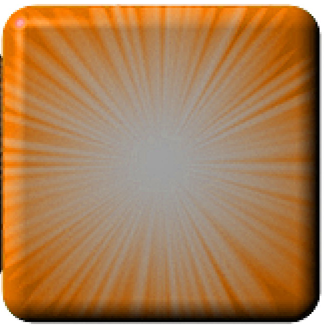
Space

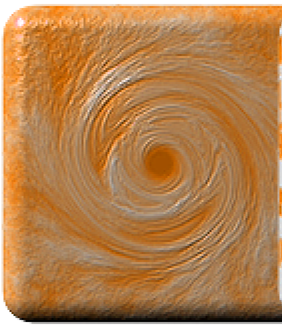

Blast

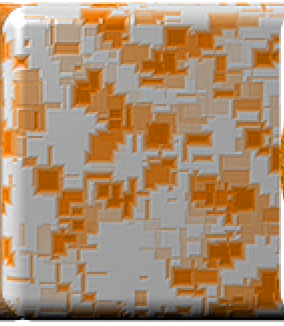

Coriolis

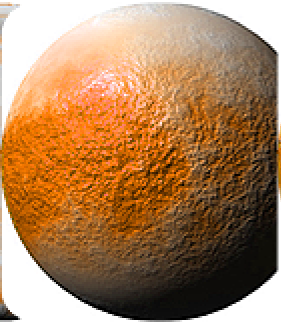

Flare

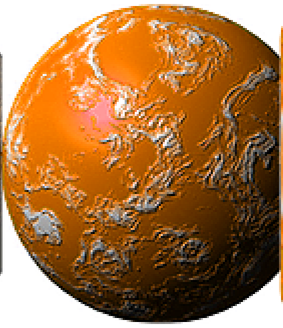

Gas Giant

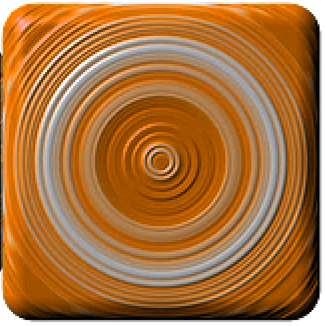

Glint


Hurricane

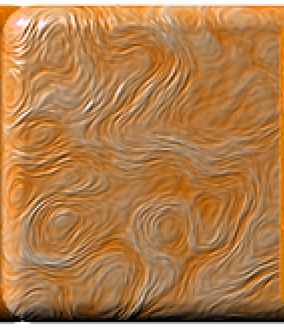

Nurnies

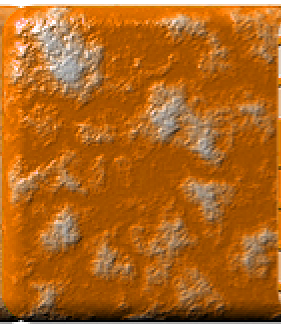

Planet

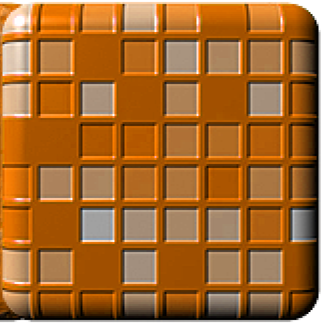

Planet Clouds


Rings

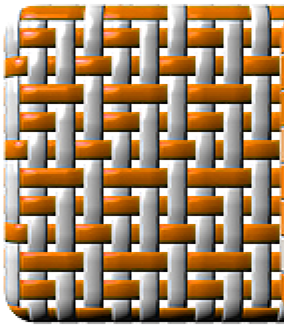
Star Field

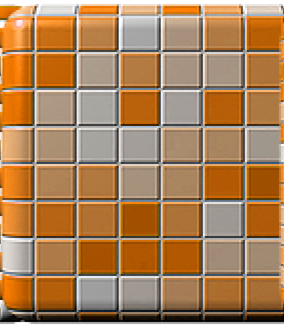

Swirl

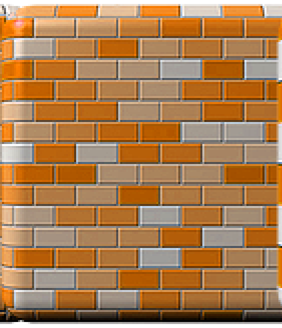

Terra

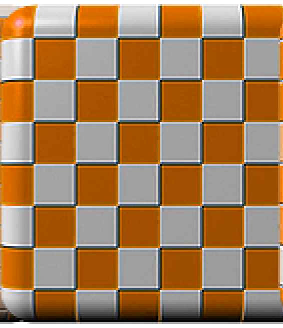

Windows

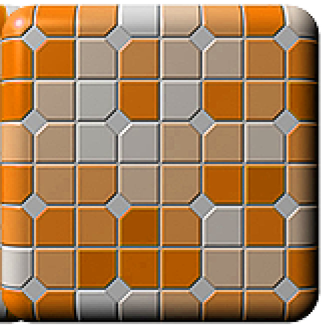
Tiles

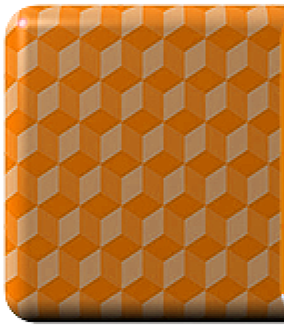

Basket

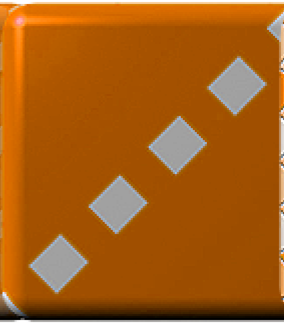

Bath Tile

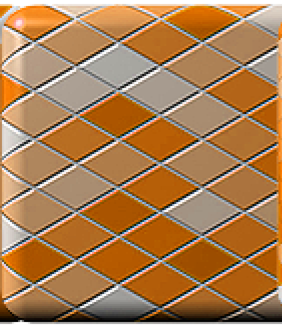

Bricks

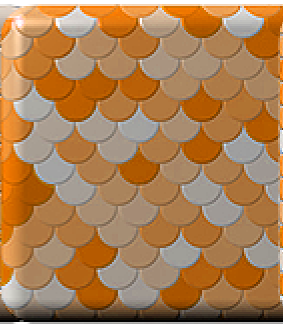

Checks

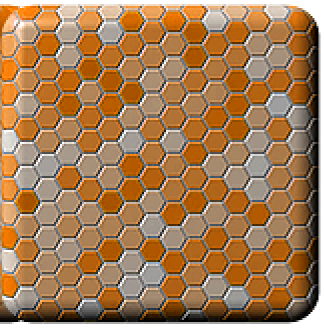

Cornerless

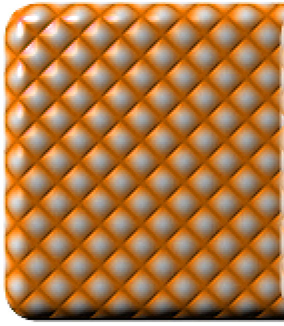

Cubes

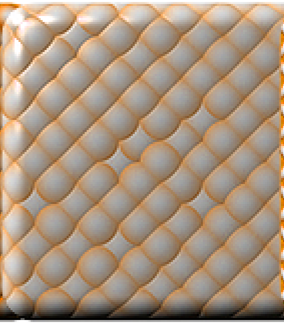

Dash Line

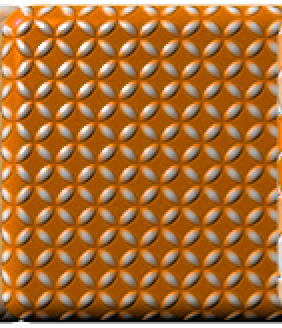

Diamond Deck

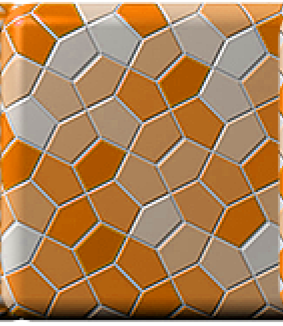

Fish Scales

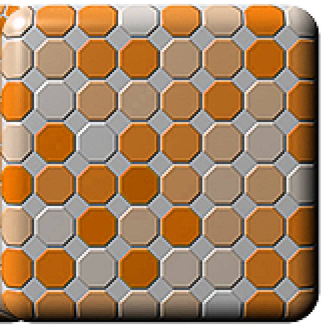

Hexagonal Tiles

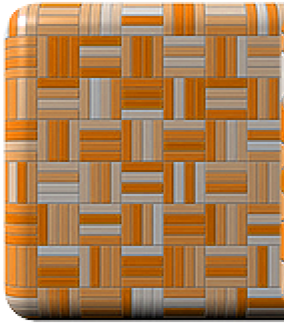

Lattice 1

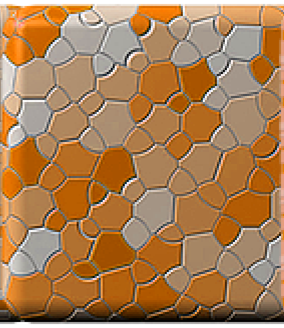

Lattice 2

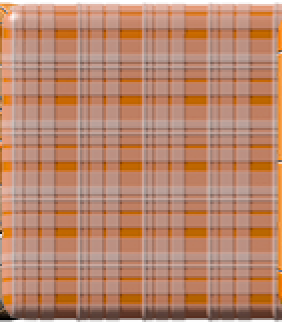

Lattice 3

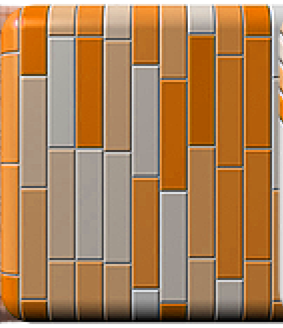

Mosaic

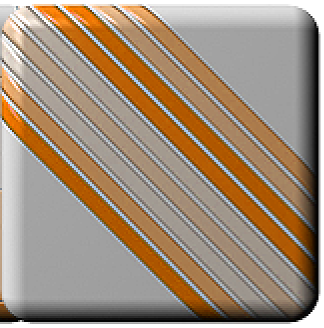

Octagonal Tile

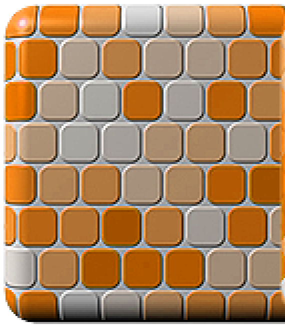

Parquet


Paving

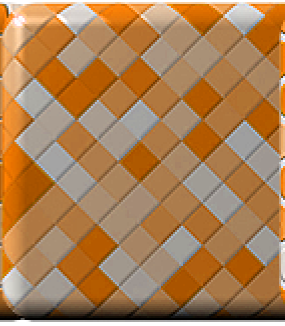

Plaid


Planks

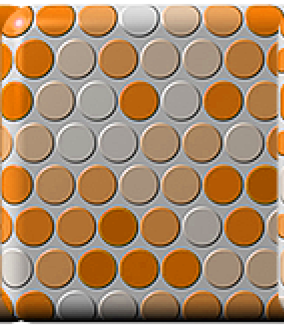

Ribs



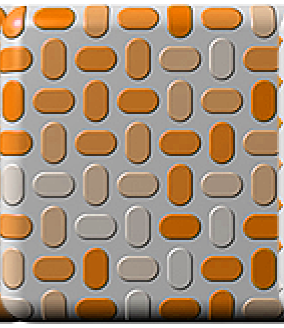
Rounded Tile



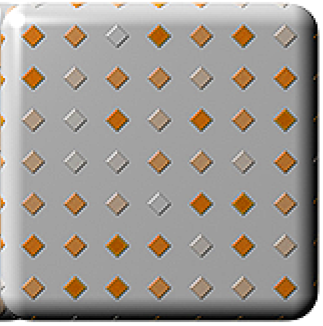
Shingles



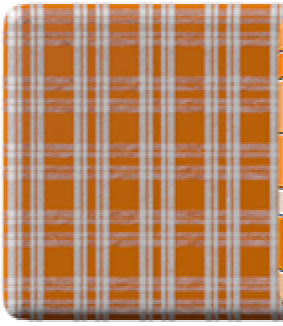
Spots



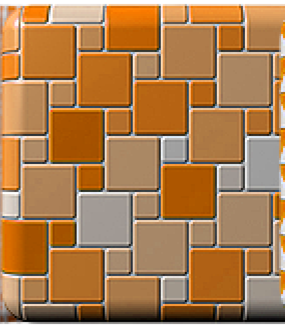
Stamped



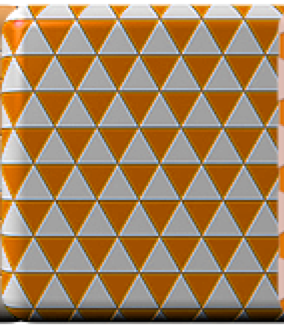
Tacos



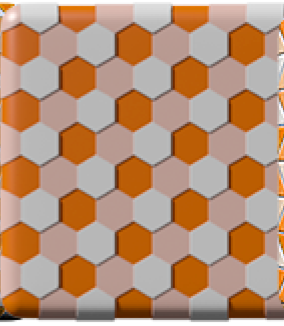
Tartan



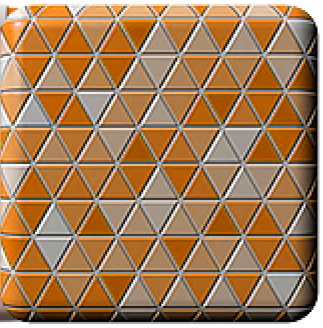
Tiler



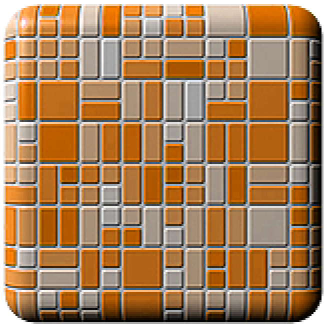
Tri Checks



Tri Hexes

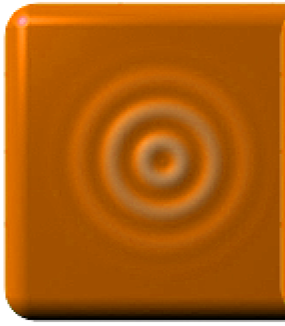


Tri Tile

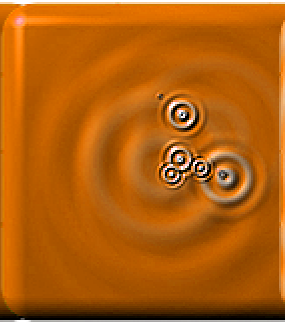


Wall

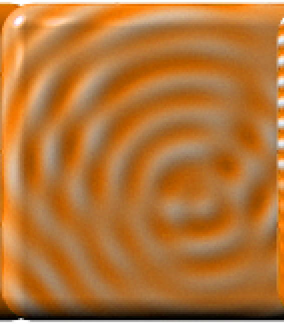
Water



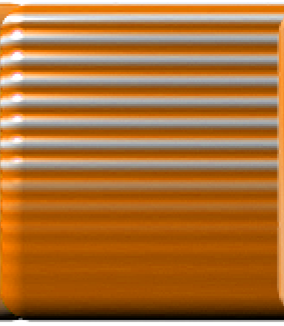
Drip Drop



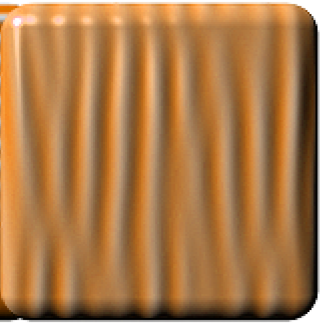
Rain



Ripples



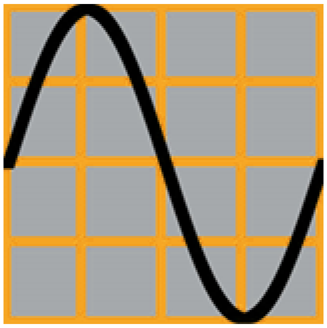
Surf



Waves

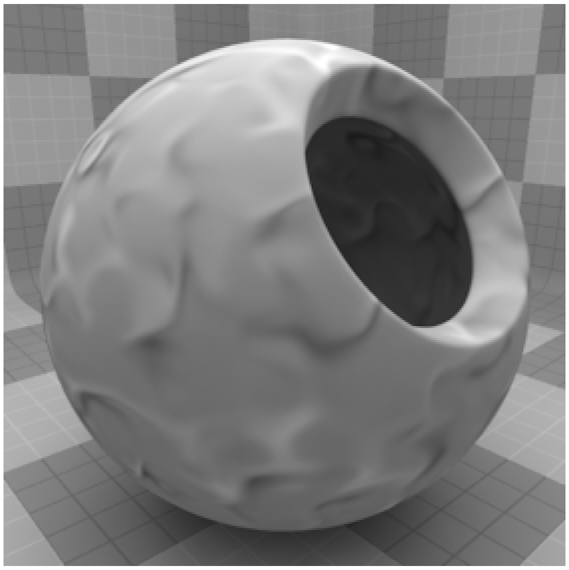
Windy Waves

Waveforms



Waveforms

Agate

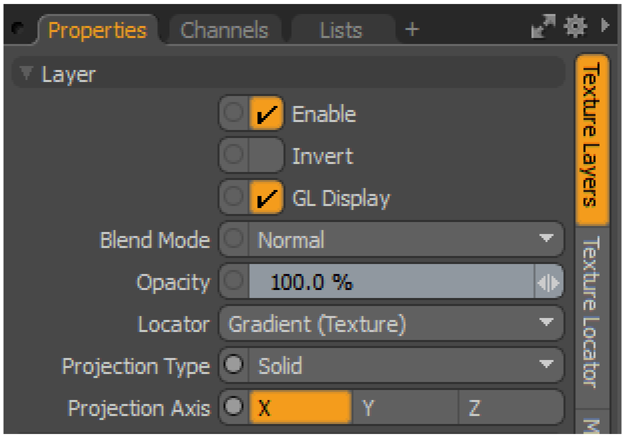


The **Agate** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Agate** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader emulates the color patterns in certain types of marble, agate, or jade.

Note: For information on adding and working with Shader Tree Items Layers, please reference the Shader Tree topic.

Layer Properties

>

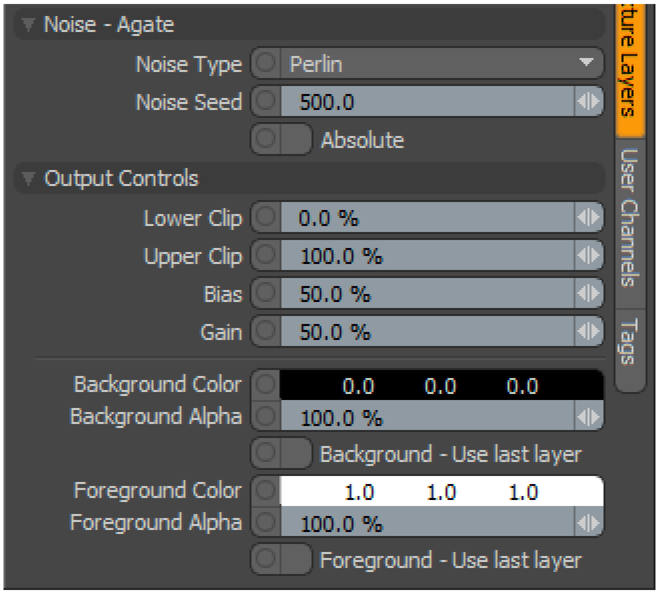


Layer
☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Agate Properties

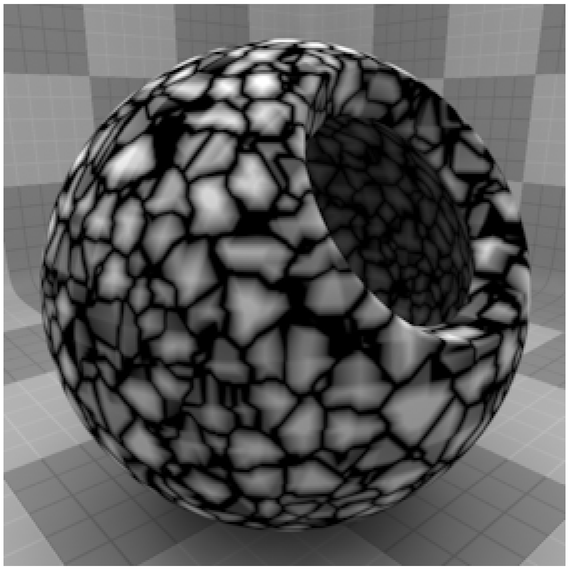

Noise - Agate
Noise Type: Perlin
Noise Seed: 500.0
Absolute
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Agate	
Noise Type	Specifies the look of the texture distortion, with several Noise function types provided: • **Perlin** • **Enhanced Perlin**

	• **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Absolute	Specifies whether or not the **Absolute** value for the noise is used, providing a stronger, more defined variation on the default texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	The **Gain** setting is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**

Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Art Deco

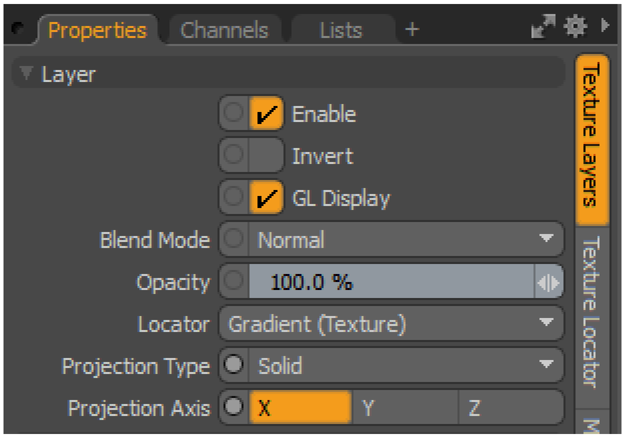


The **Art Deco** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Art Deco** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings, each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader is similar to the **Crackle** shader and provides a random crack-like pattern, muck like a dried-up river bed. The **Art Deco** shader differs from the **Crackle** shader in that distance between the cells has been changed to give them a looser, more organic feel.

Note: For information regarding adding and working with Shader Tree Items Layers, please see the Shader Tree topic.

Layer Properties

>

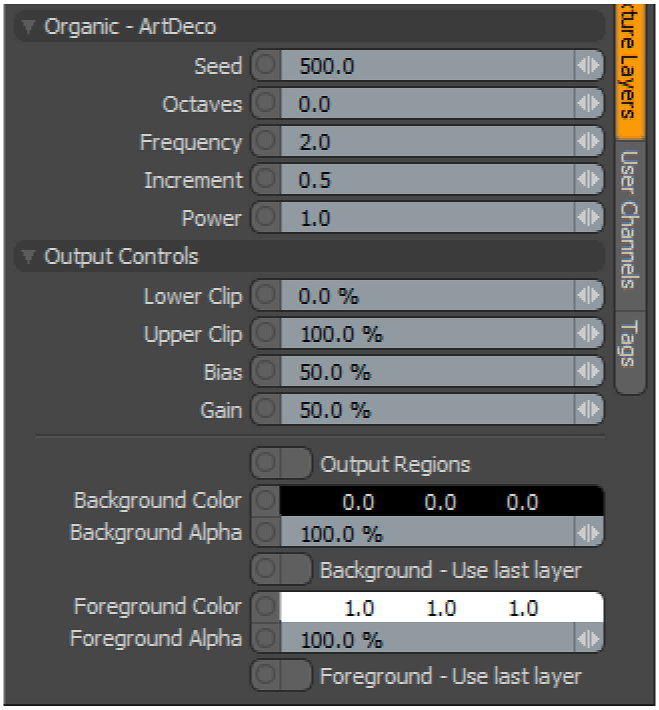


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify

	alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Art Deco Properties



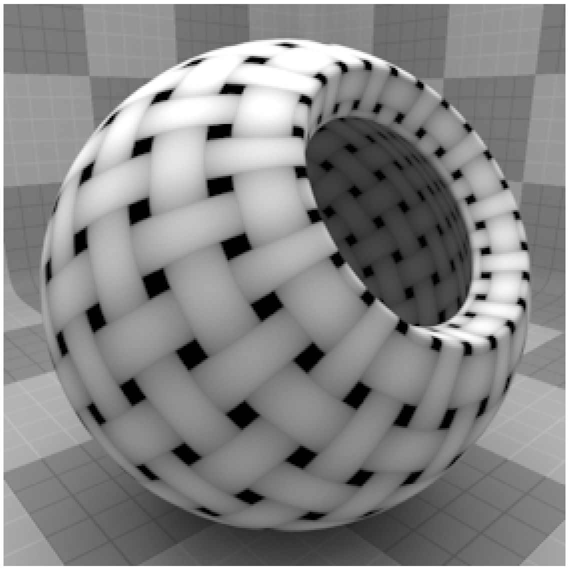
Organic - ArtDeco
Seed 500.0
Octaves 0.0
Frequency 2.0
Increment 0.5
Power 1.0
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Organic - Art Deco	
Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	Defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the

	turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Increment	Controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smoother as the parameter is increased.
Power	Controls the degree to which the edges of the texture are mixed up.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	The **Gain** setting is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Basket



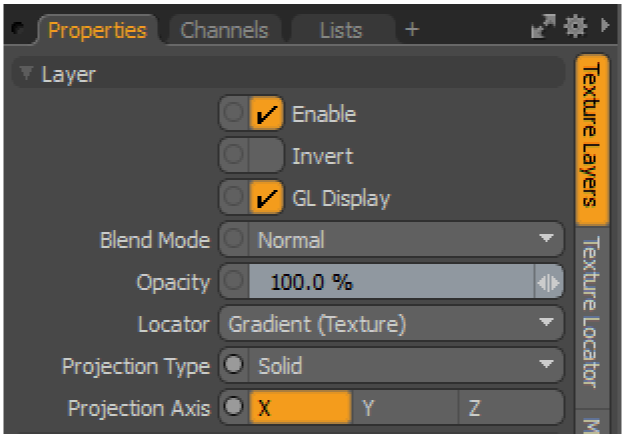
The **Basket** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Basket** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings, each zone can have either a **Value**, or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a woven pattern, the kind used to create a **Basket**, with 26 reversible weave types.

Note: For information regarding adding and working with Shader Tree Items Layers, please see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type.
To select the **Texture Locator**, click on the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a UV map (see Working with UV Maps) or choose one of the geometric projection, such as **Planar**, **Cylindrical**, and **Spherical**, and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

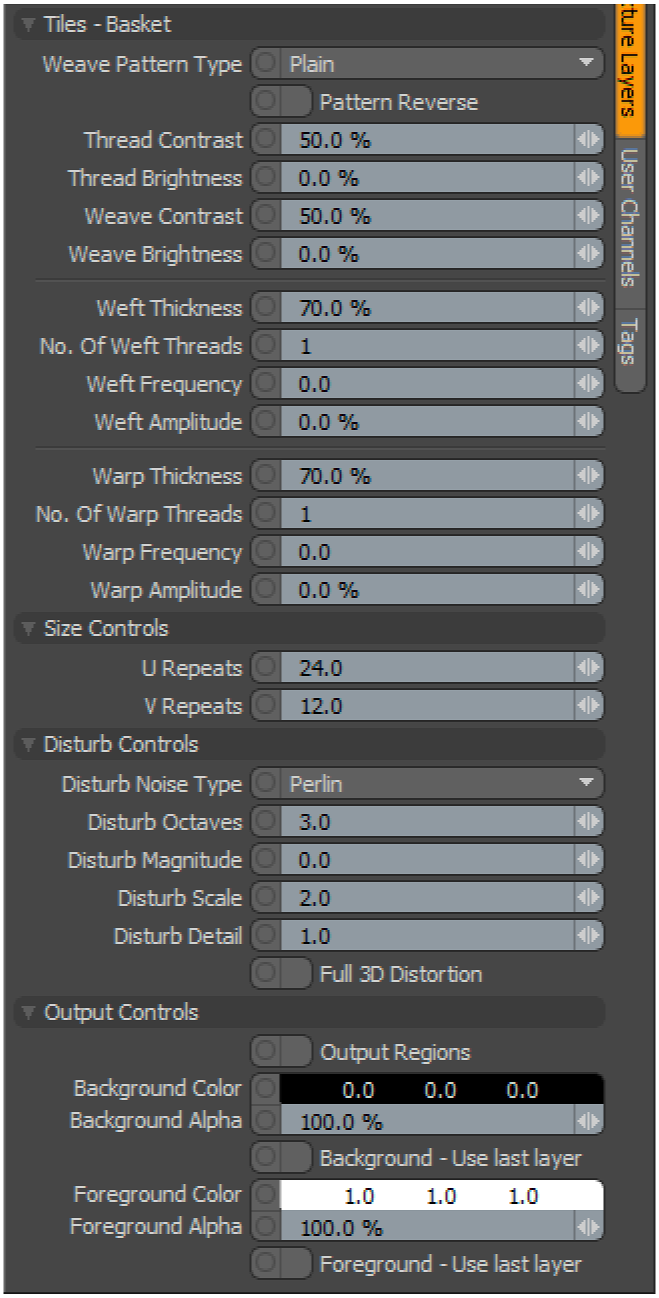
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in

	the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Basket Properties



Tiles - Basket

Weave Pattern Type: Plain

☐ Pattern Reverse

Thread Contrast: 50.0 %

Thread Brightness: 0.0 %

Weave Contrast: 50.0 %

Weave Brightness: 0.0 %

Weft Thickness: 70.0 %

No. Of Weft Threads: 1

Weft Frequency: 0.0

Weft Amplitude: 0.0 %

Warp Thickness: 70.0 %

No. Of Warp Threads: 1

Warp Frequency: 0.0

Warp Amplitude: 0.0 %

Size Controls

U Repeats: 24.0

V Repeats: 12.0

Disturb Controls

Disturb Noise Type: Perlin

Disturb Octaves: 3.0

Disturb Magnitude: 0.0

Disturb Scale: 2.0

Disturb Detail: 1.0

☐ Full 3D Distortion

Output Controls

☐ Output Regions

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

☐ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

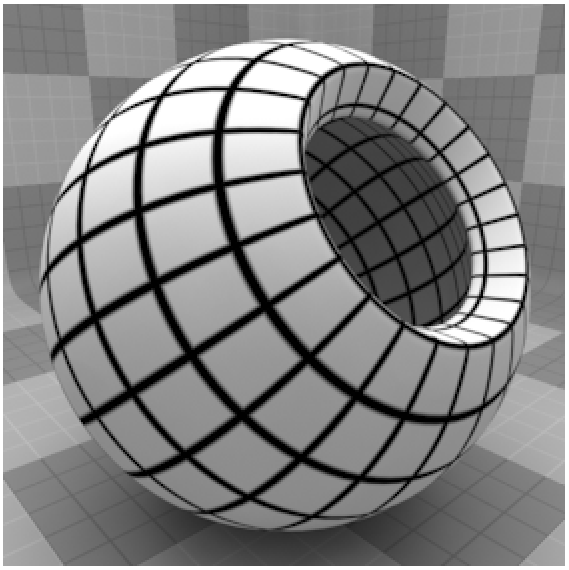
☐ Foreground - Use last layer

Tiles - Basket	
Weave Pattern Type	Specifies the weave pattern used in the texture. There are 26 options including: • Plain • Herring Bone

	• Crosses • Zig Zag
Pattern Reverse	When selected, reverses the weave and weft of the pattern.
Thread Contrast	Specifies the contrast on the threads of the weave pattern.
Thread Brightness	Specifies the brightness on the threads of the weave pattern.
Weave Contrast	Specifies the contrast on the weave pattern.
Weave Brightness	Specifies the brightness on the weave pattern.
Weft Thickness	Specifies the width of the weft, or horizontal, threads.
Number Of Weft Threads	Controls the number of weft threads.
Weft Frequency	Controls the sine frequency used to disturb the weft threads.
Weft Amplitude	Controls the amplitude used to disturb the weft threads.
Warp Thickness	Specifies the width of the warp, or vertical, threads
Number Of Warp Threads	Controls the number of warp threads.
Warp Frequency	Controls the sine frequency used to disturb the warp threads.
Warp Amplitude	Controls the amplitude used to disturb the warp threads.
Size Controls	
U Repeats	Specifies the number of the times the pattern repeats in the U direction (horizontally).
V Repeats	Specifies the number of the times the pattern repeats in the V direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several Noise function types provided: • **Perlin** • **Enhanced Perlin**

	• **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on Full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Bath Tile



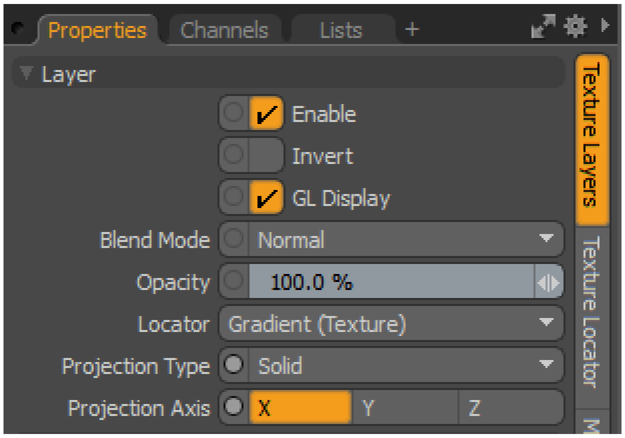
The **Bath Tile** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Bath Tile** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings, each zone can have either a **Value**, or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a square tile pattern with variable grout, tiles edges and tile height, similar to to the look of **Bath Tile**. The tiles can be staggered and random tiles can be removed.

Note: For information regarding adding and working with Shader Tree Items Layers, see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type.
To select the **Texture Locator**, click on the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a UV map (see Working with UV Maps) or choose one of the geometric projection, such as **Planar**, **Cylindrical**, and **Spherical**, and size the texture accordingly.

Layer Properties

>



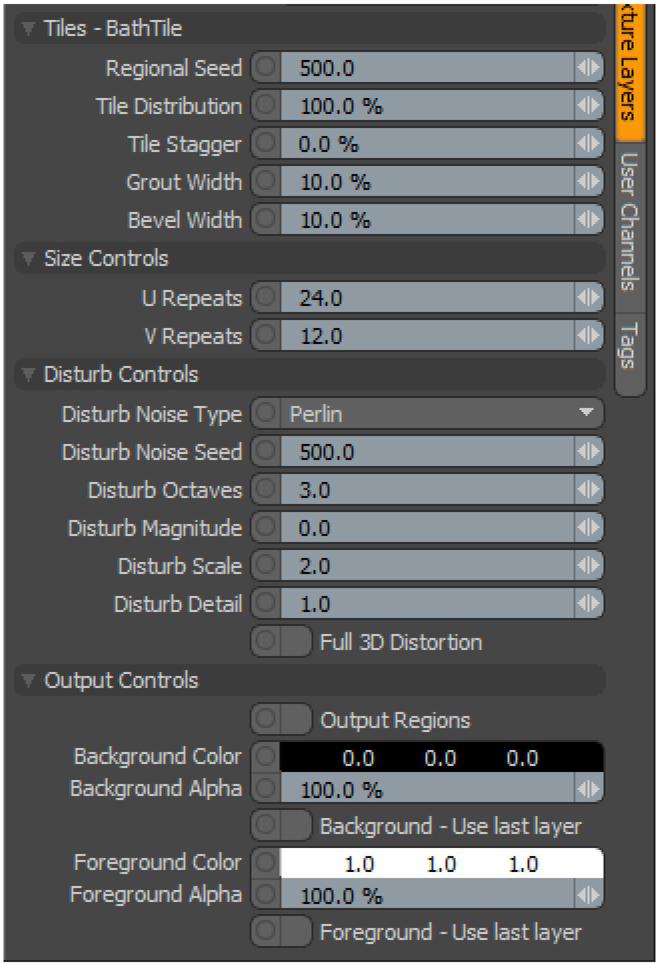
Layer

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture**

	Locator that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Bath Tile Properties



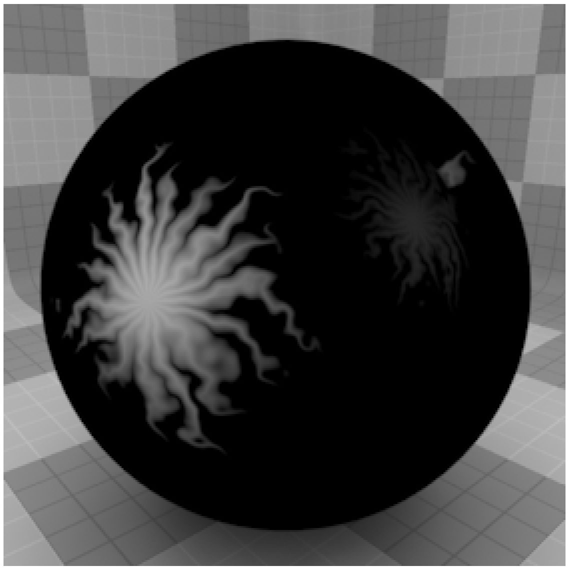
Tiles - BathTile
Regional Seed 500.0
Tile Distribution 100.0 %
Tile Stagger 0.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Bath Tile

Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Tile Distribution	Specifies the percentage of the surface covered by tiles.
Tile Stagger	Specifies the amount of staggering between tile rows. When the control is at a low value, the tiles line up vertically, with the tile rows becoming more staggered as the parameter is increased.
Grout Width	Specifies the amount of grout between the tiles.
Bevel Width	Specifies the width of the bevel on the tiles.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several Noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on Full 3D distortion, which yields better results, but takes longer to render.

Output Controls	
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Blast

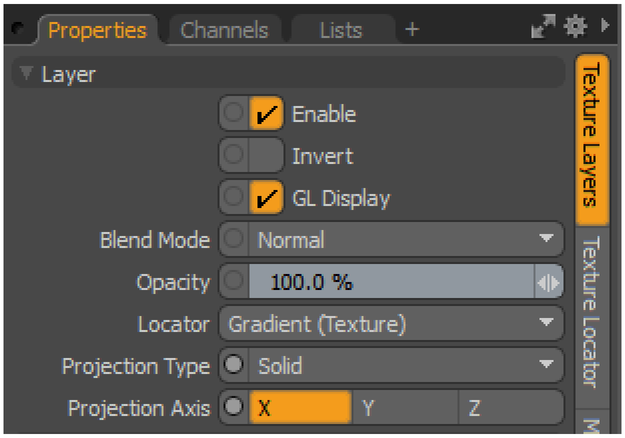


The **Blast** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Agate** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a look similar to the **Blast** marks made by a flying projectile or a futuristic phaser.

Note: For information on adding and working with Shader Tree Items Layers, please reference the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

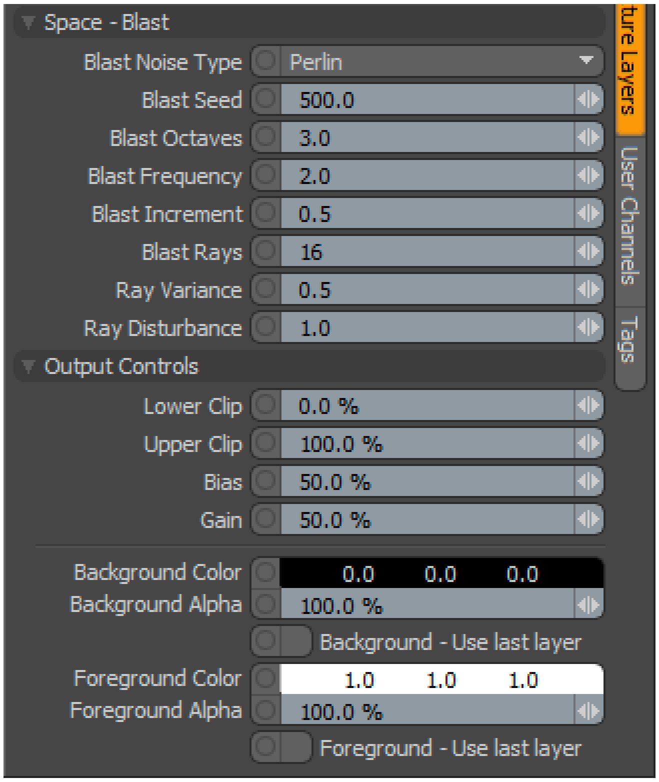
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Blast Properties



Space - Blast
Blast Noise Type: Perlin
Blast Seed: 500.0
Blast Octaves: 3.0
Blast Frequency: 2.0
Blast Increment: 0.5
Blast Rays: 16
Ray Variance: 0.5
Ray Disturbance: 1.0

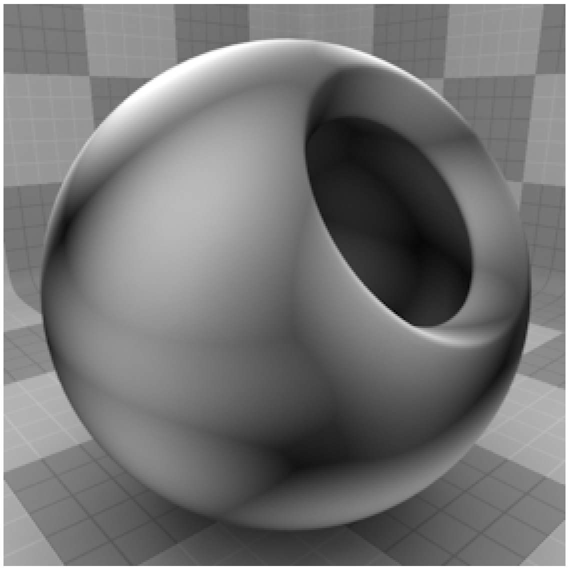
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Space - Blast

Blast Noise Type	Specifies the look of the texture distortion, with several Noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Blast Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Blast Octaves	Defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Blast Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Blast Increment	Controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smoother as the parameter is increased.
Blast Rays	Controls the number of blast waves that emanate from the blast center.
Ray Variance	Controls the variance of the blast waves as they emanate from the center. When the control is at a low value, it results in blast waves of similar distance, with the variance between the various blast waves growing as the value is increased.
Ray Disturbance	Controls the disturbance of the blast waves as they emanate from the center. When the control is at a low value, the blast rays is straighter, getting more disturbed as the value is increased.
Output Controls	

Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	The **Gain** setting is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Blister

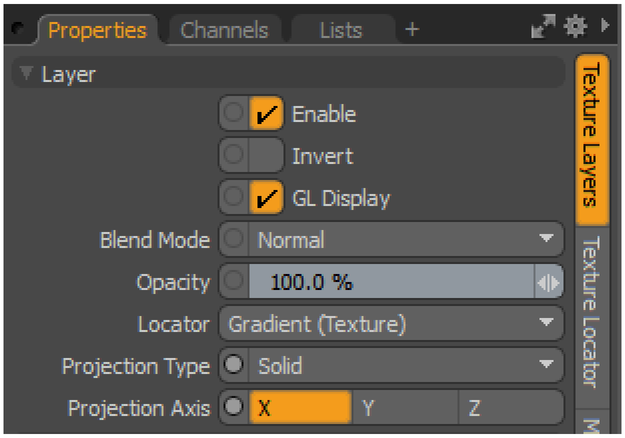


The **Blister** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Agate** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied Zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**.The look of this particular is a pattern similar to leopard or dalmatian spots. These spots can also be applied as bumps to create a smooth blistered or pitted effects.

Note: For information on adding and working with Shader Tree Items Layers, please reference the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

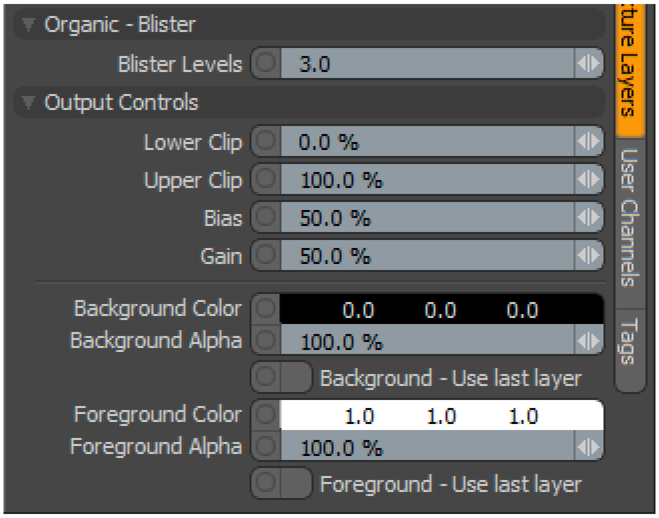
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Blister Properties

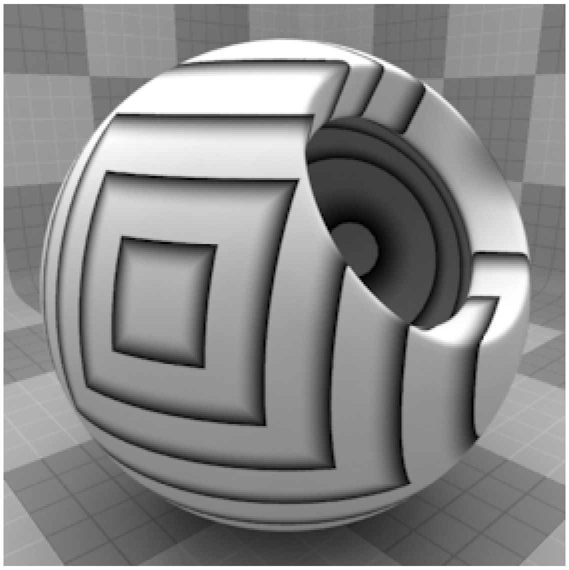


Blister Levels: 3.0
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
○ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
○ Foreground - Use last layer

Organic - Blister	
Blister	Defines the number of blister levels within the texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	The **Gain** setting is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control

	the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Box

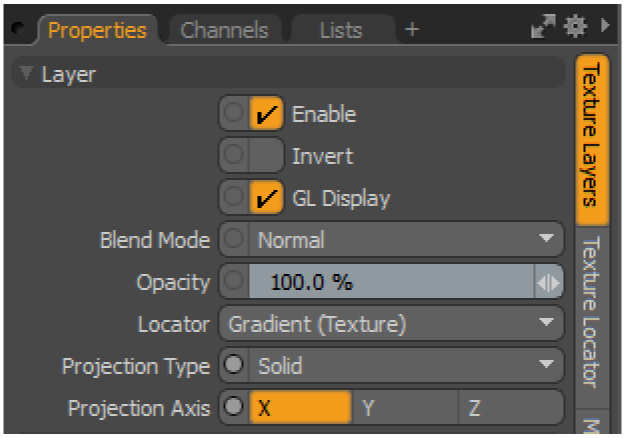


The **Box** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Basket** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings, each zone can have either a **Value**, or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a series of adjustable box-shaped patterns.

Note: For information regarding adding and working with Shader Tree Items Layers, please see the Shader Tree topic.

Layer Properties

>

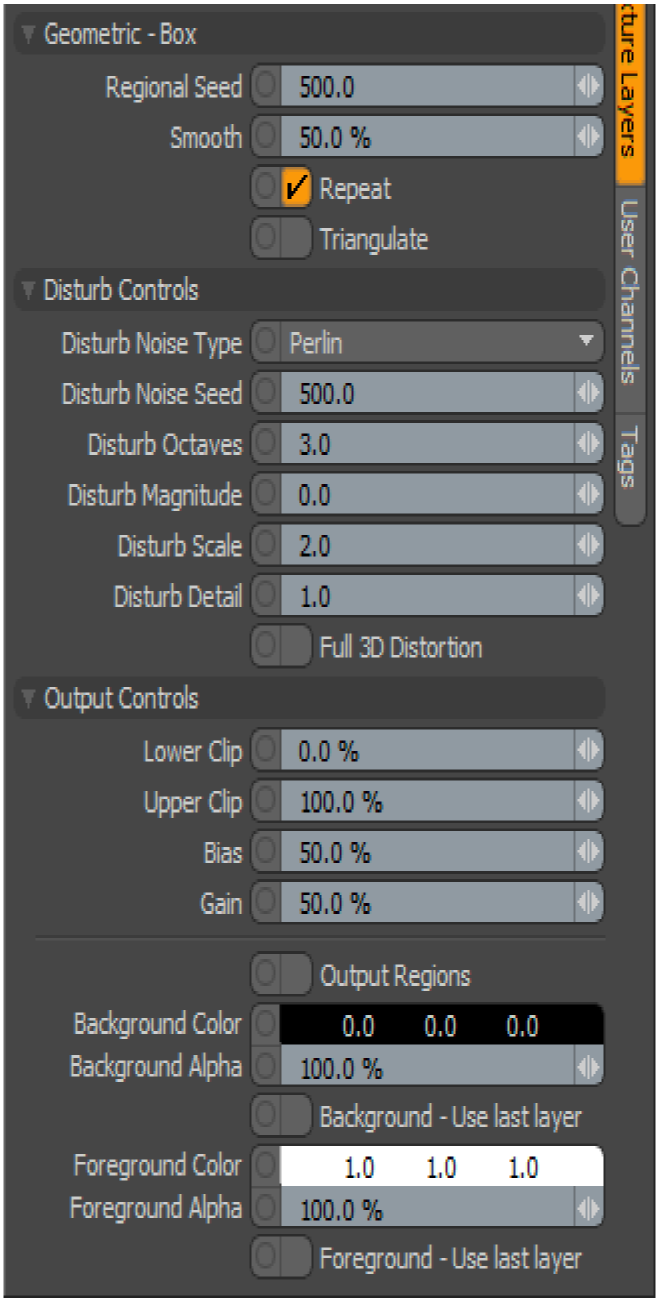


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Box Properties



Geometric - Box
Regional Seed: 500.0
Smooth: 50.0 %
☑ Repeat
☐ Triangulate

Disturb Controls
Disturb Noise Type: Perlin
Disturb Noise Seed: 500.0
Disturb Octaves: 3.0
Disturb Magnitude: 0.0
Disturb Scale: 2.0
Disturb Detail: 1.0
☐ Full 3D Distortion

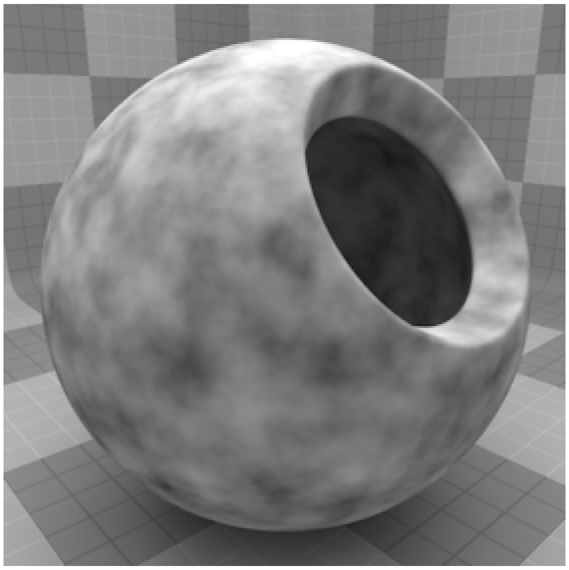
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
☐ Output Regions
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Geometric - Box	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is

	necessary for items to retain the same variations.
Smooth	Specifies the amount of smoothing around the edges of the box. Higher values result in smoother edges.
Repeat	When selected, causes the texture to repeat indefinitely.
Triangulate	When selected, causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.
Disturb Controls -Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several Noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Disturb Octaves	Defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the

	texture.
Full 3D Distortion	Turns on Full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	The **Gain** setting is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Bozo

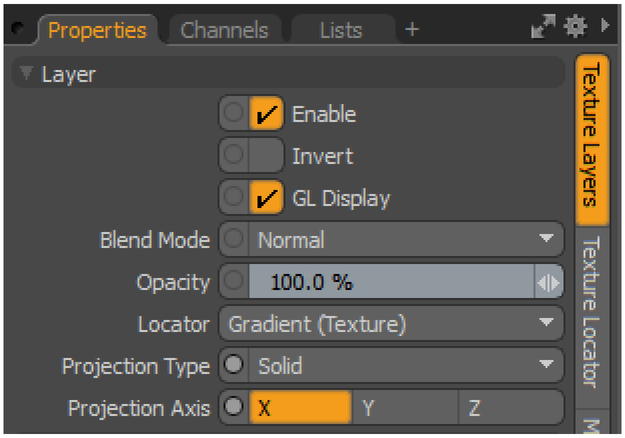


The **Bozo** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Basket** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings, each zone can have either a **Value**, or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture Effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader is ideal for textures like clouds, smoke or fog. A plastic stippled effect can also be created using very small bozo bumps. Adjusting the shader to draw larger bumps can create wave or ripple effects.

Note: For information regarding adding and working with Shader Tree Items Layers, please see the Shader Tree topic.

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

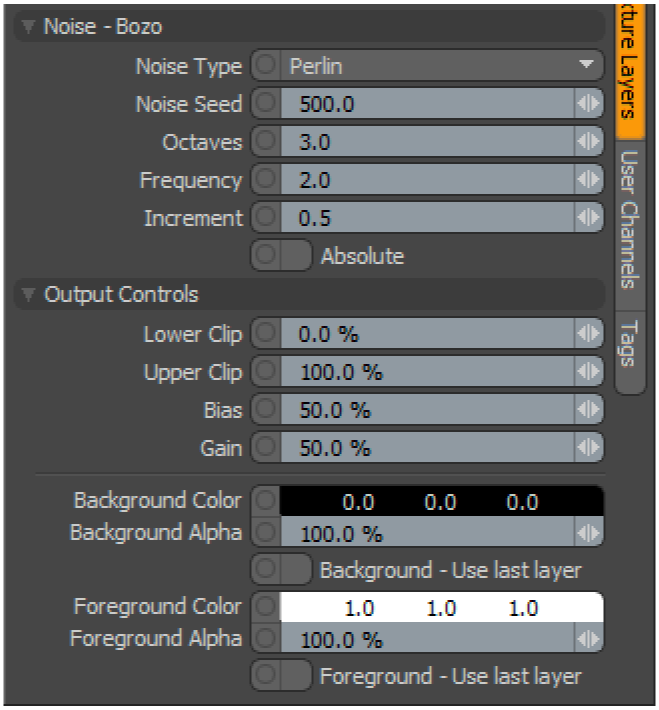
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: **X** Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Bozo Properties



Noise - Bozo

Noise Type Perlin

Noise Seed 500.0

Octaves 3.0

Frequency 2.0

Increment 0.5

☐ Absolute

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

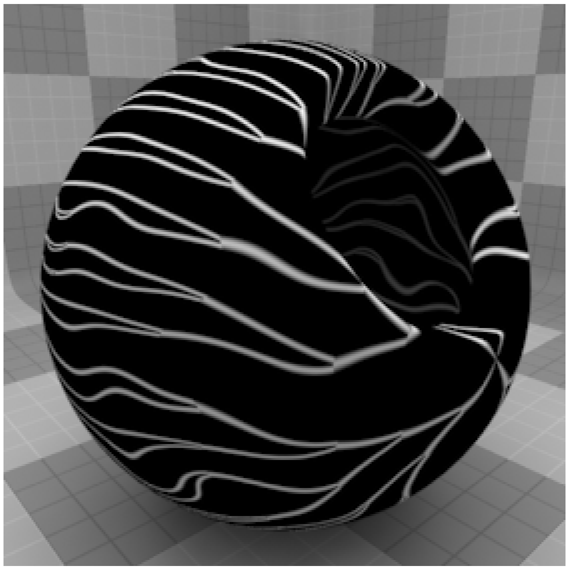
☐ Foreground - Use last layer

Noise - Bozo	
Noise Type	Specifies the look of the distortion by providing several Noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	Defines the number of iterations used to create the texture. As this number is

	increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Increment	Controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smoother as the parameter is increased.
Absolute	Specifies whether or not the **Absolute** value for the noise is used, providing a stronger, more defined variation on the default texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	The **Gain** setting is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Branches

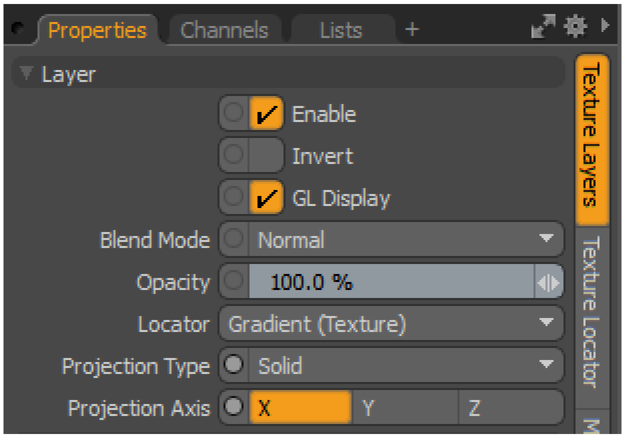


The **Branches** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Basket** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings, each zone can have either a **Value**, or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture Effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides hard-edged vein-like patterns that branch out from the center of the texture.

> **Note:** For information regarding adding and working with Shader Tree Items Layers, please see the Shader Tree topic.

Layer Properties

>

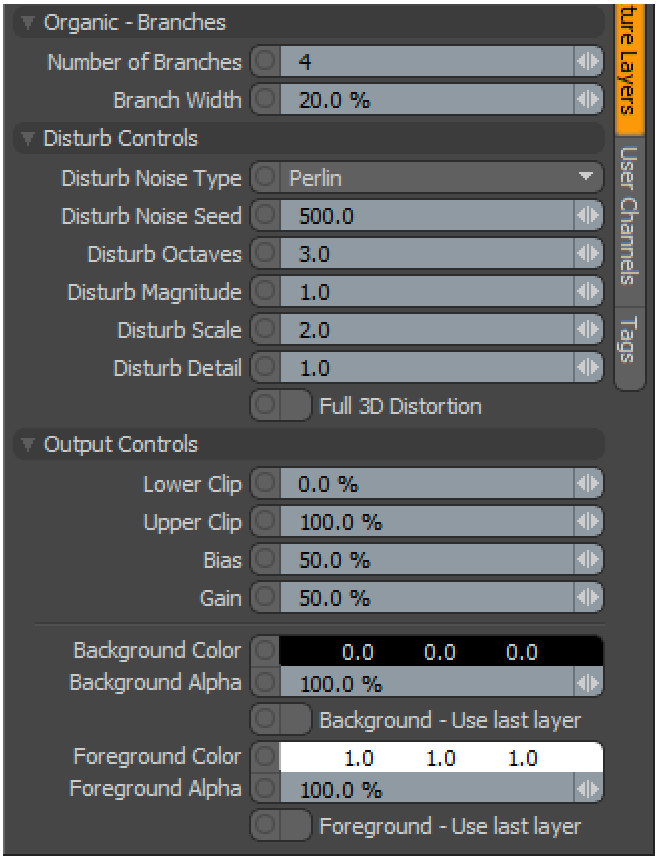


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Branches Properties



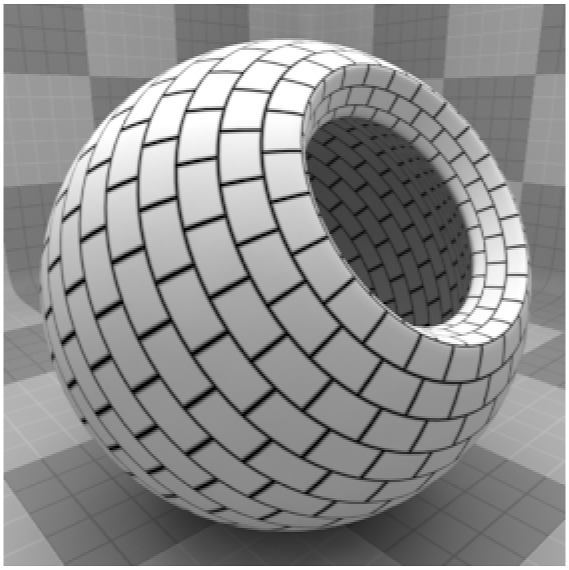
Organic - Branches
Number of Branches 4
Branch Width 20.0 %
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 1.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Organic - Branches	
Number Of Branches	Defines the number of branches that emanate from the center.
Branch Width	Defines the width of the individual branches.
Disturb Controls -Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise	Specifies the look of the texture distortion, with several Noise function types

Type	provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Disturb Octaves	Defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.

Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	The **Gain** setting is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles providing you a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's Background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's **Foreground** area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Bricks



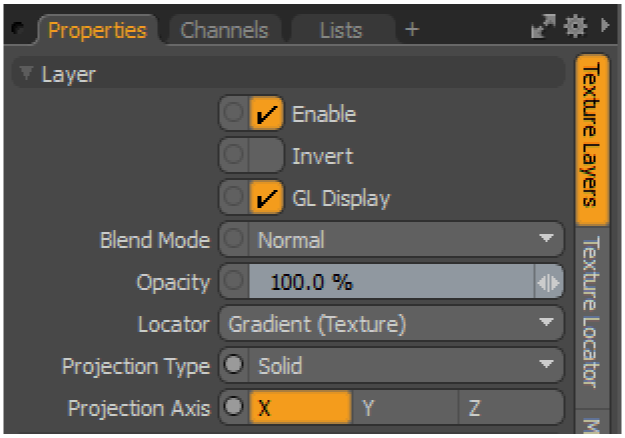
The **Bricks** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Bricks** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a rectangular, brick-like pattern. The shader has the same controls as the **Bath Tile** shader but the aspect ratio of the bricks can be modified.

Note: For information regarding adding and working with Shader Tree item layers the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the texture locator, press the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a **UV map** (highly recommended) or choose one of the geometric projections, such as **Planar**, **Cylindrical**, **Spherical**, and so on, and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

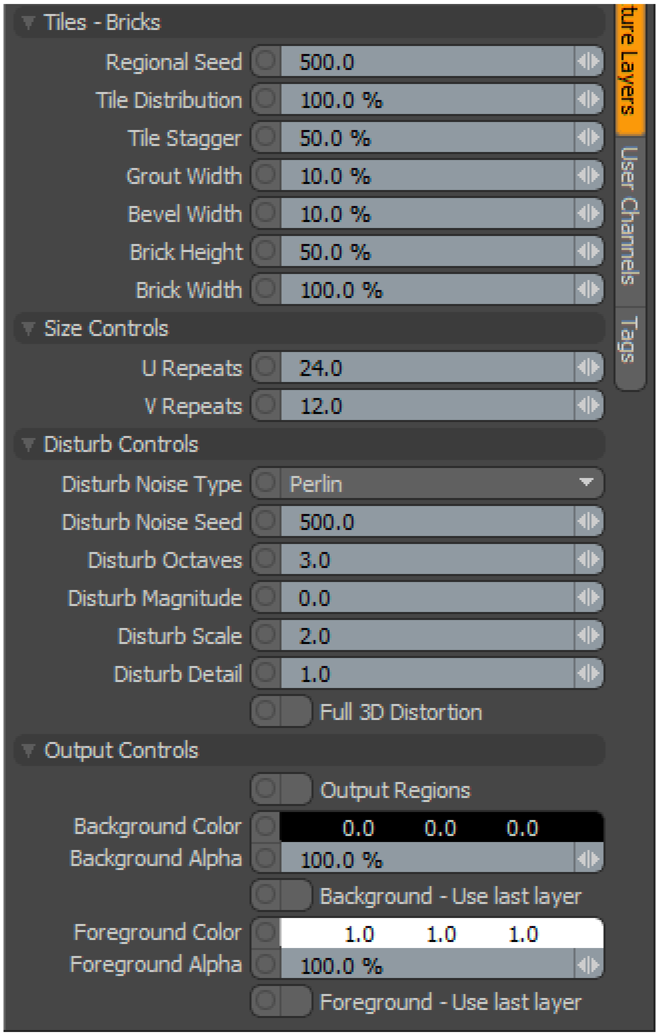
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Bricks Properties



Tiles - Bricks

Regional Seed 500.0
Tile Distribution 100.0 %
Tile Stagger 50.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Brick Height 50.0 %
Brick Width 100.0 %

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
○ Full 3D Distortion

Output Controls

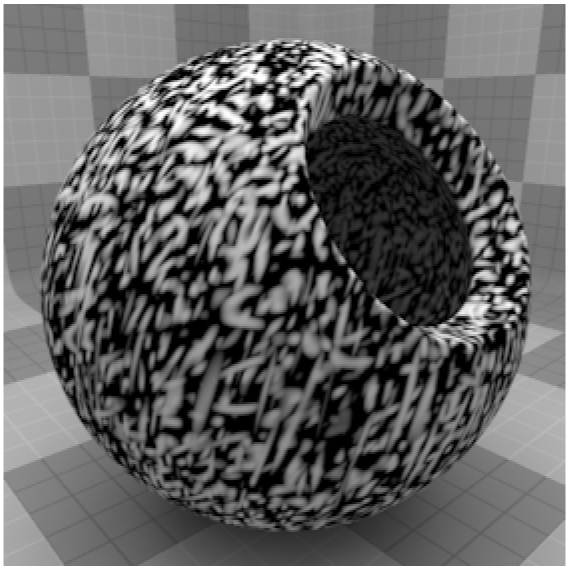
○ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
○ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
○ Foreground - Use last layer

Tiles - Bricks	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Tile Distribution	Specifies the percentage of the surface covered by bricks.
Tile Stagger	Specifies the amount of staggering between brick rows. When the control is at a low value, the bricks line up vertically, with the brick rows becoming more staggered as the parameter is increased.

Grout Width	Specifies the amount of grout between the bricks.
Bevel Width	Specifies the width of the bevel on the bricks.
Brick Height	Determines the height of the individual bricks.
Brick Width	Determines the width of the individual bricks.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the U direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the V direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	

Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Camo

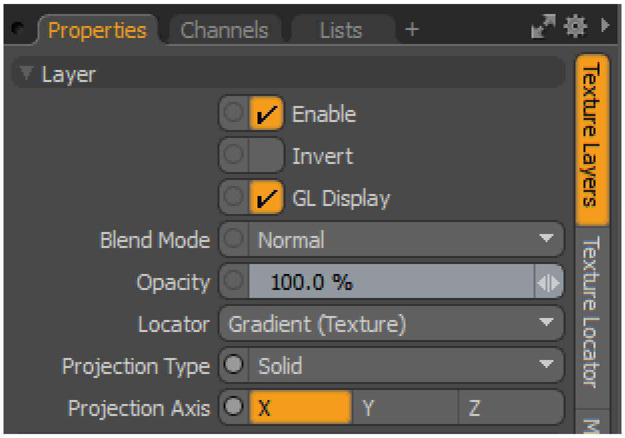


The **Camo** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Camo** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader is a random desert camouflage kind of pattern.

Note: For information regarding adding and working with Shader Tree item layers see the Shader Tree topic.

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

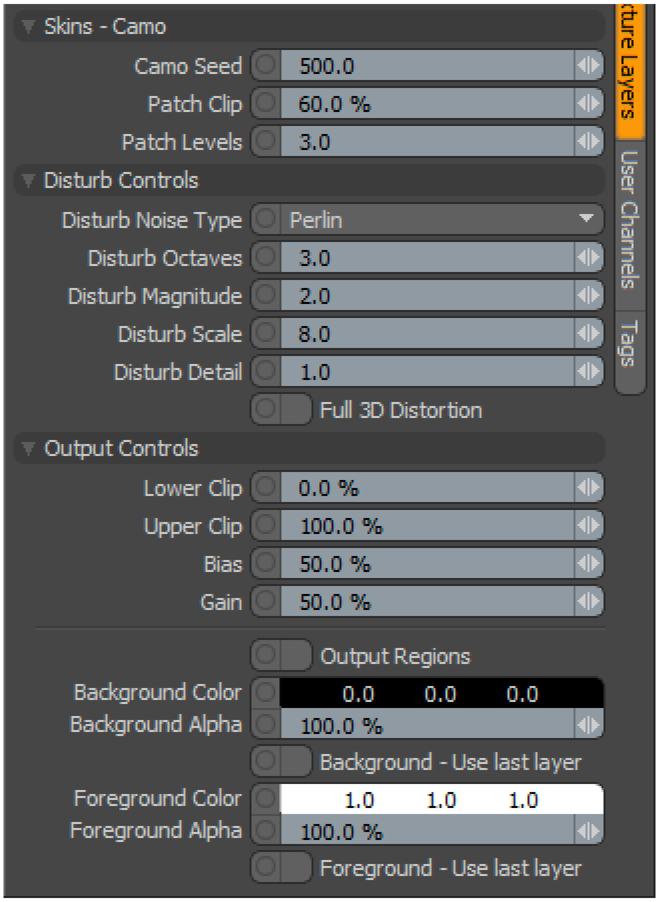
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Camo Properties



Skins - Camo

Camo Seed 500.0
Patch Clip 60.0 %
Patch Levels 3.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Octaves 3.0
Disturb Magnitude 2.0
Disturb Scale 8.0
Disturb Detail 1.0
Full 3D Distortion

Output Controls

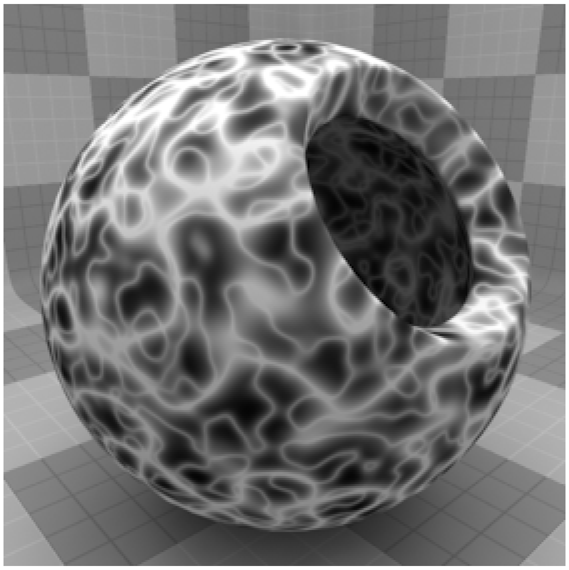
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Skins - Camo	
Camo Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Patch Clip	Determines where the texture shading transitions to the background color. The higher the value, the quicker the transition.
Patch Levels	Determines the number of layers of the camo texturing. Additional levels add additional detail, while also lightening the overall look of the texture.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	

Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with

	sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Caustic

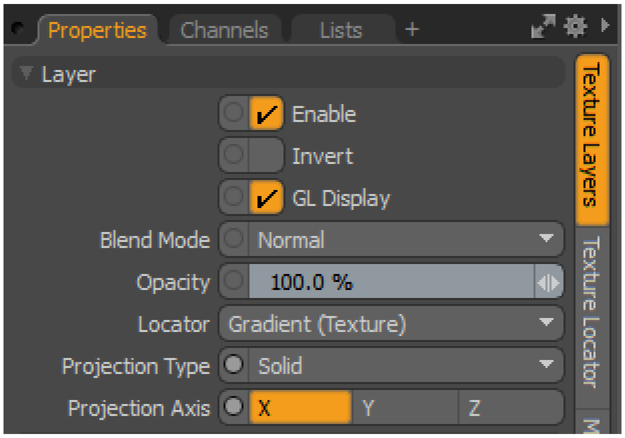


The **Caustic** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Caustic** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides a watery, caustic noise, similar to how light plays benetath the surface of a swimming pool.

Note: For information regarding adding and working with Shader Tree item layers see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

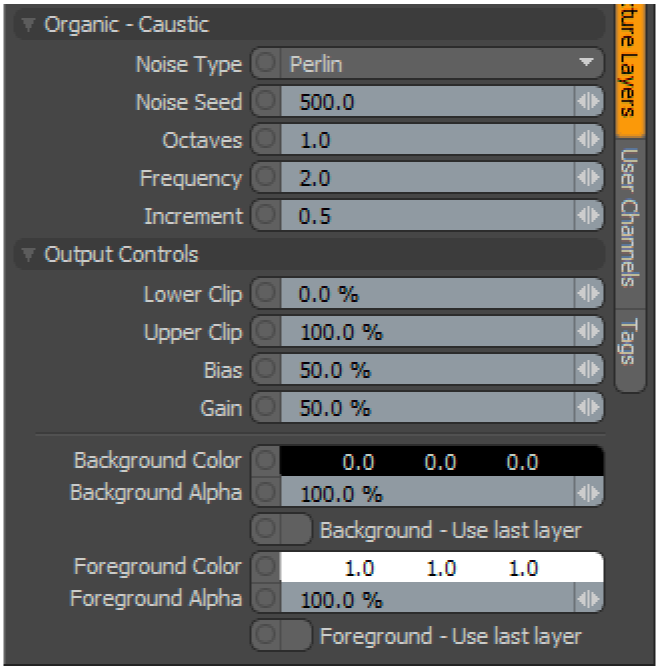
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Caustic Properties



Organic - Caustic
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 1.0
Frequency: 2.0
Increment: 0.5

Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %

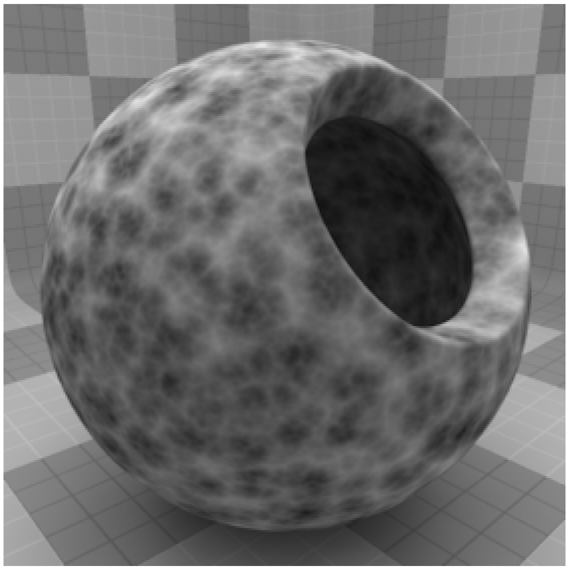
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Organic - Caustic	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided:

	• **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Cellular

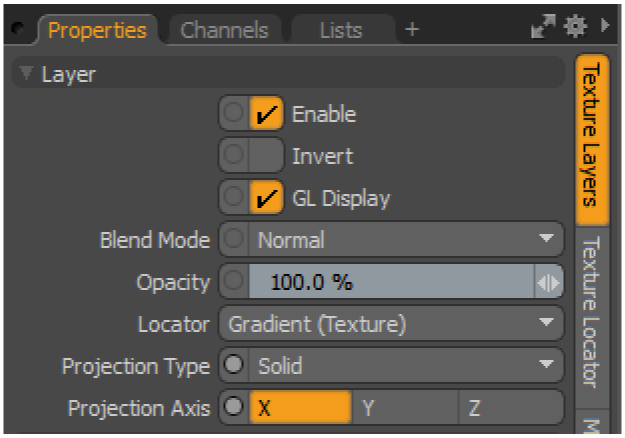


The **Cellular** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Cellular** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides a random cell-like pattern, similar to clusters of varying sized circles. The texture can also be bumped and colored to provide a wide variation of organic patterns.

Note: For information regarding adding and working with Shader Tree item layers see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

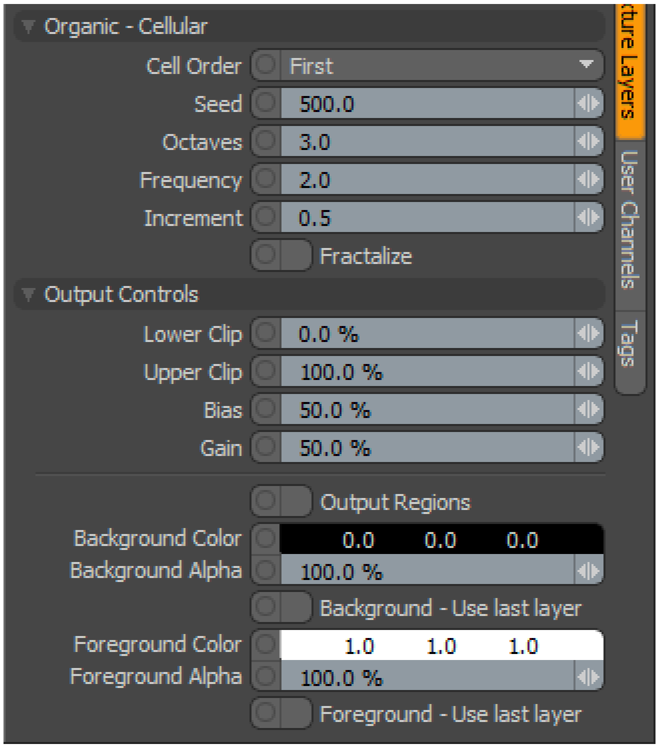
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Cellular Properties



Organic - Cellular
Cell Order: First
Seed: 500.0
Octaves: 3.0
Frequency: 2.0
Increment: 0.5
Fractalize

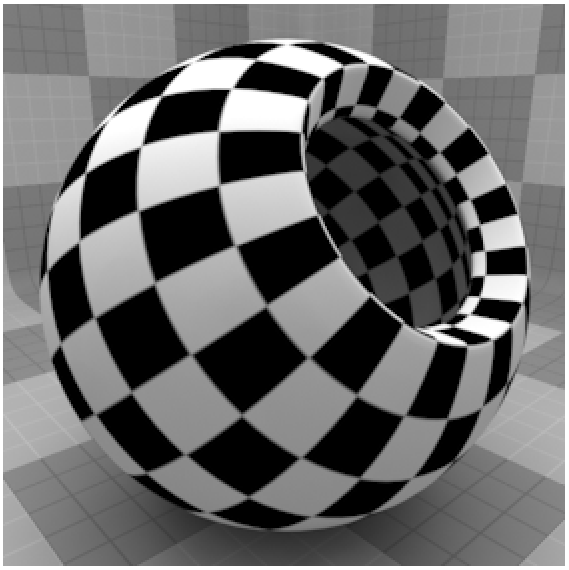
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Output Regions
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Organic - Cellular	
Cell Order	This option determines the cell order used for texture creation. • **First**: Nearest Cell • **Second**: Next Nearest Cell • **Third**: Next Nearest Cell • **Fourth**: Furthest Cell
Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates

	more turbulence, stirring up the pattern.
Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Fractalize	Determines whether or not the texture uses the fractalize algorithm, based on Fractal Brownian Motion (FBM).
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward

Color/Value	the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Checks



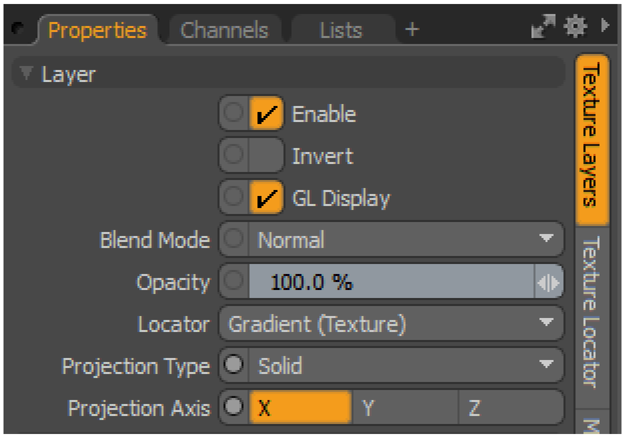
The **Checks** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Checks** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a standard check pattern, perfect for a chess board or marble floor-like covering.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the texture locator, press the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a **UV map** (highly recommended) or choose one of the geometric projections, such as **Planar**, **Cylindrical**, **Spherical**, and so on, and size the texture accordingly.

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

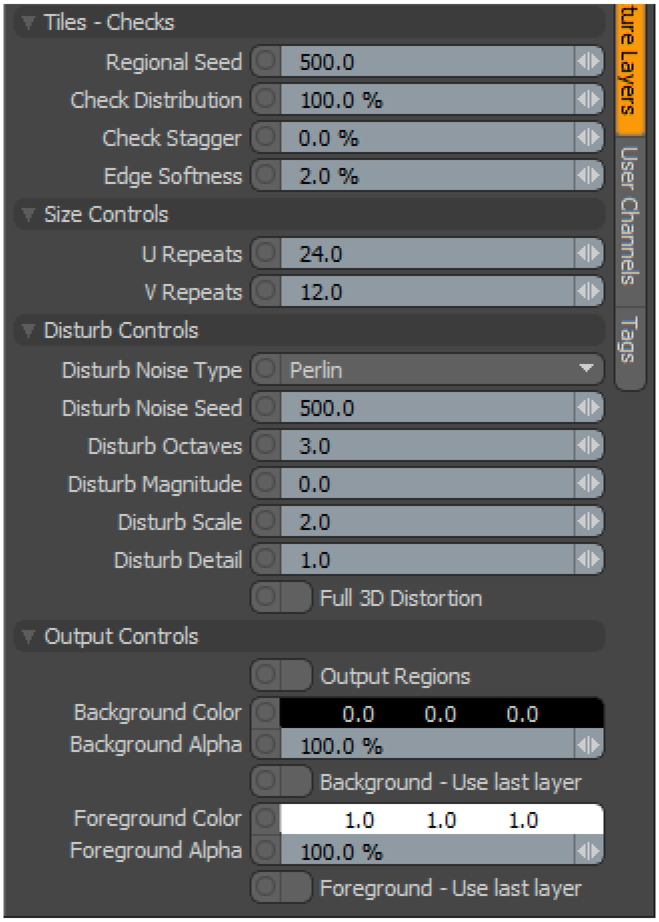
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Tiles Properties



Tiles - Checks
Regional Seed 500.0
Check Distribution 100.0 %
Check Stagger 0.0 %
Edge Softness 2.0 %

Size Controls
U Repeats 24.0
V Repeats 12.0

Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

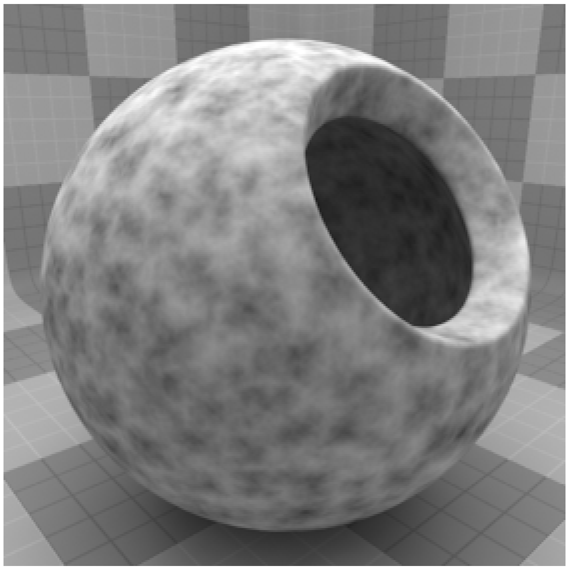
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Checks	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in

	changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Check Distribution	Specifies the percentage of the surface covered by checks.
Check Stagger	Specifies the amount of staggering between check rows. When the control is at a low value, the tiles line up vertically, with the tile rows becoming more staggered as the parameter is increased.
Edge Softness	Controls the blurring, or softness at the edges of the checks.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the U direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the V direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

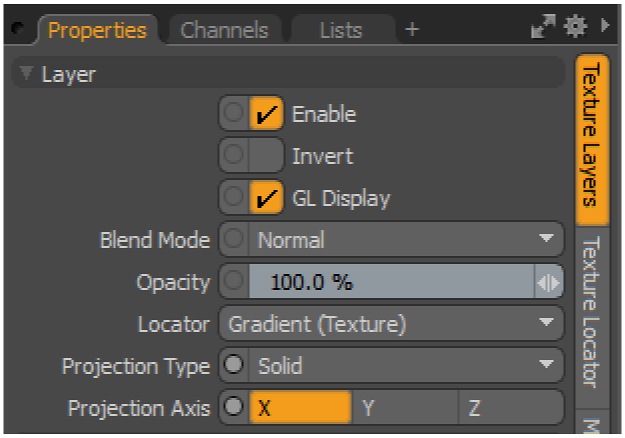
Cheesy



The **Cheesy** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Cheesy** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader is similar to the **Cellular** shader, except that the distance between cells have been squared off. This texture can be used to create a rectangular pattern for machinery-like surfaces.

Note: For information regarding adding and working with Shader Tree item layers see the Shader Tree topic.

Layer Properties

>


Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

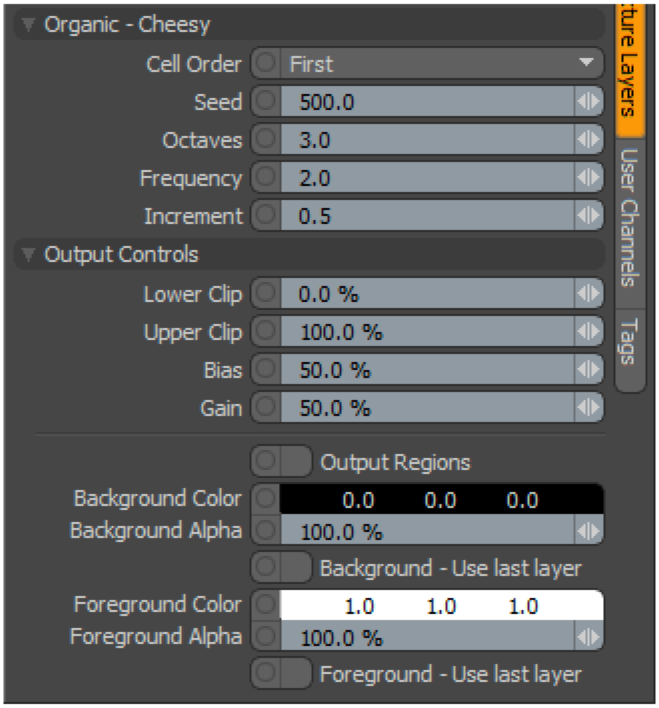
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Cheesy Properties



Organic - Cheesy

Cell Order: First

Seed: 500.0

Octaves: 3.0

Frequency: 2.0

Increment: 0.5

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Output Regions

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

Background - Use last layer

Foreground Color: 1.0 1.0 1.0

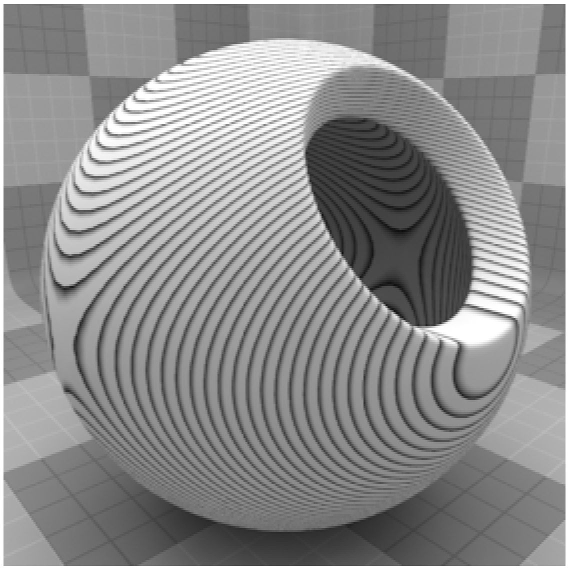
Foreground Alpha: 100.0 %

Foreground - Use last layer

Organic - Cheesy	
Cell Order	This option determines the cell order used for texture creation. • **First**: Nearest Cell • **Second**: Next Nearest Cell • **Third**: Next Nearest Cell • **Fourth**: Furthest Cell
Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.

Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Circular

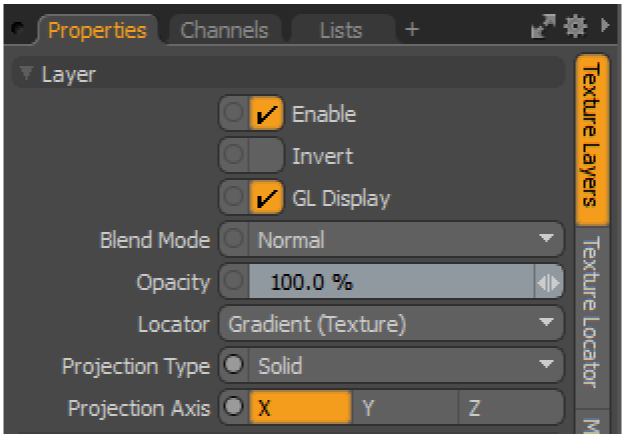


The **Circular** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Circular** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a series of adjustable circular patterns.

Note: For information regarding adding and working with Shader Tree item layers see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

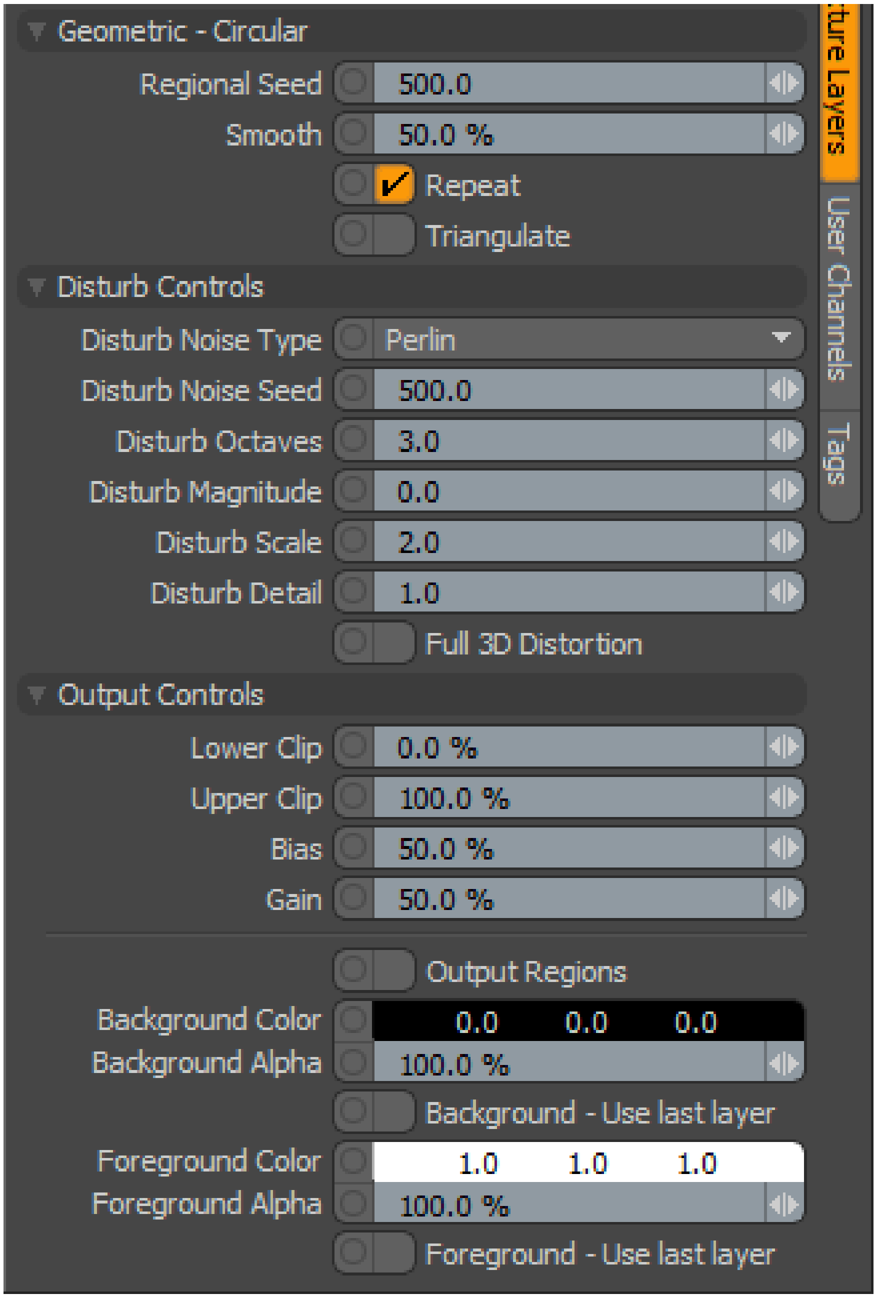
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Circular Properties



Geometric - Circular

Regional Seed 500.0
Smooth 50.0 %
☑ Repeat
☐ Triangulate

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion

Output Controls

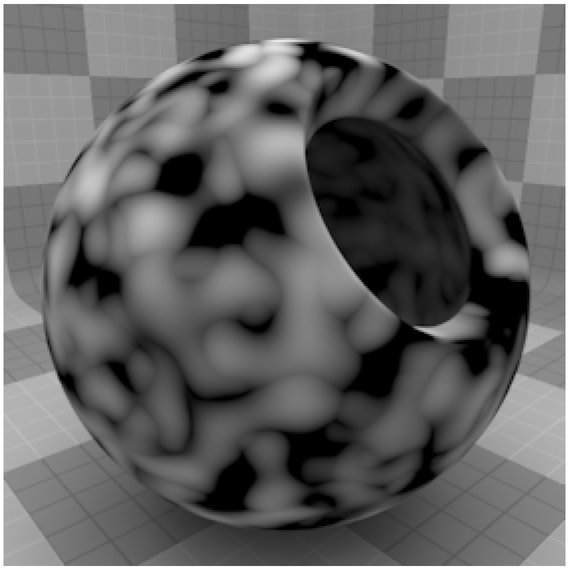
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Geometric - Circular	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is

	necessary for items to retain the same variations.
Smooth	Determines the amount of smoothing around the edges of the circle. Higher values result in smoother edges.
Repeat	When enabled, this option causes the texture to repeat indefinitely.
Triangulate	When enabled, this option causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Concrete

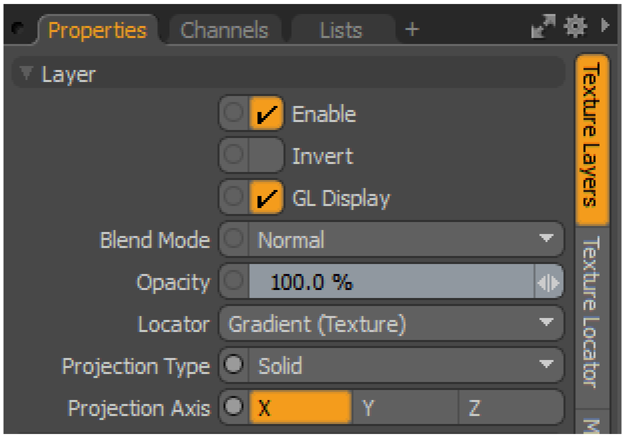


The **Concrete** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Concrete** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this provides a look similar to concrete, with three interleaved bump map patterns.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

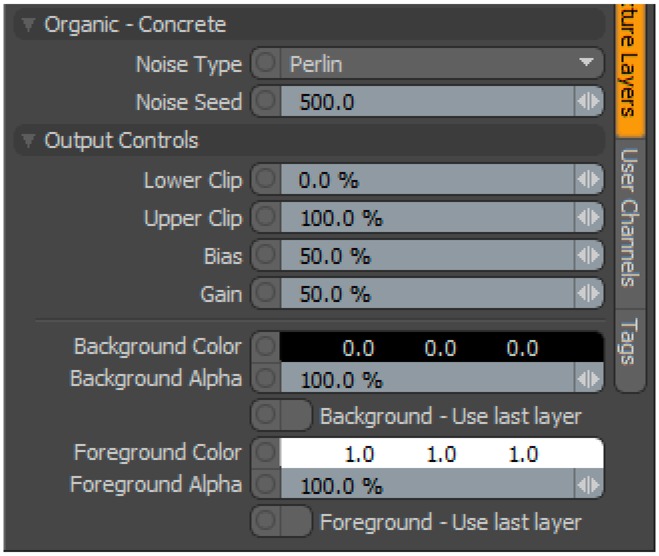
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

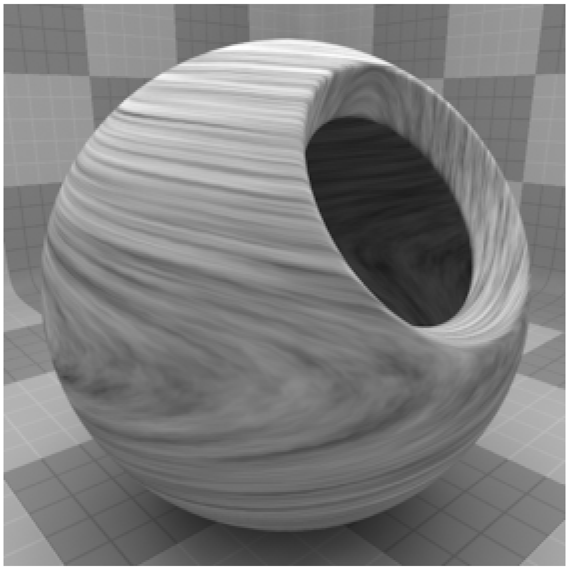
Concrete Properties


Organic - Concrete
Noise Type: Perlin
Noise Seed: 500.0
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Organic - Concrete	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.

Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Coriolis

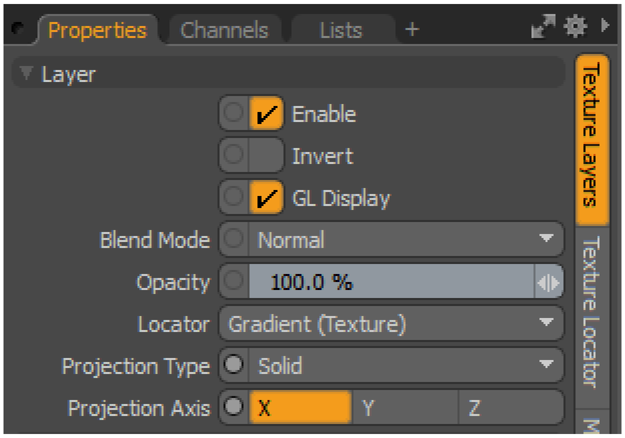


The **Coriolis** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Coriolis** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader applies a pattern similar to the swirling, windswept clouds of the Coriolis effect. The amount of twist or swirling can be controlled, along with the detail of the clouds.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

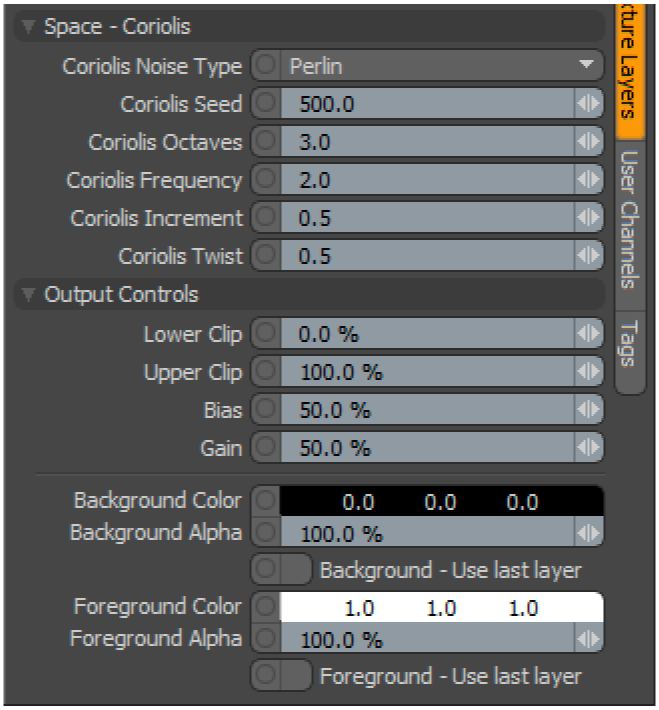
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Coriolis Properties



Space - Coriolis

Coriolis Noise Type: Perlin
Coriolis Seed: 500.0
Coriolis Octaves: 3.0
Coriolis Frequency: 2.0
Coriolis Increment: 0.5
Coriolis Twist: 0.5

Output Controls

Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %

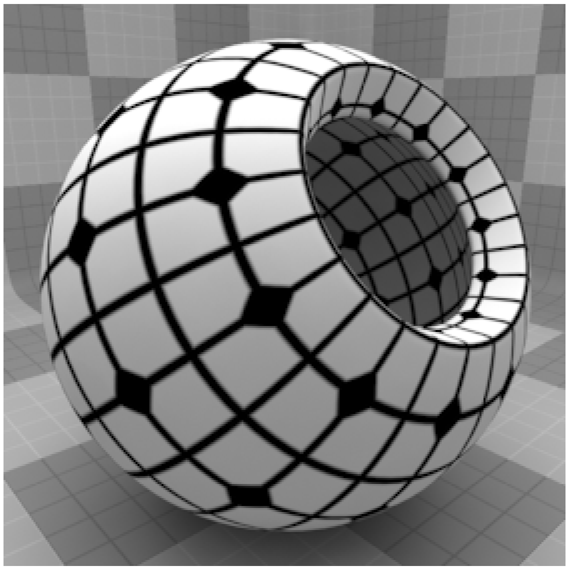
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Space - Coriolis	
Coriolis Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Coriolis Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.

Coriolis Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Coriolis Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Coriolis Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Coriolis Twist	This value controls the amount of Coriolis effect, or twist applied to the surface.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Cornerless

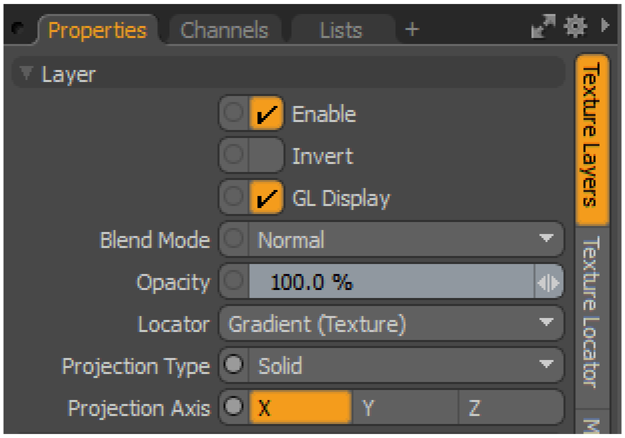


The **Cornerless** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Cornerless** texture can be addressed by its two zones,the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a square tile pattern with variable grout, tiles edges and tile height. The tiles may be staggered and random tiles can be removed. The major difference with this tile shader is that one corner is missing, which, combined with the **Taco** shader, allows for complex tile patterns to be created.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the texture locator, press the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a **UV map** (highly recommended) or choose one of the geometric projections, such as **Planar**, **Cylindrical**, **Spherical**, and so on, and size the texture accordingly.

Layer Properties



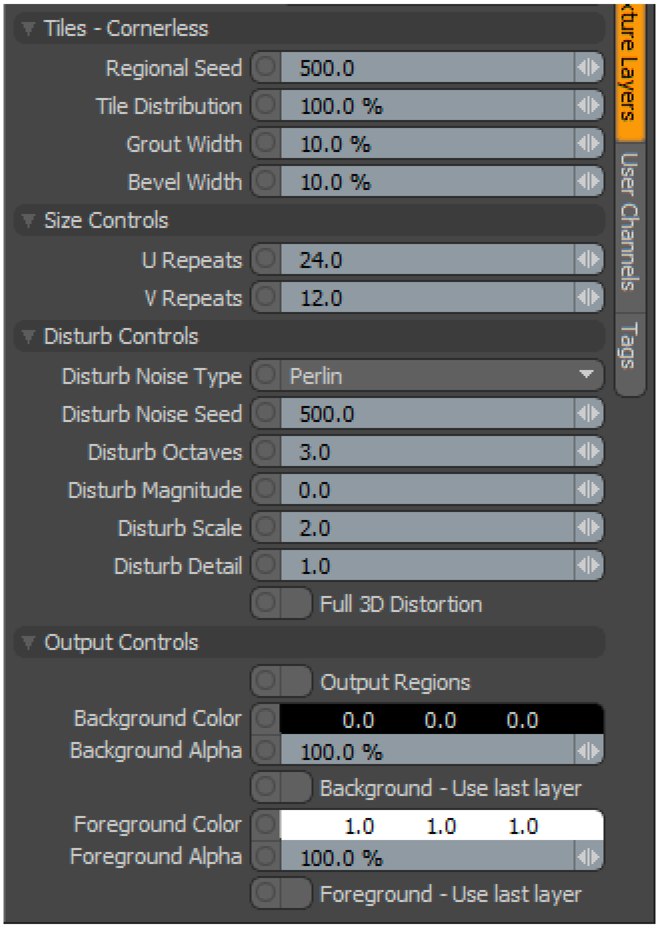
☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects.

	For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Cornerless Properties



Tiles - Cornerless
Regional Seed 500.0
Tile Distribution 100.0 %
Grout Width 10.0 %
Bevel Width 10.0 %

Size Controls
U Repeats 24.0
V Repeats 12.0

Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

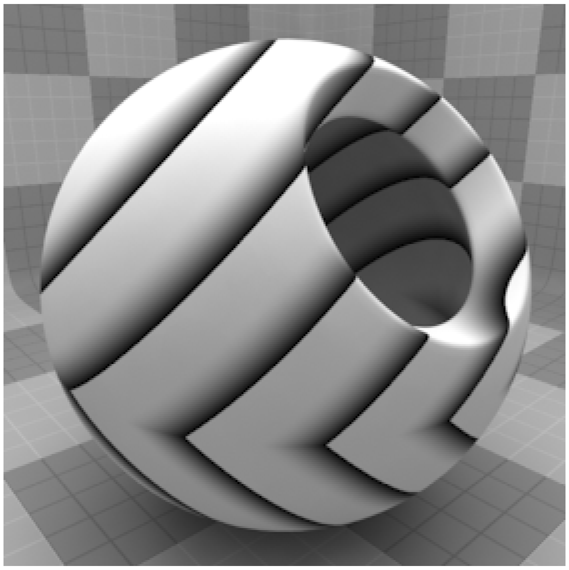
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Cornerless	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Tile Distribution	Specifies the percentage of the surface covered by tiles.
Grout Width	Specifies the amount of grout between the tiles.
Bevel Width	Specifies the width of the bevel on the tiles.
Size Controls	

U Repeats	Determines the number of times the pattern repeats in the U direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the V direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.

Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Corners

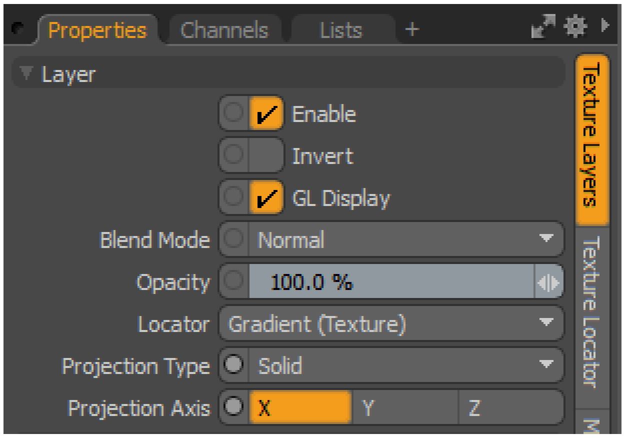


The **Corners** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Corners** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a series of adjustable corner patterns.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

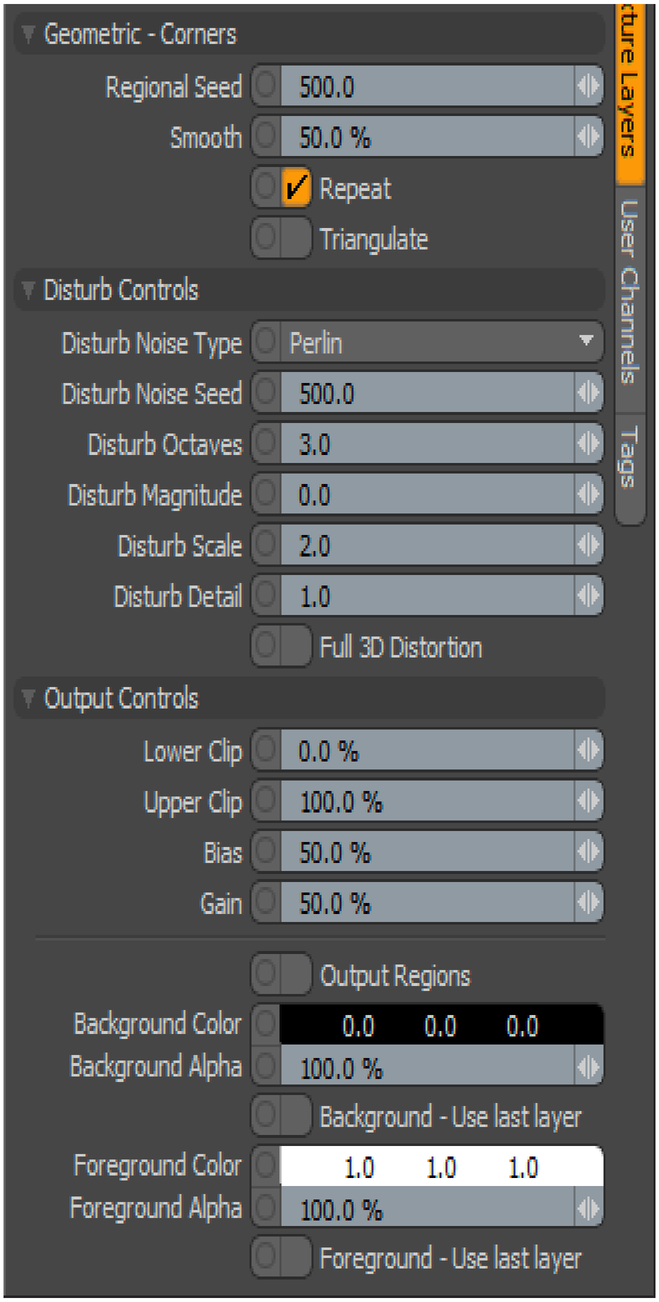
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Corners Properties



Geometric - Corners

Regional Seed 500.0

Smooth 50.0 %

☑ Repeat

☐ Triangulate

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

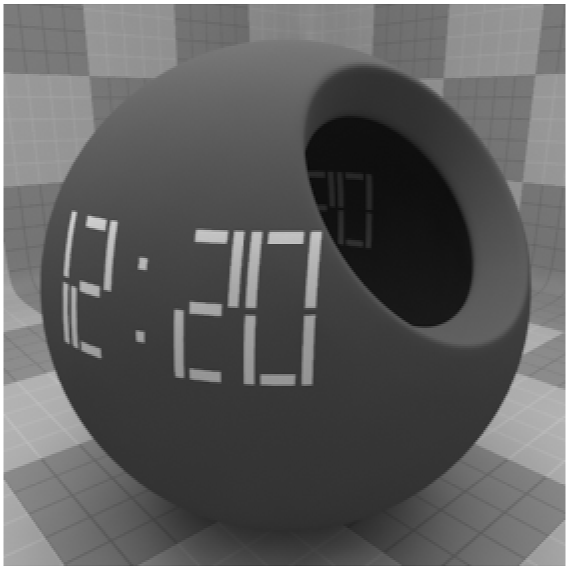
☐ Foreground - Use last layer

Geometric - Corners	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is

	necessary for items to retain the same variations.
Smooth	Determines the amount of smoothing around the edges of the circle. Higher values result in smoother edges.
Repeat	When enabled, this option causes the texture to repeat indefinitely.
Triangulate	When enabled, this option causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Counter 1

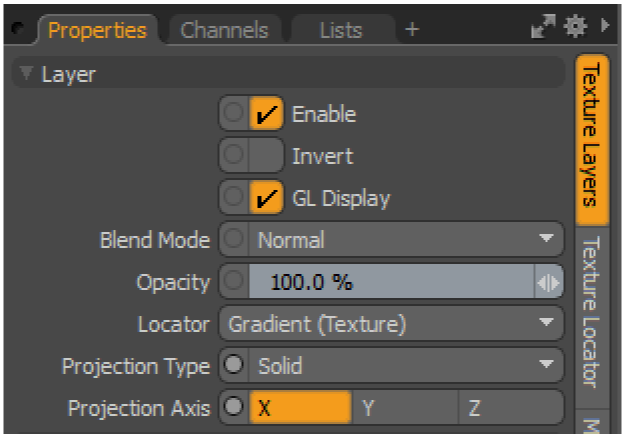


The **Counter 1** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The shaders in the **Display** group differ from the other **Enhance:Modo** textures in that they present procedurally generated text rather than traditional textures. Similar to the other shaders, **Counter 1** can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader is a digital timer readout, such as those seen on a speedometer or a clock.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

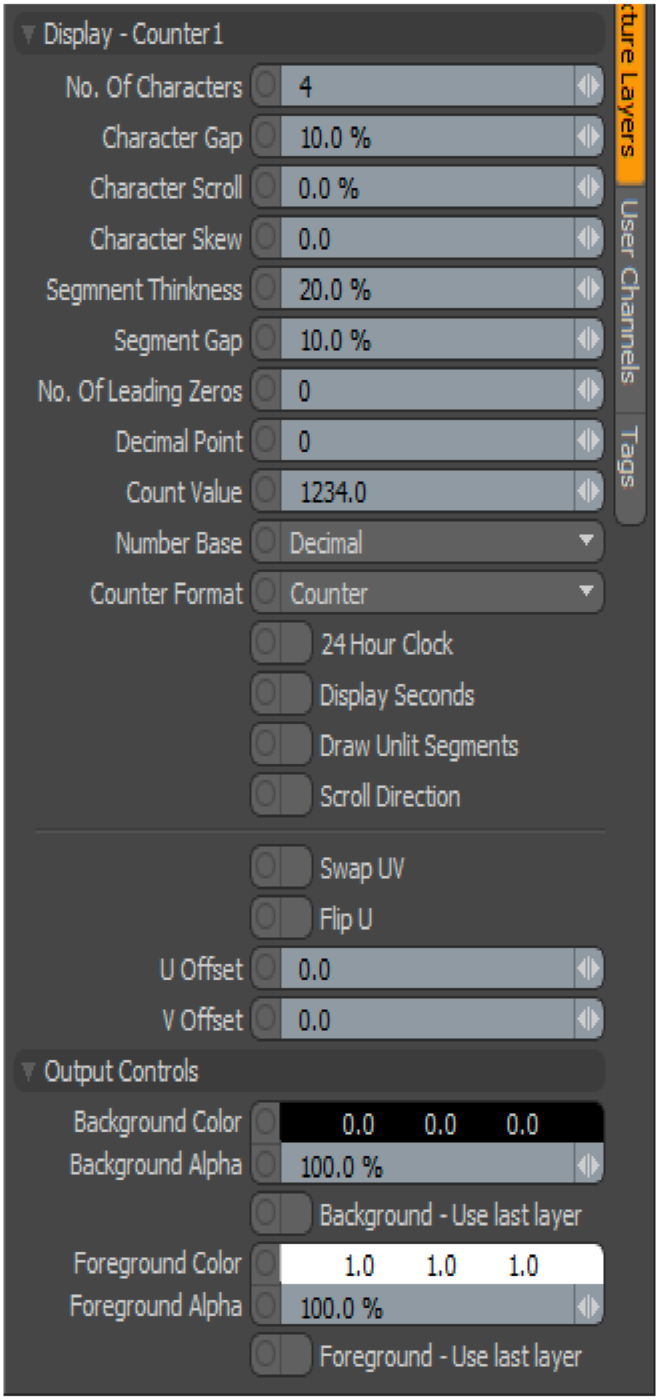
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Counter 1 Properties



Display - Counter 1

No. Of Characters: 4

Character Gap: 10.0 %

Character Scroll: 0.0 %

Character Skew: 0.0

Segmnent Thinkness: 20.0 %

Segment Gap: 10.0 %

No. Of Leading Zeros: 0

Decimal Point: 0

Count Value: 1234.0

Number Base: Decimal

Counter Format: Counter

☐ 24 Hour Clock

☐ Display Seconds

☐ Draw Unlit Segments

☐ Scroll Direction

☐ Swap UV

☐ Flip U

U Offset: 0.0

V Offset: 0.0

Output Controls

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

☐ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

☐ Foreground - Use last layer

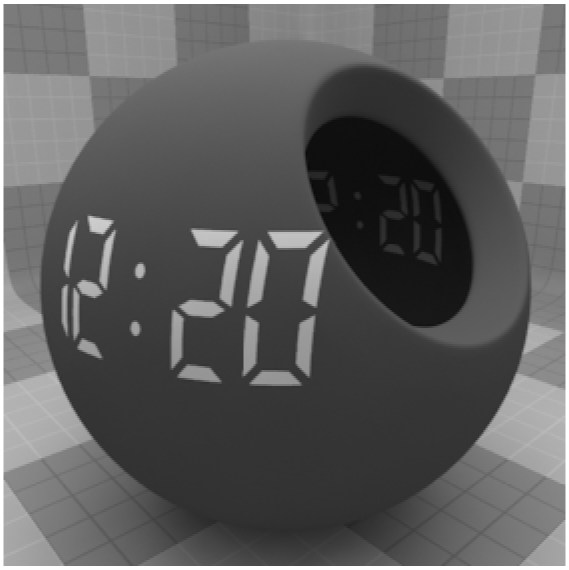
Display - Counter 1	
Number of	Specifies the total number of characters to display.

Characters	
Character Gap	Specifies the spacing between the individual characters within the overall content dimensions.
Character Scroll	Allows the characters to scroll on a character-by-character basis. As this value is increased, the characters scroll from left to right.
Character Skew	Applies a skew to all of the display characters. Positive values skew the characters to the right, while negative values skew the characters to the left.
Segment Thickness	Defines the thickness of the individual segments for each display character.
Segment Gap	Determines the spacing between the individual segments for each character.
No. Of Leading Zeros	Prefixes the counter with a defined number of zeros.
Decimal Point	Specifies the number of zeros to be shown after a decimal point.
Count Value	Specifies the number that is shown in the display. In the case of time, the value is represented in seconds, then converted to hours:minutes in the display.
Number Base	Specifies the number base used to display the count value. The options are **Decimal**, **Hexadecimal**, **Octal** and **Binary**.
Counter Format	Specifies the format of the count for display. Options are **Simple Count**, **Realtime Clock**,or **SMPTE** time code (either **SMPTE24**, **SMPTE25** or **SMPTE30**.)
24 Hour Clock	Forces a 24-hour clock display when the **Realtime Clock** option is selected in the **Counter Format**.
Display Seconds	When this option is enabled and the **Counter Format** is set to **Realtime Clock**, the character for seconds is included in the display.
Draw Unlit Segments	When enabled, all of the unlit segments and decimals are included in the display.
Scroll Direction	When enabled, the scroll direction switches from the default setting of left to right and scroll from right to left.
Swap UV	When enabled, the UV coordinates of the display text are swapped.

Flip U	When enabled, it flips the U coordinates of the display text.
U Offset	Specifies U (horizontal) offset of the display text.
V Offset	Specifies V (vertical) offset of the display text.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground	Specifies the **Alpha** transparency of the **Foreground Color**.

Alpha	
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

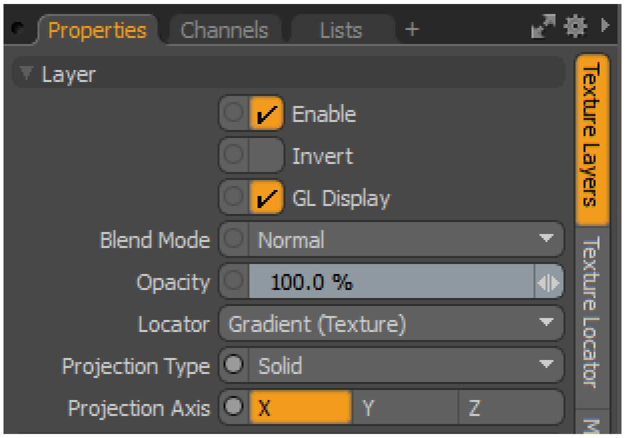
Counter 2



The **Counter 2** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The shaders in the **Display** group differ from the other **Enhance:Modo** textures in that they present procedurally generated text rather than traditional textures. Similar to the other shaders, **Counter 2** can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader is a digital timer readout, such as those seen on a speedometer or clock. The difference between **Counter 1** and **Counter 2** is that **Counter 2** provides a more modern display face, with beveled corners.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

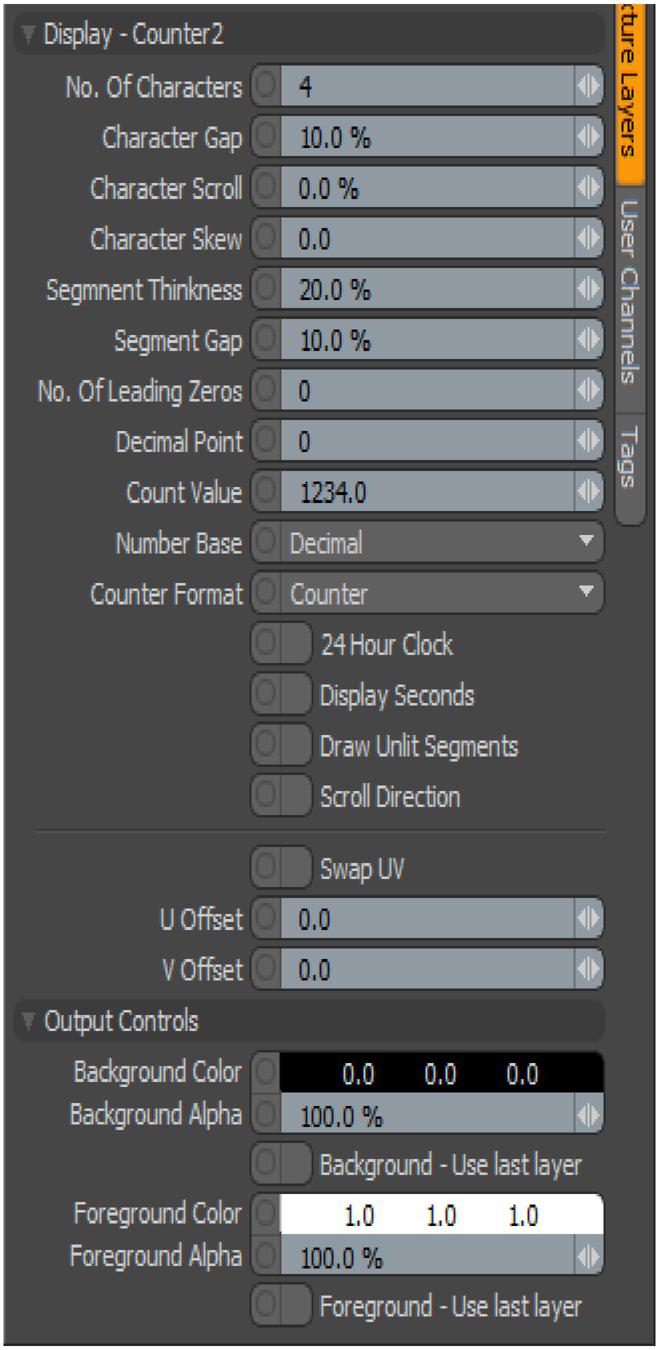
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X (selected) Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Counter 2 Properties



Display - Counter2

No. Of Characters: 4
Character Gap: 10.0 %
Character Scroll: 0.0 %
Character Skew: 0.0
Segmnent Thinkness: 20.0 %
Segment Gap: 10.0 %
No. Of Leading Zeros: 0
Decimal Point: 0
Count Value: 1234.0
Number Base: Decimal
Counter Format: Counter
☐ 24 Hour Clock
☐ Display Seconds
☐ Draw Unlit Segments
☐ Scroll Direction
☐ Swap UV
U Offset: 0.0
V Offset: 0.0

Output Controls

Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

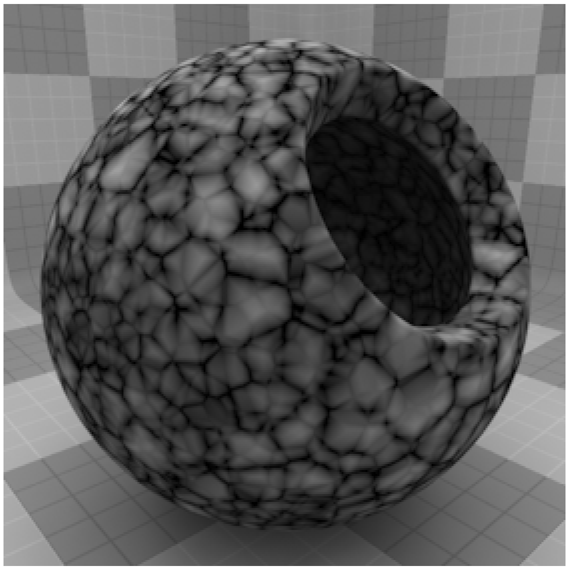
Option	Description
Layer	

Option	Description
Enable	Toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
Invert	Inverts the RGB values for the layer producing a negative effect.
Blend Mode	Affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, please see the Layer Blend Modes page of the documentation.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the Shader Tree if present, or dim the effect of the layer itself on the surface.
Locator	Most texture layers also have an associated Texture Locator that is automatically created in the **Item List**. This defines the mapping of the texture, the way it is applied, to the surface. The **Locator** option sets that association. While you can choose alternate locators, the need to do so is very rare, still, there are some possible instances where you may want multiple texture items to share a single locator.
Display - Counter 2	
Number of Characters	Specifies the total number of characters to display.
Character Gap	Specifies the spacing between the individual characters within the overall content dimensions.
Character Scroll	Allows the characters to scroll on a character-by-character basis. As this value is increased, the characters scroll from left to right.
Character Skew	Applies a skew to all of the display characters. Positive values skew the characters to the right, while negative values skew the characters to the left.
Segment Thickness	Defines the thickness of the individual segments for each display character.
Segment Gap	Determines the spacing between the individual segments for each character.

Option	Description
No. Of Leading Zeros	Prefixes the counter with a defined number of zeros.
Decimal Point	Specifies the number of zeros to be shown after a decimal point.
Count Value	Specifies the number that is shown in the display. In the case of time, the value is represented in seconds, then converted to hours:minutes in the display.
Number Base	Specifies the number base used to display the count value. The options are **Decimal**, **Hexadecimal**, **Octal** and **Binary**.
Counter Format	Specifies the format of the count for display. Options are **Simple Count**, **Realtime Clock**,or **SMPTE** time code (either **SMPTE24**, **SMPTE25** or **SMPTE30**.)
24 Hour Clock	Forces a 24-hour clock display when the **Realtime Clock** option is selected in the **Counter Format**.
Display Seconds	When this option is enabled and the **Counter Format** is set to **Realtime Clock**, the character for seconds is included in the display.
Draw Unlit Segments	When enabled, all of the unlit segments and decimals are included in the display.
Scroll Direction	When enabled, the scroll direction switches from the default setting of left to right and scroll from right to left.
Swap UV	When enabled, the UV coordinates of the display text are swapped.
Flip U	When enabled, it flips the U coordinates of the display text.
U Offset	Specifies U (horizontal) offset of the display text.
V Offset	Specifies V (vertical) offset of the display text.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.

Option	Description
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Crackle

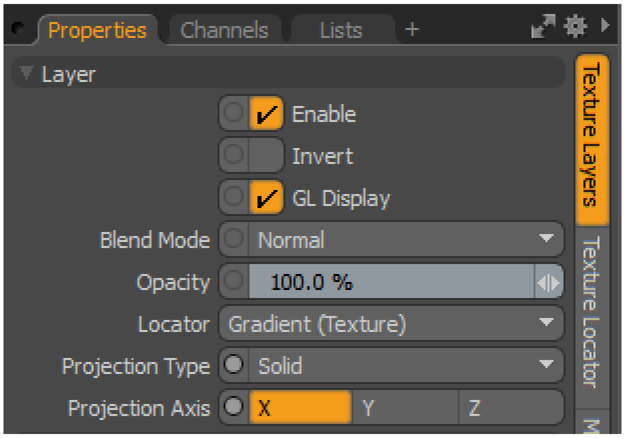


The **Crackle** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Crackle** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides a random crackle pattern, similar to a dried-up river bed. The shader can also be used to create veins or random lattice-type patterns.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

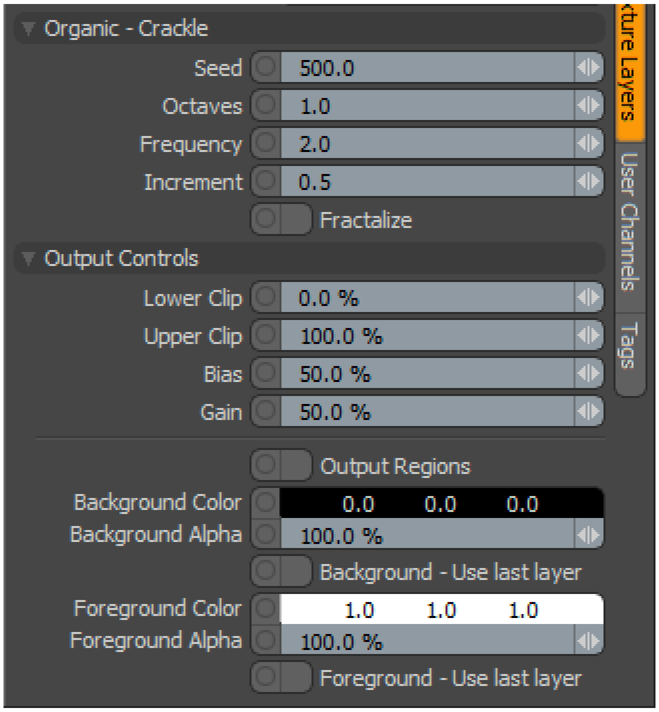
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Crackle Properties



Organic - Crackle

Seed 500.0
Octaves 1.0
Frequency 2.0
Increment 0.5
Fractalize

Output Controls

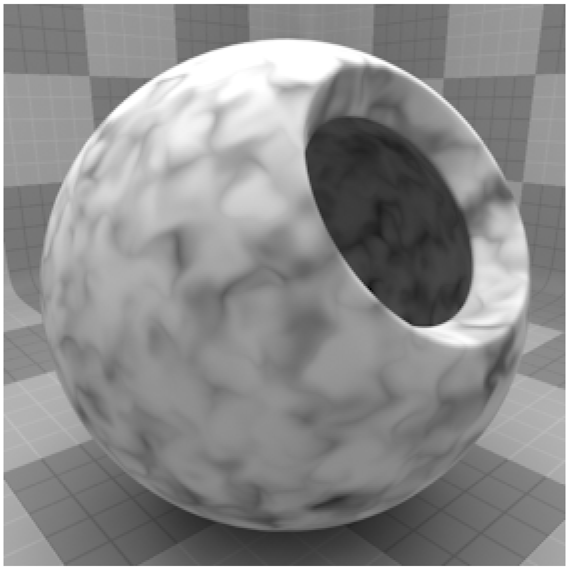
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %

Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Organic - Crackle	
Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Fractalize	Determines whether or not the texture uses the fractalize algorithm, based on Fractal Brownian Motion (FBM).

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Cruddy

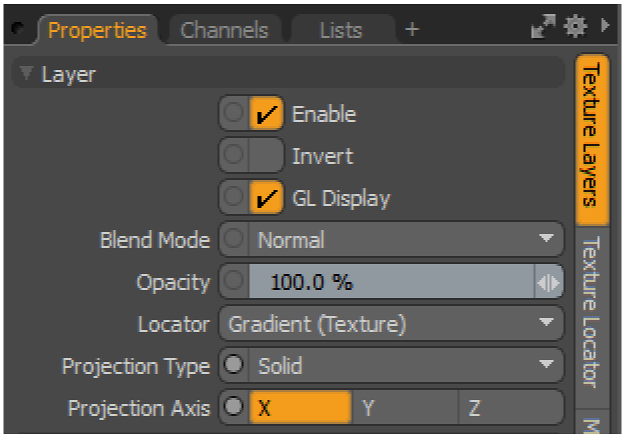


The **Cruddy** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Cruddy** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader creates a very rough crusty, cruddy type of pattern, much like the accumulation of dirt or rust on a surface.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

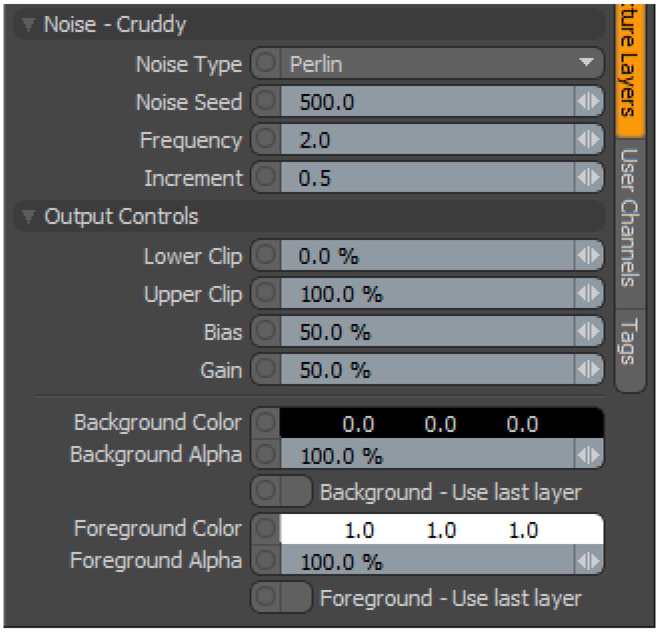
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.

Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Cruddy Properties



Noise - Cruddy

Noise Type: Perlin
Noise Seed: 500.0
Frequency: 2.0
Increment: 0.5

Output Controls

Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %

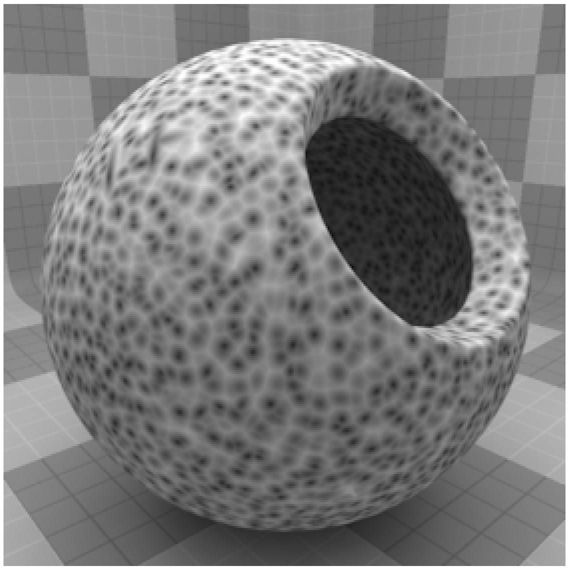
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
○ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
○ Foreground - Use last layer

Noise - Cruddy	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is

	necessary for items to retain the same variations.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward

Color/Value	the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Crumpled

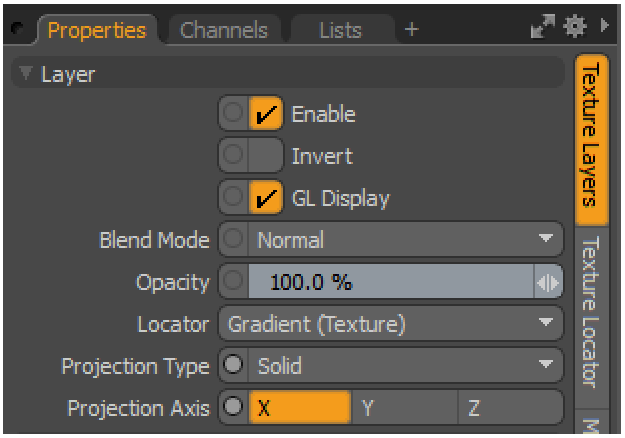


The **Crumpled** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Crumpled** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides an effect similar to used tinfoil, or a crumpled-and-flattened paper bag. By adding color, this can take on a planetary look, like the canals of Mars.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

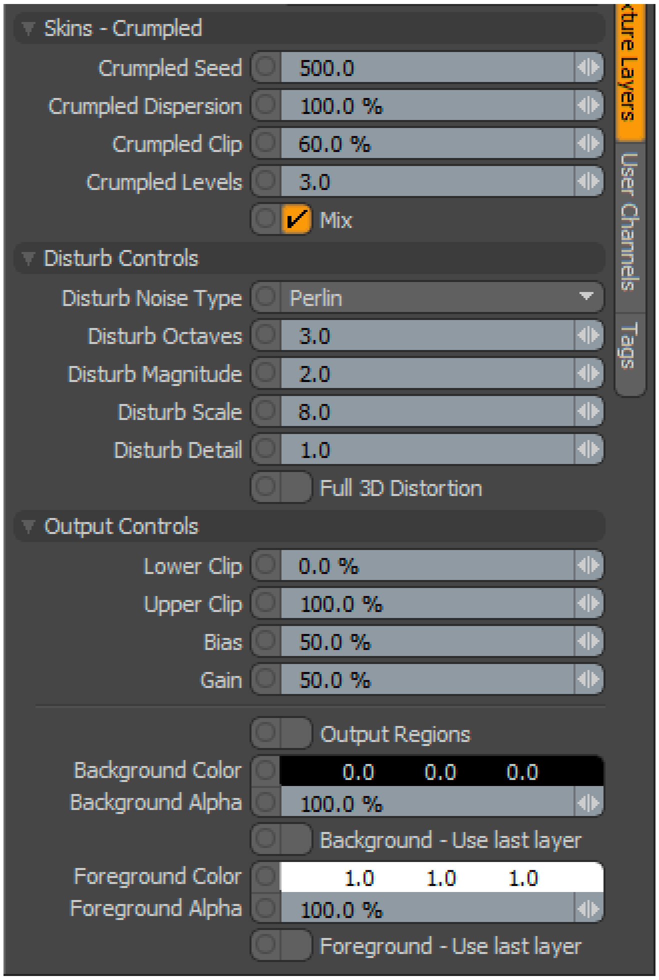
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Crumpled Properties



Skins - Crumpled

Crumpled Seed 500.0

Crumpled Dispersion 100.0 %

Crumpled Clip 60.0 %

Crumpled Levels 3.0

☑ Mix

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 2.0

Disturb Scale 8.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

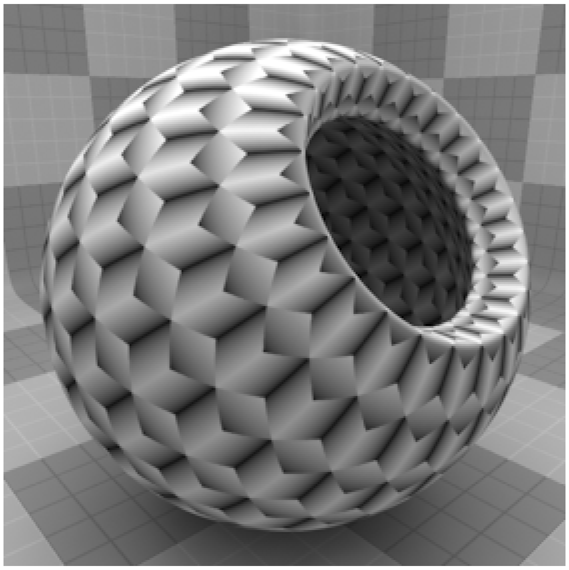
☐ Foreground - Use last layer

Skins - Crumpled	
Crumpled Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Crumpled	Determines how many of the crumpled sections are drawn.

Dispersion	
Crumpled Clip	Specifies where the texture shading transitions to the background color. The higher the value, the quicker the transition.
Crumpled Levels	Specifies the number of layers of the texture. Additional levels add additional detail, in turn also lightening the overall look of the texture.
Mix	When enabled, this option mixes the texture layers. When disabled, the texture layers are stacked one upon the other.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to

	extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Cubes



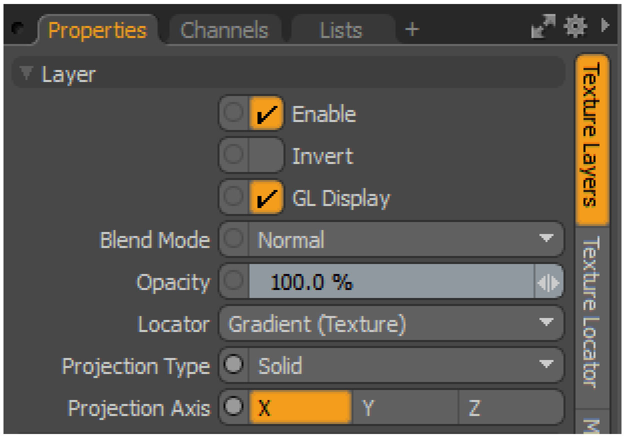
The **Cubes** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Cubes** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides a tri-colored isometric cube pattern.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the texture locator, press the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a **UV map** (highly recommended) or choose one of the geometric projections, such as **Planar**, **Cylindrical**, **Spherical**, and so on, and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

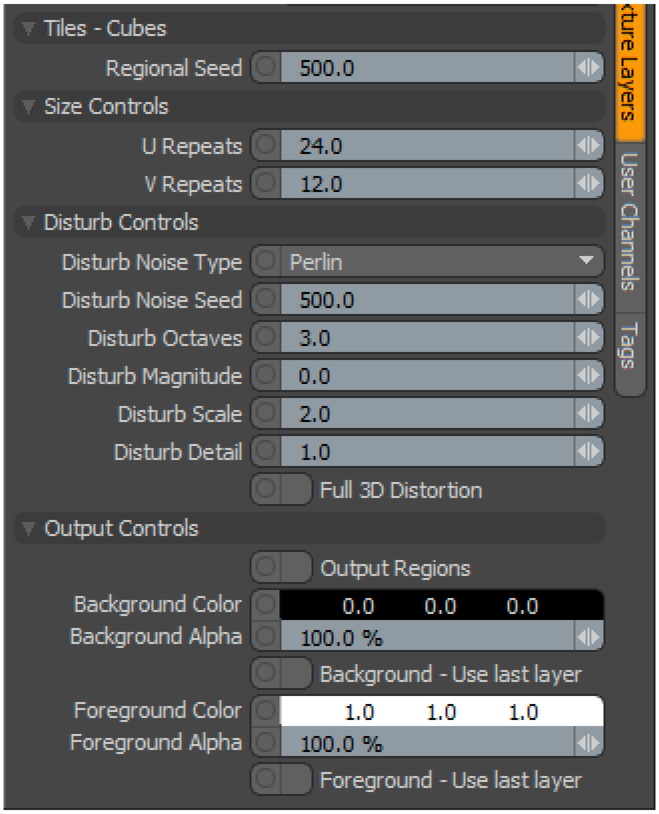
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Cubes Properties



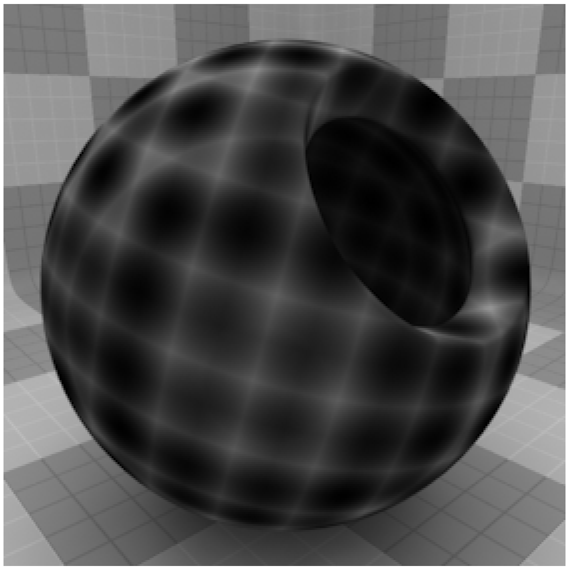
Tiles - Cubes
Regional Seed 500.0
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foregound - Use last layer

Tiles - Cubes	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the U direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the V direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided:

	• **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.

Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Cubic

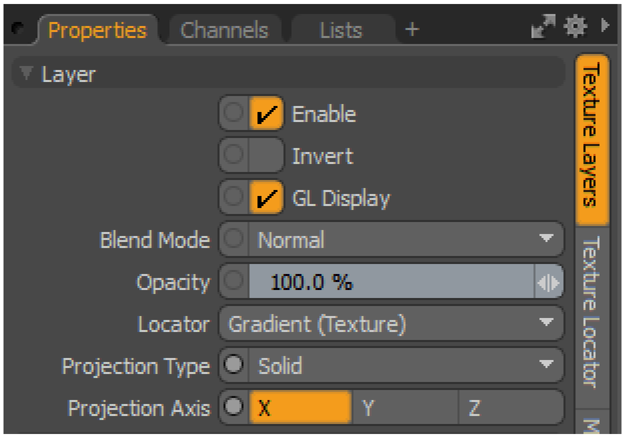


The **Cubic** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Cubic** texture can be addressed by its two zones, the 'Background' and 'Foreground' colors. the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a series of adjustable cubic patterns.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

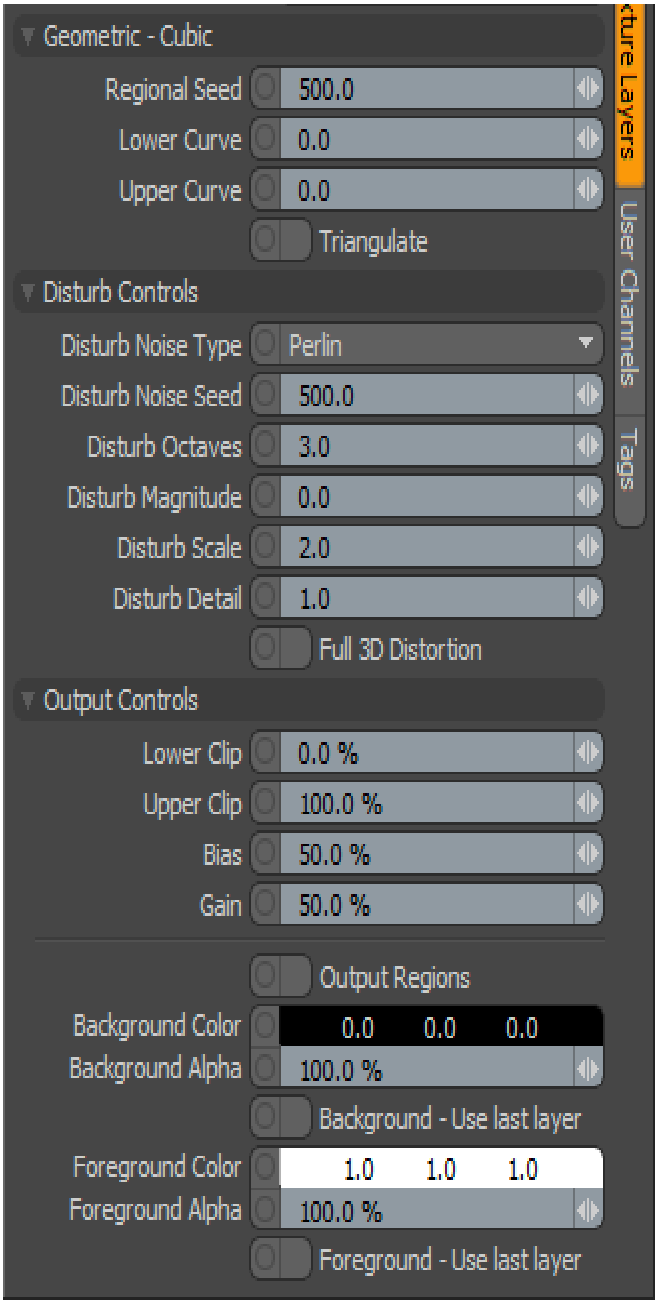


Enable ☑
Invert ☐
GL Display ☑
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Cubic Properties



Geometric - Cubic

Regional Seed 500.0

Lower Curve 0.0

Upper Curve 0.0

Triangulate

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

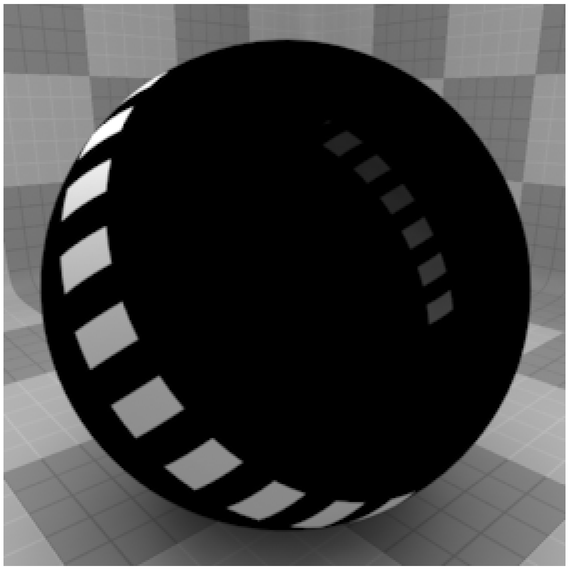
Foreground - Use last layer

Geometric - Cubic	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is

	necessary for items to retain the same variations.
Lower Curve	Determines the shape of the curve at the bottom of each cube.
Upper Curve	Determines the shape of the curve at the top of each cube.
Triangulate	When enabled, this option causes the pattern to have a triangle-based edging pattern, as opposed to the default sawtooth blend.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Dash Line



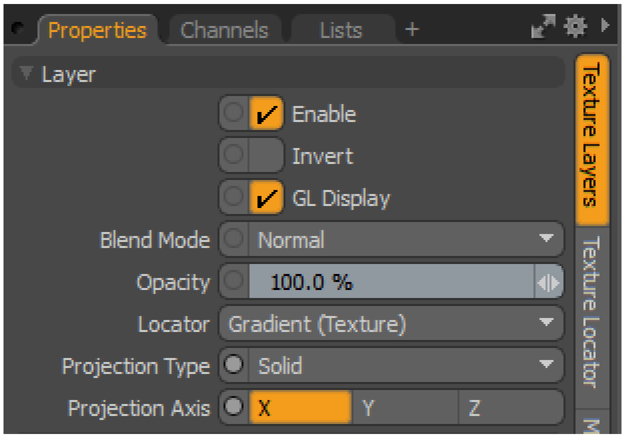
The **Dash Line** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Dash Line** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a dashed line, similar to the lane separator on highway. You can alter the layout of the dashes using a binary pattern.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the texture locator, press the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a **UV map** (highly recommended) or choose one of the geometric projections, such as **Planar**, **Cylindrical**, **Spherical**, and so on, and size the texture accordingly.

Layer Properties

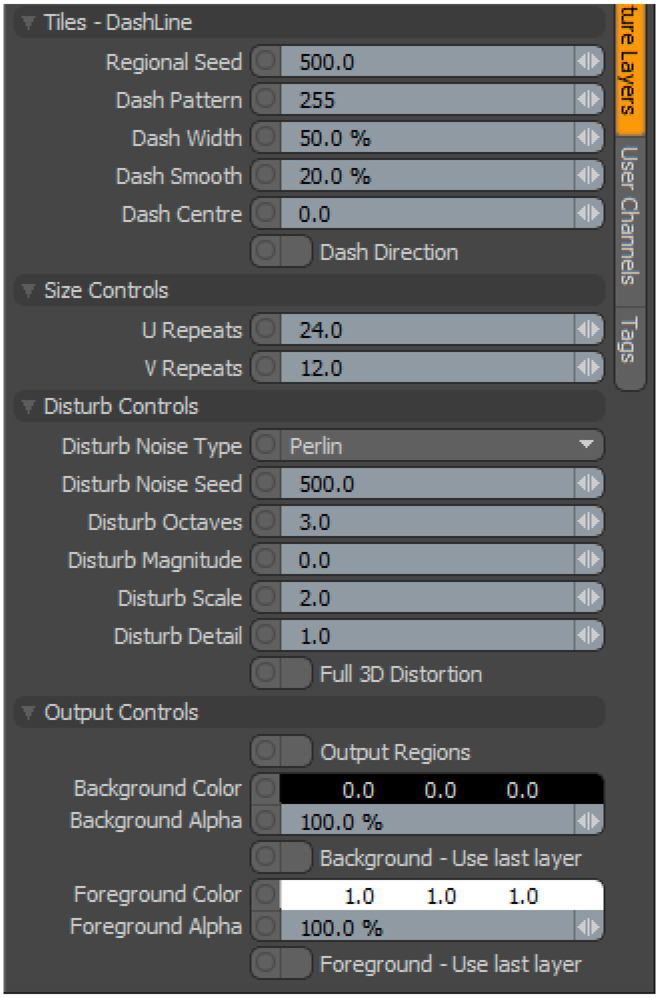
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: [✓ X] Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Dash Line Properties



Tiles - DashLine

Regional Seed 500.0
Dash Pattern 255
Dash Width 50.0 %
Dash Smooth 20.0 %
Dash Centre 0.0
○ Dash Direction

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
○ Full 3D Distortion

Output Controls

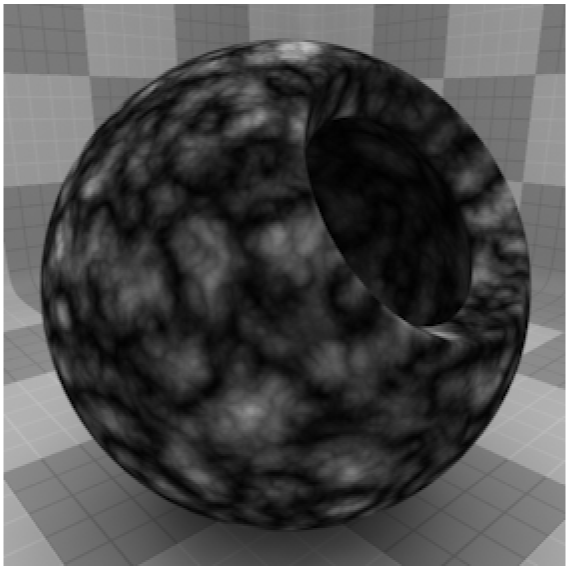
○ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
○ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
○ Foreground - Use last layer

Tiles - Dash Line	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Dash Pattern	Specifies the dash pattern for the texture.
Dash Width	Specifies the width of the dashed line.
Dash Smooth	Specifies the smoothness on the edges of the dashed line.

Dash Centre	Specifies the center point of the dashed line.
Dash Direction	Reverses the direction of the dash.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the U direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the V direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.

Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Dented

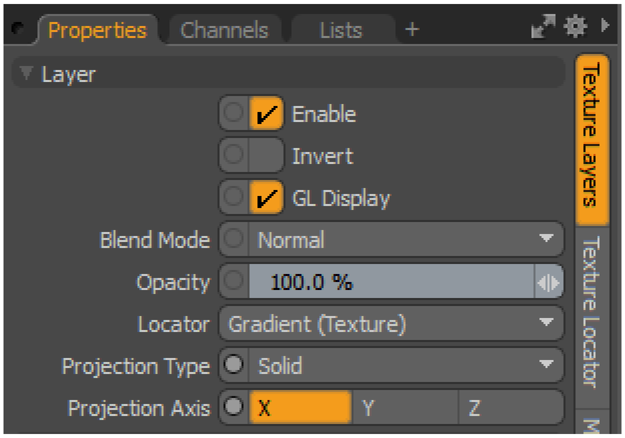


The **Dented** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Dented** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader can be used to provide variety of bumps that resemble dents. The bumps range from the kind of dents you might find in beaten up sheet metal to the deep craggy craters of an asteroid. Alternatively, the dents can be inverted to create pimples or detailed growths.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

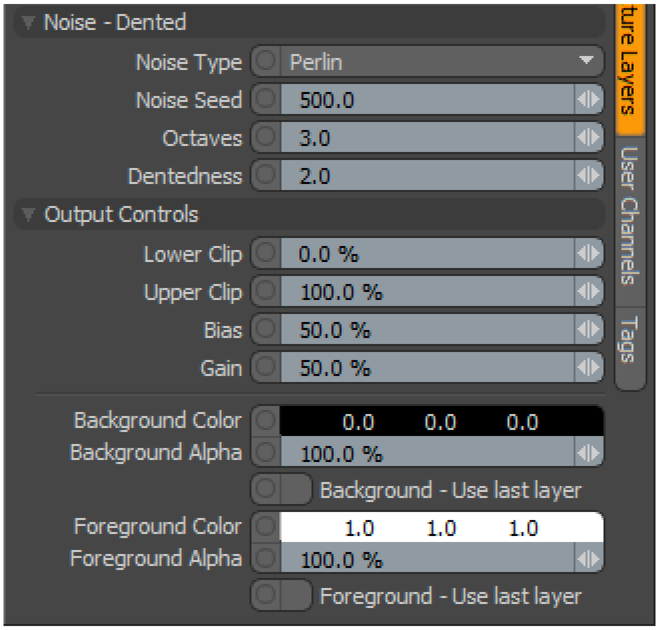
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Dented Properties



Noise - Dented

Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Dentedness: 2.0

Output Controls

Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %

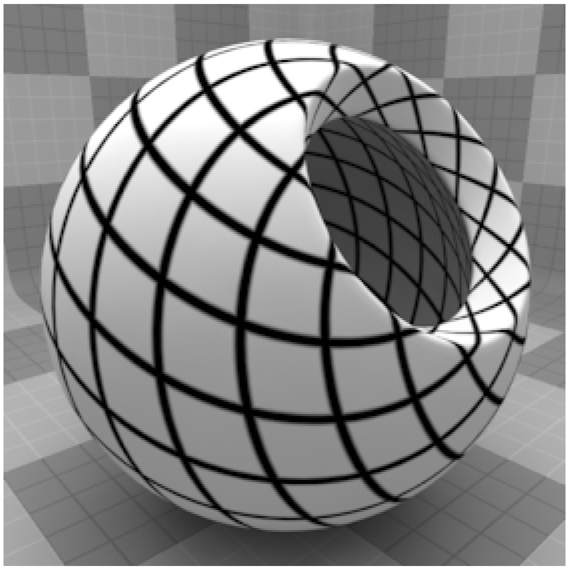
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
○ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
○ Foreground - Use last layer

Noise - Dented	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise

	being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Dentedness	Controls how dented the texture appears, with higher values generating smaller, more detailed dents.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground	Specifies the **Alpha** transparency of the **Foreground Color**.

Alpha	
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Diamond Deck



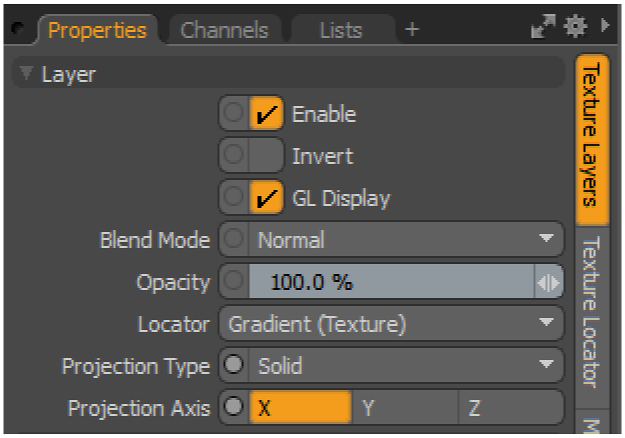
The **Diamond Deck** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Diamond Deck** texture can be addressed by its two zones, the the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a diamond tile pattern, similar to those found on certain types of man hole covers.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the texture locator, press the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a **UV map** (highly recommended) or choose one of the geometric projections, such as **Planar**, **Cylindrical**, **Spherical**, and so on, and size the texture accordingly.

Layer Properties

>

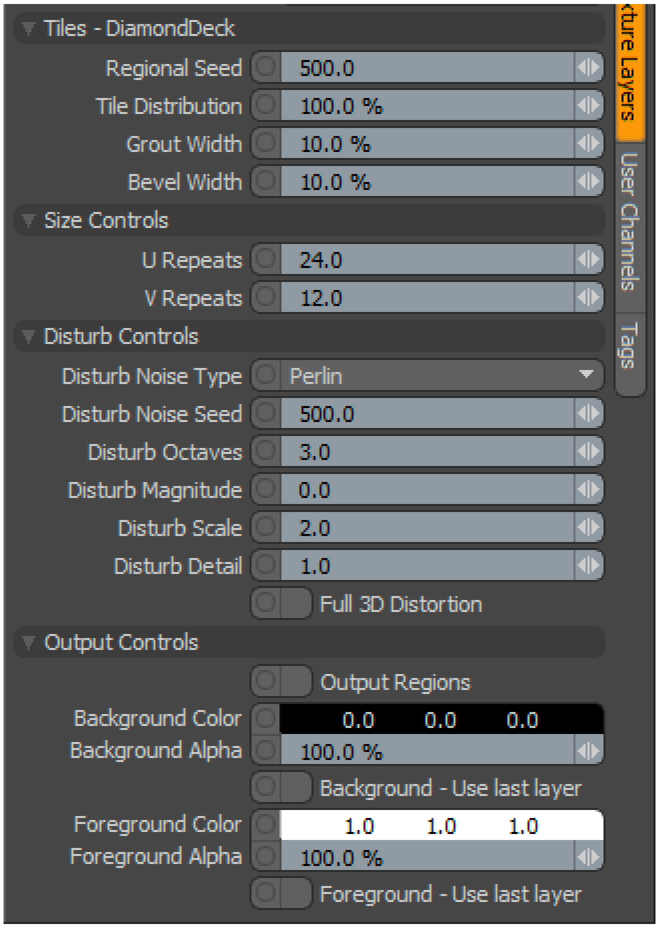


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Diamond Deck Properties



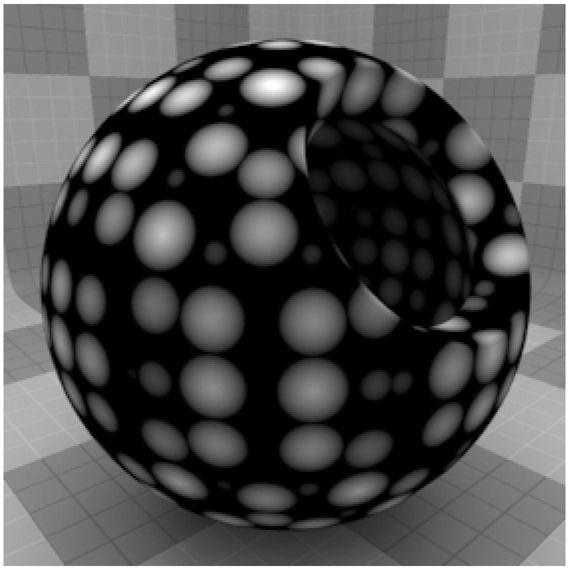
Tiles - DiamondDeck
Regional Seed 500.0
Tile Distribution 100.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
○ Full 3D Distortion
Output Controls
○ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
○ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
○ Foreground - Use last layer

Tiles - Diamond Deck	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in

	changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Tile Distribution	Specifies the percentage of the surface that is covered by tiles.
Grout Width	Specifies the amount of grout between the tiles.
Bevel Width	Specifies the width of the bevel on the tiles.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the U direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the V direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Dimples

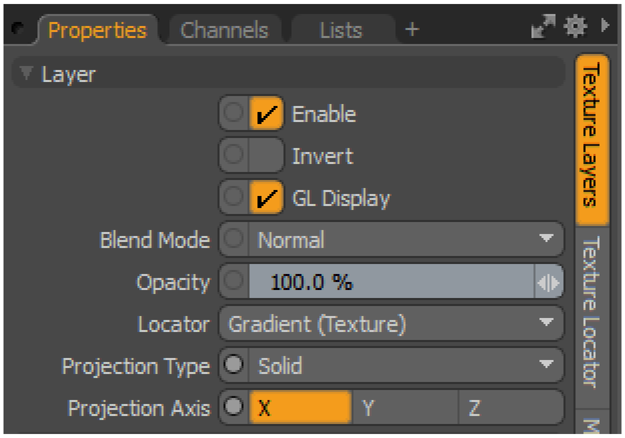


The **Dimples** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Dimples** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a series of adjustable dimple patterns.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

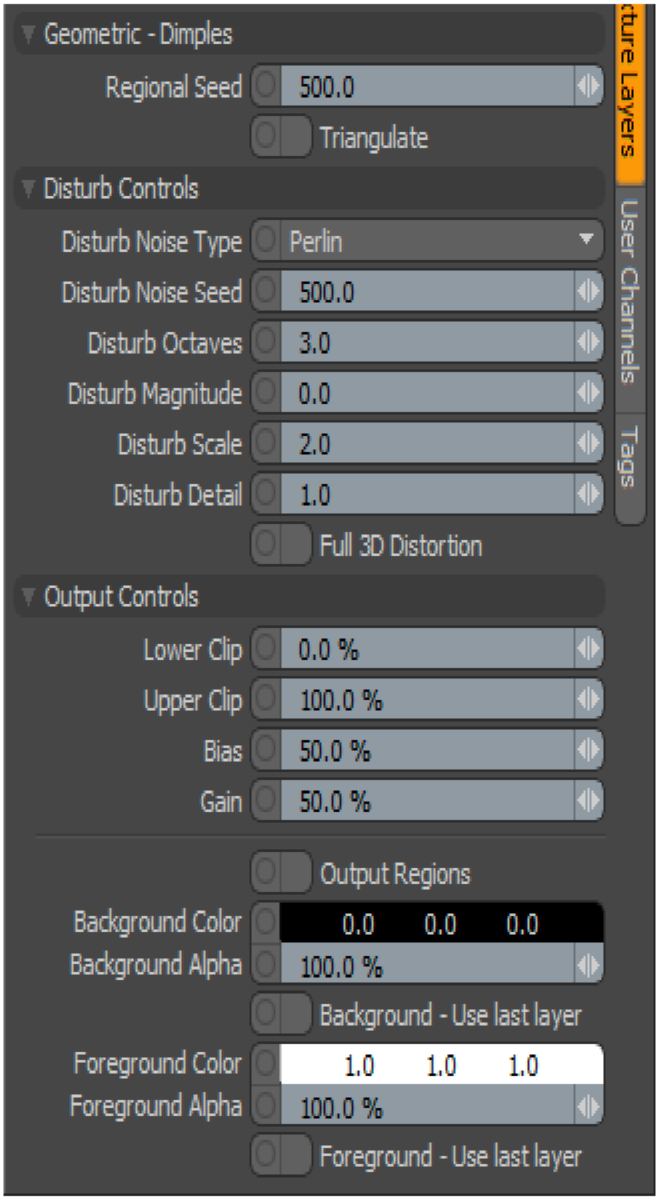
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Dimples Properties



Geometric - Dimples

Regional Seed 500.0

☐ Triangulate

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

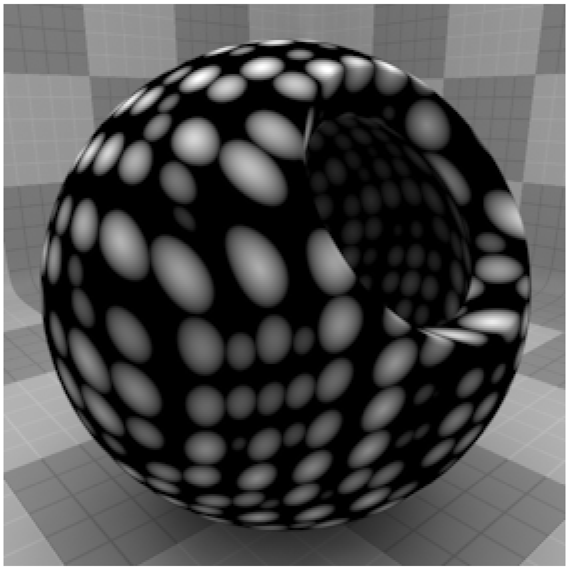
☐ Foreground - Use last layer

Geometric - Dimples	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Triangulate	When enabled, this option causes the pattern to have a triangle-based edging

	pattern, as opposed to the default sawtooth blend.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Disturb Octaves	Specifies the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to

	extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Dino Skin

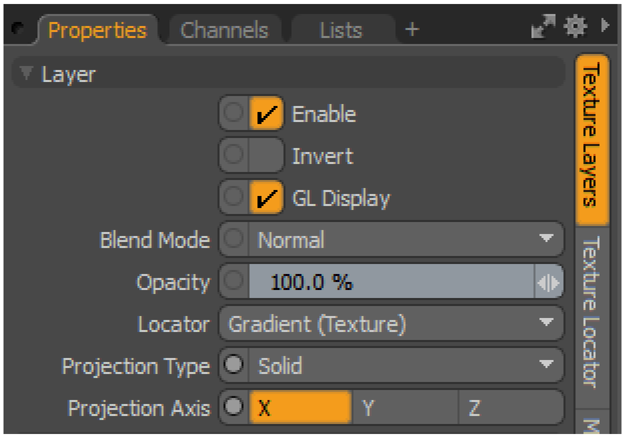


The **Dino Skin** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Dino Skin** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides a scaly/bumpy/wrinkly dinosaur skin. The various controls can be used to make any number of alien or reptile skins.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

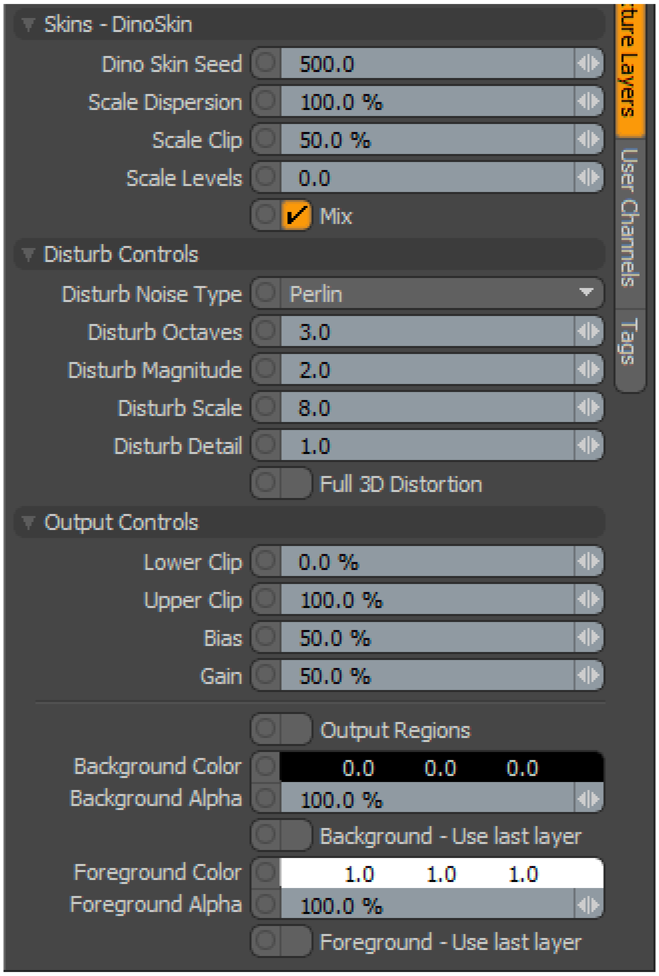
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Dino Skin Properties



Skins - DinoSkin

Dino Skin Seed 500.0

Scale Dispersion 100.0 %

Scale Clip 50.0 %

Scale Levels 0.0

☑ Mix

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 2.0

Disturb Scale 8.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

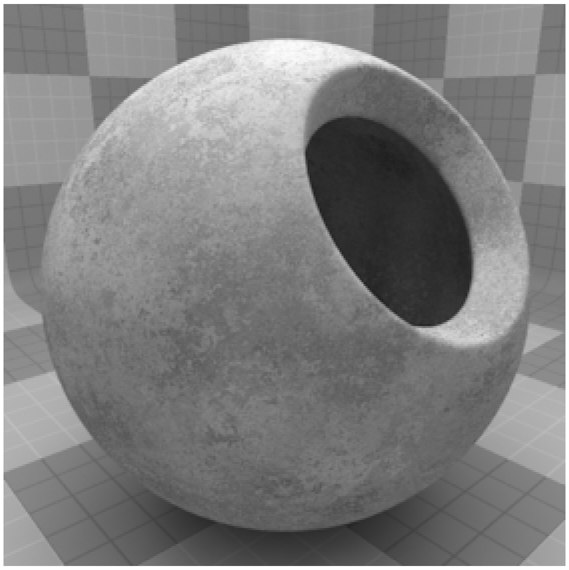
☐ Foreground - Use last layer

Skins - DinoSkin	
Dino Skin Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Scale Dispersion	Specifies how many of the bump sections are actually drawn.
Scale Clip	Determines where the texture shading transitions to the background color. The higher the value, the quicker the transition.
Scale Levels	Determines the number of layers of the texture. Additional levels add additional detail, while also lightening the overall look of the texture.

Mix	When enabled, this option mixes the texture layers. When disabled, the texture layers are stacked one upon the other.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the

	background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Dirt

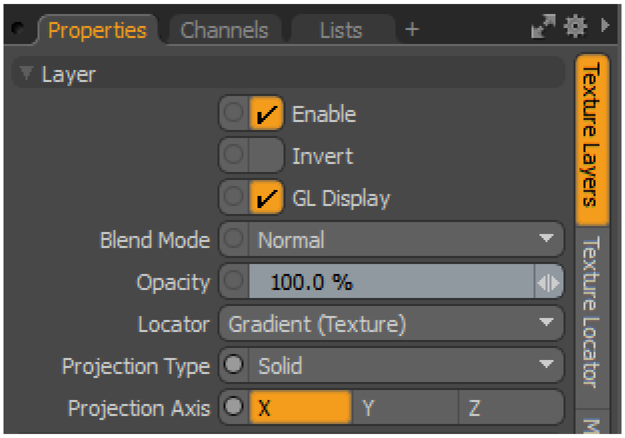


The **Dirt** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Dirt** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a way to dirty up objects so they don't look quite so pristine. The dirt function can be applied in different ways, providing a splotchy, dusty look as well as a more even, scattered dirt.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

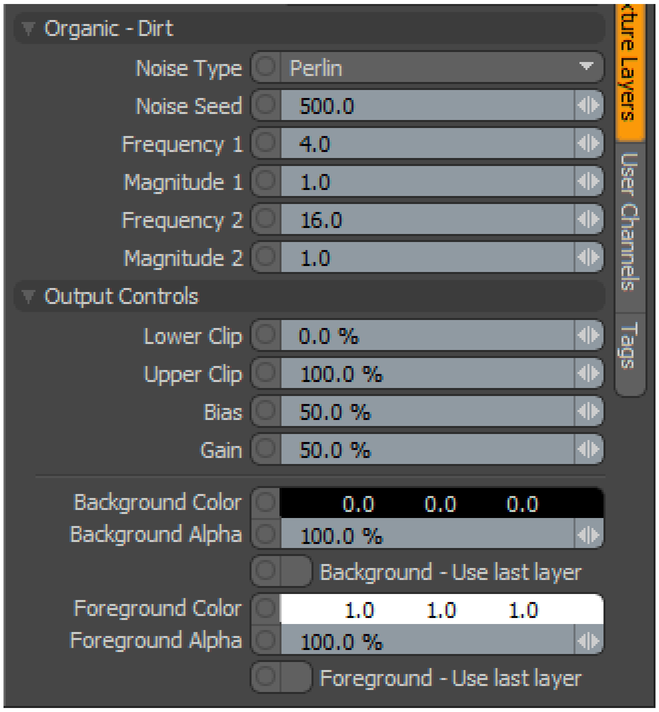
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Dirt Properties



Organic - Dirt

Noise Type: Perlin
Noise Seed: 500.0
Frequency 1: 4.0
Magnitude 1: 1.0
Frequency 2: 16.0
Magnitude 2: 1.0

Output Controls

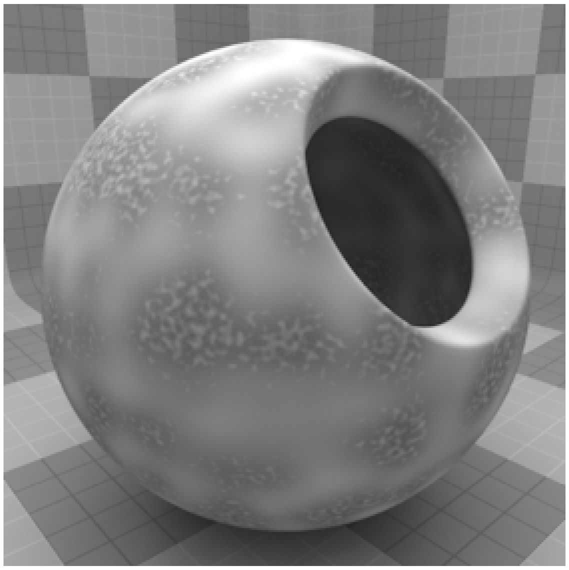
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Organic - Dirt	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.

Frequency 1	Controls the first frequency of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Magnitude 1	Controls the first magnitude of each iteration of the noise. Increasing this value tightens the delineation between the background and foreground color.
Frequency 2	Controls the second frequency of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Magnitude 2	Controls the second magnitude of each iteration of the noise. Increasing this value tightens the delineation between the background and foreground color.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use	When enabled, the **Background Color** area is completely transparent, revealing the

Last Layer	shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Disease

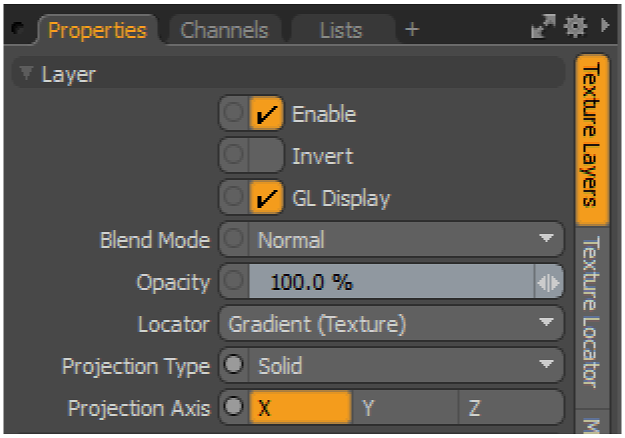


The **Disease** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Disease** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides a look similar to a skin *disease*. The controls can be used to create clumps of various sizes to make scab-like areas.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

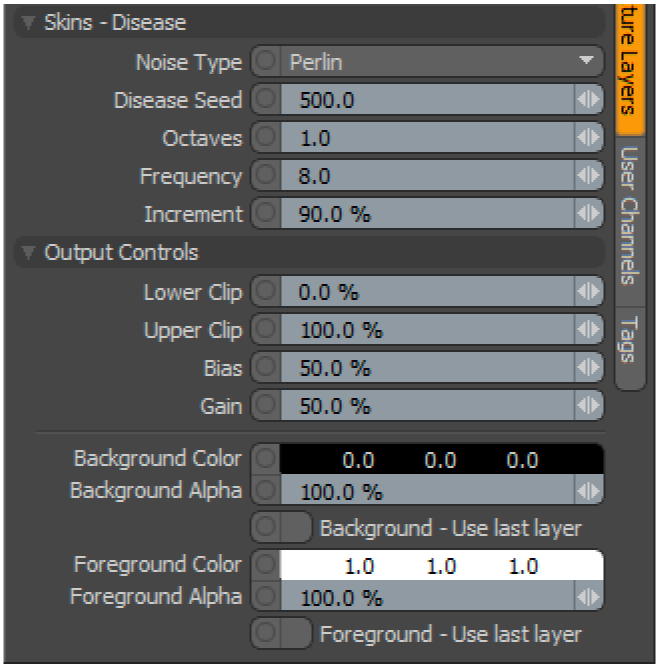
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Disease Properties

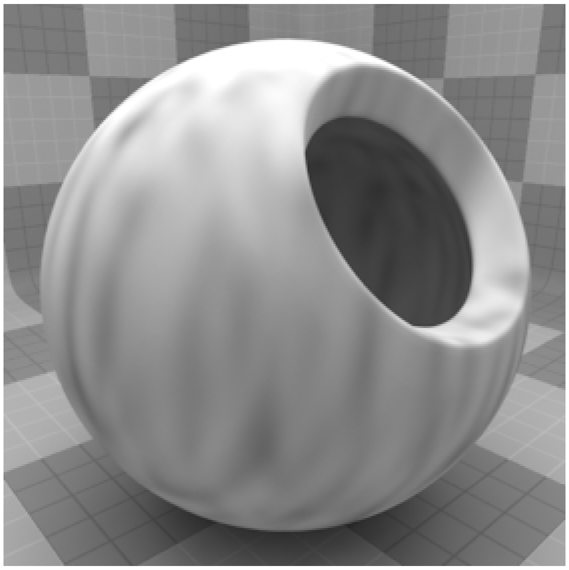

Skins - Disease
Noise Type: Perlin
Disease Seed: 500.0
Octaves: 1.0
Frequency: 8.0
Increment: 90.0 %
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Skins - Disease	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Disease Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Increment	Controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background	Specifies the **Alpha** transparency of the **Background Color**.

Alpha	
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Disturbed

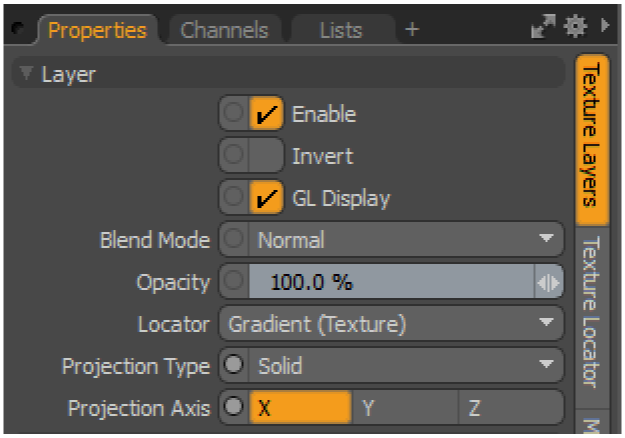


The **Disturbed** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Disturbed** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader produces a two-wave pattern, with each wave affecting and potentially overlapping the other. This texture can be used to create a water surface or criss-cross pattern.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

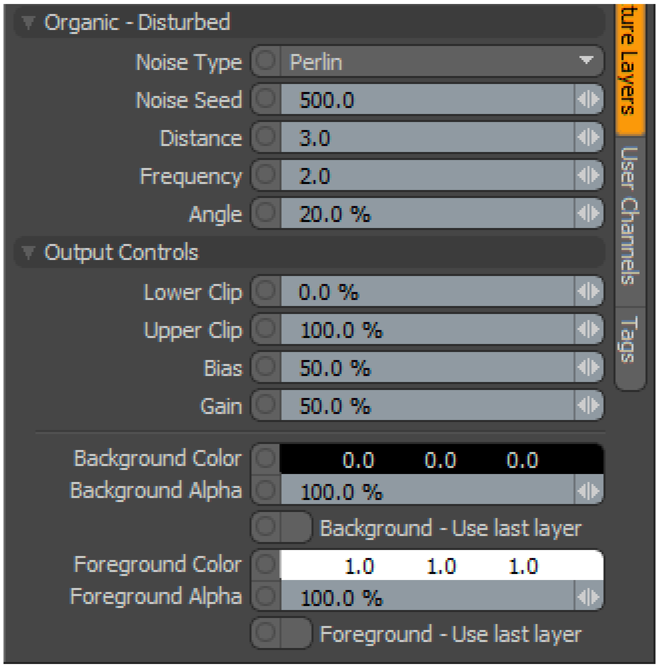
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify

	alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Disturbed Properties

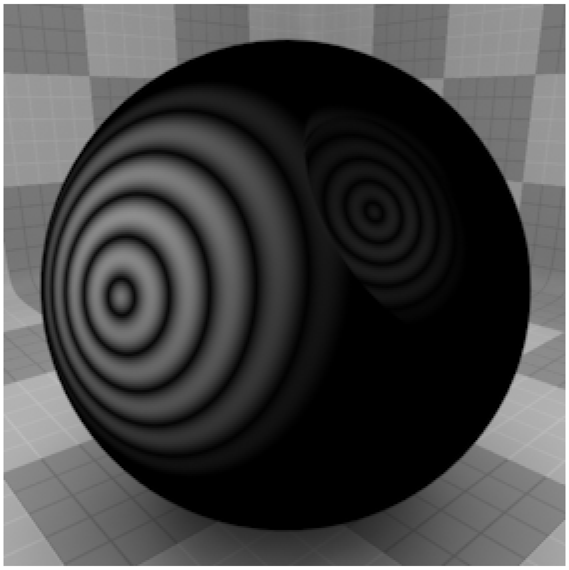

Organic - Disturbed
Noise Type: Perlin
Noise Seed: 500.0
Distance: 3.0
Frequency: 2.0
Angle: 20.0 %
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Organic - Disturbed	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse**

	• **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Distance	Controls the distance traveled by the wave sources.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Angle	Controls the angle between the two wave sources.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background	Specifies the **Alpha** transparency of the **Background Color**.

Alpha	
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

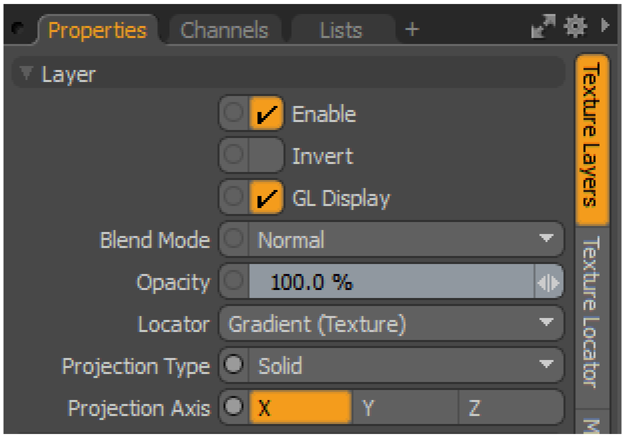
Drip Drop



The **Drip Drop** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Drip Drop** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader creates a pattern of concentric circles, similar to the ripple effect when a pebble is dropped into water. All of the parameters are keyed to the time function, allowing for easy animation.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



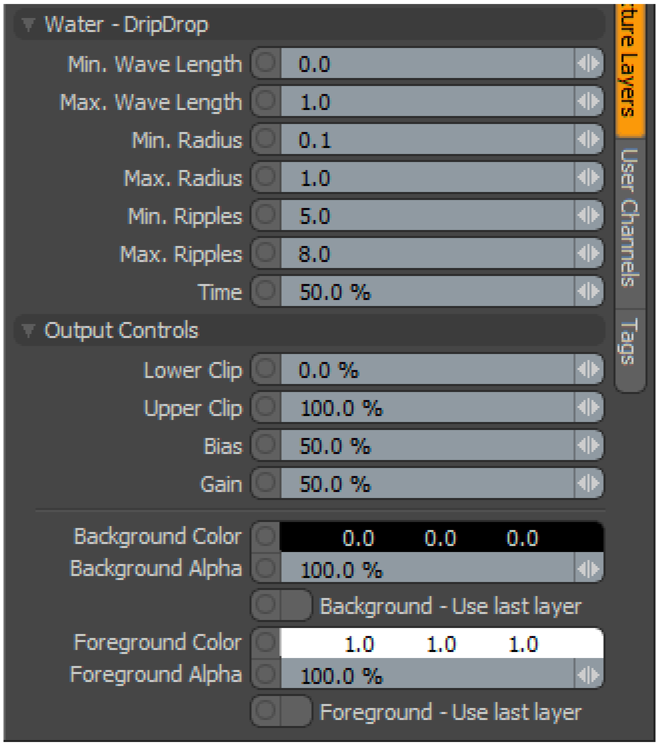
☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify

	alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Drip Drop Properties

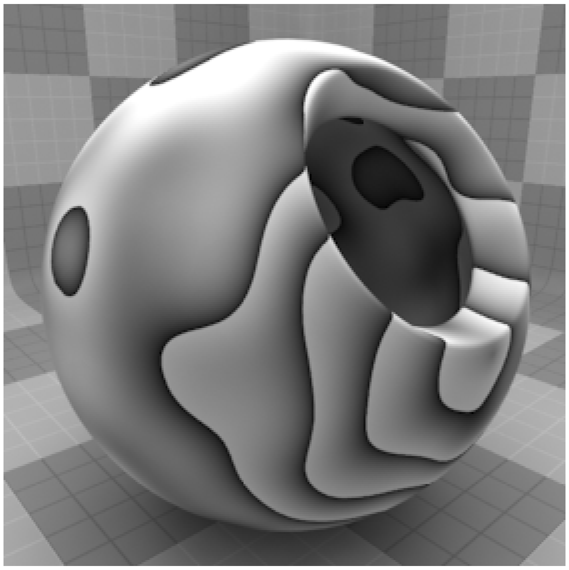

Water - DripDrop
Min. Wave Length 0.0
Max. Wave Length 1.0
Min. Radius 0.1
Max. Radius 1.0
Min. Ripples 5.0
Max. Ripples 8.0
Time 50.0 %
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Water - DripDrop	
Minimum Wave Length	Specifies the wave length of the ripples at the start of the animation (when the time parameter is set to 0%).
Maximum Wave Length	Specifies the end wave length of the ripples at the end of the animation (when the time parameter is set to 100%).
Minimum Radius	Specifies the radius of the ripples at the start of the animation (when the time parameter is set to 0%).

Maximum Radius	Specifies the radius of the ripples at the end of the animation (when the time parameter is set to 100%).
Minimum Ripples	Specifies the number or ripples at the start of the animation (when the time parameter is set to 0%).
Maximum Ripples	Specifies the number or ripples at the end o the animation (when the time parameter is set to 100%).
Time	Controls the animation progress of the other settings through time. The animation morphs from the beginning of the water drop (0%), when the concentric circles start, to the end (100%), when the circles have been exhausted.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

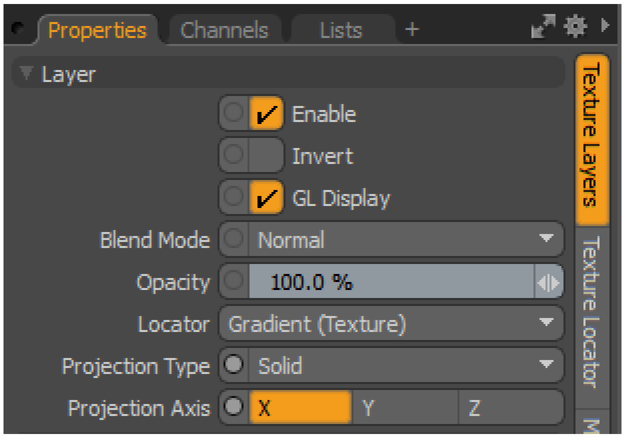
Easy Wood



The Easy Wood texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Easy Wood** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader creates a wood grain pattern with concentric rings around the specified axis. A bump can also be applied to generate some relief to the wood grain.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

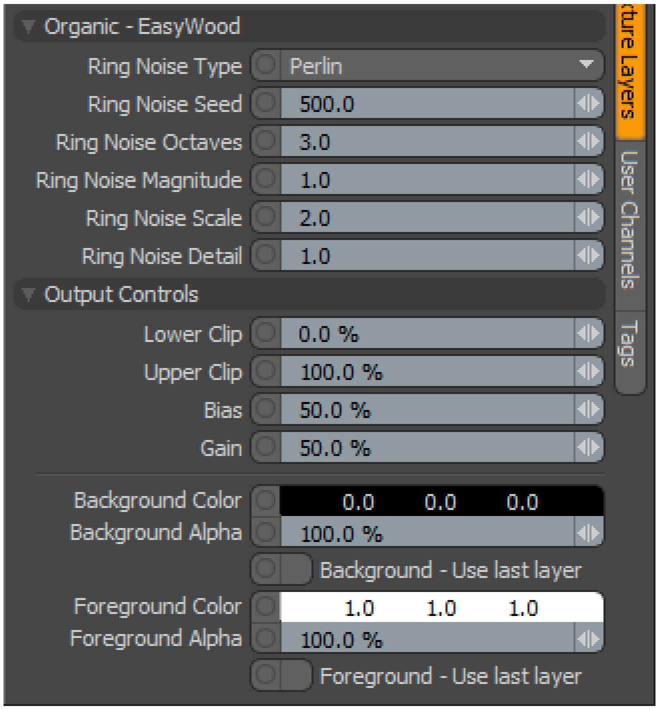
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Easy Wood Properties



Organic - EasyWood

Ring Noise Type: Perlin

Ring Noise Seed: 500.0

Ring Noise Octaves: 3.0

Ring Noise Magnitude: 1.0

Ring Noise Scale: 2.0

Ring Noise Detail: 1.0

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

☐ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

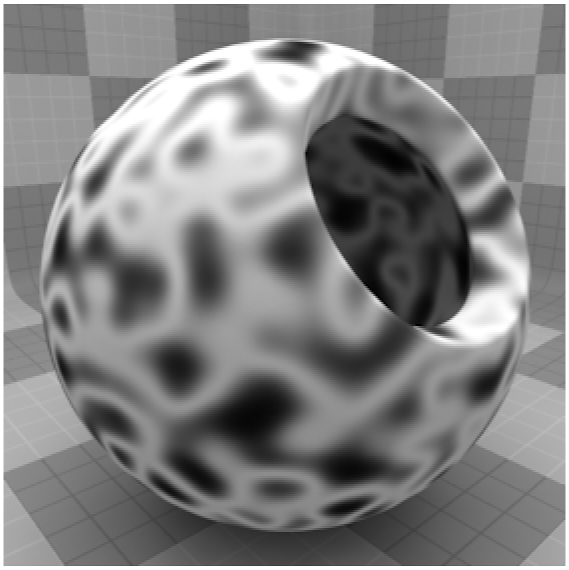
☐ Foreground - Use last layer

Organic - EasyWood	
Ring Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Ring Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Ring Noise Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Ring Noise Magnitude	Controls the size or magnitude of each iteration of the noise.
Ring Noise Scale	Controls the scale of each subsequent iteration of the noise.
Ring Noise Detail	Controls the step size between each iteration of the noise.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**

Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Electric

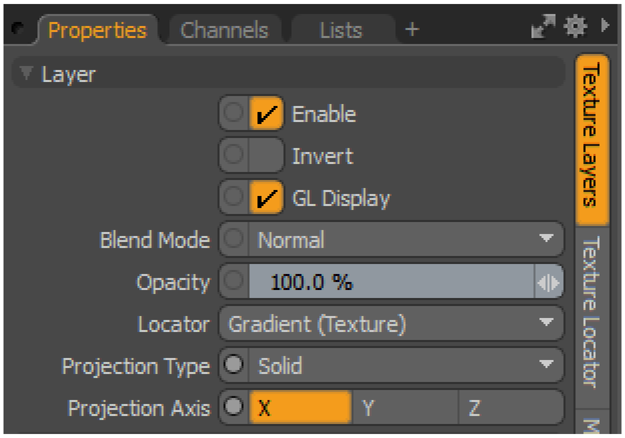


The **Electric** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Electric** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and

Alpha settings for **Background** and **Foreground**. The look of this particular shader provides a noisy pattern similar to large electrical splashes or discharges.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



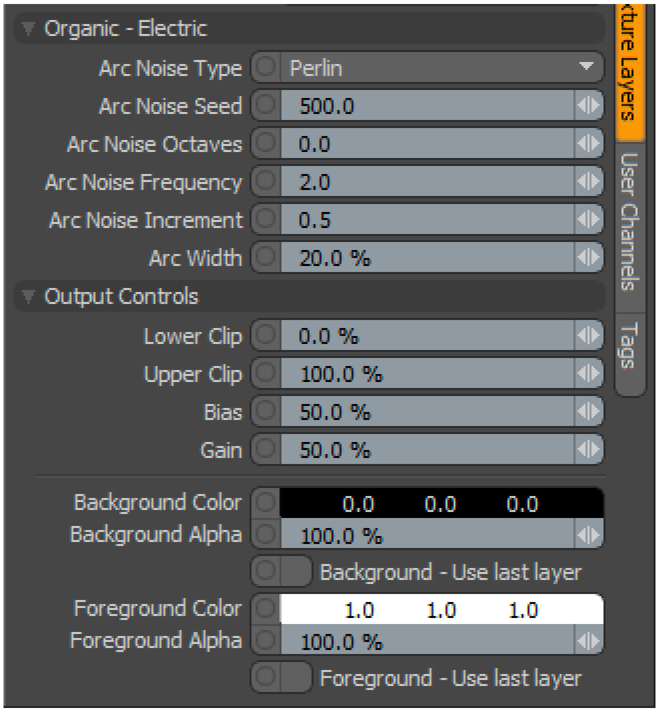
☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify

	alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Electric Properties



Organic - Electric

Arc Noise Type: Perlin

Arc Noise Seed: 500.0

Arc Noise Octaves: 0.0

Arc Noise Frequency: 2.0

Arc Noise Increment: 0.5

Arc Width: 20.0 %

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

☐ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

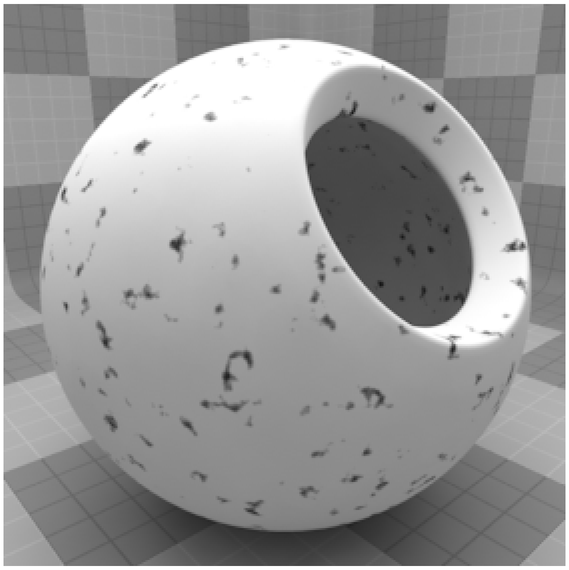
☐ Foreground - Use last layer

Organic - Electric	
Arc Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value**

	• **Impulse** • **Lattice** • **Bubble**
Arc Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Arc Noise Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Arc Noise Frequency	Controls the size or magnitude of each iteration of the noise.
Arc Noise Increment	Controls the scale of each subsequent iteration of the noise.
Arc Width	Controls the width of the electrical arc.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs

	random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Etched

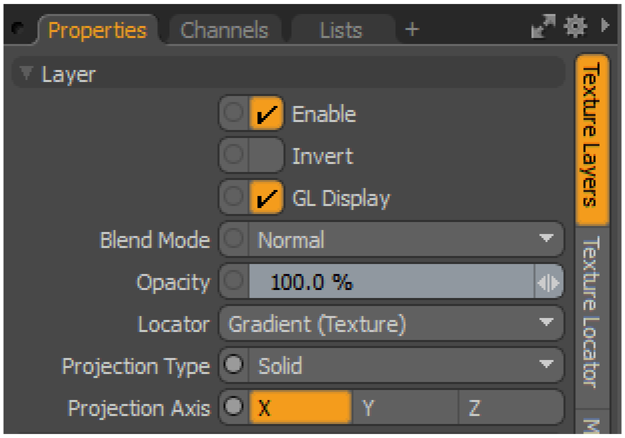


The **Etched** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Etched** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides an etched appearance, as though the surface has been treated with acid.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

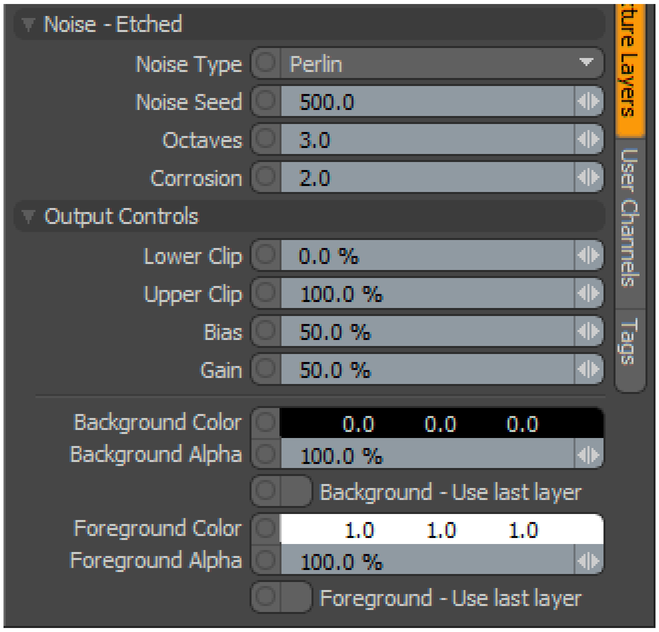
Projection Type: Solid

Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Etched Properties



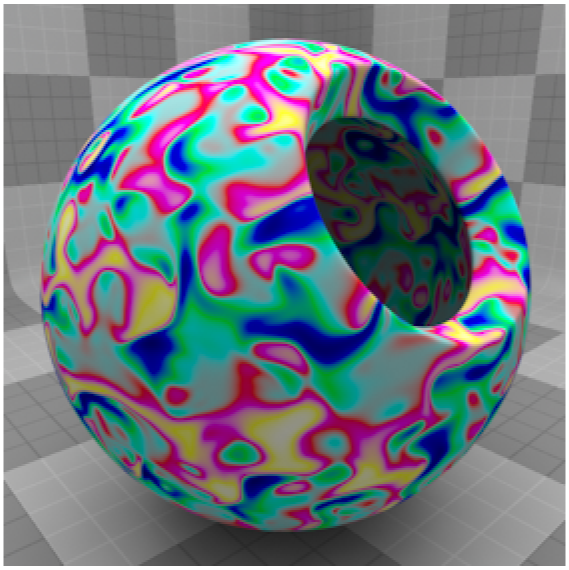
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Corrosion: 2.0
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Etched	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient**

	• **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Corrosion	Specifies the amount of corrosion used to create the texture. As this number is increased, more corrosion is introduced into the texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the

	amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

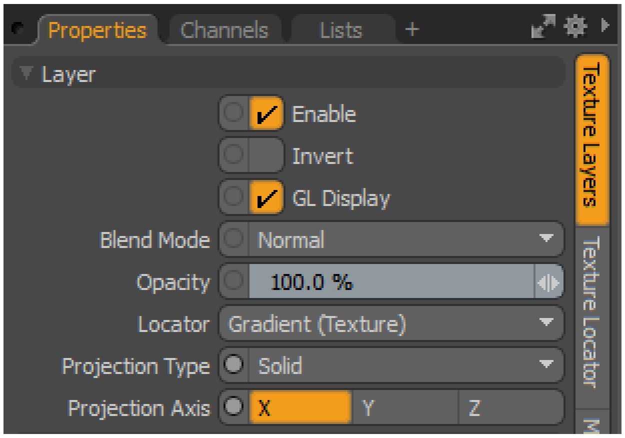
Easy Gradient



The **Easy Gradient** process layer applies a user-defined color gradation across the gray values of any lower Shader Tree layer set to the same **Effect** type, this makes it easy to colorize procedural textures that are generally defined by two colors (foreground/background). In the example above, an **Agate** texture layer is applied, set as **Diffuse Color** effect, with an **Easy Gradient** layer at the default values, placed directly above it in the tree also set to **Diffuse Color**. The color gradations are applied as zones represented by the **Blend Values**. You can define up to 8 colors with corresponding blend zones; representing the entire gray spectrum from 0% black to 100% white.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

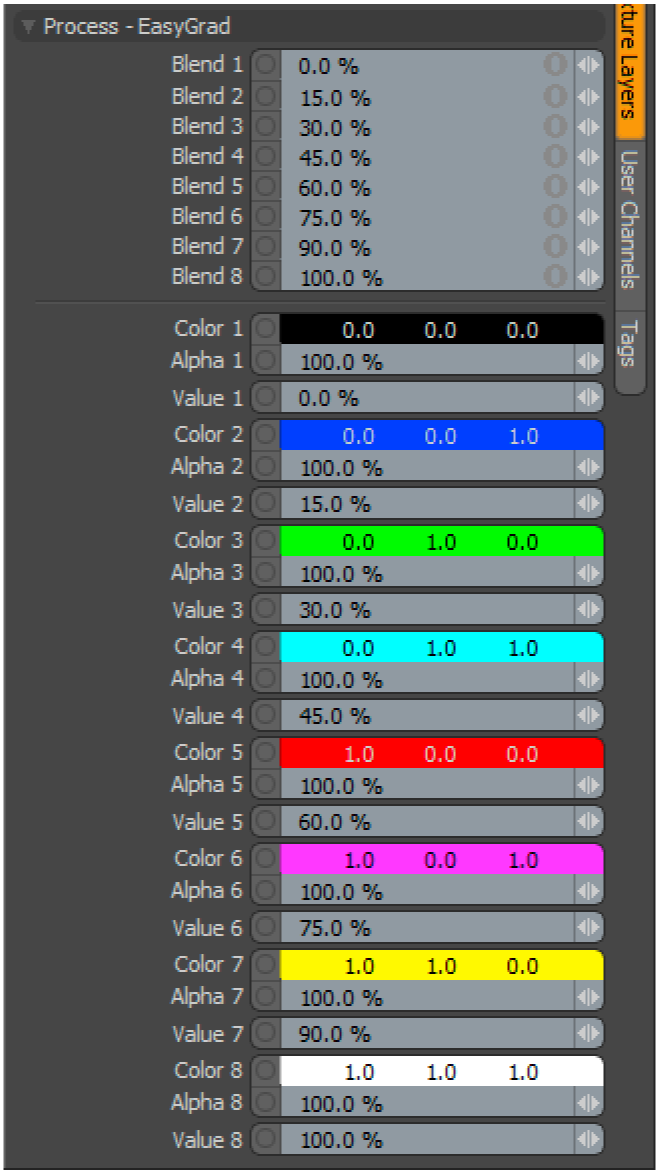
>


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Easy Gradient Properties



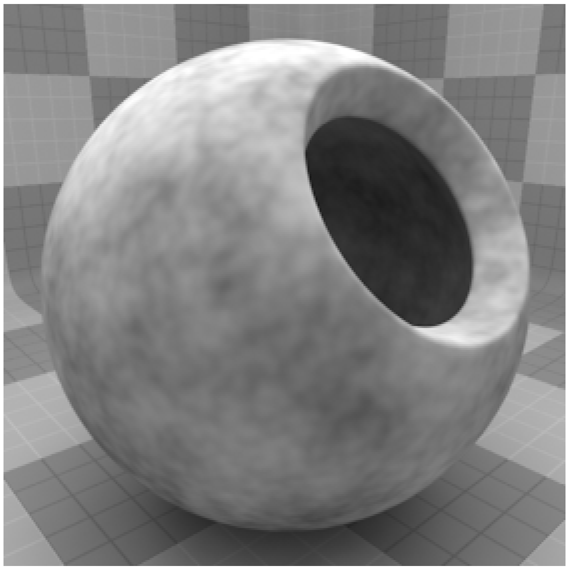
Process - EasyGrad

Blend 1 0.0 %
Blend 2 15.0 %
Blend 3 30.0 %
Blend 4 45.0 %
Blend 5 60.0 %
Blend 6 75.0 %
Blend 7 90.0 %
Blend 8 100.0 %

Color 1 0.0 0.0 0.0
Alpha 1 100.0 %
Value 1 0.0 %
Color 2 0.0 0.0 1.0
Alpha 2 100.0 %
Value 2 15.0 %
Color 3 0.0 1.0 0.0
Alpha 3 100.0 %
Value 3 30.0 %
Color 4 0.0 1.0 1.0
Alpha 4 100.0 %
Value 4 45.0 %
Color 5 1.0 0.0 0.0
Alpha 5 100.0 %
Value 5 60.0 %
Color 6 1.0 0.0 1.0
Alpha 6 100.0 %
Value 6 75.0 %
Color 7 1.0 1.0 0.0
Alpha 7 100.0 %
Value 7 90.0 %
Color 8 1.0 1.0 1.0
Alpha 8 100.0 %
Value 8 100.0 %

Process - EasyGrad	
Blend 1-8	These parameters define the blend points of the gradient.
Color 1-8	These parameters define the colors of the gradient.
Alpha 1-8	These parameters define the alpha channel of the gradient.
Value 1-8	These parameters define the opacity of the gradient.

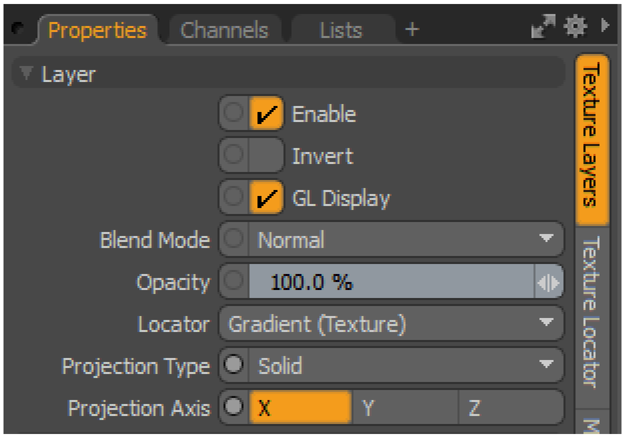
FBM



The **FBM** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **FBM** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides noisy bumps based on Fractal Brownian Motion (FBM). This textures is great for noisy, organic surfaces such as rocks or rough stippled surfaces.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

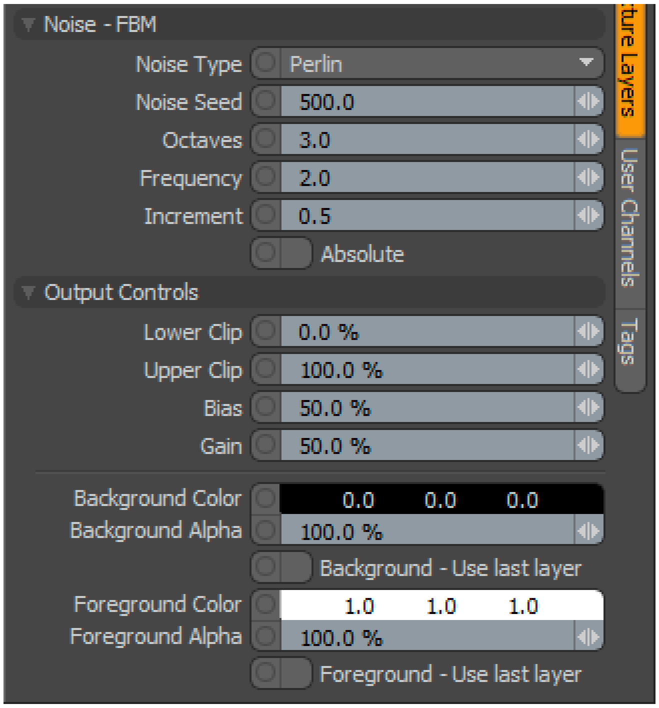
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

FBM Properties



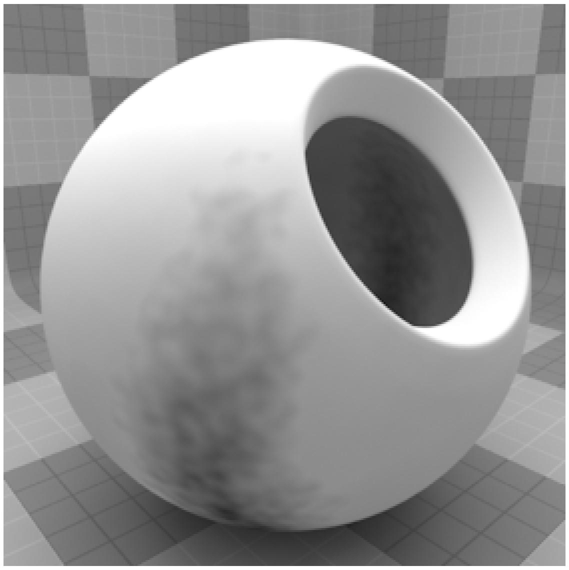
Noise - FBM
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Frequency: 2.0
Increment: 0.5
☐ Absolute
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Noise - FBM	
Noise Type	Specifies the look of the distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Increment	Controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Absolute	Determines whether or not the **Absolute** value for the noise is used, providing a stronger, more defined variation on the default texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the

	amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Fire

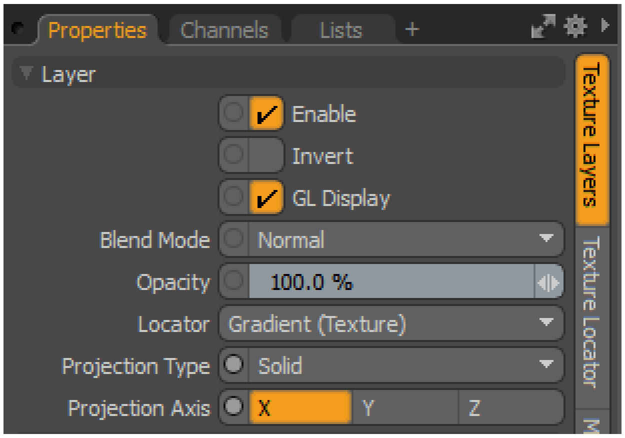


The **Fire** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Fire** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader simulates a bonfire or campfire type of effect.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

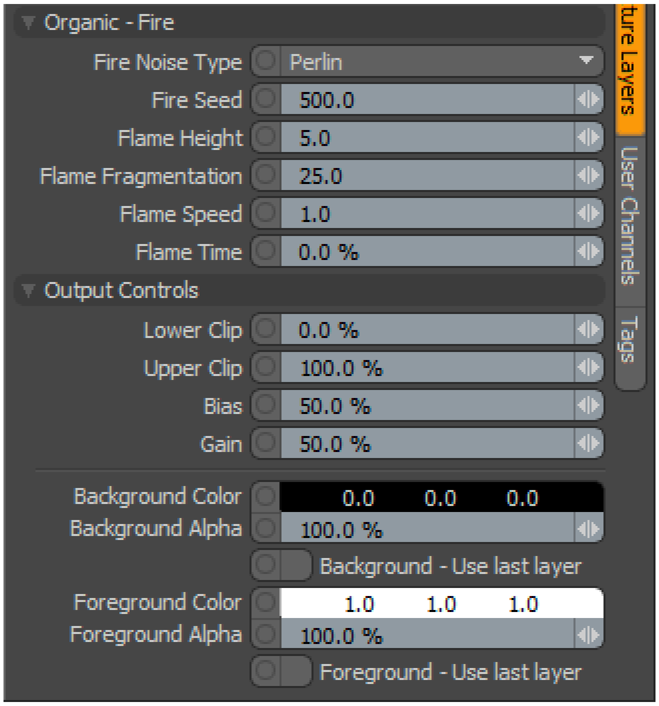


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Fire Properties



Organic - Fire
Fire Noise Type: Perlin
Fire Seed: 500.0
Flame Height: 5.0
Flame Fragmentation: 25.0
Flame Speed: 1.0
Flame Time: 0.0 %

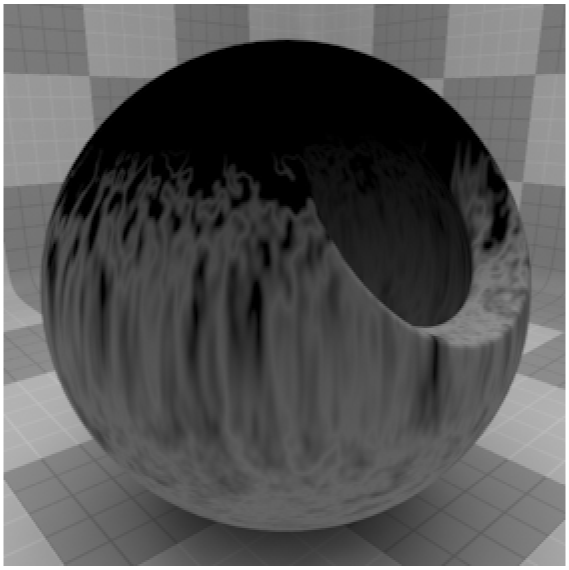
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Organic - Fire	
Fire Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient**

	• **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Fire Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Flame Height	Controls the height of the flame texture.
Flame Fragmentation	Controls the fragmentation, or detail, of the flames.
Flame Speed	Controls the speed of the flame when it is animated.
Flame Time	Controls the overall timing of the animation, when used in conjunction with the other available flame parameters.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to

	add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Fire Wall

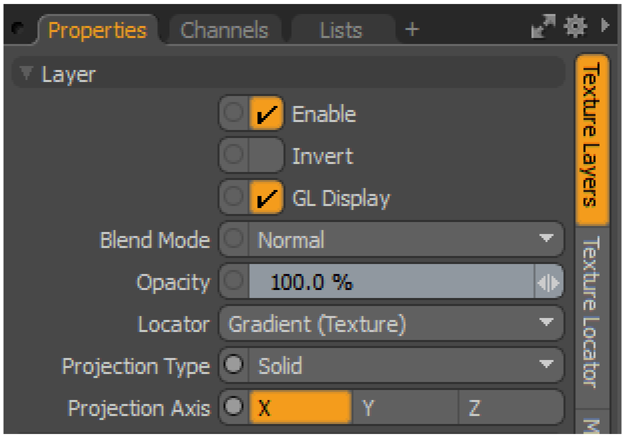


The **Fire Wall** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Fire Wall** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader simulates a bonfire or campfireFor information regarding adding and working with Shader Tree item layers see Shader Tree. type of effect.

Note: For information regarding adding and working with Shader Tree Items Layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

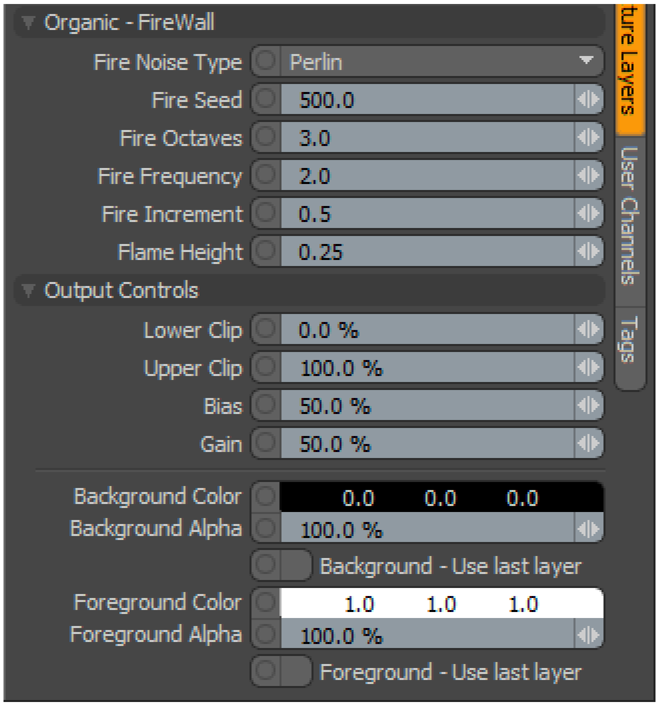
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Firewall Properties



Organic - FireWall
Fire Noise Type: Perlin
Fire Seed: 500.0
Fire Octaves: 3.0
Fire Frequency: 2.0
Fire Increment: 0.5
Flame Height: 0.25

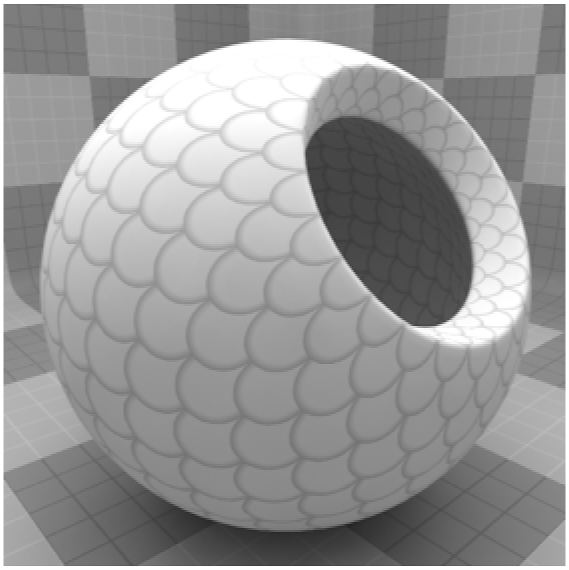
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Organic - FireWall	
Fire Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin**

	• **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Fire Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Fire Octaves	This value defines the number of iterations used to create the texture. As this number is increased, the texture takes on more detail.
Fire Frequency	This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Fire Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Flame Height	This value controls the height of the flame texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas

	setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Fish Scales



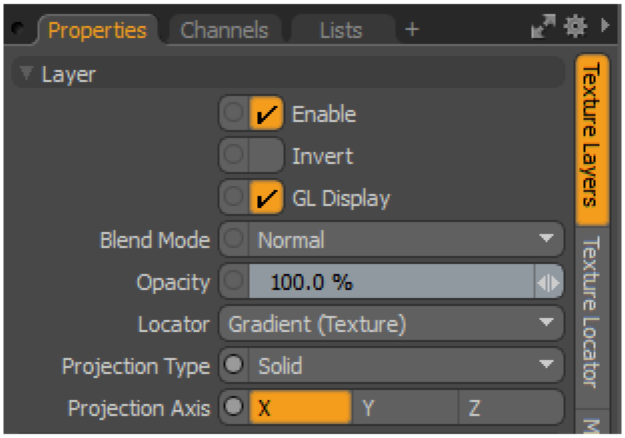
The **Fish Scales** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Fish Scales** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a pattern similar to the scales on a fish. The radius, edges and height of the scales are all variable.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Note: Due to the nature of this procedural, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the texture locator, press the small **+** icon preceding the layer name in the Shader Tree and click on the layer once revealed. In the Properties Panel, change the **Projection Type** to a **UV map** (highly recommended) or choose one of the geometric projections, such as **Planar**, **Cylindrical**, **Spherical**, and so on, and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

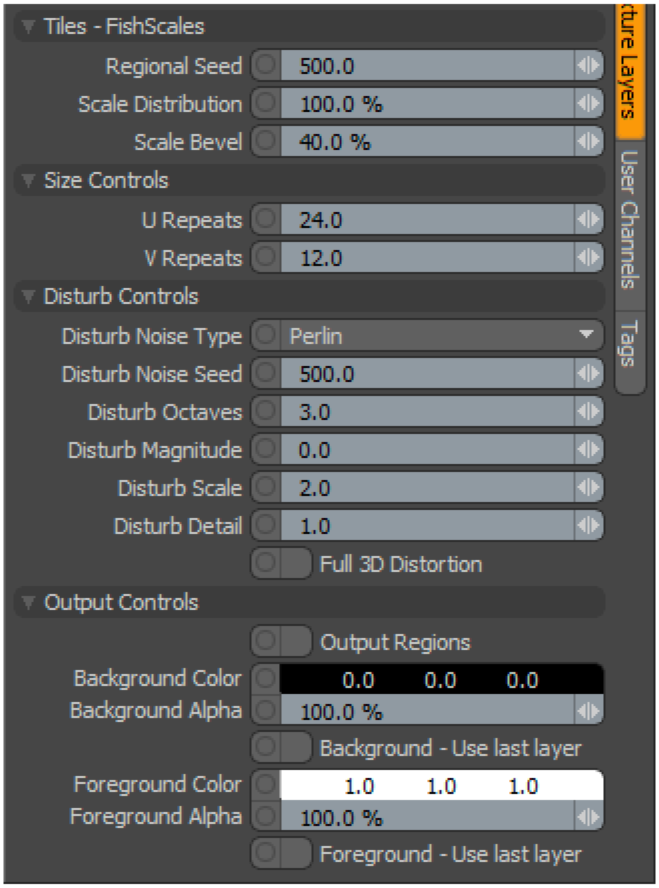
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Fish Scales Properties



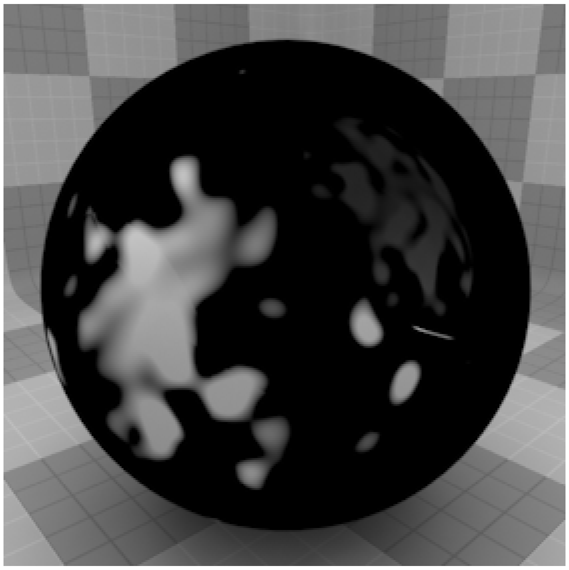
Tiles - FishScales
Regional Seed 500.0
Scale Distribution 100.0 %
Scale Bevel 40.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion
Output Controls
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Tiles - FishScales	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Scale Distribution	Specifies the percentage of the surface that is covered by scales.
Scale Bevel	Specifies the amount of bevel between the scales.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the U direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the V direction (vertically).

Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.

Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Flare

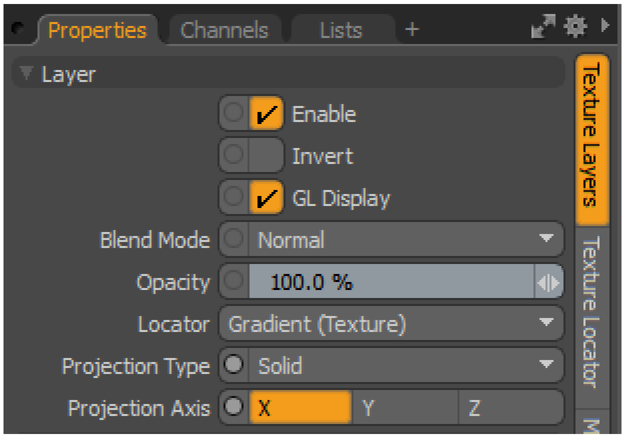


The **Flare** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Flare** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader provides a lens flare-like star pattern. It can be used for highlights or glints, as well as the more traditional lens flare effects.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

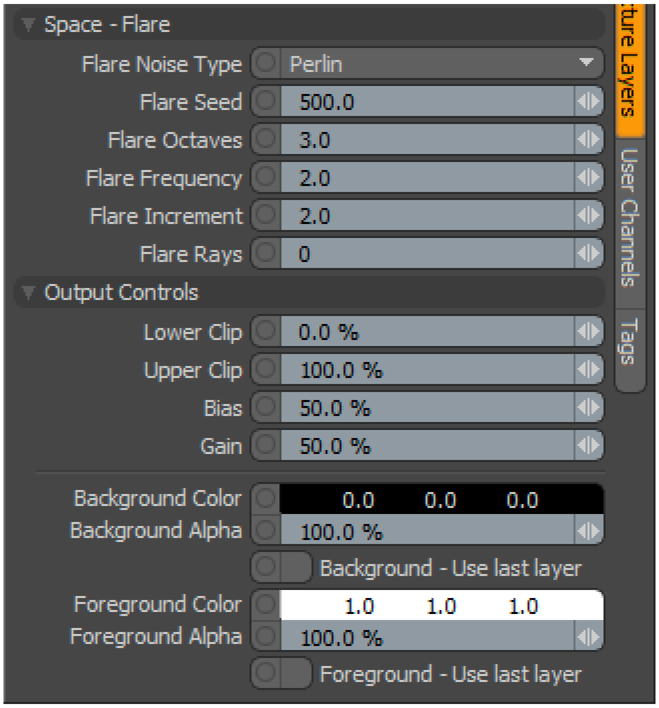
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Flare Properties



Space - Flare

Flare Noise Type: Perlin

Flare Seed: 500.0

Flare Octaves: 3.0

Flare Frequency: 2.0

Flare Increment: 2.0

Flare Rays: 0

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

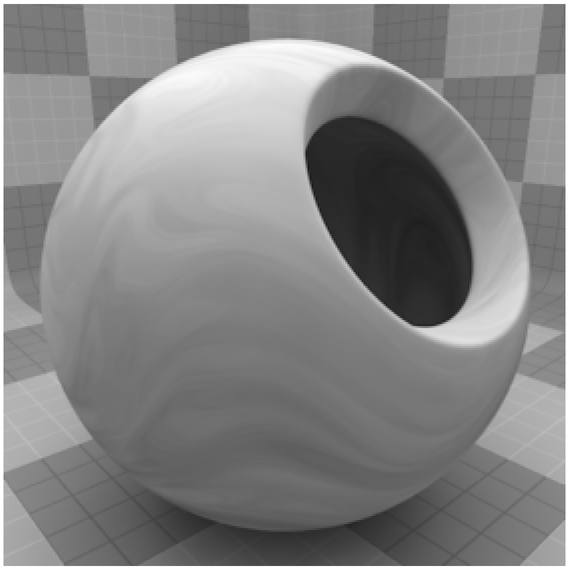
Foreground - Use last layer

Space - Flare	
Flare Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Flare Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.

Flare Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Flare Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Flare Increment	This value controls the scaling of each successive iteration. When the control is at a low value, the turbulence is tight and crinkled, becoming increasingly smooth as the parameter is increased.
Flare Rays	This value controls the number of rays used for the flare pattern.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Flow Bozo

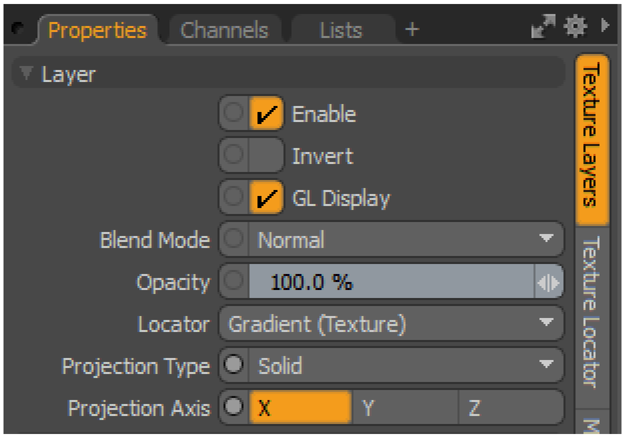


The **Flow Bozo** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Flow Bozo** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader creates a pattern similar to flowing gas or liquid.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

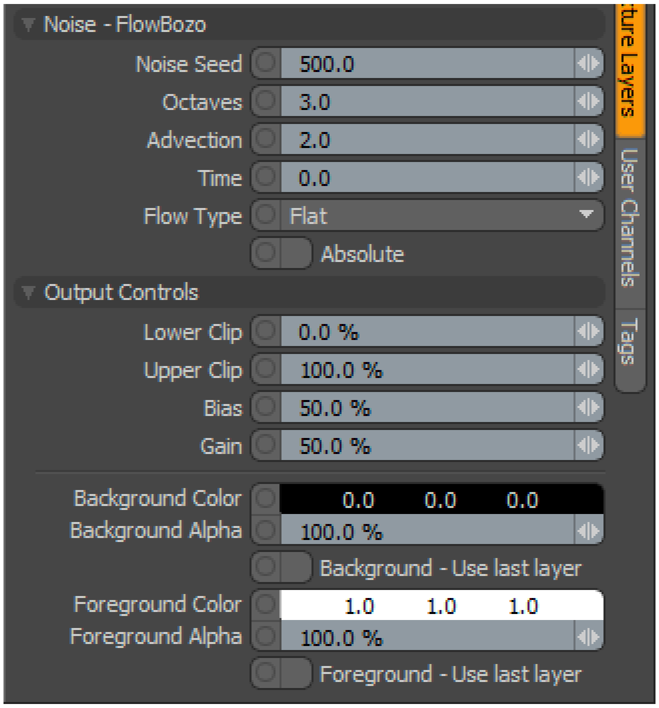
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Flow Bozo Properties



Noise - FlowBozo

Noise Seed 500.0

Octaves 3.0

Advection 2.0

Time 0.0

Flow Type Flat

☐ Absolute

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

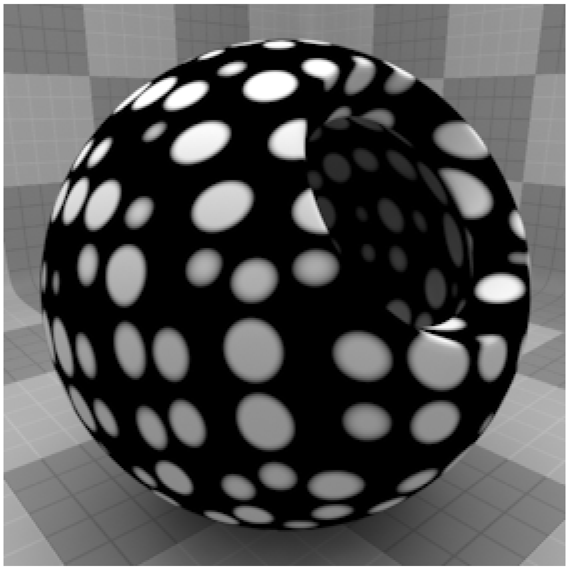
☐ Foreground - Use last layer

Noise - FlowBozo	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Advection	This value controls the noise of the flow effect, allowing you to stir up the texture.
Time	This value allows you to adjust the noise at selected key frames to control the animation of the flow.
Flow Type	This option controls the type of flow when using the flow time to create an animation. Choose from Flat, Square or Inverse.
Absolute	This option determines whether or not the **Absolute** value for the noise is used, providing a stronger, more defined variation on the default texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the

	amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Frog Skin

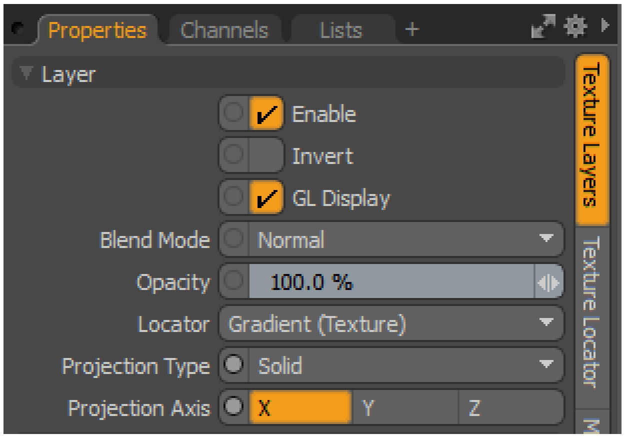


The **Frog Skin** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Frog Skin** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. The look of this particular shader provides a speckled, bumpy, spotted pattern, similar to the skin of a frogFor information regarding adding and working with Shader Tree item layers see Shader Tree. or toad.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

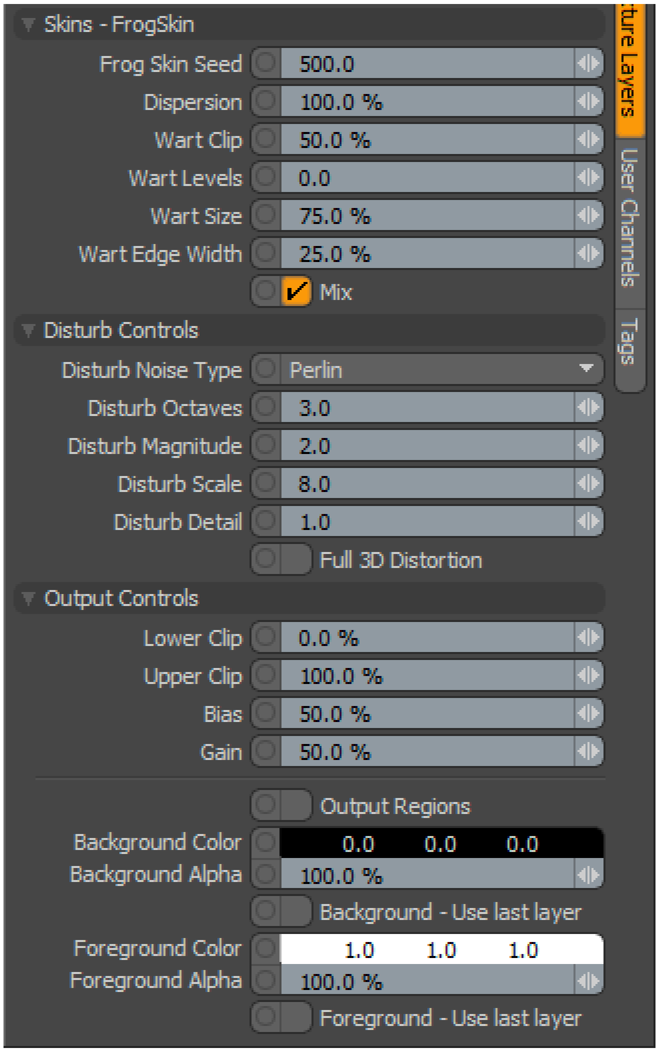
Projection Type: Solid

Projection Axis: [✓ X] Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Frog Skin Properties



Skins - FrogSkin

Frog Skin Seed 500.0
Dispersion 100.0 %
Wart Clip 50.0 %
Wart Levels 0.0
Wart Size 75.0 %
Wart Edge Width 25.0 %
☑ Mix

Disturb Controls

Disturb Noise Type Perlin
Disturb Octaves 3.0
Disturb Magnitude 2.0
Disturb Scale 8.0
Disturb Detail 1.0
☐ Full 3D Distortion

Output Controls

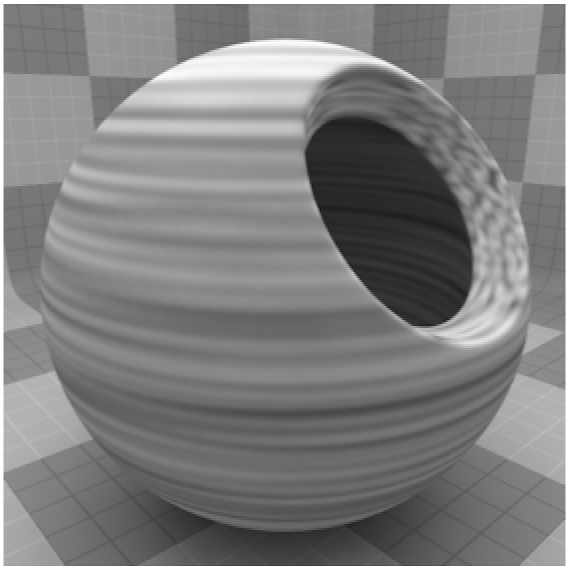
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Skins - FrogSkin	
Frog Skin Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Dispersion	Specifies how many of the wart sections are actually drawn.
Wart Clip	Determines where the texture shading transitions to the background color. The higher the value, the quicker the transition.
Wart Levels	Determines the number of layers of the texture. Additional levels add additional detail, in turn also lightening the overall look of the texture.
Wart Size	Determines the overall size of the warts.
Mix	When enabled, this option mixes the texture layers. When disabled, the texture layers are stacked one upon the other.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.

Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.

Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Gas Giant

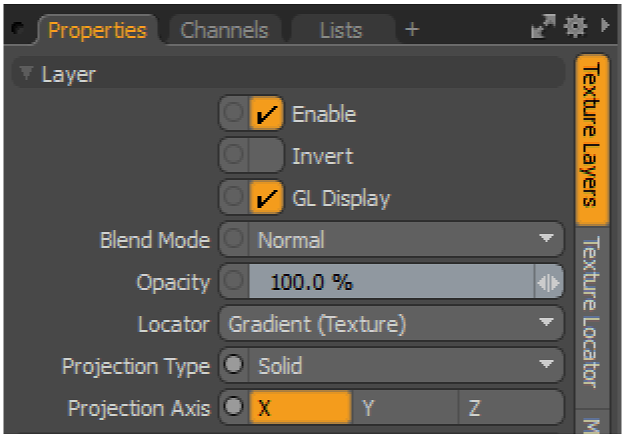


The **Gas Giant** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Gas Giant** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader applies rings of color to an object, along with a noise function to disturb the banding. By selecting various base colors, you can create planets similar to the gas giants, Saturn, Jupiter and Uranus.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

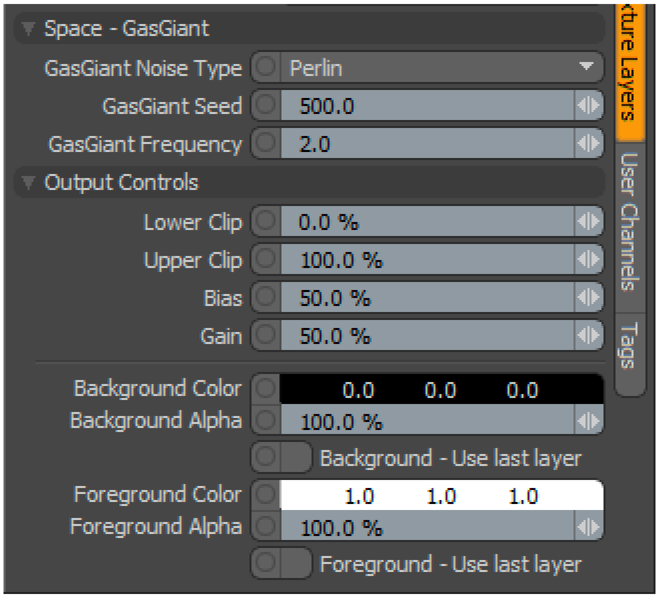
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Gas Giant Properties



Space - GasGiant

GasGiant Noise Type: Perlin

GasGiant Seed: 500.0

GasGiant Frequency: 2.0

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

Background - Use last layer

Foreground Color: 1.0 1.0 1.0

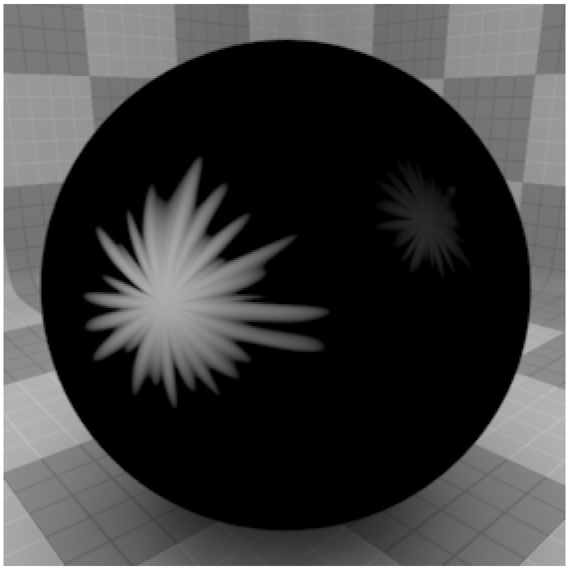
Foreground Alpha: 100.0 %

Foreground - Use last layer

Space - GasGiant	
GasGiant Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
GasGiant Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
GasGiant Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Glint

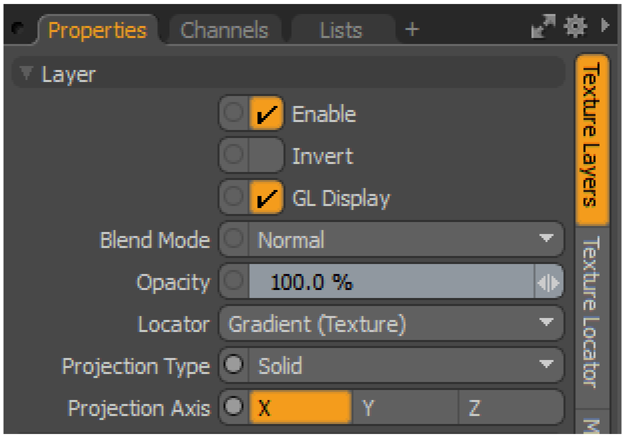


The **Glint** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Glint** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This shader creates a glint pattern, similar to a specular highlight, as seen on highly polished metals. It can also be used to create stars and other space related patterns.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

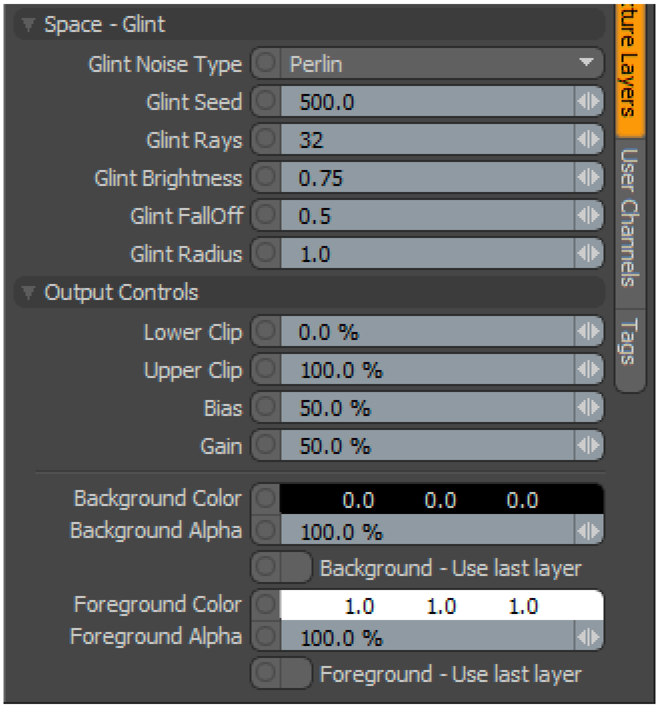
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Glint Properties



Space - Glint

Glint Noise Type: Perlin
Glint Seed: 500.0
Glint Rays: 32
Glint Brightness: 0.75
Glint FallOff: 0.5
Glint Radius: 1.0

Output Controls

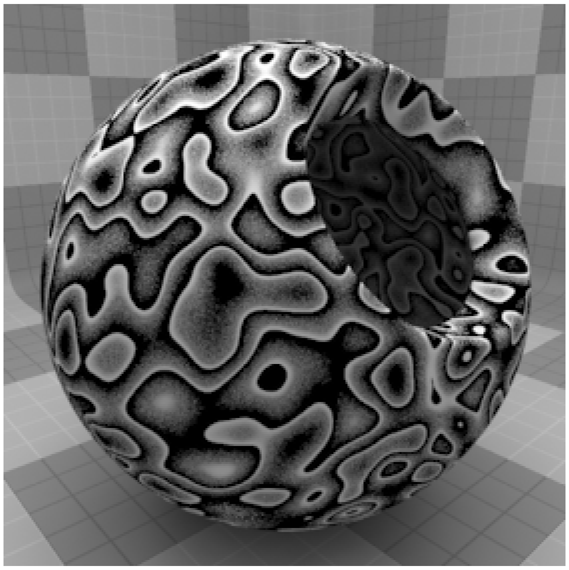
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Space - Glint	
Glint Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Glint Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.

Glint Rays	This value controls the number of rays that are used to create the glint pattern.
Glint Brightness	This value controls the overall brightness of the glint flare.
Glint Falloff	This value controls the intensity of the falloff as the rays emanate from the glint fare. Increasing this value creates thinner, longer rays.
Glint Radius	This value controls the radius of the center of the glint flare. When the control is at a low value, the center of the flare is denser, becoming less dense as the parameter is increased.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.

Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Grainy Wood

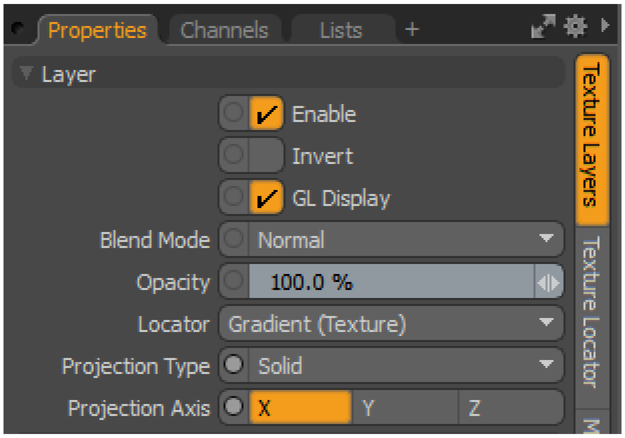


The **Grainy Wood** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Grainy Wood** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader creates a very grainy wood pattern. The depth of the grain can be varied along with the wood ring spacing.

Note: For information regarding adding and working with Shader Tree Items Layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

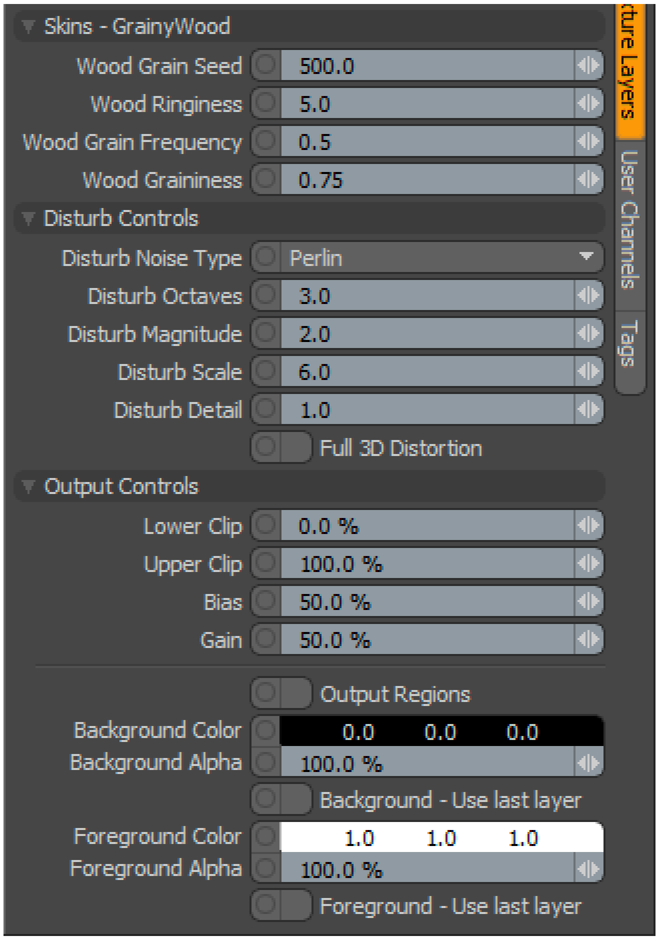
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Grainy Wood Properties



Skins - GrainyWood

Wood Grain Seed 500.0
Wood Ringiness 5.0
Wood Grain Frequency 0.5
Wood Graininess 0.75

Disturb Controls

Disturb Noise Type Perlin
Disturb Octaves 3.0
Disturb Magnitude 2.0
Disturb Scale 6.0
Disturb Detail 1.0
Full 3D Distortion

Output Controls

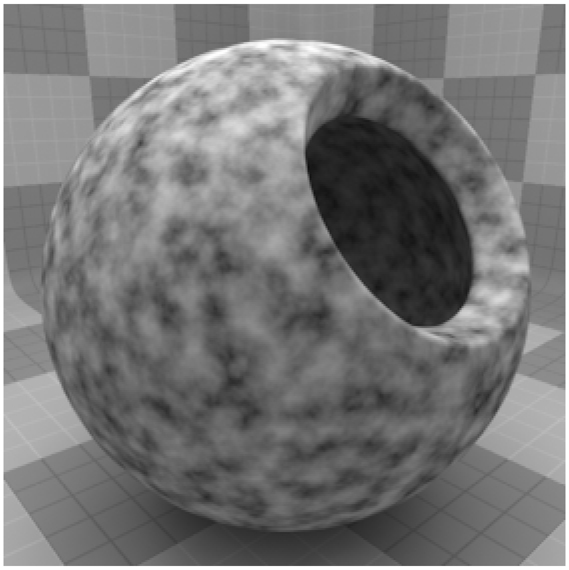
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Skins - GrainyWood	
Wood Grain Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Wood Ringiness	This value controls the amplitude of the wood rings.

Wood Grain Frequency	This value controls the overall frequency or scale of the grain. Increasing this value creates more turbulence, stirring up the grain in the pattern.
Wood Graininess	This value controls the graininess of the wood rings.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Similar to the **Patch Levels** setting, the **Disturb Octaves** value specifies the number of layers of noise when distorting the texture, in effect producing greater detail.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.

Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

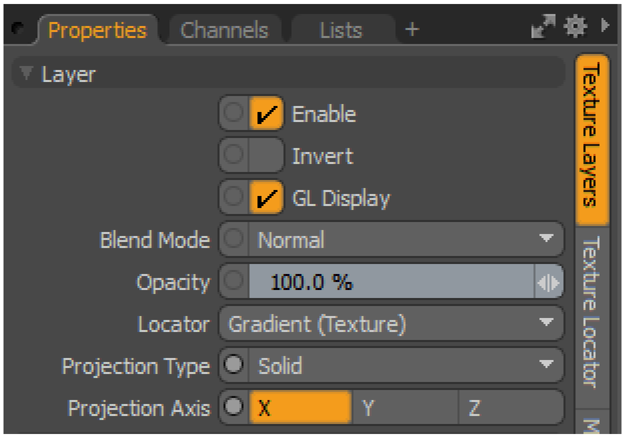
Granite



The **Granite** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Granite** texture can be addressed by its two zones, the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a pattern similar to granite. This texture can also be used for making star fields by defining very narrow color bands.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

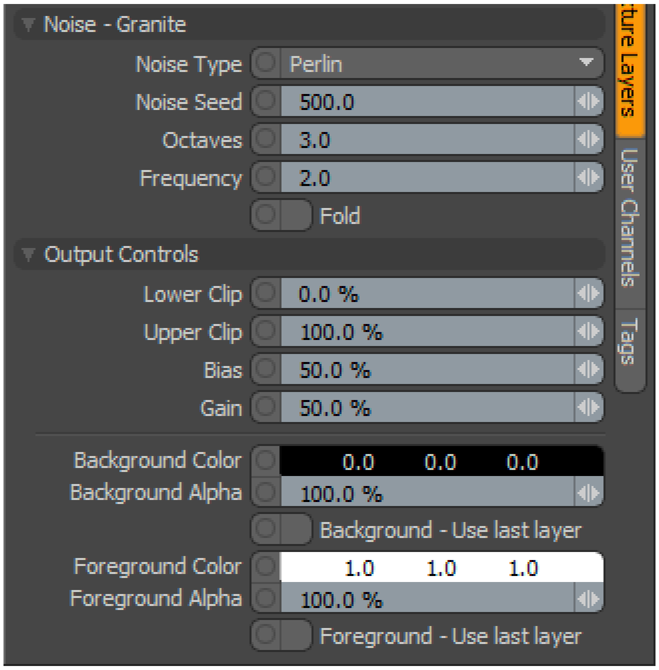
Layer Properties



Enable: ☑
Invert: ☐
GL Display: ☑
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Granite Properties



Noise - Granite

Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Frequency: 2.0
☐ Fold

Output Controls

Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %

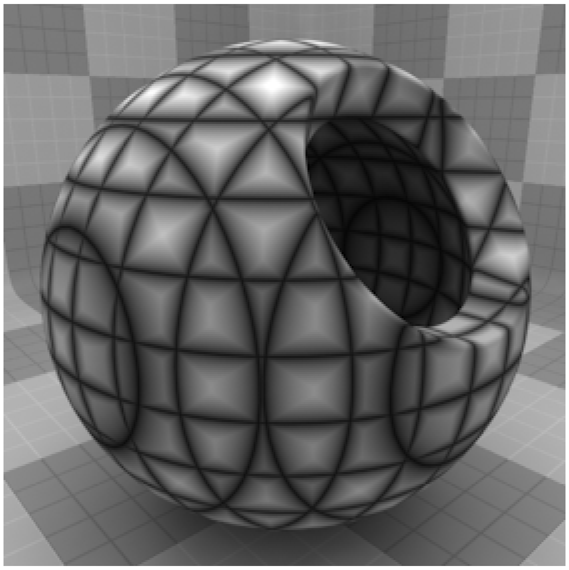
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Noise - Granite	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Octaves	This value defines the number of iterations used to create the texture. As this number

	is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Frequency	This value controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Fold	This option creates a folded variant, resulting in a denser noise pattern.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward

Color/Value	the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Grid

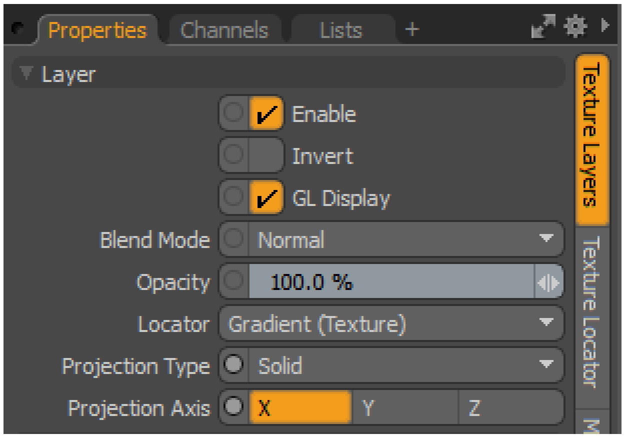


The **Grid** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution, they can be magnified nearly infinitely with no visual loss in detail. The **Grid** texture can be addressed by its two zones, the the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For instance, if the texture is applied as a **Displacement**, the **Value** settings would be used, whereas setting the texture effect to **Diffuse Color** would use the **Color** and **Alpha** settings for **Background** and **Foreground**. This particular shader provides a series of adjustable grid patterns.

Note: For information regarding adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

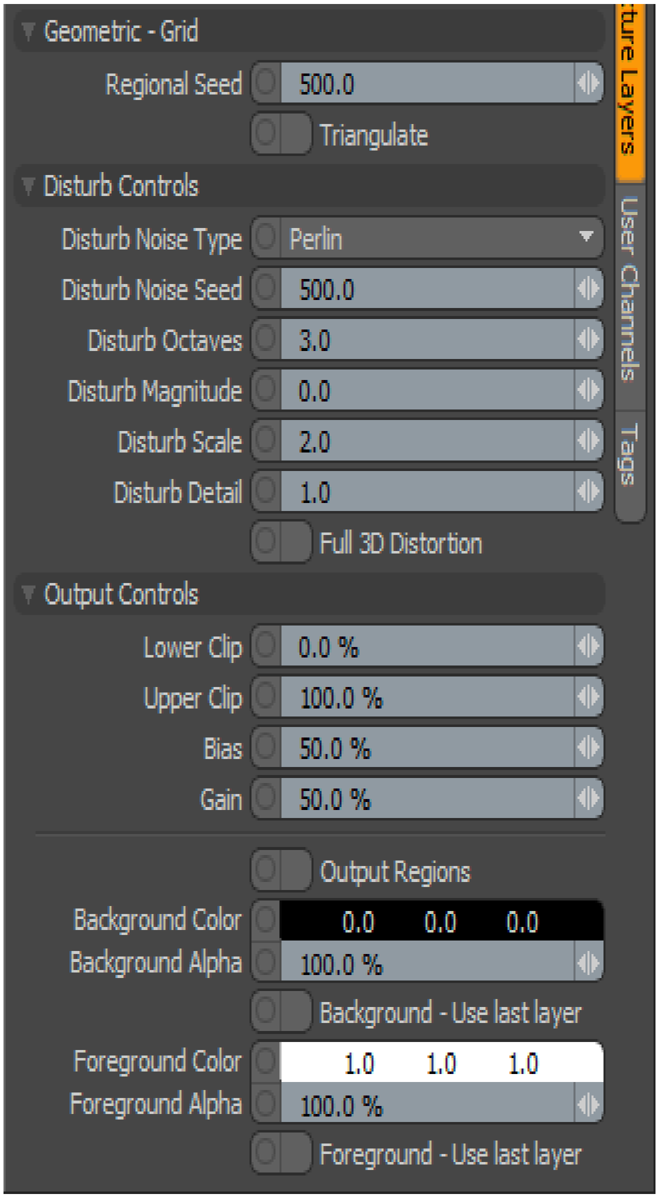
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Grid Properties



Regional Seed 500.0

○ Triangulate

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

○ Full 3D Distortion

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

○ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

○ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

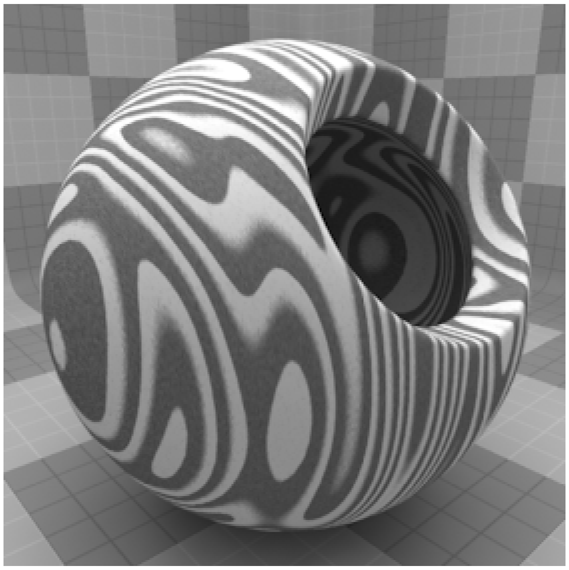
○ Foreground - Use last layer

Geometric - Grid	
Regional Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Triangulate	When enabled, this option causes the pattern to have a triangle-based edging

	pattern, as opposed to the default sawtooth blend.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the Disturb Magnitude.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	The **Seed** value is the initial number used when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result, however, you need to use the same **Seed** value when it is necessary for items to retain the same variations.
Disturb Octaves	This value defines the number of iterations used to create the texture. As this number is increased, turbulence is generated by summing noise, with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother, increasing in detail as the number of octaves is increased.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence, stirring up the pattern.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value**, truncating values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to

	extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value**, truncating values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Increasing this value causes the texture to favor the foreground color over the background color, whereas decreasing the value causes the background color to be favored.
Gain	Similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient.
Output Regions	When the **Output Regions** option is enabled, the procedural texture outputs random gray shades per region rather than outlines for tiles, providing a means to add random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value.**
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, the **Background Color** area is completely transparent, revealing the shading results of any lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, the **Foreground Color** area is completely transparent, revealing the shading results of any lower layers.

Hard Wood

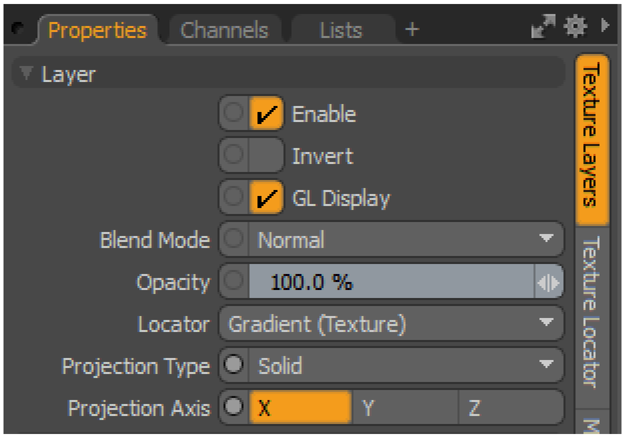


The **Hard Wood** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Hard Wood** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. The look of this texture creates a wood-like effect. It differs from the **Easy Wood** texture in that it provides a grainier, noisier wood pattern.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

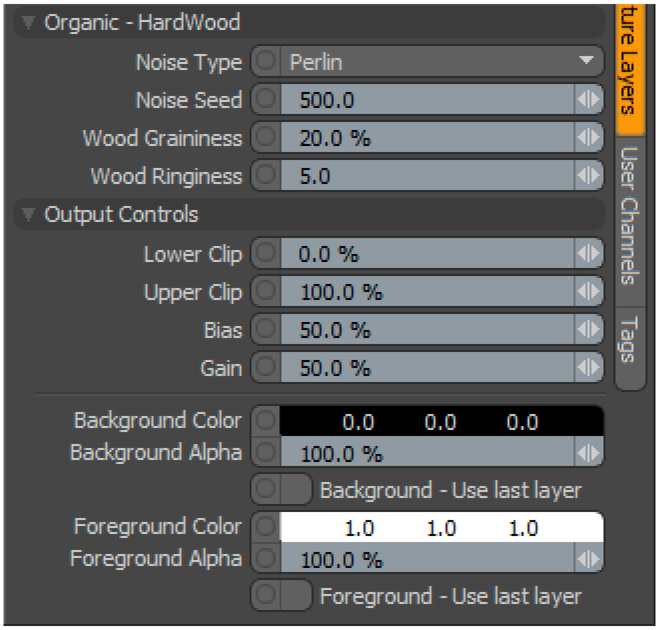
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Hardwood Properties



Organic - HardWood

Noise Type Perlin

Noise Seed 500.0

Wood Graininess 20.0 %

Wood Ringiness 5.0

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

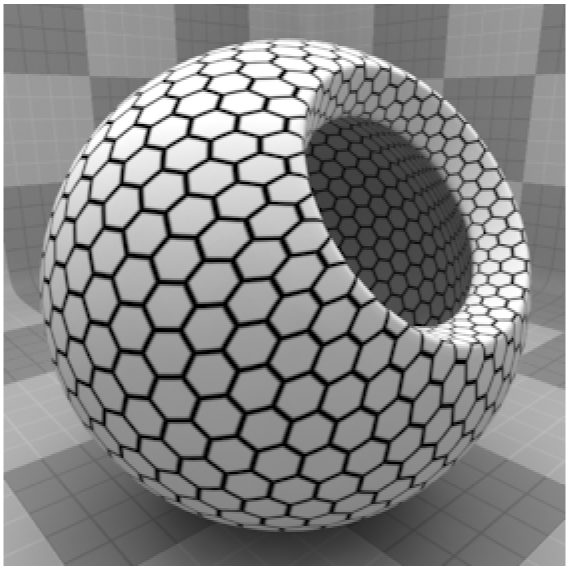
Foreground Alpha 100.0 %

Foreground - Use last layer

Organic - Hard Wood—	
Noise Type	Specifies the look of the texture distortion with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Determines the random variation for the texture. The **Seed** value is the initial number Modo uses when generating the procedural values. Different values produce different random variations and can be useful in changing the appearance of the texture; however, use the same value when you want items to retain the same variation.
Wood Graininess	Controls the graininess of the wood rings.

Wood Ringiness	Controls the amplitude of the wood rings.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Hexagonal Tiles



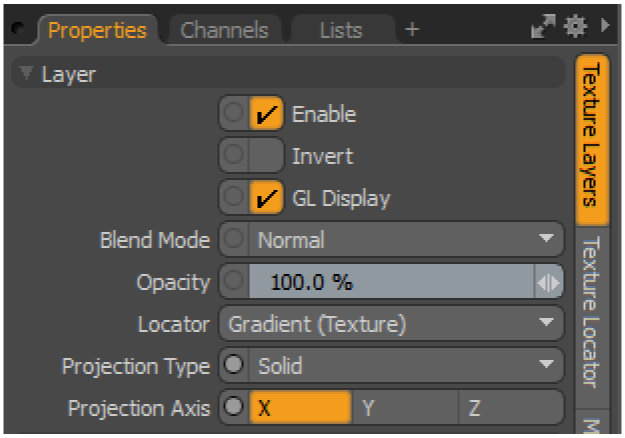
The **Hexagonal Tiles** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Hexagonal Tiles** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a hexagonal tile pattern with variable grout, tiles edges, and tile height.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties panel change the **Projection Type** to a UV Map (recommended) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

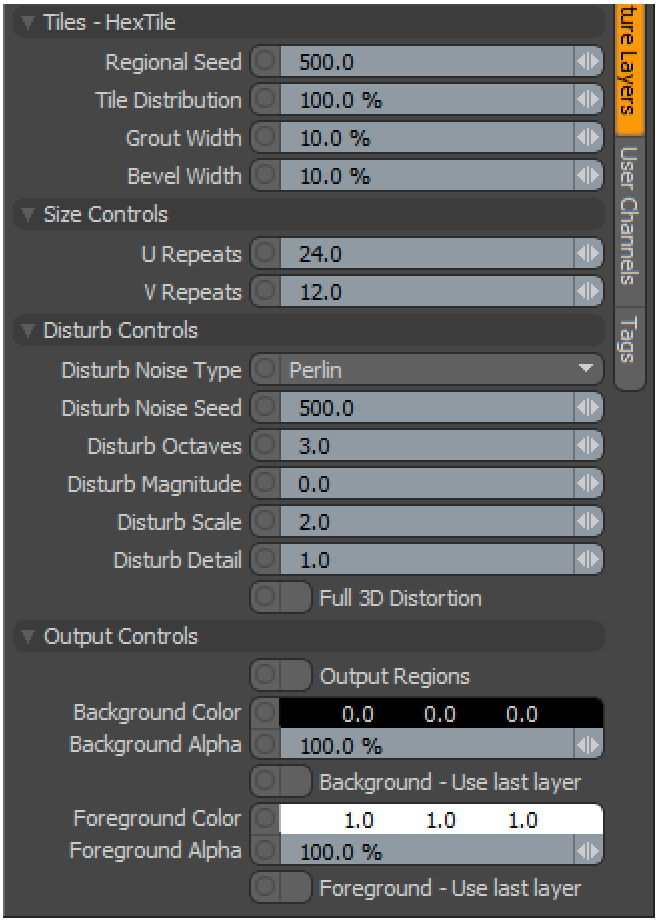
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Hex Tile Properties



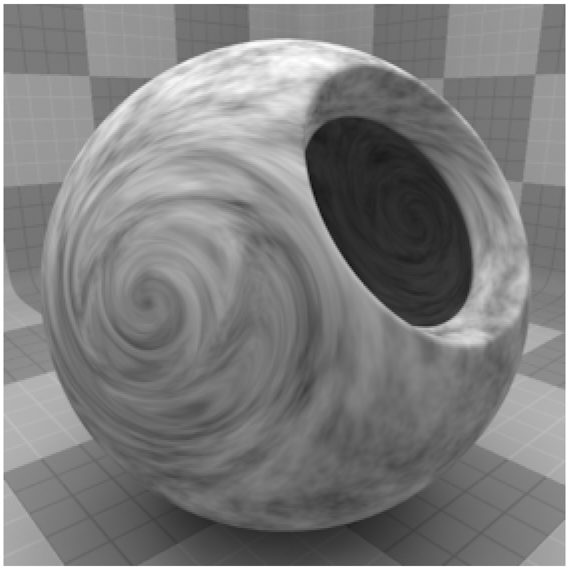
Tiles - HexTile
Regional Seed 500.0
Tile Distribution 100.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - HexTile	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Tile Distribution	Determines the percentage of the surface that is covered by tiles.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Size Controls	

U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.

Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Hurricane

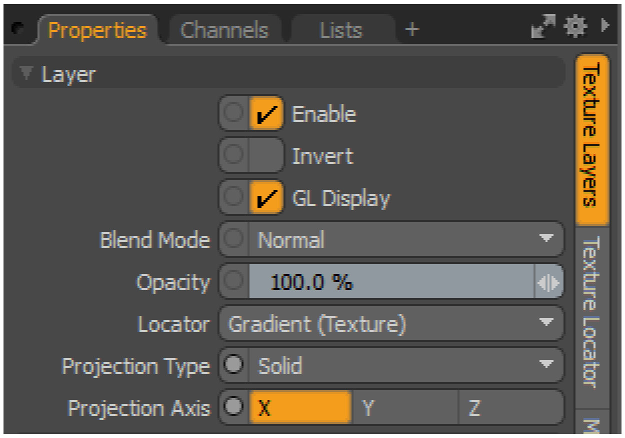


The **Hurricane** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Hurricane** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture applies a pattern similar to the swirling clouds present in a hurricane. You can control the amount of twist or swirling — along with the size of the eye of the hurricane and detail of the clouds.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode Normal

Opacity 100.0 %

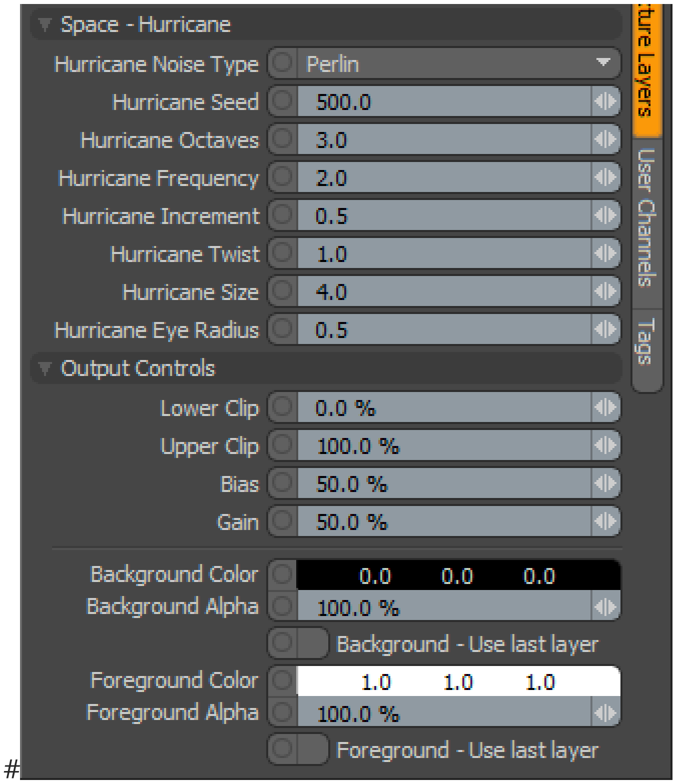
Locator Gradient (Texture)

Projection Type Solid

Projection Axis X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Hurricane Properties



Space - Hurricane

Hurricane Noise Type: Perlin
Hurricane Seed: 500.0
Hurricane Octaves: 3.0
Hurricane Frequency: 2.0
Hurricane Increment: 0.5
Hurricane Twist: 1.0
Hurricane Size: 4.0
Hurricane Eye Radius: 0.5

Output Controls

Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %

Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

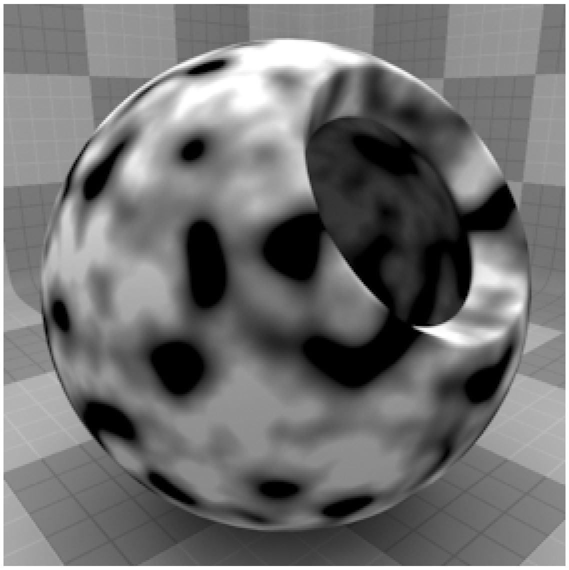
#

Space - Hurricane	
Hurricane Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Hurricane Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when

	you want items to retain the same variations.
Hurricane Octaves	Defines the number of iterations used to create the texture. As you increase this number, Modo generates turbulence by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother and increases in detail as you increase the number of octaves.
Hurricane Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence and, therefore, "stirs up" the pattern.
Hurricane Increment	Controls the scaling of each successive iteration. When the value is low, the turbulence is tight and crinkled. As you increase the value, the turbulence becomes smoother.
Hurricane Twist	Controls the amount of twist applied to the surface.
Hurricane Size	Controls the overall size of the hurricane.
Hurricane Eye Radius	Controls the size of the eye of the hurricane.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.

Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Hybrid

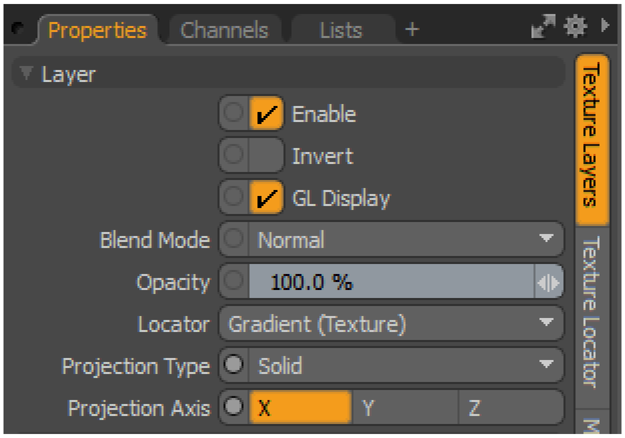


The **Hybrid** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Hybrid** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture is similar to the noisy bumps provided by the **FMB** (Fractal Brownian Motion) texture, but the **Hybrid** texture has smoother valleys to make it suitable for craggy mountain tops.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

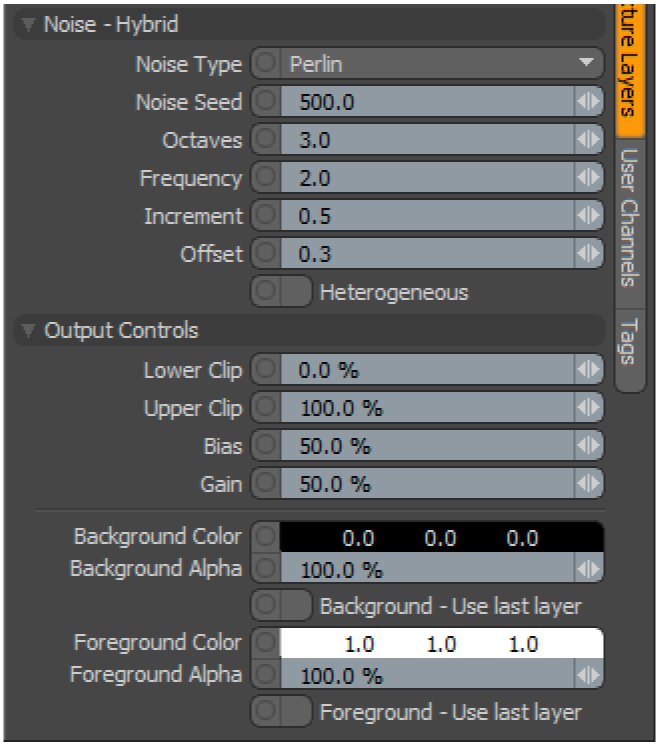
Projection Type: Solid

Projection Axis: X

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Hybrid Properties

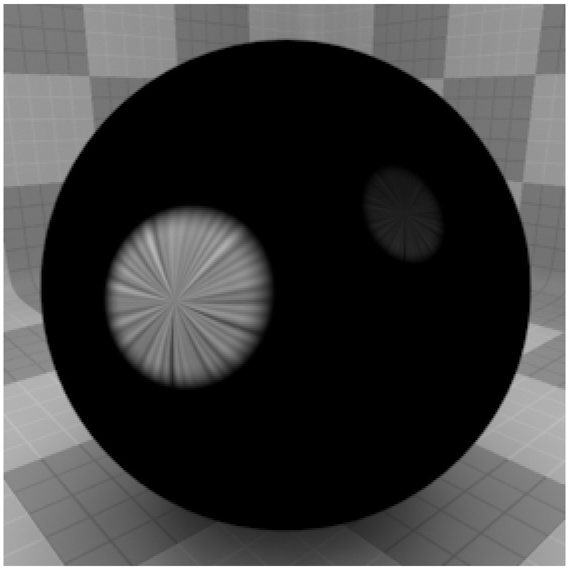

Noise - Hybrid
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Frequency: 2.0
Increment: 0.5
Offset: 0.3
☐ Heterogeneous
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Noise - Hybrid	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin**

	• **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As this number increases, Modo generates turbulence by summing noise — with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother. It increases in detail as you increase the number of octaves.
Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence to "stir up" the pattern.
Increment	Controls the scaling of each successive iteration. When you set the control to a low value, the turbulence is tight and crinkled. It becomes increasingly smoother as you increase the value.
Offset	Controls the detail of the texture. When you set this at a low value, the detail is very sharp. The details become increasingly faded as you increase the value.
Heterogeneous	When enabled, switches from using the default fractal model to a heterogeneous terrain model.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color.

	Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Iris

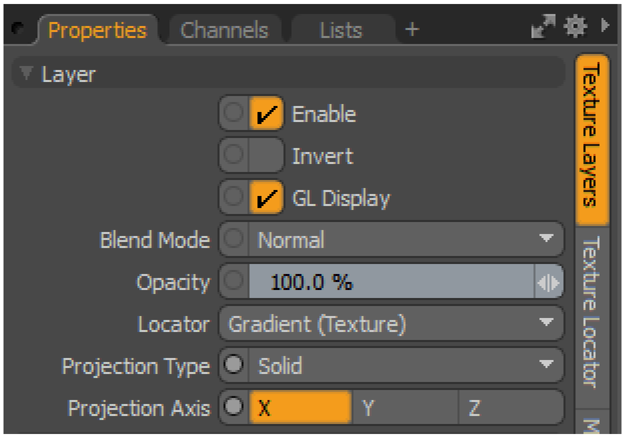


The **Iris** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Iris** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This particular texture provides a pattern similar in shape and texture to an iris (of an eye). This texture is great for making eyeballs or for adding a lens flare-type element.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

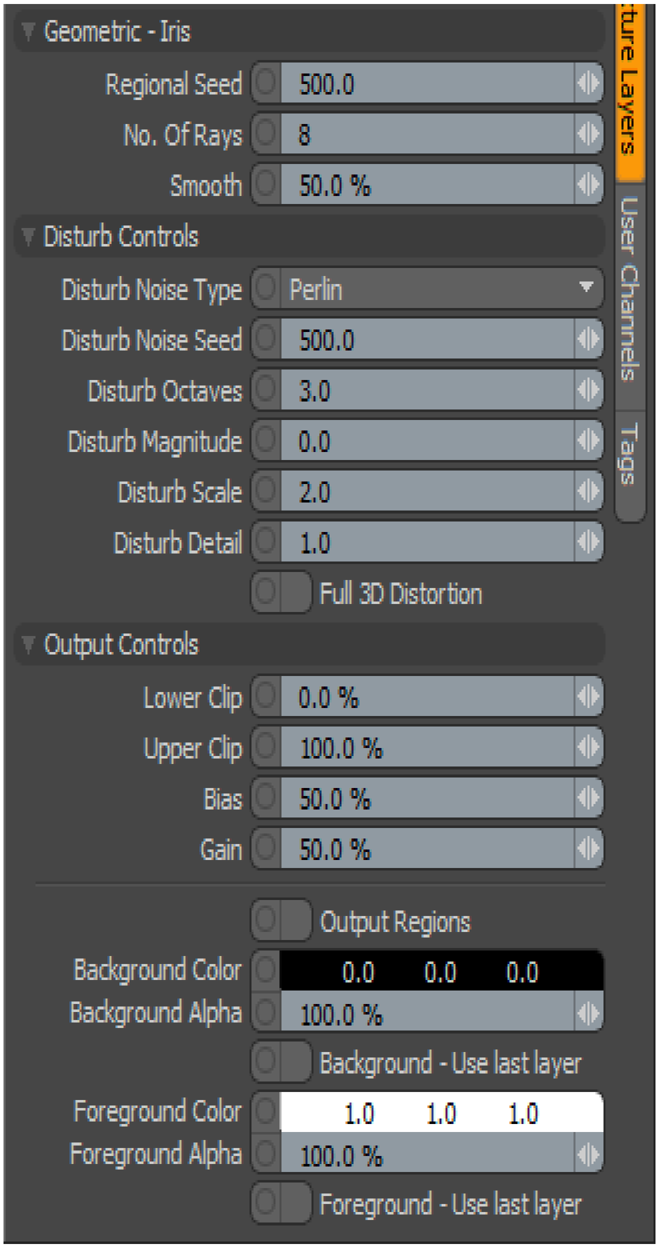
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Iris Properties



Geometric - Iris

Regional Seed 500.0

No. Of Rays 8

Smooth 50.0 %

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

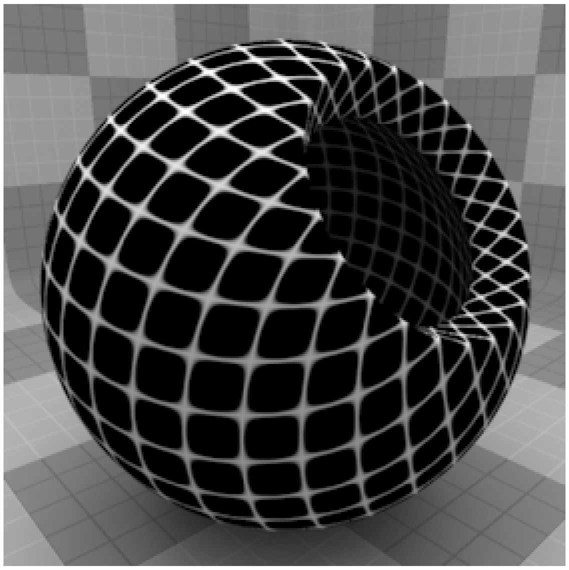
Foreground - Use last layer

Geometric - Iris	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

Number Of Rays	Determines the amount of smoothing around the edges of the iris. Higher values produce smoother edges.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence to stir up the pattern.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to

	extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Lattice 1



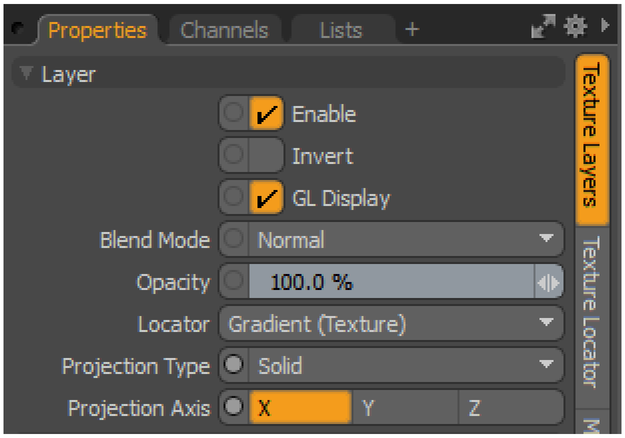
The **Lattice 1** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Lattice 1** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a lattice pattern with variable thickness, which can be used to texture nets or fences.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties panel change the **Projection Type** to a UV Map (recommended) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

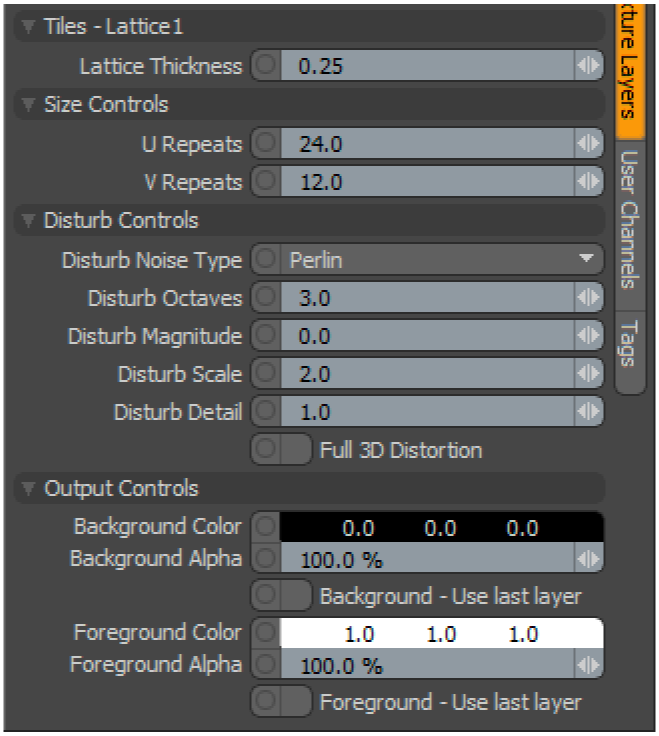
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Lattice 1 Properties



Tiles - Lattice1

Lattice Thickness 0.25

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

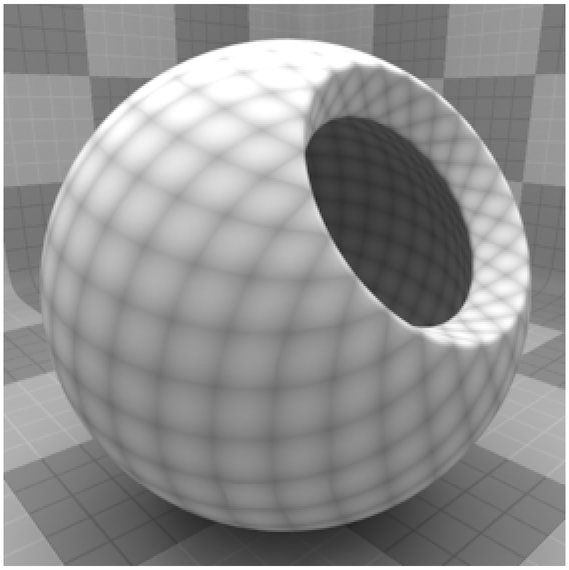
Foreground Alpha 100.0 %

Foreground - Use last layer

Tiles - Lattice 1	
Lattice Thickness	Determines the overall thickness of the lattice.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value**

	• **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Lattice 2



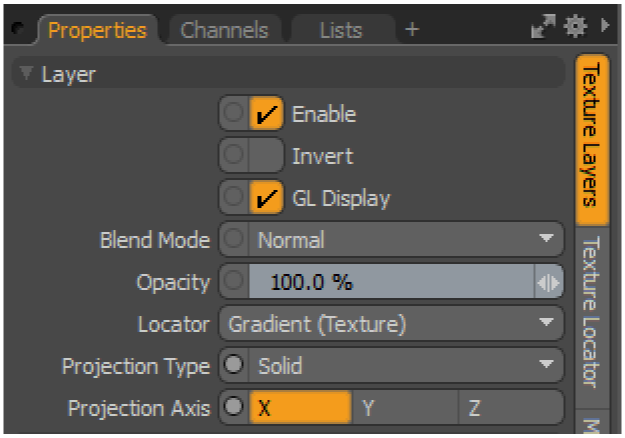
The **Lattice 2** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Lattice 2** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a lattice pattern with variable thickness, which can be used to texture nets or fences.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties panel change the **Projection Type** to a UV Map (recommended) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

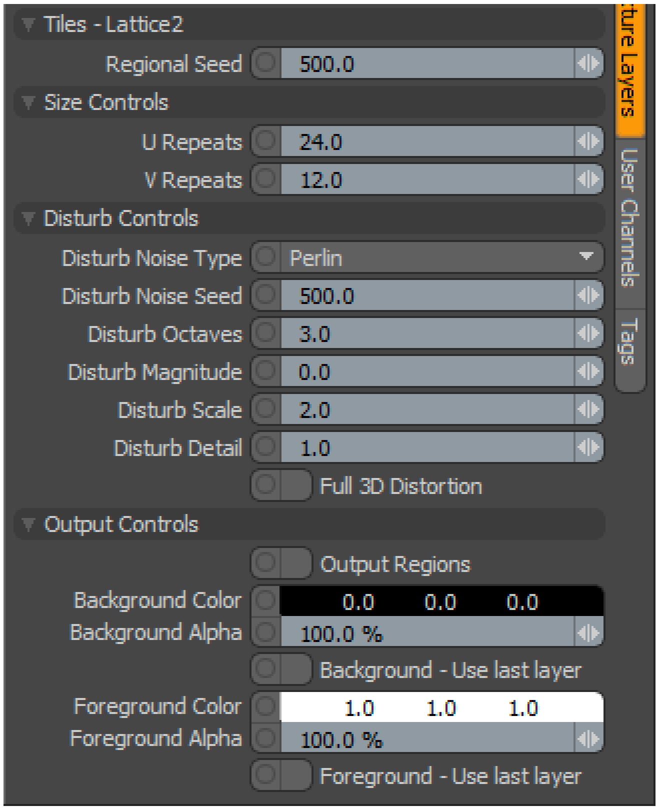
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Lattice 2 Properties



Tiles - Lattice2

Regional Seed 500.0

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

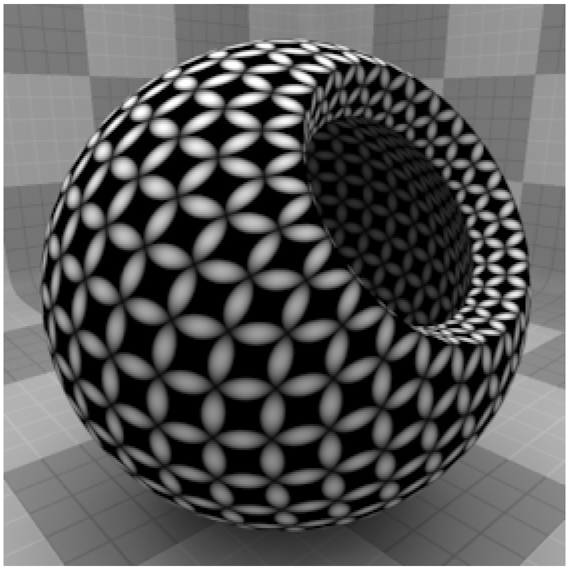
Foreground - Use last layer

Tiles - Lattice 2	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin**

	• **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.

Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Lattice 3



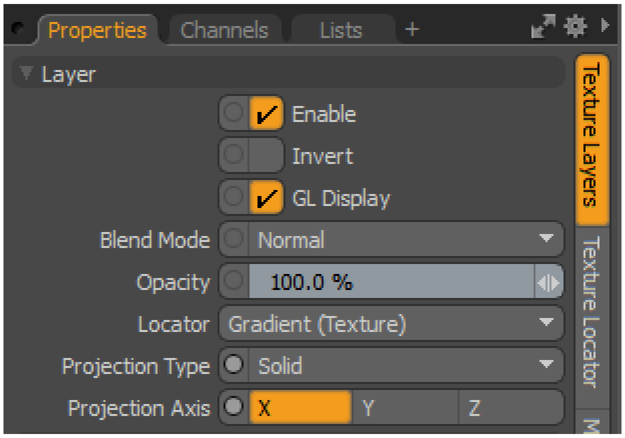
The **Lattice 3** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Lattice 3** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a lattice pattern with variable thickness, which can be used to texture nets or fences.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties panel change the **Projection Type** to a UV Map (recommended) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>

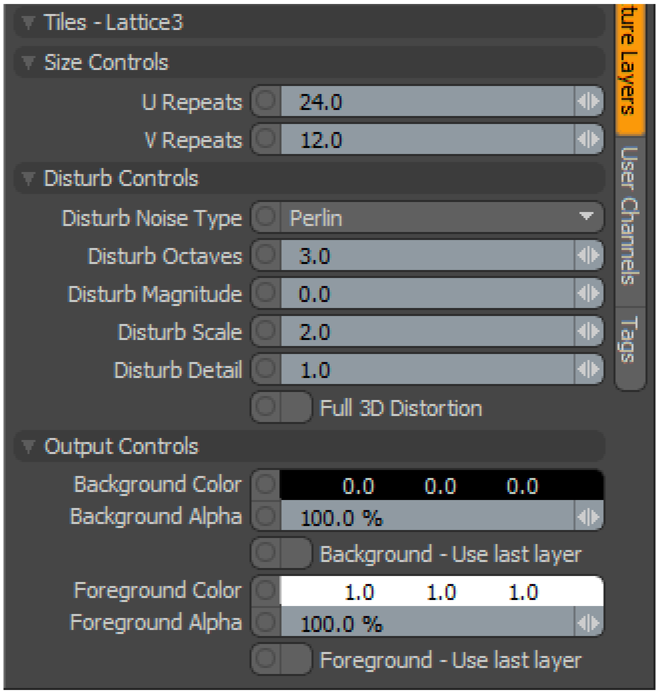


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Lattice 3 Properties



Tiles - Lattice3

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

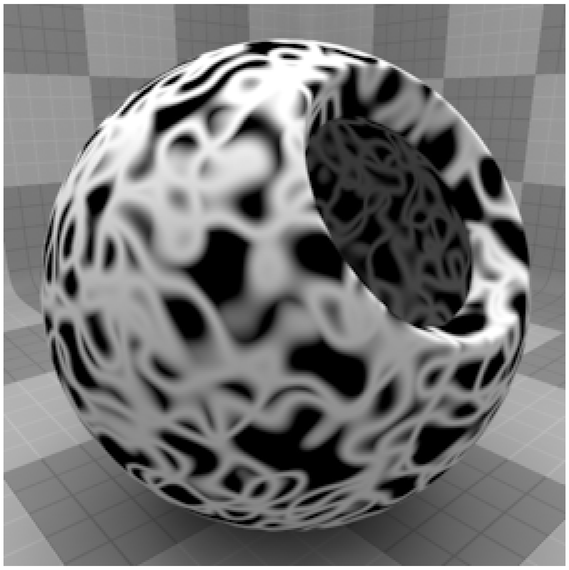
Foreground Alpha 100.0 %

Foreground - Use last layer

Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided:

	- **Perlin** - **Enhanced Perlin** - **Gradient** - **Value** - **Gradient Value** - **Impulse** - **Lattice** - **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Leather

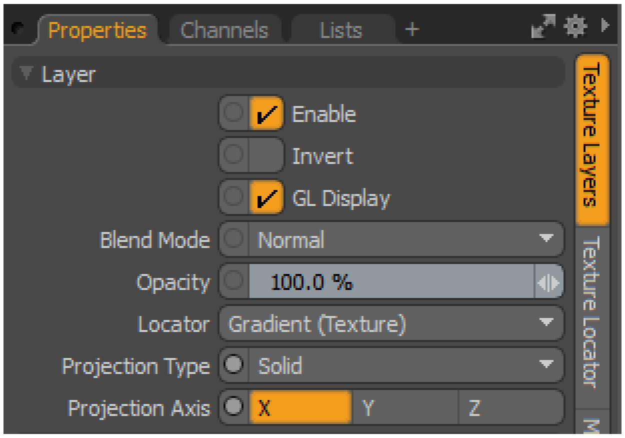


The **Leather** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Leather** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. The look of this texture emulates the grain of leather or wrinkles. The various controls provide multiple variations of grain width, depth, and color.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

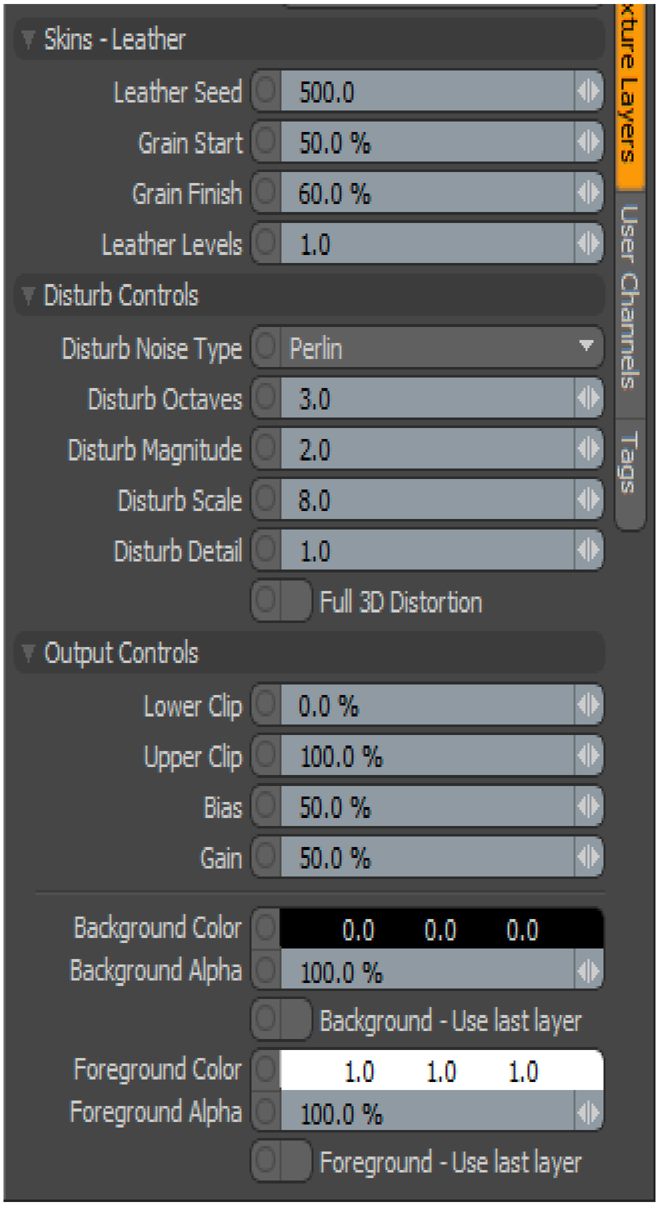
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Leather Properties



Skins - Leather

Leather Seed 500.0

Grain Start 50.0 %

Grain Finish 60.0 %

Leather Levels 1.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 2.0

Disturb Scale 8.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

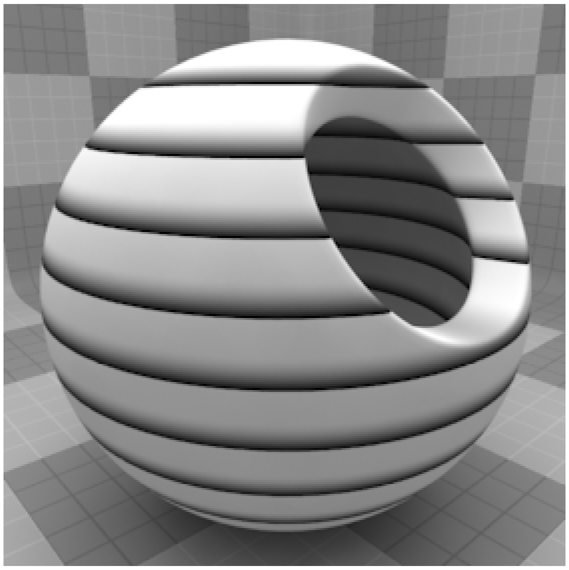
Foreground - Use last layer

Skins - Leather	
Leather Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

Grain Start	Controls where the valleys of the leather grain pattern start.
Grain Finish	Controls where the valleys of the leather grain pattern finish.
Leather Levels	Controls the number of leather patch levels.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base procedural.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to

	extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Linear

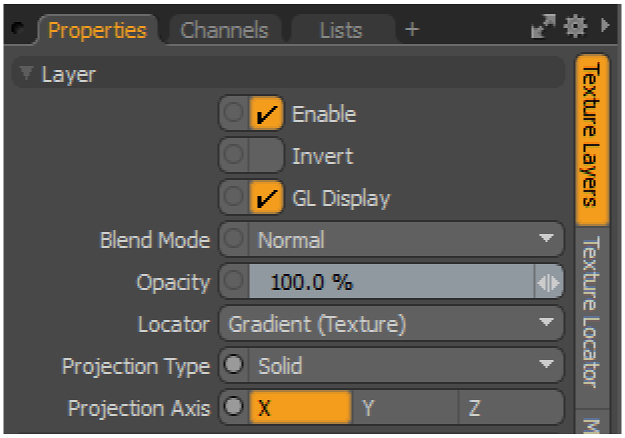


The **Linear** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Linear** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This particular texture provides a series of adjustable linear patterns.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

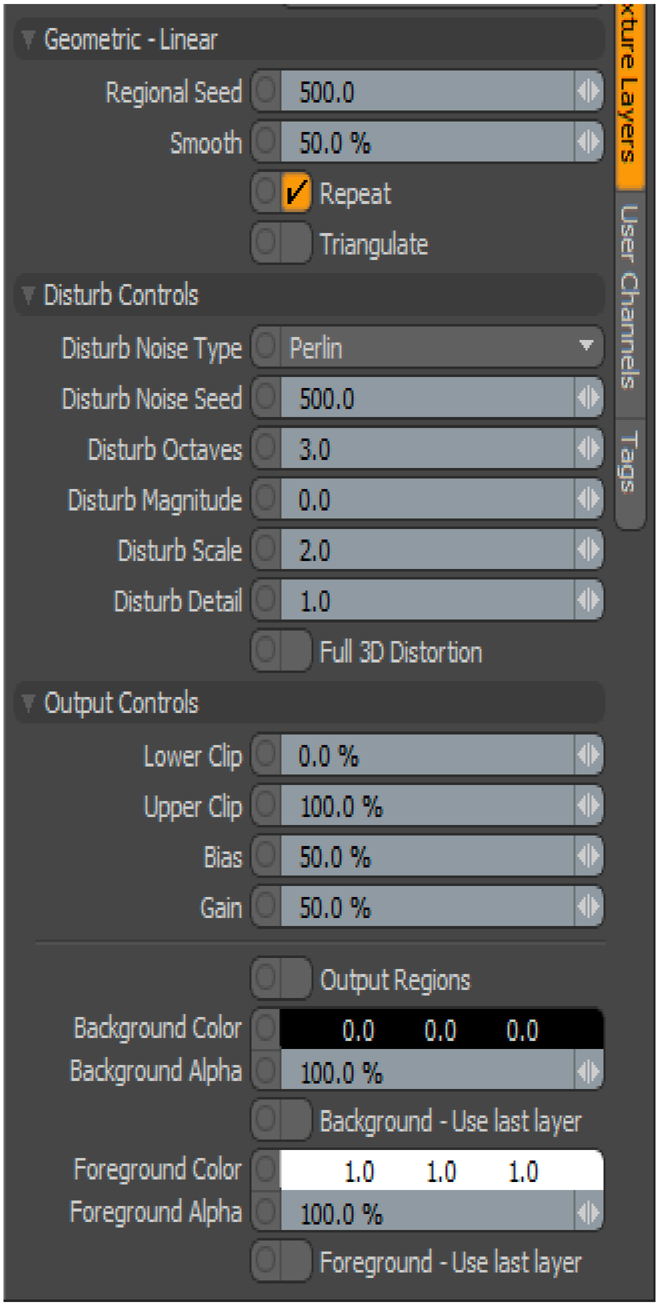
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Linear Properties



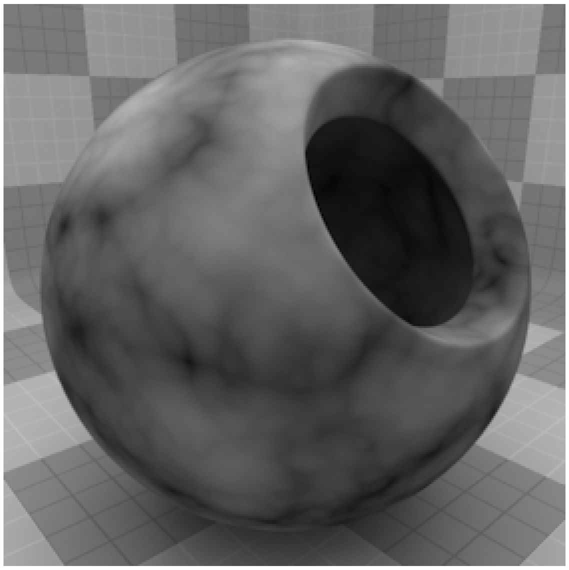
Geometric - Linear
Regional Seed 500.0
Smooth 50.0 %
☑ Repeat
☐ Triangulate
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Geometric - Linear	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when

	you want items to retain the same variations.
Smooth	Determines the amount of smoothing around the edges of the liner patterns. Higher values produce smoother edges.
Repeat	Causes the texture to repeat indefinitely.
Triangulate	Causes the pattern to have a triangle-based edging pattern instead of the default sawtooth blend.
Disturb Controls - Alters the look of the generated procedural. The noise layer distorts the base texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base procedural.
Disturb Scale	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir up the pattern.
Disturb Detail	Specifies the step size between each iteration of the noise used to disturb the texture.

Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Lump

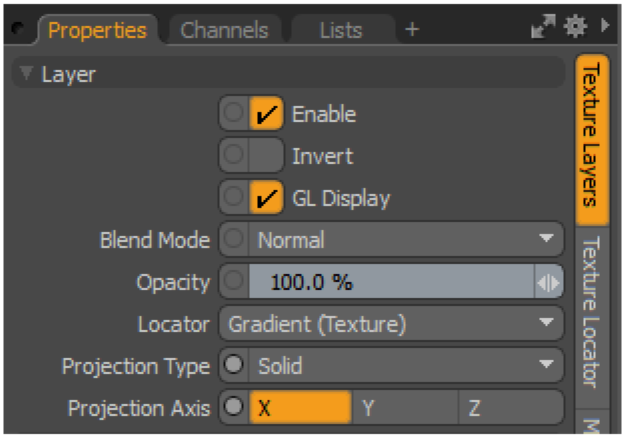


The **Lump** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Lump** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. The look of this particular texture provides lumpy noise that is similar to billowing clouds.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

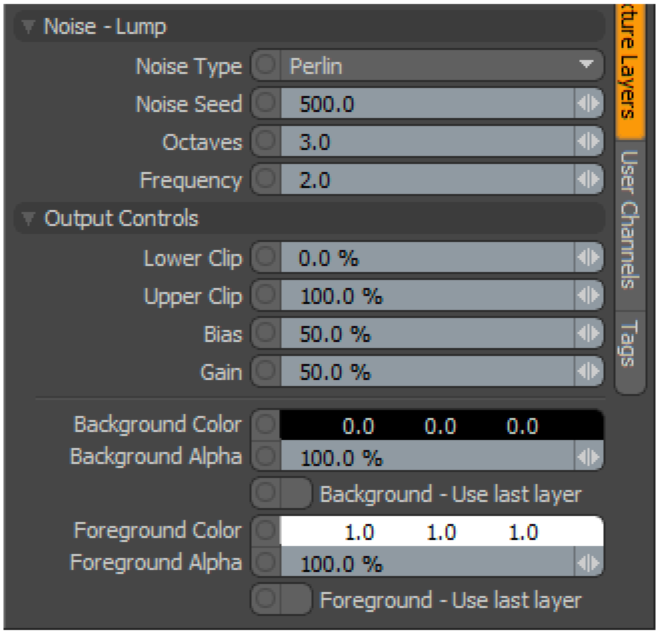
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Lump Properties



Noise - Lump

Noise Type: Perlin

Noise Seed: 500.0

Octaves: 3.0

Frequency: 2.0

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

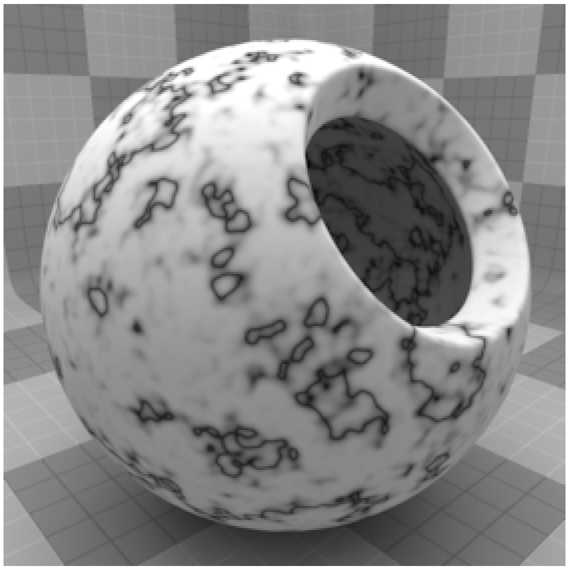
Foreground - Use last layer

Noise - Lump	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations used to create the texture. As you increase this number, Modo generates turbulence by summing noise with each summed noise

	being half the magnitude of the previously summed noise. At low values, the turbulence is smoother and increases in detail as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir up the pattern.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.

Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Marble Noise

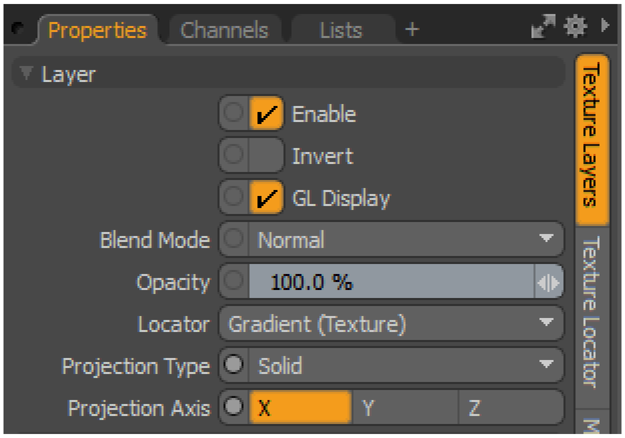


The **Marble Noise** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Marble Noise** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. The look of this particular texture provides a marble-like pattern. However, unlike the **Marble Vein** texture, this is a noise-based texture, which gives it much more detail.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode Normal

Opacity 100.0 %

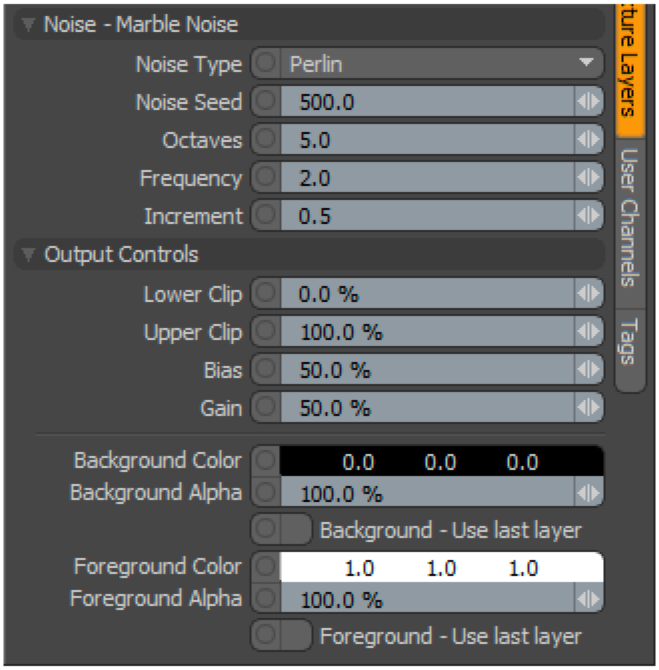
Locator Gradient (Texture)

Projection Type Solid

Projection Axis X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Marble Noise Properties



Noise - Marble Noise

Noise Type: Perlin

Noise Seed: 500.0

Octaves: 5.0

Frequency: 2.0

Increment: 0.5

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

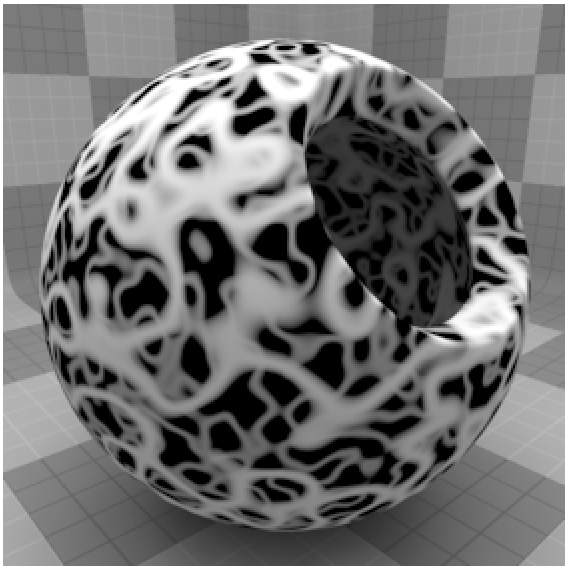
Foreground - Use last layer

Noise - Marble Noise	
Noise Type	Specifies the look of the texture distortion, with several noise function types provided: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations used to create the texture. As you increase this

	number, Modo generates turbulence by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smoother and increases in detail as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir up the pattern.
Increment	Controls the scaling of each successive iteration. When the value is low, the turbulence is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or contract the total range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.

Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Marble Vein

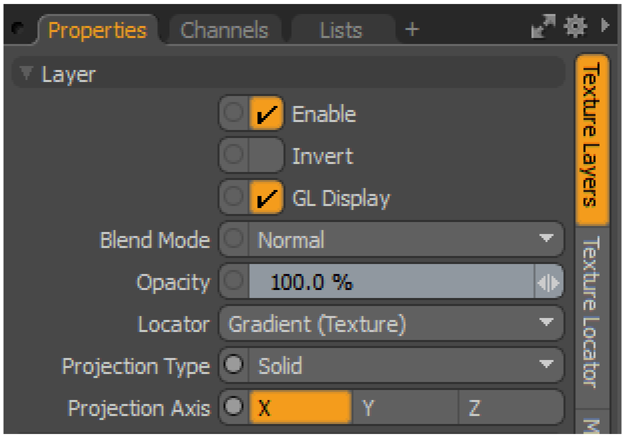


The **Marble Vein** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Marble Vein** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a veiny marble pattern. This texture is less detailed than the **Marble Noise** texture.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

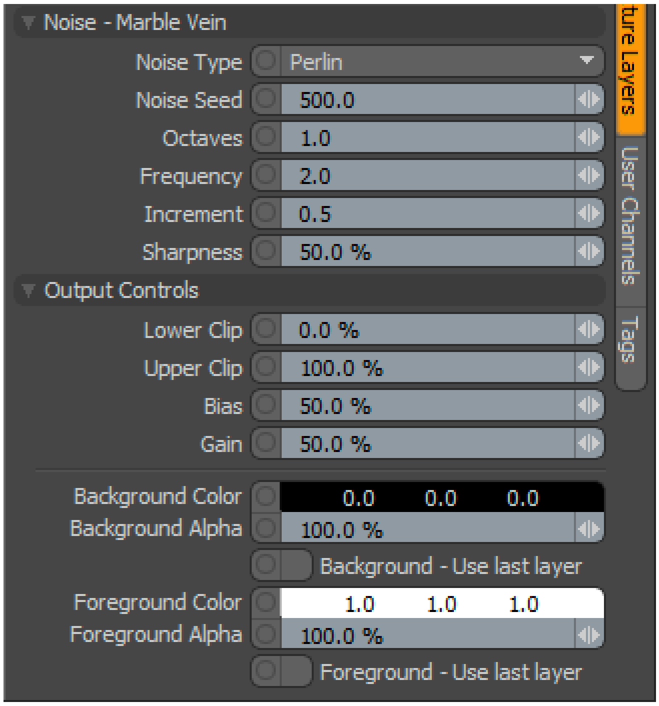
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Marble Vein Properties



Noise - Marble Vein

Noise Type Perlin

Noise Seed 500.0

Octaves 1.0

Frequency 2.0

Increment 0.5

Sharpness 50.0 %

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

○ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

○ Foreground - Use last layer

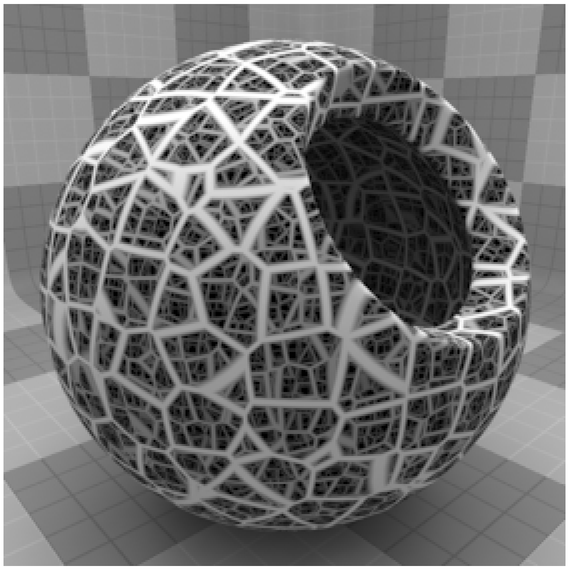
s

Noise - Marble Vein	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Sharpness	Controls the point at which the texture becomes more detailed.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles.

	This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Membrane

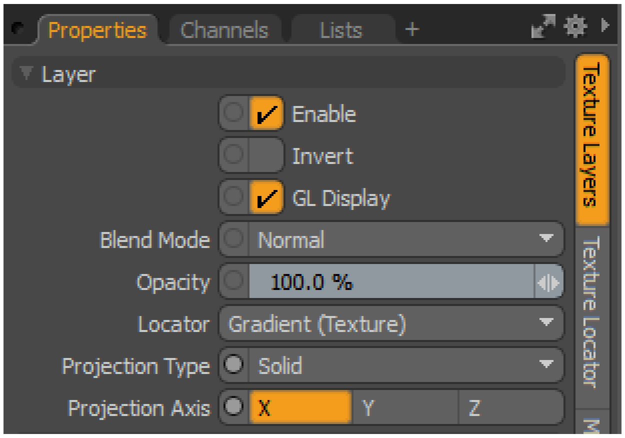


The **Membrane** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Membrane** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. The look of this particular texture is a layered membrane that is similar to the pith found in an orange.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

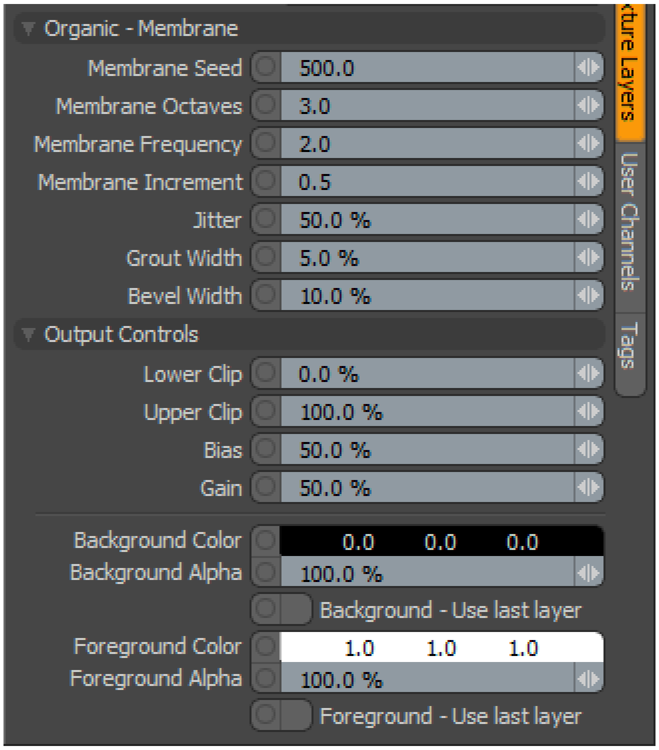
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Membrane Properties



Organic - Membrane

Membrane Seed 500.0
Membrane Octaves 3.0
Membrane Frequency 2.0
Membrane Increment 0.5
Jitter 50.0 %
Grout Width 5.0 %
Bevel Width 10.0 %

Output Controls

Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %

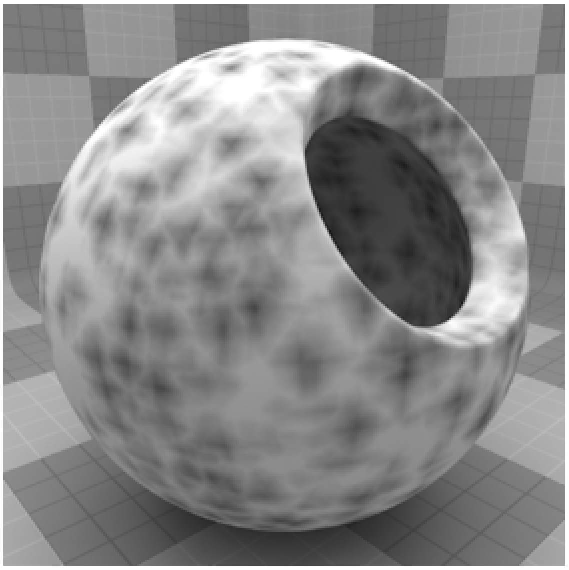
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Organic - Membrane	
Membrane Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

Membrane Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Membrane Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Membrane Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Jitter	Controls the jitter. Higher values cause more jitter in the membrane.
Grout Width	Controls the width between cells.
Bevel Width	Controls the bevel between the edges of the cells.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward

Color/Value	the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Minky



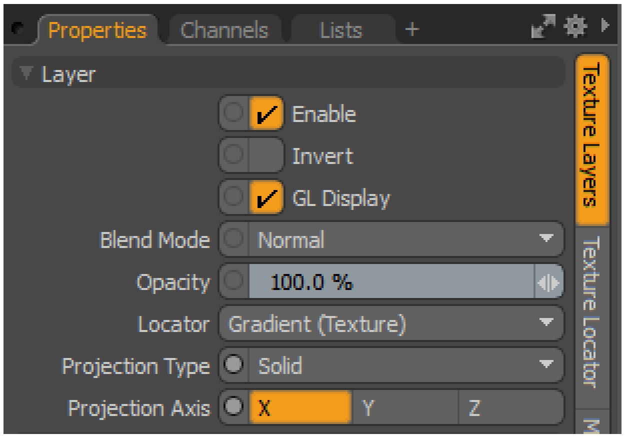
The **Minky** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Minky** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then

Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. The look of this particular texture is based on the **Cellular** texture, except that the distance between the cells is derived in a linear manner. This creates a diamond-like surface pattern.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

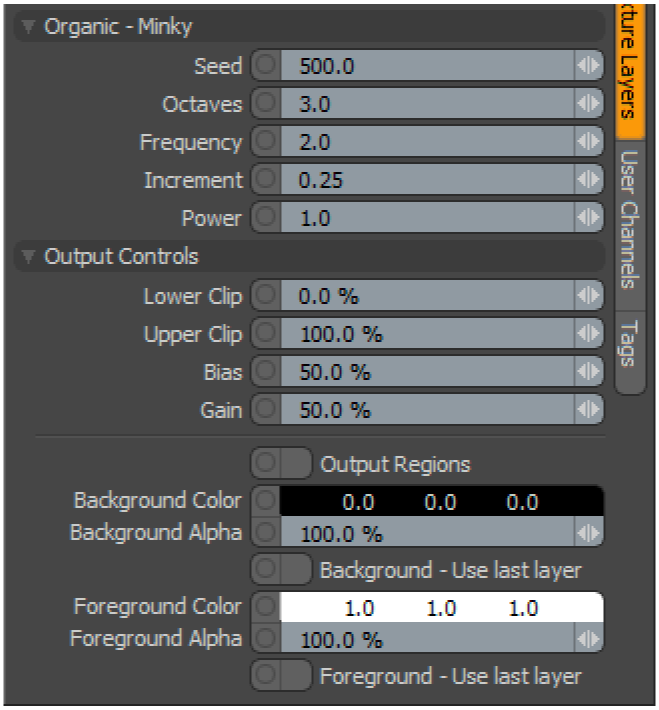
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture**

	Locator that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Minky Properties



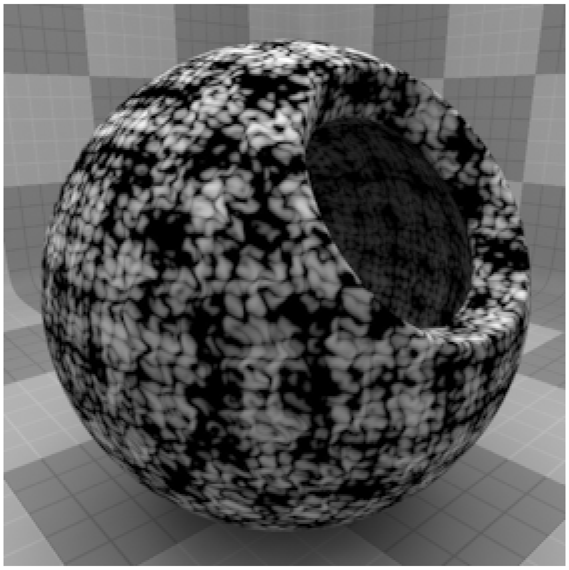
Organic - Minky
Seed 500.0
Octaves 3.0
Frequency 2.0
Increment 0.25
Power 1.0
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Organic - Minky	
Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase

	this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Power	Controls the degree to which Modo mixes up the edges of the texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Monster

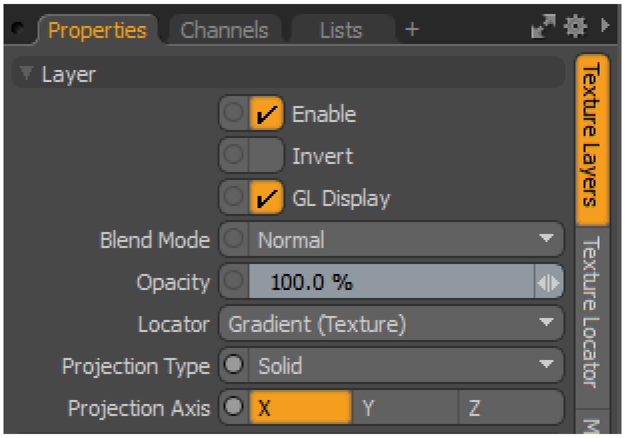


The **Monster** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Monster** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This particular texture provides a look similar to a monster's skin. The texture shows groups of bumps that can display large, scab-like areas.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

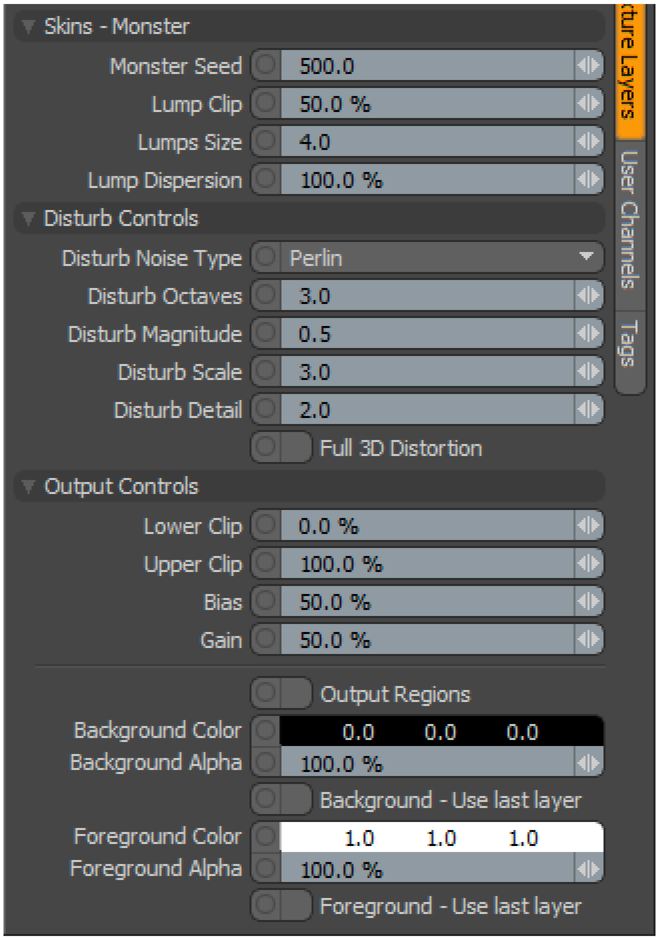
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Monster Properties



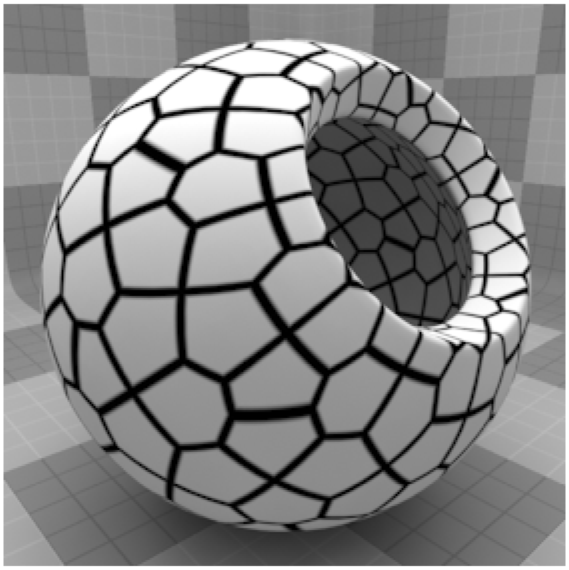
Skins - Monster
Monster Seed 500.0
Lump Clip 50.0 %
Lumps Size 4.0
Lump Dispersion 100.0 %
Disturb Controls
Disturb Noise Type Perlin
Disturb Octaves 3.0
Disturb Magnitude 0.5
Disturb Scale 3.0
Disturb Detail 2.0
Full 3D Distortion
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Skins - Monster	
Monster Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Lump Clip	Determines where the texture shading transitions to the background color. The higher the value is, the quicker the transition occurs.

Lump Size	Determines the size of the smaller lumps.
Lump Dispersion	Determines how many of the lumps Modo actually draws.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.

Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Mosaic



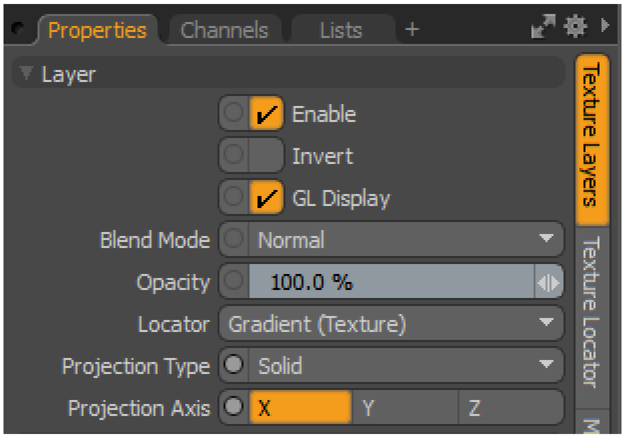
The **Mosaic** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Mosaic** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a mosaic pattern of randomly shaped tiles with variable grout, tile scale, and tile height.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties panel change the **Projection Type** to a UV Map (recommended) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

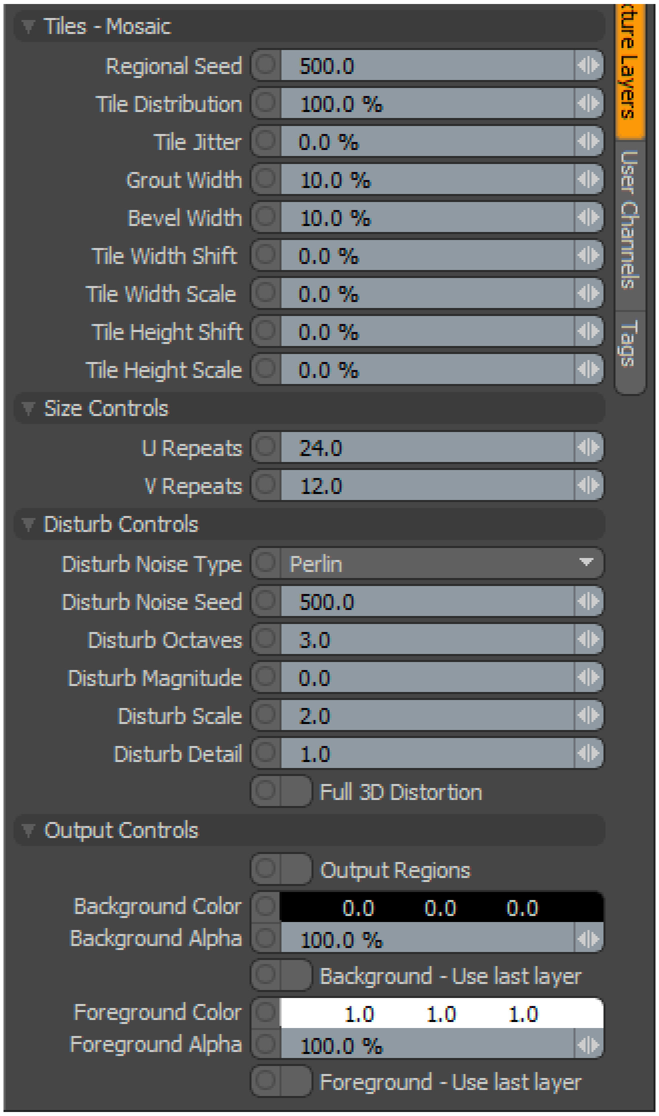
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Mosaic Properties



Tiles - Mosaic

Regional Seed 500.0
Tile Distribution 100.0 %
Tile Jitter 0.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Tile Width Shift 0.0 %
Tile Width Scale 0.0 %
Tile Height Shift 0.0 %
Tile Height Scale 0.0 %

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

Output Controls

Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

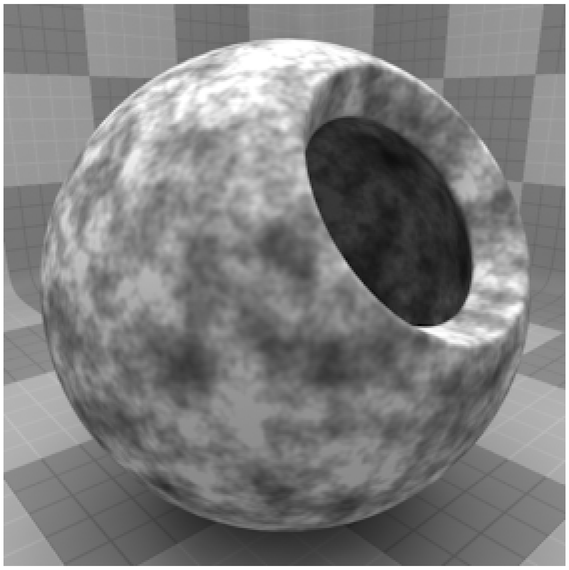
Tiles - Mosaic	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Tile Distribution	Determines the percentage of the surface that is covered by tiles.
Tile Jitter	Determines the amount of randomness in the tiles' placement. When you set this to a

	low value, the tiles are bit more uniform. The tiles become more random as you increase the value.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Tile Width Shift	Controls the horizontal shift of the tiles.
Tile Width Scale	Controls the horizontal scale of the tiles.
Tile Height Shift	Controls the vertical shift of the tiles.
Tile Height Scale	Controls the vertical scale of the tiles.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.

Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.

Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Multi-Fractal

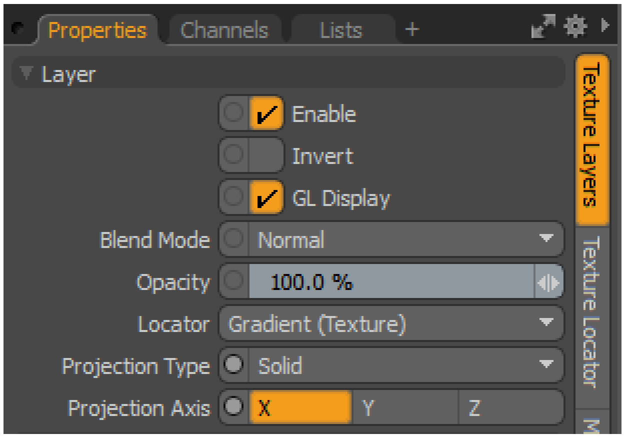


The **Multi-Fractal** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Multi-Fractal** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides noisy bumps based on Fractal Brownian Motion (the **FBM** texture). This texture is different from the **FBM** texture in that the detail increases at the peaks.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

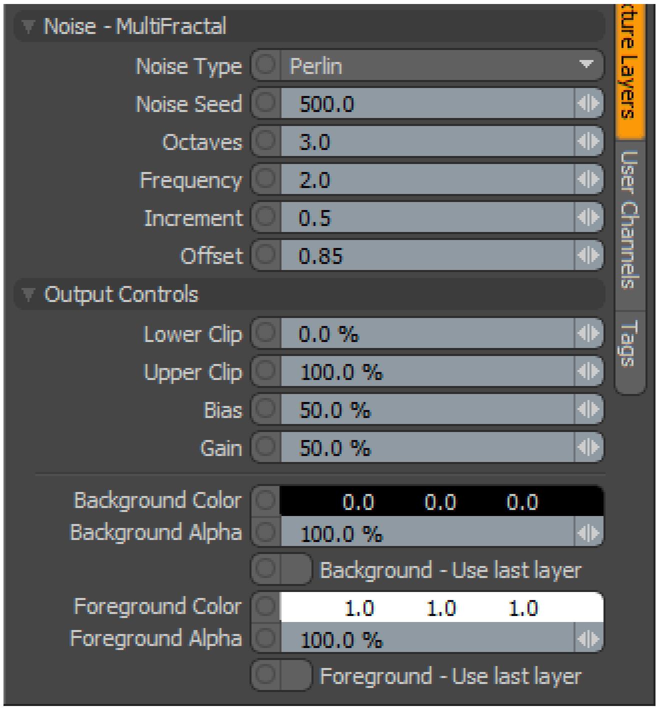


Layer

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Multi-Fractal Properties



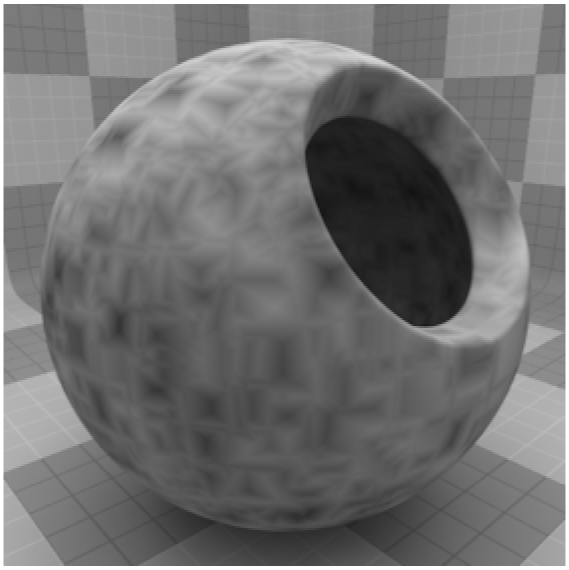
Noise - MultiFractal
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Frequency: 2.0
Increment: 0.5
Offset: 0.85
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Multi-Fractal	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Offset	Controls the point at which the texture becomes more detailed.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Nurnies

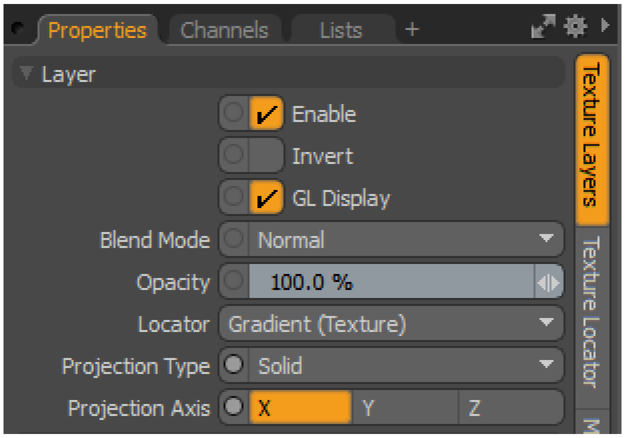


The **Nurnies** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Nurnies** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This shader creates a random tile-type of pattern, which makes it useful for adding detail to a surface (such as the hull of a space ship).

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

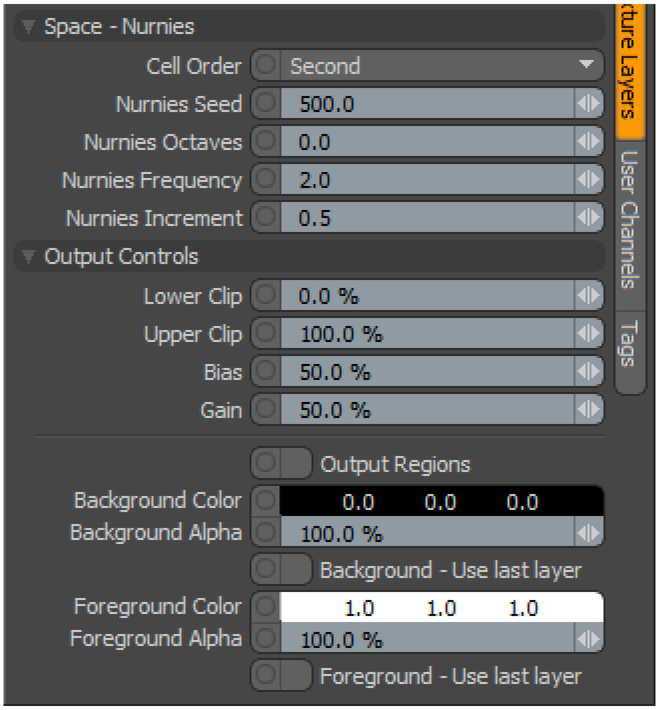
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Nurnies Properties

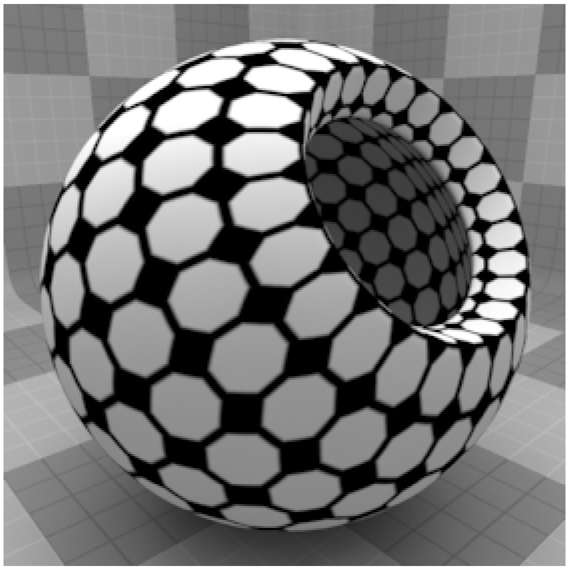

Space - Nurnies
Cell Order: Second
Nurnies Seed: 500.0
Nurnies Octaves: 0.0
Nurnies Frequency: 2.0
Nurnies Increment: 0.5
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Output Regions
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Nurnies	
Cell Order	Determines the cell order that Modo uses for creating the texture. • **First - Nearest Cell** • **Second - Next Nearest Cell** • **Third - Next Nearest Cell** • **Fourth - Furthest Cell**
Nurnies Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

Nurnies Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Nurnies Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Nurnies Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use	When enabled, makes the **Background Color** area completely transparent to reveal

Last Layer	the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Octagonal Tile

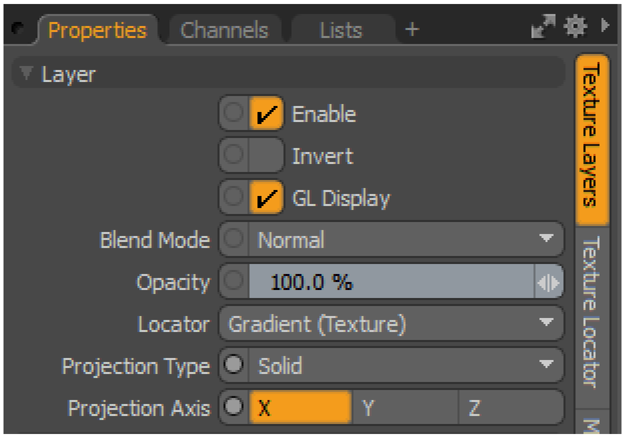


The **Octagonal Tile** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Octagonal Tile** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides an octagonal tile pattern with variable grout, tile edges, and tile height.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties panel change the **Projection Type** to a UV Map (recommended) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties



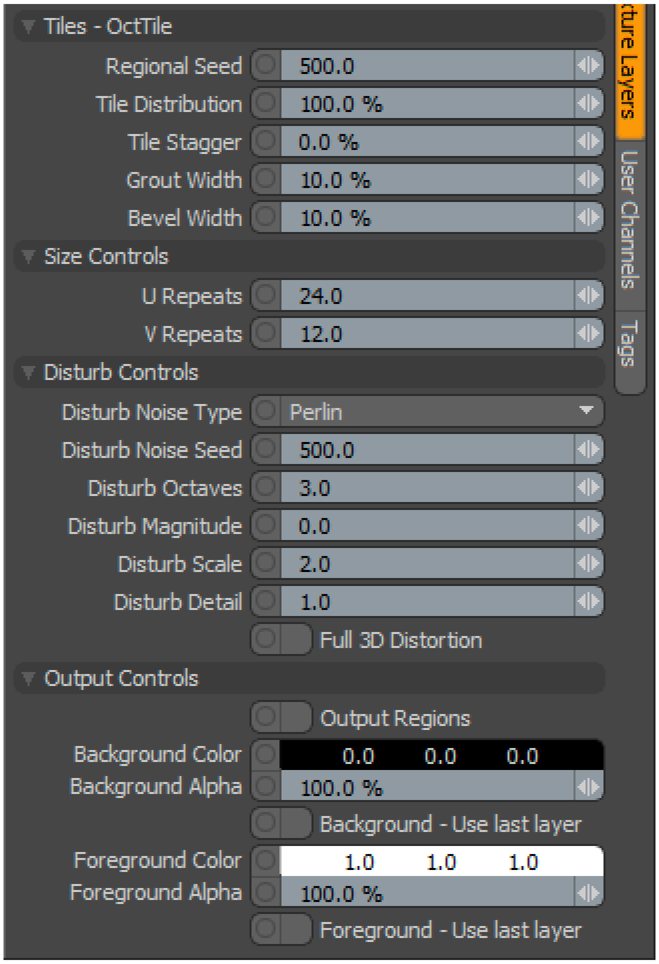
☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Oct Tile Properties



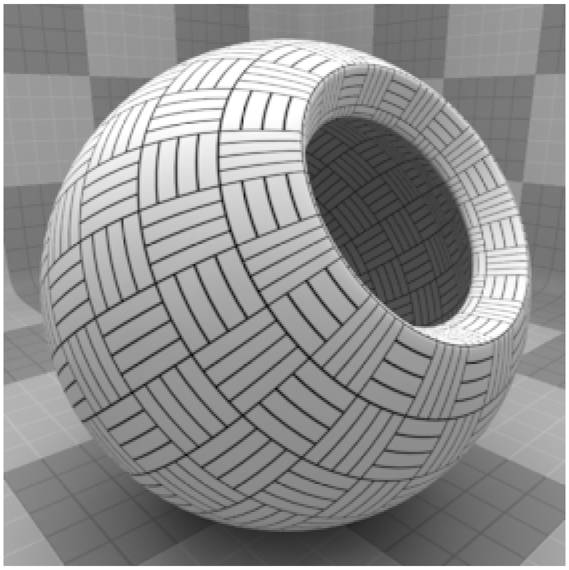
Tiles - OctTile
Regional Seed 500.0
Tile Distribution 100.0 %
Tile Stagger 0.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Oct Tiles	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Tile Distribution	Determines the percentage of the surface that is covered by tiles.
Tile Stagger	Determines how staggered the rows of tiles are in the texture. When you set a low value, the tiles line up vertically. The tile rows become more staggered as you increase the value.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.

Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.

Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Parquet



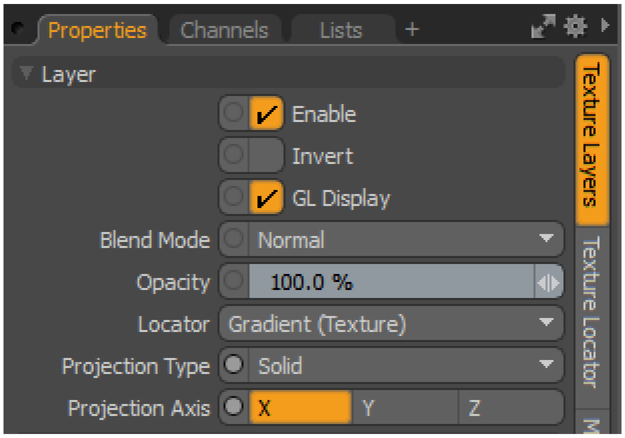
The **Parquet** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Parquet** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a parquet (or blocked wood-type) texture, which you can use for floor coverings. You can modify the number of blocks used, the graininess of the wood, the visibility of the wood rings, and the block edges.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

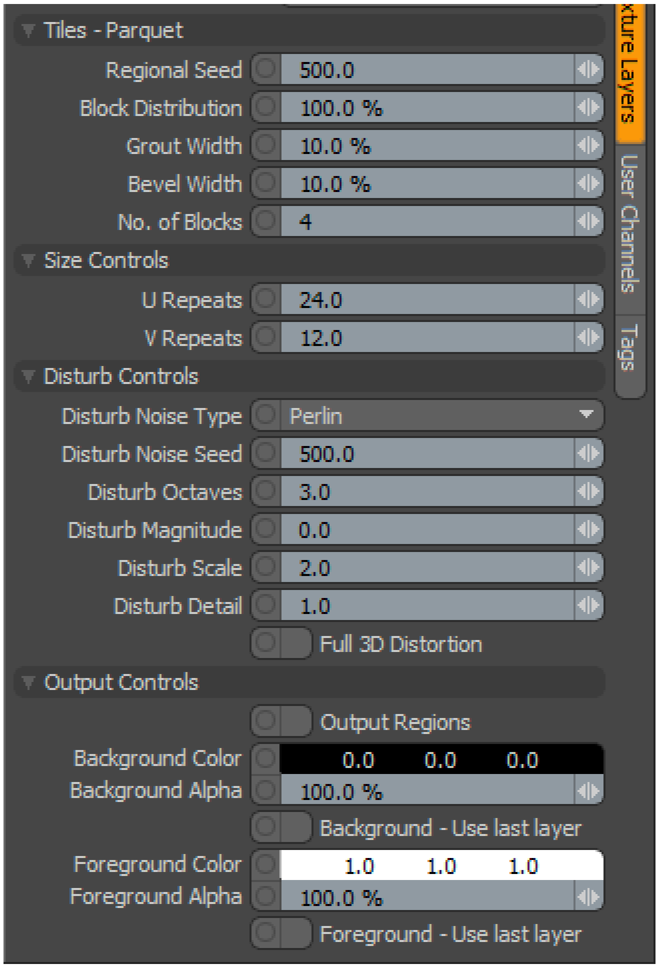
Projection Type: Solid

Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Parquet Properties



Tiles - Parquet
Regional Seed 500.0
Block Distribution 100.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
No. of Blocks 4
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

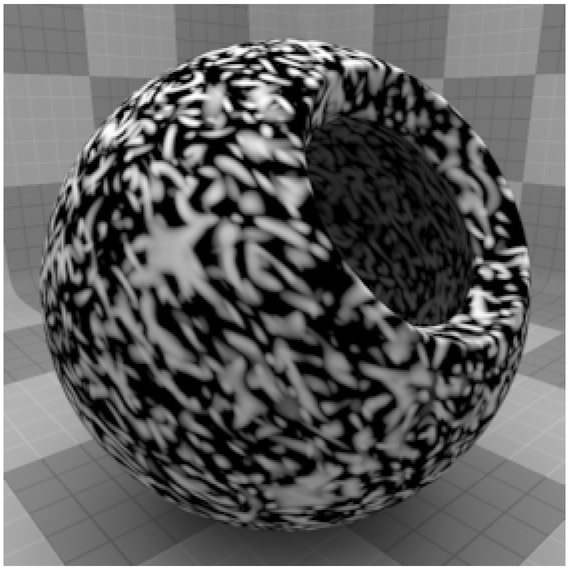
Tiles - Parquet	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values.

	Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Scale Dispersion	Specifies how many of the bump sections are actually drawn.
Block Distribution	Determines the percentage of the surface that is covered by tiles.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Number Of Blocks	Controls the number of blocks in each parquet tile.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.

Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.

Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

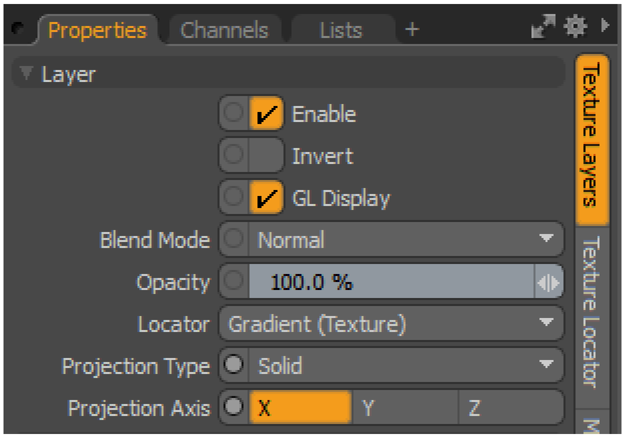
Pastella



The **Pastella** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Pastella** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture produces a pleasant, noisy sine wave effect. By specifying the seed parameter, you can create many variations.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

> 

Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

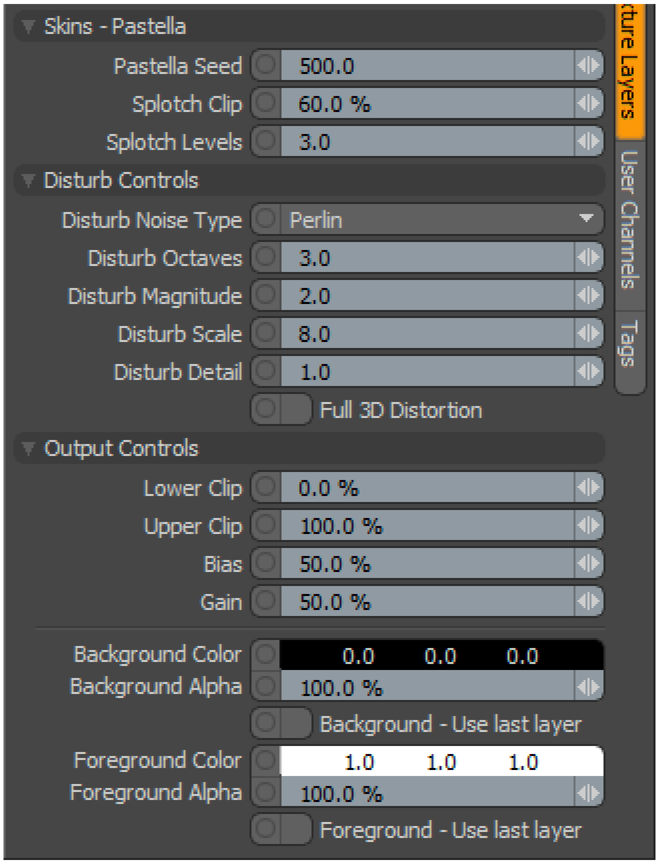
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Pastella Properties



Skins - Pastella

Pastella Seed 500.0

Splotch Clip 60.0 %

Splotch Levels 3.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 2.0

Disturb Scale 8.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

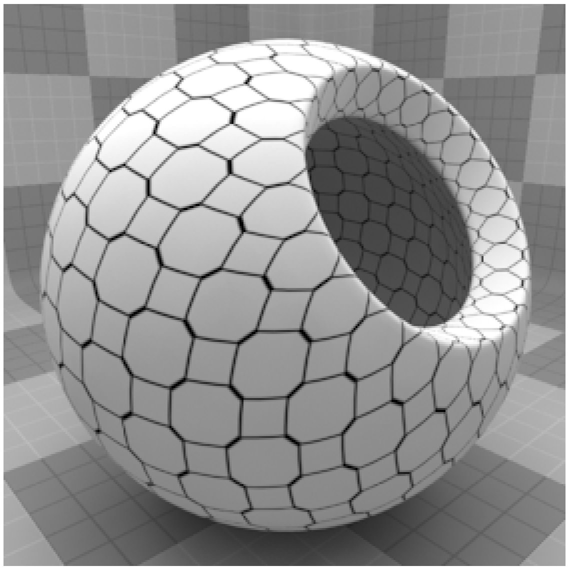
Foreground - Use last layer

Skins - Pastella	
Pastella Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Splotch Clip	Determines where the texture shading transitions to the background color. The higher the value is, the quicker the transition occurs.
Splotch Levels	Determines the number of layers of the texture. Additional levels add more detail and, in turn, also lighten the overall look of the texture.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise	Specifies the look of the texture's distortion and provides several noise function

Type	types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau

	around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Paving

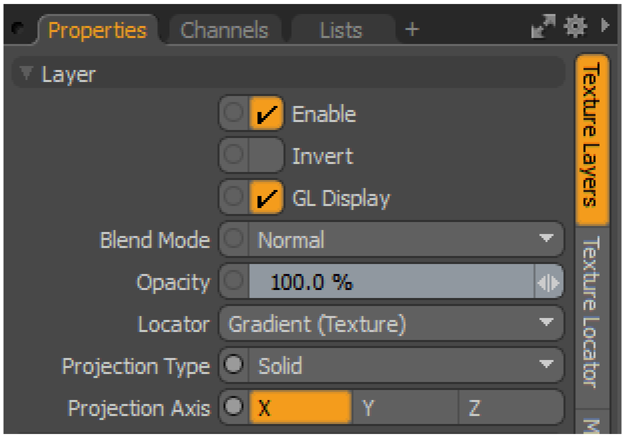


The **Paving** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Paving** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a tile pattern with adjustable shapes, variable grout, and tile edges that is similar to paving stones. You can stagger the tiles and remove random tiles.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

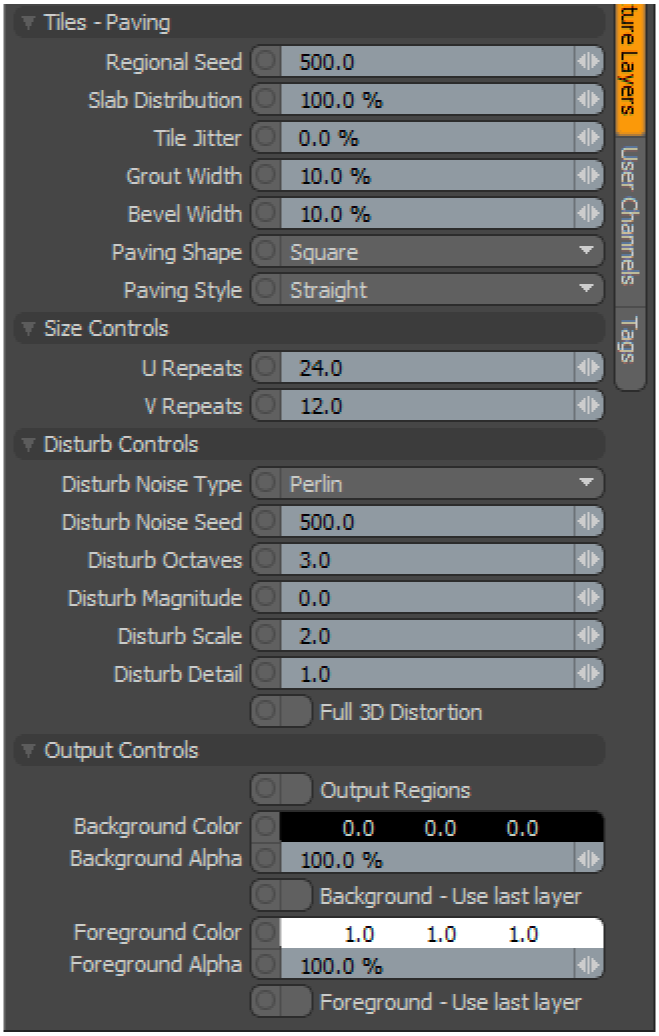
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of

	toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Paving Properties



Tiles - Paving

Regional Seed 500.0

Slab Distribution 100.0 %

Tile Jitter 0.0 %

Grout Width 10.0 %

Bevel Width 10.0 %

Paving Shape Square

Paving Style Straight

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

Foreground - Use last layer

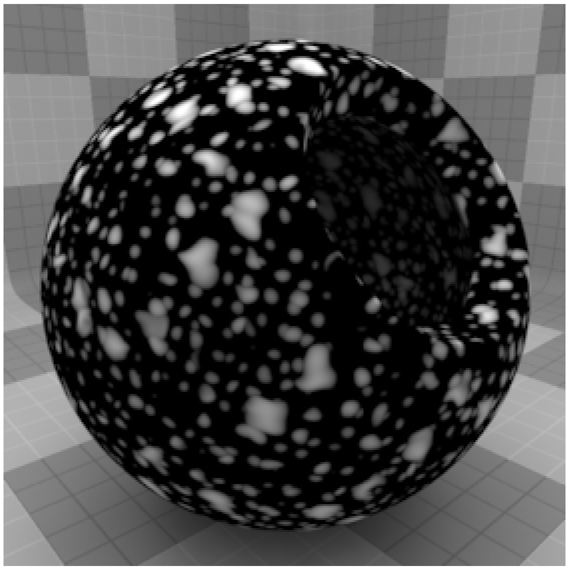
Tiles - Paving	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Slab Distribution	Determines the percentage of the surface that is covered by tiles.
Tile Jitter	Determines how randomly placed the tiles are. When you specify a low value, Modo places the tiles a bit more uniformly. The placement becomes more random as you increase the value.

Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Paving Shape	Determines the shape of the paving tiles. The options include: • **Square** • **Hexagon** • **Octagon** • **Triangle** • **Wedges**
Paving Style	Determines the layout of the paving tiles. The options include: • **Straight** • **Crazy** • **Ornate** • **Triangle** • **Uneven**
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Pebbles

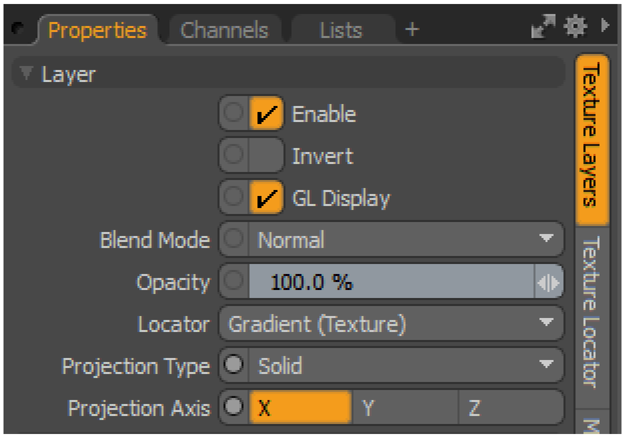


The **Pebbles** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Pebbles** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture shows a random selection of differently sized pebbles (or small stones).

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

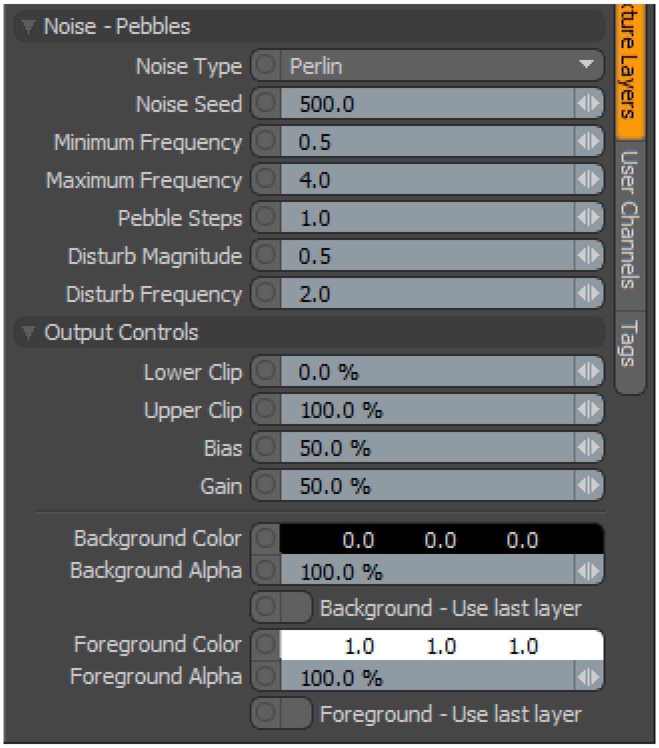
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Pebbles Properties



Noise - Pebbles

Noise Type: Perlin
Noise Seed: 500.0
Minimum Frequency: 0.5
Maximum Frequency: 4.0
Pebble Steps: 1.0
Disturb Magnitude: 0.5
Disturb Frequency: 2.0

Output Controls

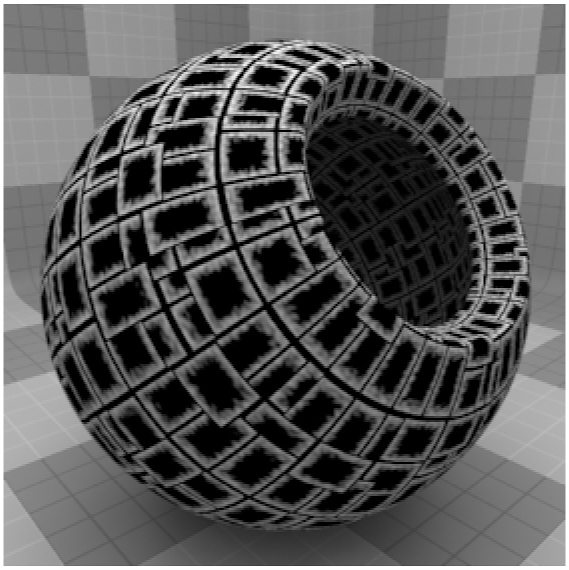
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Pebbles	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice**

	• **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Minimum Frequency	Controls the minimum frequency of the pebbles. Increasing this value decreases the number of large pebbles.
Maximum Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more small pebbles with the foreground color.
Pebble Steps	Controls the steps between the maximum and minimum frequency of the noise. Increasing this value creates more spacing between the background and foreground colors.
Disturb Magnitude	Controls the magnitude of the noise used the disturb the foreground color. Increasing this value creates a texture with more swirl.
Disturb Frequency	Controls the frequency of noise used the disturb the foreground color. Decreasing this value creates a texture with a more rounded appearance.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles.

	This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Peel



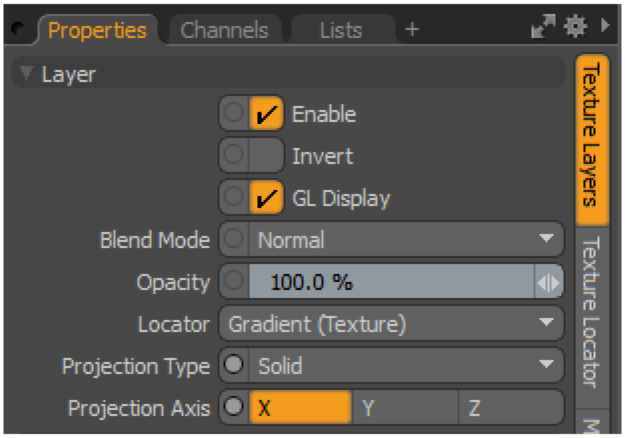
The **Peel** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Peel** texture can be addressed by its two zones:

the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of rectangular panels with a peeling, paint-like texture applied.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



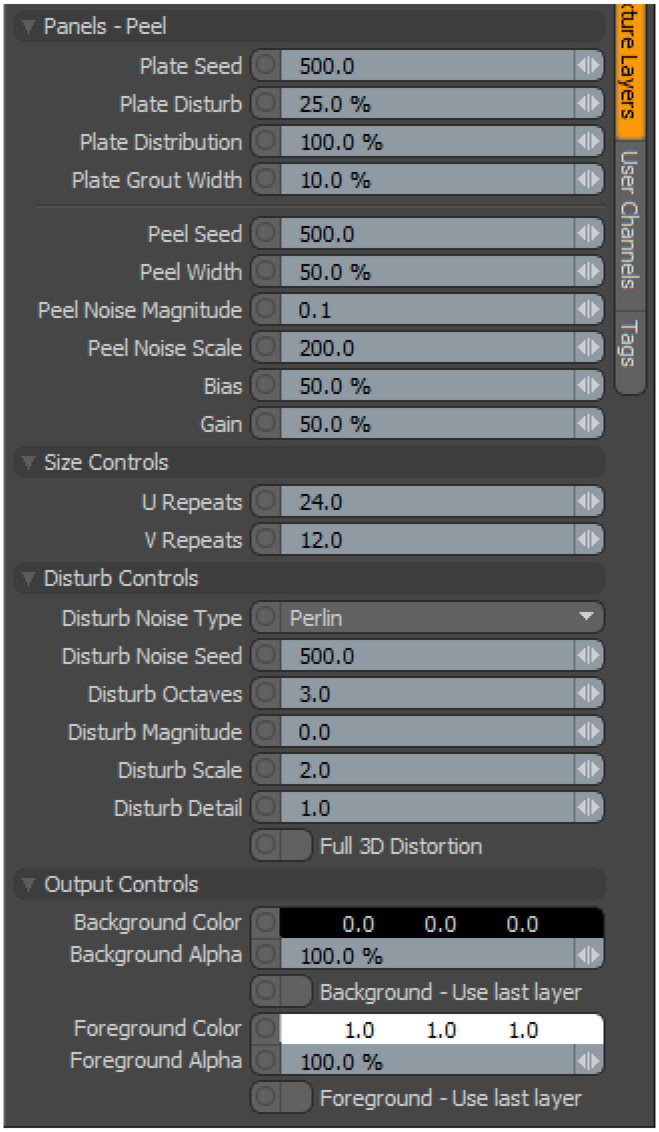
Layer

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Peel Properties

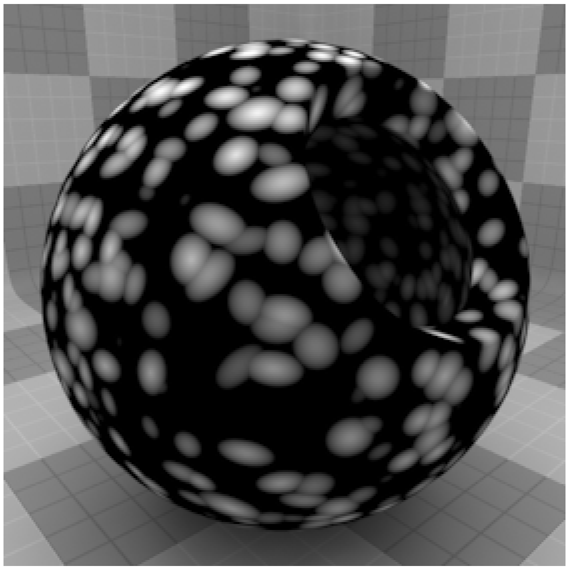

Panels - Peel
Plate Seed 500.0
Plate Disturb 25.0 %
Plate Distribution 100.0 %
Plate Grout Width 10.0 %
Peel Seed 500.0
Peel Width 50.0 %
Peel Noise Magnitude 0.1
Peel Noise Scale 200.0
Bias 50.0 %
Gain 50.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Panels - Peel	
Plate Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Plate Disturb	Determines the amount of distortion among the plates (or panels). When you set this to a low value, the plates are very uniform in size. As you increase the value, the panel sizes vary more.
Plate Distribution	Determines the percentage of the surface that is covered by the plates (or panels).
Plate Grout Width	Determines the percentage of the peel that shows as defined from the outer edge of the plate (or panel).
Peel Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Peel Width	Determines the strength of the distortion effect on the base texture.
Peel Noise Magnitude	Determines the strength of the distortion effect for the peeling.
Peel Noise Scale	Determines the scale (or frequency) of the distortion effect for the peeling.
Bias	Determines the brightness of the overall texture.
Gain	Controls the gain (or contrast) of the overall texture.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin**

	• **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.

Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Peened

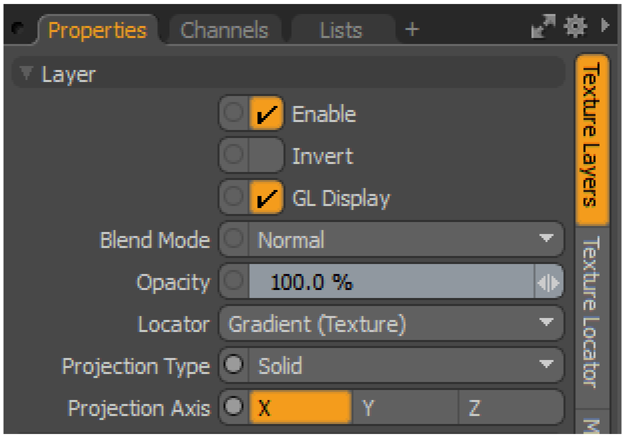


The **Peened** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Peened** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This textures provides a peened- or hammered-looking surface as seen on finished metals. The texture applies a series of tightly packed and overlapping spherical dents to the surface. By adding color to the dents, you can create texture that seems like an antique finish.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

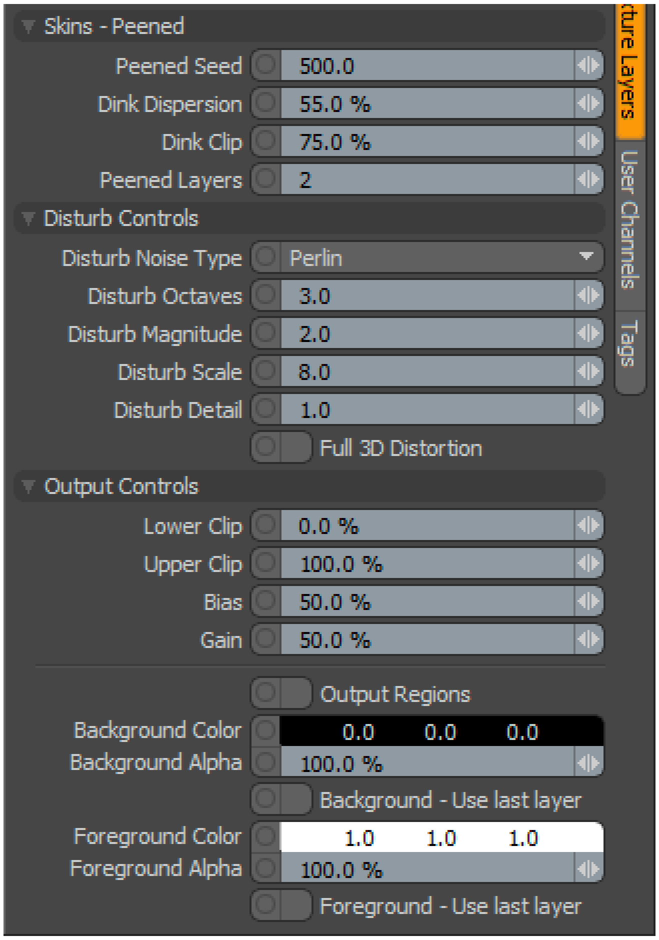
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Peened Properties

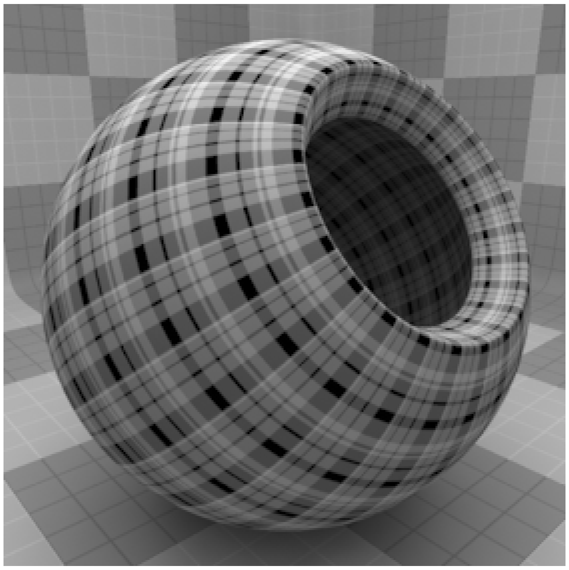


Skins - Peened
Peened Seed 500.0
Dink Dispersion 55.0 %
Dink Clip 75.0 %
Peened Layers 2
Disturb Controls
Disturb Noise Type Perlin
Disturb Octaves 3.0
Disturb Magnitude 2.0
Disturb Scale 8.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Skins - Peened	
Peened Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Dink Dispersion	Determines how many dinks (or dents) are actually drawn.
Dink Clip	Determines where the texture shading transitions to the background color. The higher the value is, the quicker the transition occurs.
Peened Layers	Determines the number of layers of the texture. Additional levels add additional details, but they also lighten the overall look of the texture.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes

	longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Plaid



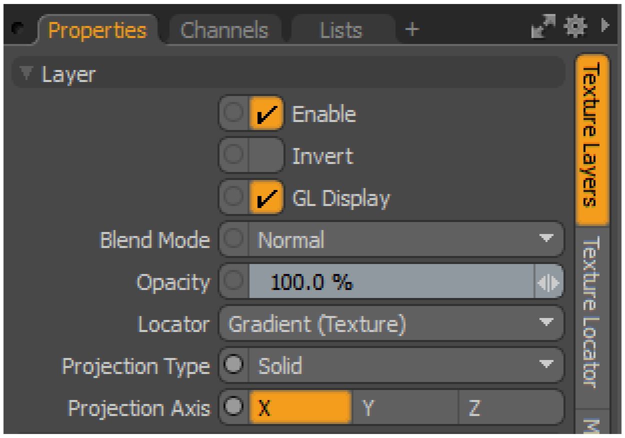
The **Plaid** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Plaid** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a standard plaid effect with an overlapping stripe pattern. This texture works best when you apply it to a black object.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

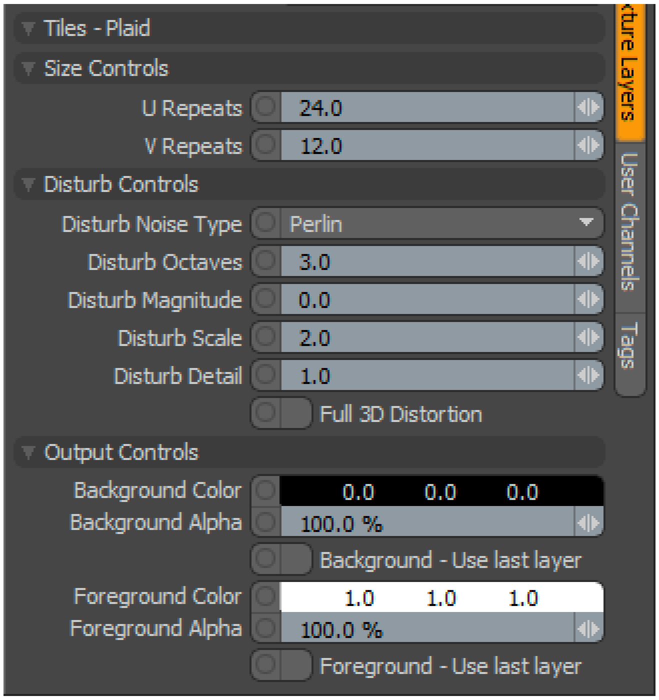
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Plaid Properties



Tiles - Plaid

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

○ Full 3D Distortion

Output Controls

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

○ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

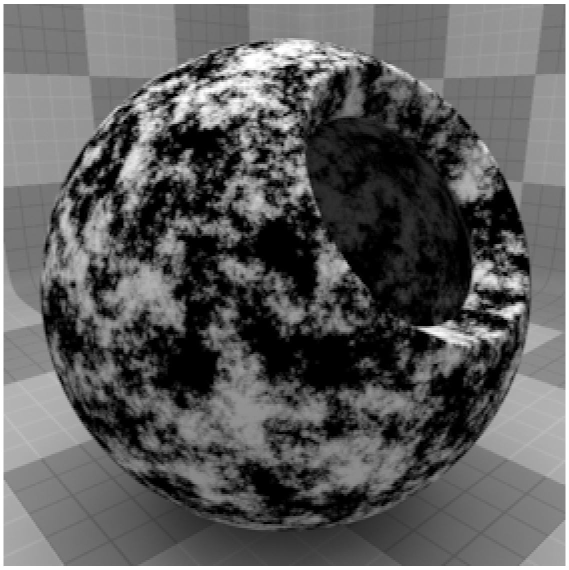
○ Foreground - Use last layer

Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Planet

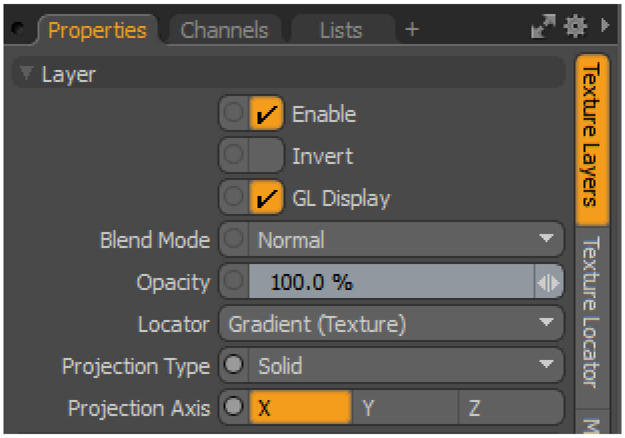


The **Planet** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Planet** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a pattern similar to the atmosphere of an Earth-like planet.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

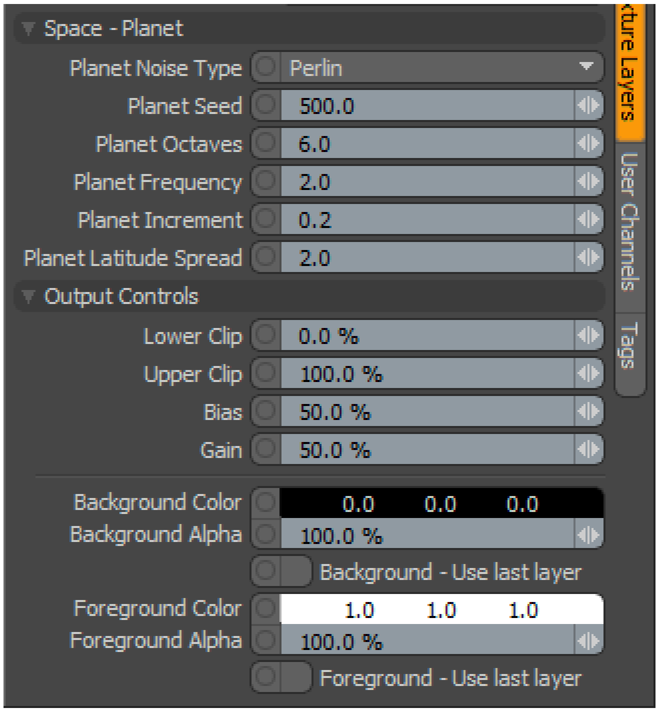
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Planet Properties



Space - Planet

Planet Noise Type: Perlin

Planet Seed: 500.0

Planet Octaves: 6.0

Planet Frequency: 2.0

Planet Increment: 0.2

Planet Latitude Spread: 2.0

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

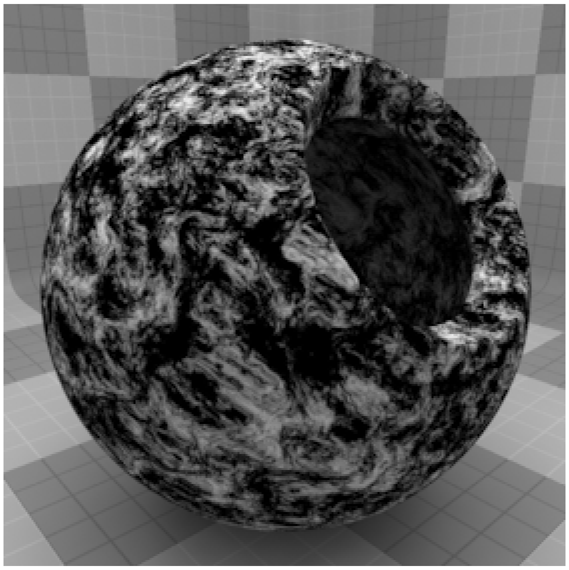
Foreground - Use last layer

Space - Planet	
Planet Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Planet Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Planet Octaves	Defines the number of iterations used to create the texture. As you increase this number, Modo generates turbulence by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is smooth, and it increases in detail as you increase the number of octaves.
Planet Frequency	Controls the overall frequency or scale of the texture. Increasing this value creates more turbulence to stir up the pattern.
Planet Increment	Controls the scaling of each successive iteration. When set at a low value, the turbulence is tight and crinkled. It becomes increasingly smoother as the you increase the value.
Planet Latitude Spread	Controls the scaling between each of the polar regions.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control

	the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Planet Clouds

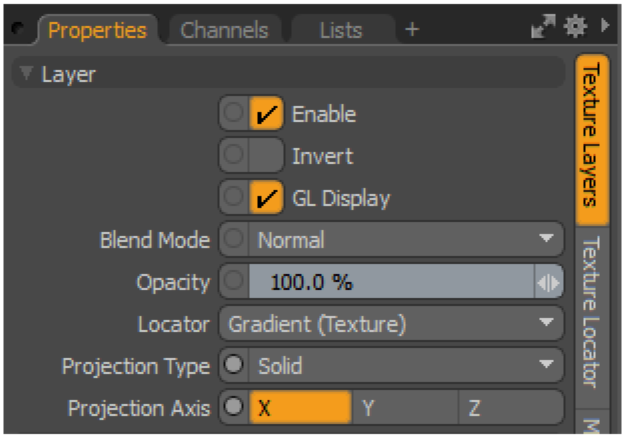


The **Planet Clouds** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Planet Clouds** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a pattern similar to the atmosphere of an Earth-like planet. It differs from the **Planet** texture by providing more of a swirling-cloud effect.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

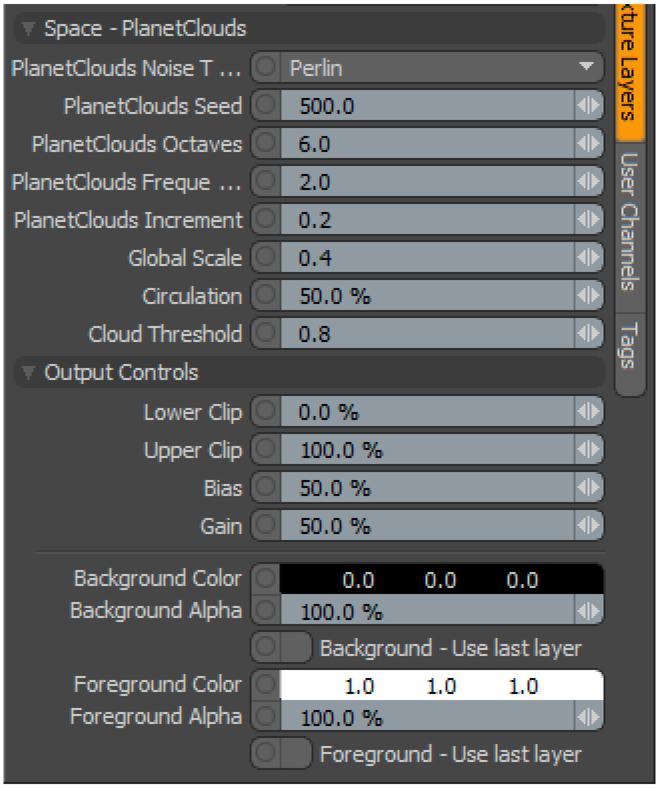
Projection Type: Solid

Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Planet Clouds Properties



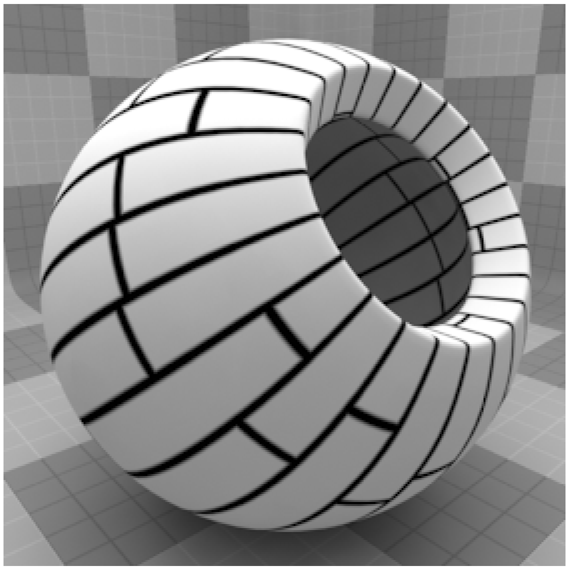
Space - PlanetClouds
PlanetClouds Noise T ... Perlin
PlanetClouds Seed 500.0
PlanetClouds Octaves 6.0
PlanetClouds Freque ... 2.0
PlanetClouds Increment 0.2
Global Scale 0.4
Circulation 50.0 %
Cloud Threshold 0.8
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Space - Planet Clouds	
Planet Clouds Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin**

	• **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Planet Clouds Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Planet Clouds Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates turbulence by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is less to create a smooth texture, but the turbulence detail increases as you increase the number of octaves.
Planet Clouds Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir up the pattern.
Planet Clouds Increment	Controls the scaling of each successive iteration. When you set the value low, the turbulence is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Global Scale	Controls how much the global circulation disturbs the clouds. When you set this at a low value, the clouds are long and sweeping. They become increasingly broken up and smaller as you increase the value.
Circulation	Controls the global circulation of the pattern. Increasing this value causes the clouds have more swirl.
Cloud Threshold	Determines where the texture shading transitions to the background color. The higher the value is, the smoother the transition seems.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.

Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Planks

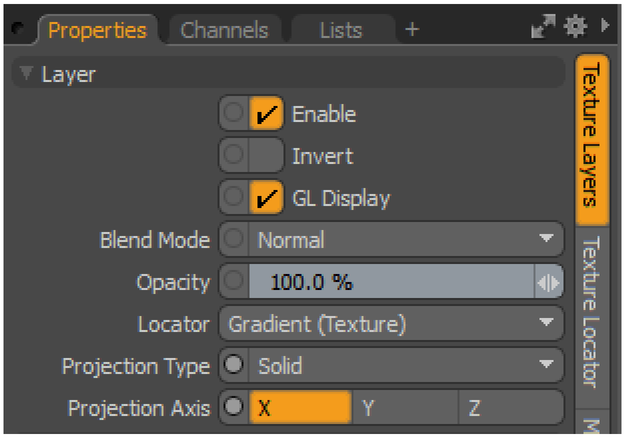


The **Planks** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Planks** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a floor board- or plank-type of texture. You can adjust several attributes of the texture including how much the planks are staggered, the direction of the planks, the length of the planks, and the plank edges.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties



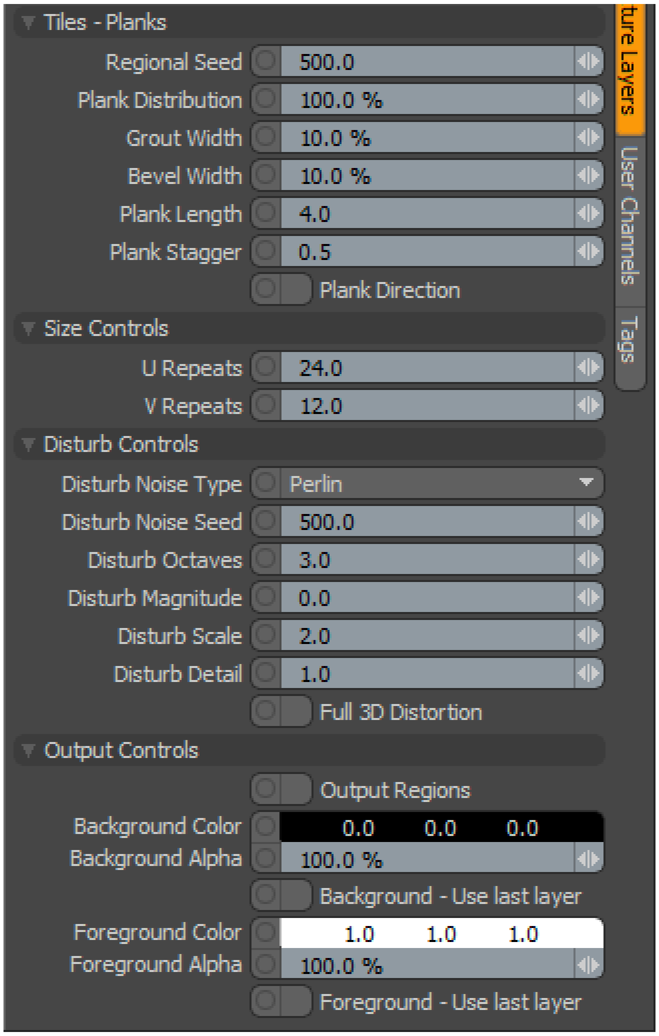
Layer
☑ Enable
☐ Invert
☑ GL Display
Blend Mode Normal
Opacity 100.0 %
Locator Gradient (Texture)
Projection Type Solid
Projection Axis X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Planks Properties



Tiles - Planks

Regional Seed 500.0
Plank Distribution 100.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Plank Length 4.0
Plank Stagger 0.5
Plank Direction

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

Output Controls

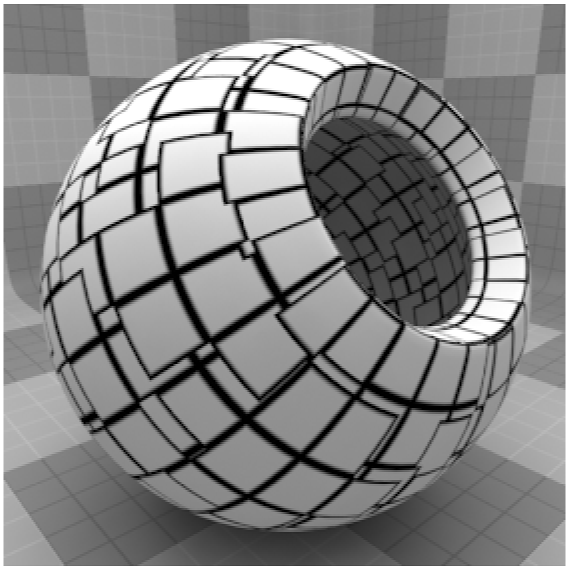
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Planks	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Plank Distribution	Determines the percentage of the surface that is covered by planks.
Grout Width	Determines the amount of grout between the planks.
Bevel Width	Determines the width of the bevel on the planks.
Plank Length	Determines the length of the individual planks.
Plank Stagger	Controls the randomization of the planks' stagger. When you set this at a low value, the planks line up horizontally. The rows become more staggered as the you increase the value.
Plank Direction	Switches the planks' orientation between vertical and horizontal.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This,

	in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use	When enabled, makes the **Background Color** area completely transparent to reveal

Last Layer	the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

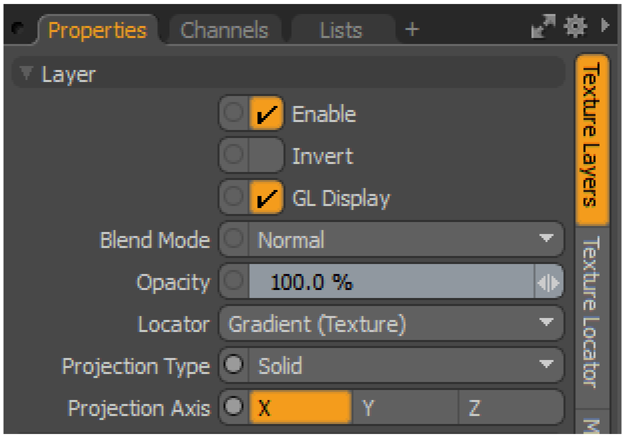
Plates



The **Plates** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Plates** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of rectangular plates, which can be used to create floor tiles or metal armor.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

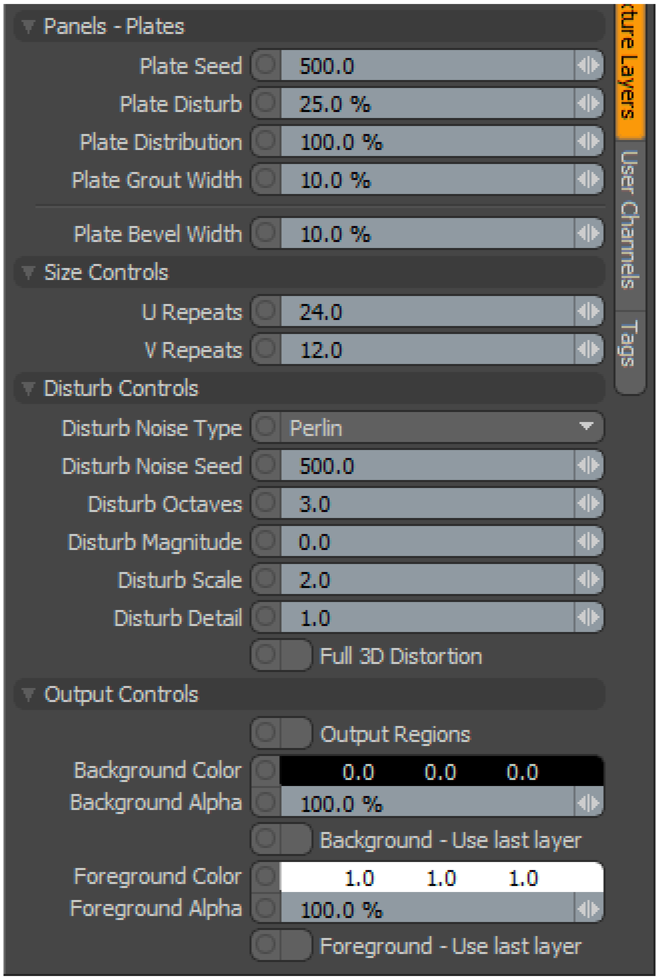
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Plates Properties



Panels - Plates

Plate Seed 500.0

Plate Disturb 25.0 %

Plate Distribution 100.0 %

Plate Grout Width 10.0 %

Plate Bevel Width 10.0 %

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

○ Full 3D Distortion

Output Controls

○ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

○ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

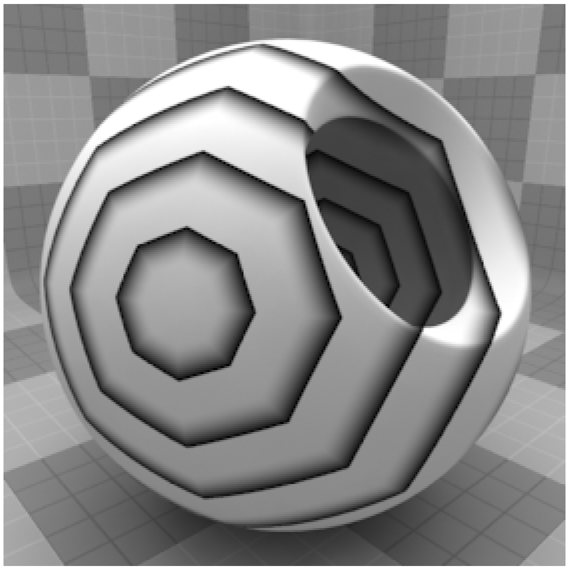
○ Foreground - Use last layer

Panels - Plates	
Plate Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Plate Disturb	Determines the amount of distortion among the plates. When you set this at a low

	value, the plates are very uniform. The plates sizes have more variation as you increase the value.
Plate Distribution	Determines the percentage of the surface that is covered by plates.
Plate Grout Width	Determines the amount of grout between the individual plates.
Plate Bevel Width	Determines the length of the bevel on the indviudal plates as measured from the outer edge to the center of the plate.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes

	longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

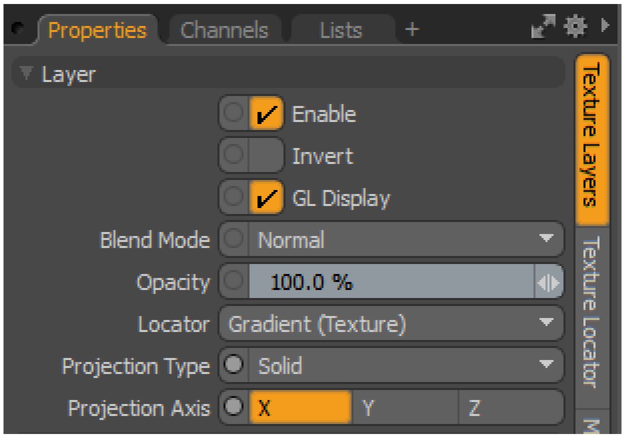
Polygon



The **Polygon** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Polygon** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a series of adjustable polygonal patterns.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

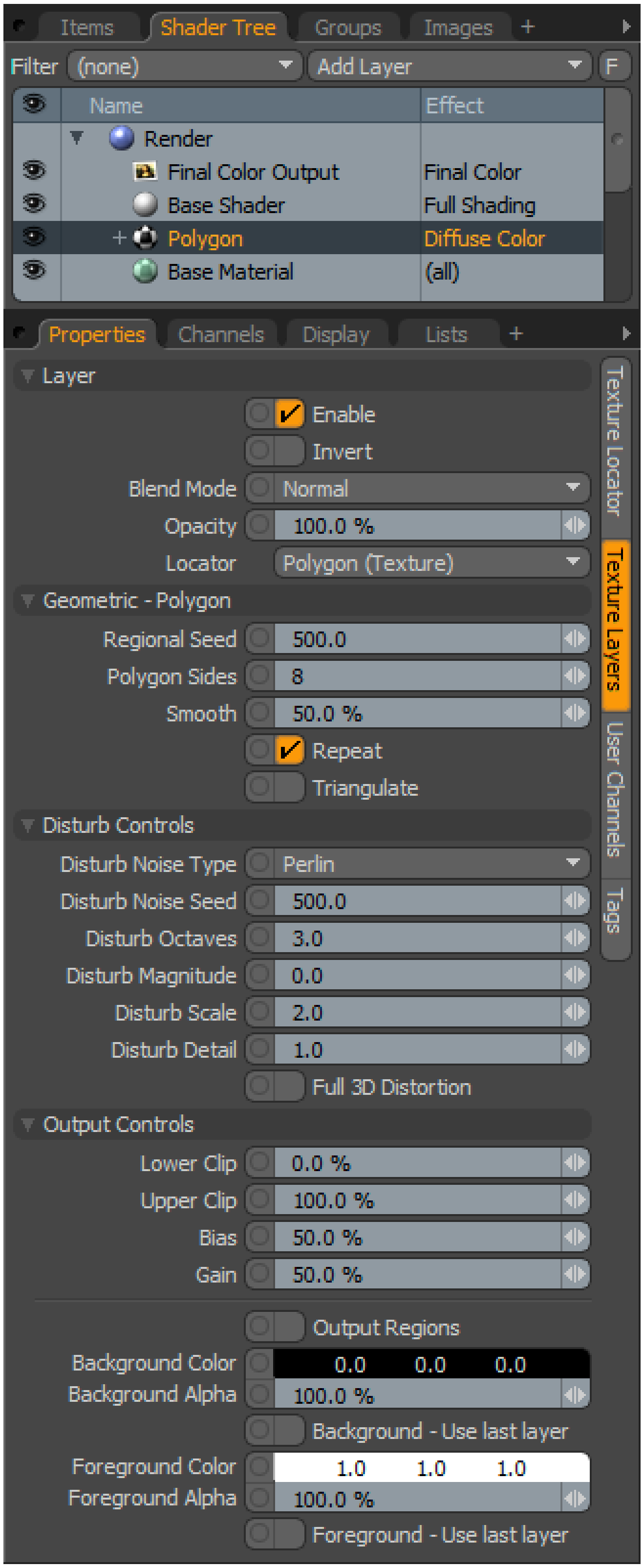


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Polygon Properties


Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer
F
Properties
Channels
Display
Lists
+
Texture Locator
Texture Layers
User Channels
Tags

Name	Effect
Render	
Final Color Output	Final Color
Base Shader	Full Shading
+ Polygon	Diffuse Color
Base Material	(all)

Layer

☑ Enable
☐ Invert
Blend Mode: Normal
Opacity: 100.0 %
Locator: Polygon (Texture)

Geometric - Polygon

Regional Seed: 500.0
Polygon Sides: 8
Smooth: 50.0 %
☑ Repeat
☐ Triangulate

Disturb Controls

Disturb Noise Type: Perlin
Disturb Noise Seed: 500.0
Disturb Octaves: 3.0
Disturb Magnitude: 0.0
Disturb Scale: 2.0
Disturb Detail: 1.0
☐ Full 3D Distortion

Output Controls

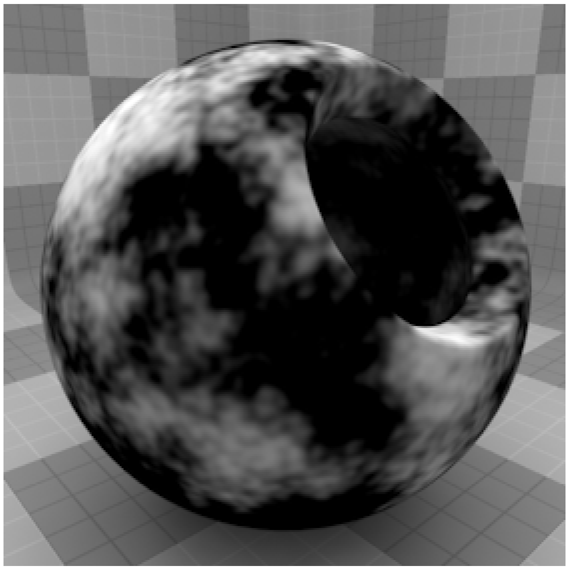
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
☐ Output Regions
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Geometric - Polygon	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Polygon Sides	Determines the number of sides on the polygon(s).
Smooth	Determines the amount of smoothing around the edges of the polygon. Higher values create smoother edges.
Repeat	When enabled, causes the texture to repeat indefinitely.
Triangulate	When enabled, causes the pattern to have a triangle-based edging pattern, instead of the default sawtooth blend.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you

	increase the number of octaves.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Controls the overall frequency (or scale) of the texture. Increasing this value creates more distortion to stir up the pattern.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Puffy Clouds

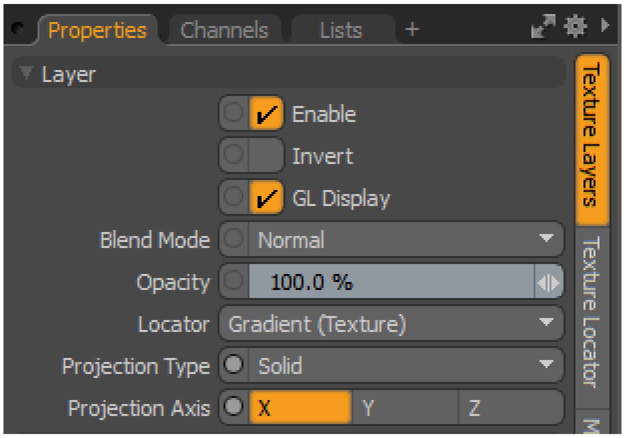


The **Puffy Clouds** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Puffy Clouds** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a noise pattern that you can use for a light, puffy-cloud effect.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

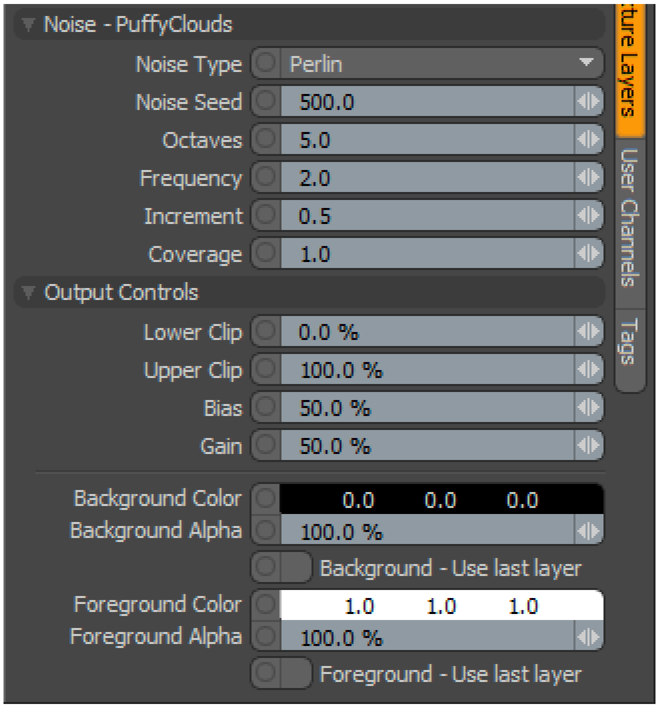
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Puffy Clouds Properties



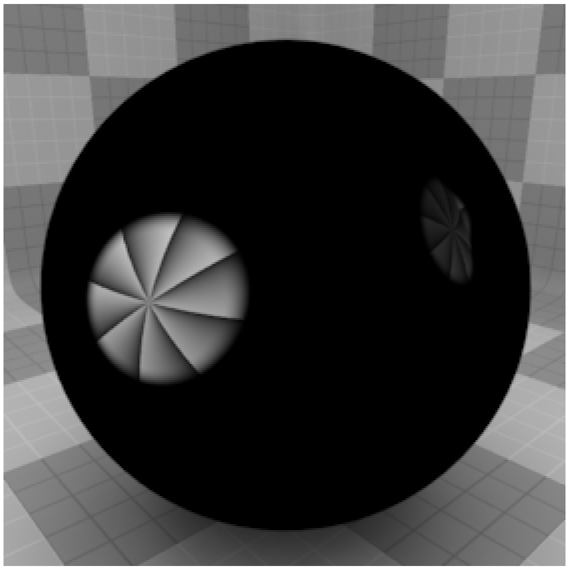
Noise - PuffyClouds
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 5.0
Frequency: 2.0
Increment: 0.5
Coverage: 1.0
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Puffy Clouds	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir up the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the turbulence is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Coverage	Controls the amount of the foreground color. Decreasing this value increases the overall coverage of the foreground color.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control

	the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Radial

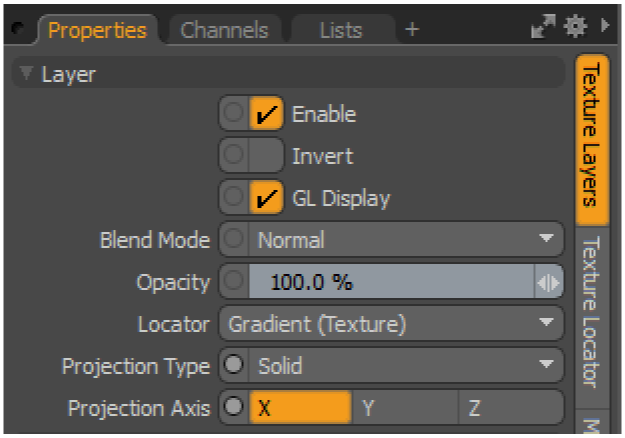


The **Radial** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Radial** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a customizable radial pattern.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

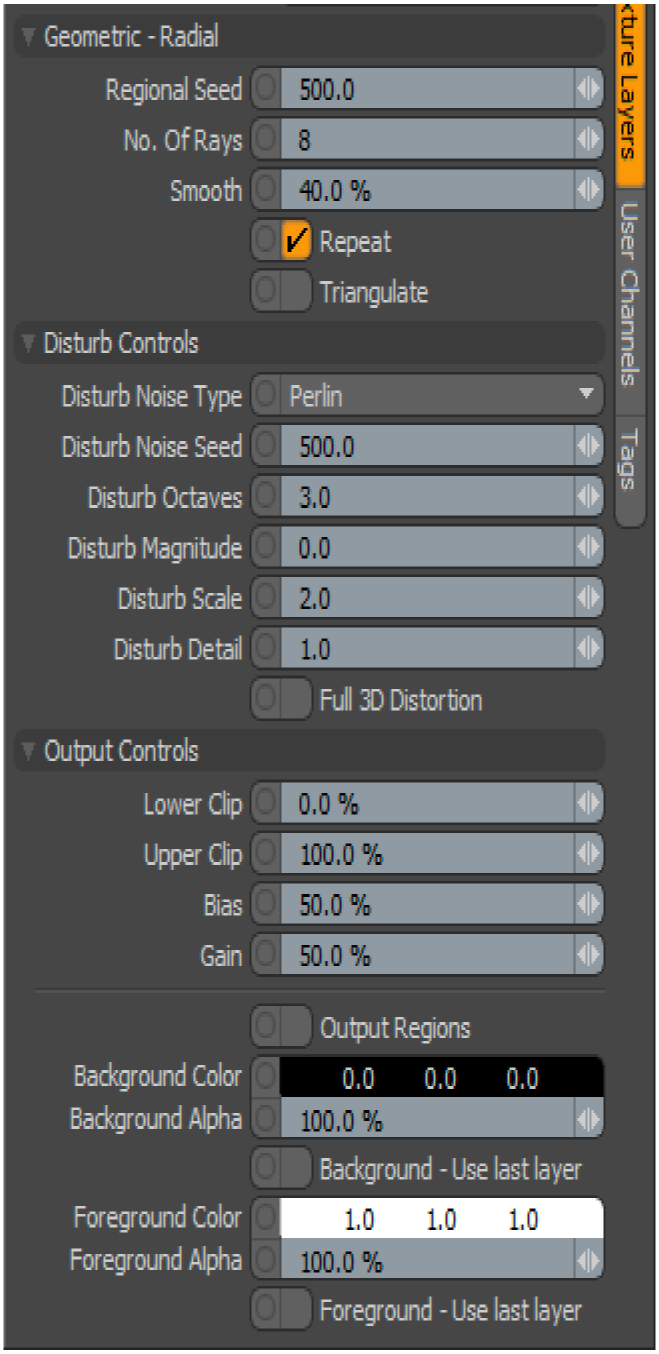
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture**

	Locator that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Radial Properties



Geometric - Radial

Regional Seed 500.0

No. Of Rays 8

Smooth 40.0 %

☑ Repeat

☐ Triangulate

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

☐ Foreground - Use last layer

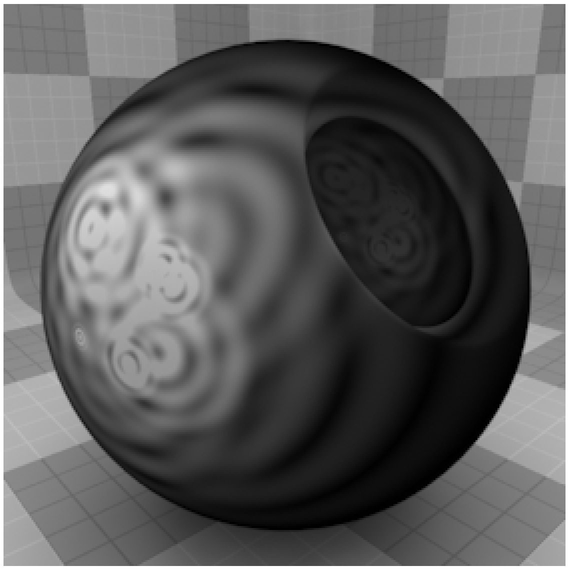
Geometric - Radial	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in

	changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
No. Of Rays	Determines the number of rays in the radial pattern.
Smooth	Determines the amount of smoothing around the edges of the radial pattern. Higher values create smoother edges.
Repeat	When enabled, causes the texture to repeat indefinitely.
Triangulate	When enabled, causes the pattern to have a triangle-based edging pattern instead of the default sawtooth blend.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.

Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.

Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Rain

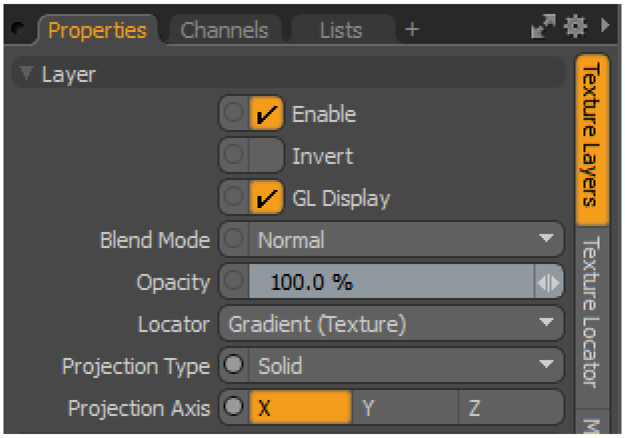


The **Rain** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Rain** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series random concentric circles that are similar to when raindrops fall into a body of water. All the parameters are keyed to the time function for easy animation.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

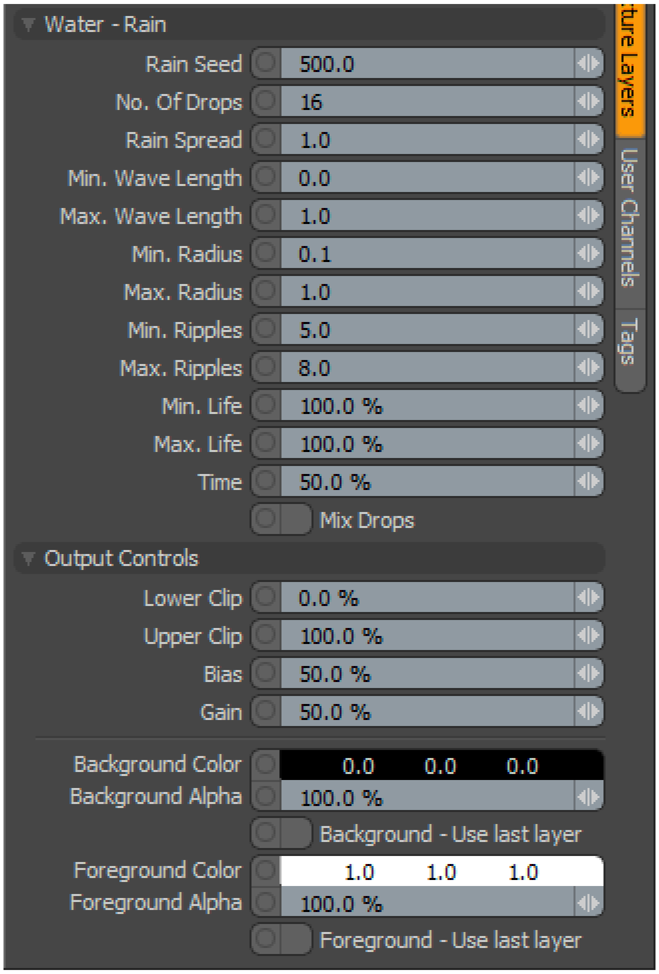
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Rain Properties



Water - Rain

Rain Seed 500.0

No. Of Drops 16

Rain Spread 1.0

Min. Wave Length 0.0

Max. Wave Length 1.0

Min. Radius 0.1

Max. Radius 1.0

Min. Ripples 5.0

Max. Ripples 8.0

Min. Life 100.0 %

Max. Life 100.0 %

Time 50.0 %

Mix Drops

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

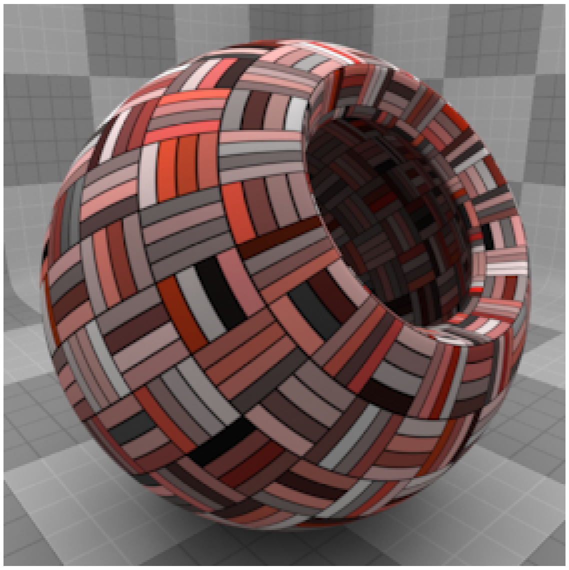
Foreground - Use last layer

Water - Rain	
Rain Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
No. Of Drops	Determines the number of rain drops in the pattern.
Rain Spread	Determines the spread of the rain drops. The drops move farther from one another as you increase the value.
Min. Wave Length	Determines the wavelength of the ripples at the start of the animation (when **Time** is set to **0%**).

Max. Wave Length	Determines the wavelength of the ripples at the end of the animation (when **Time** is set to **100%**).
Min. Radius	Determines the radius of the ripples at the start of the animation (when **Time** is set to **0%**).
Max. Radius	Determines the radius of the ripples at the end of the animation (when **Time** is set to **100%**).
Min. Ripples	Determines the number of ripples at the start of the animation (when **Time** is set to **0%**).
Max. Ripples	Determines the number of ripples at the end of the animation (when **Time** is set to **100%**).
Min. Life	Determines the minimum life expectancy of the rain drops.
Max. Life	Determines the maximum life expectancy of the rain drops.
Time	Controls the progress of the animation (based on the other settings) through time. The animation morphs through multiple batches of rain drops as the timer goes from **0%** to **100%**.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles.

	This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Regional HSV

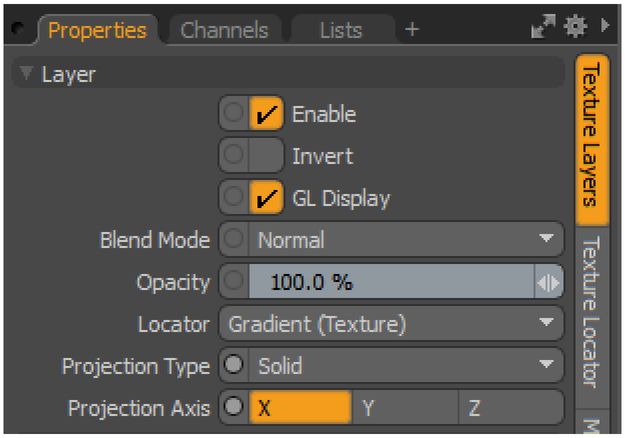


The **Regional HSV** process layer provides you with a way to control variations in e:Modo textures that output regions (such as the Tiles procedural layers). For the **Regional HSV** layer to work properly, you must enable **Output Regions** in the originating procedural texture layer. To use **Regional HSV**, define a layer that supports the **Output Regions** options and enable **Output Regions**. Add the **Regional HSV** layer set to the same effect type directly above the originating layer in the Shader Tree. You can then adjust the minimum and maximum **Hue**, **Saturation**, and **Value** settings to control the amount of variations in the regions.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

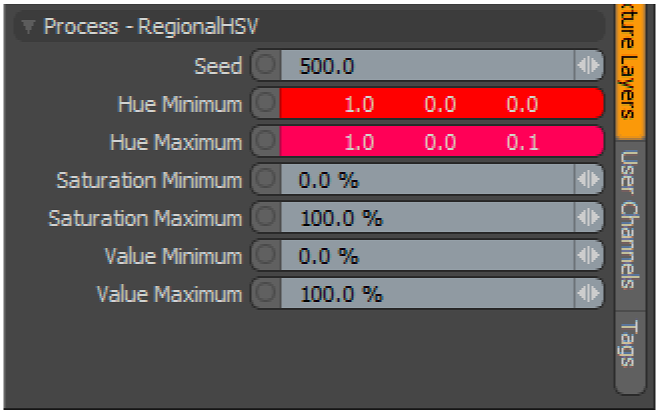
Projection Type: Solid

Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Regional HSV Properties



Process - RegionalHSV

Seed 500.0

Hue Minimum 1.0 0.0 0.0

Hue Maximum 1.0 0.0 0.1

Saturation Minimum 0.0 %

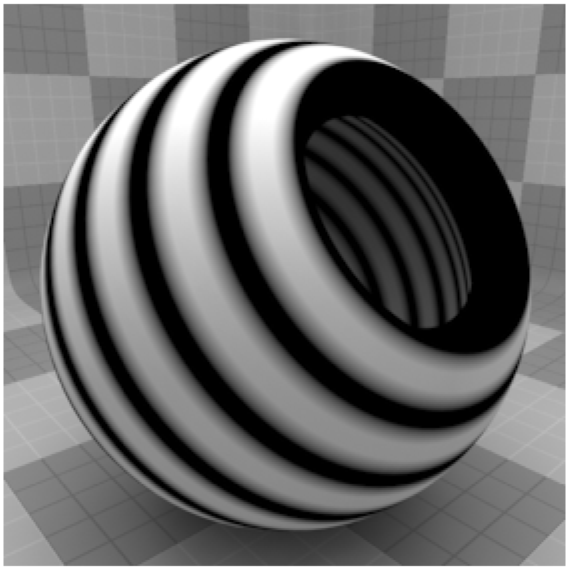
Saturation Maximum 100.0 %

Value Minimum 0.0 %

Value Maximum 100.0 %

Process - Regional HSV	
Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Hue Minimum	Specifies the low-end hue value that Modo uses to mix the final output.
Hue Maximum	Specifies the high-end hue value that Modo uses to mix the final output.
Saturation Minimum	Specifies the low-end saturation value that Modo uses to mix the final output.
Saturation Maximum	Specifies the high-end saturation value that Modo uses to mix the final output.
Value Minimum	Specifies the low-end intensity amount that Modo uses to mix the final output.
Value Maximum	Specifies the high-end intensity amount that Modo uses to mix the final output.

Ribs

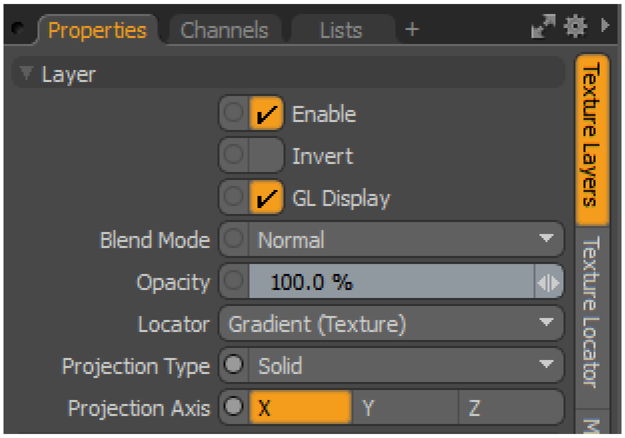


The **Ribs** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Ribs** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a series of straight ribs running along an object. You can modify the number of ribs drawn as well as their location in relation to the object's center line.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

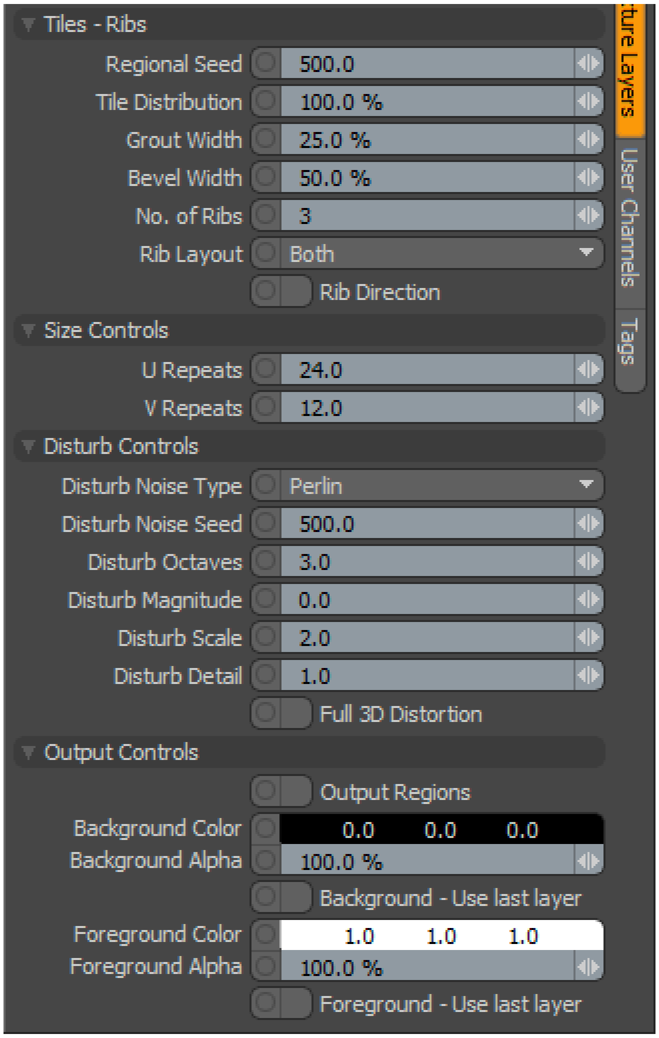
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Ribs Properties



Tiles - Ribs
Regional Seed 500.0
Tile Distribution 100.0 %
Grout Width 25.0 %
Bevel Width 50.0 %
No. of Ribs 3
Rib Layout Both
☐ Rib Direction

Size Controls
U Repeats 24.0
V Repeats 12.0

Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion

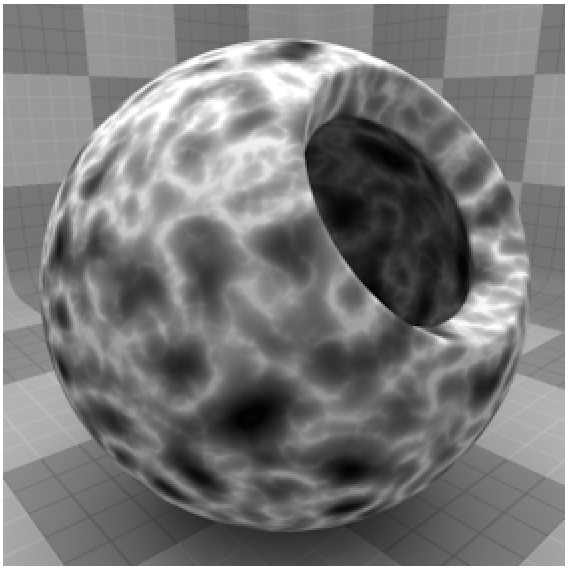
Output Controls
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Option	Description
Tiles - Ribs	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Tile Distribution	Determines the percentage of the surface that is covered by ribs.
Grout Width	Determines the amount of grout between the ribs.
Bevel Width	Determines the width of the bevel on the ribs.
Number of Ribs	Determines the number of ribs drawn in the texture.
Rib Layout	Controls which ribs Modo draws in the texture. These options include: • **Both** • **All Ribs** • **Positive** • **Negative**
Rib Direction	Switches the ribs' orientation between vertical and horizontal.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse**

Option	Description
	• **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward

Option	Description
Color/Value	the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Ridged

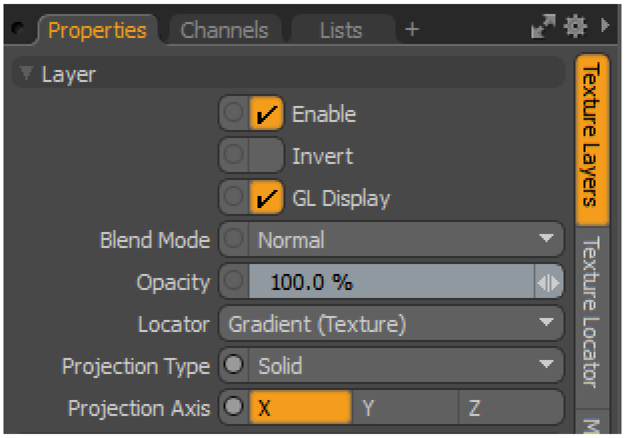


The **Ridged** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Ridged** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides noisy bumps based on Fractal Brownian Motion (FBM). The **Ridged** texture differs from the **FBM** texture in that its peaks are more like ridges.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

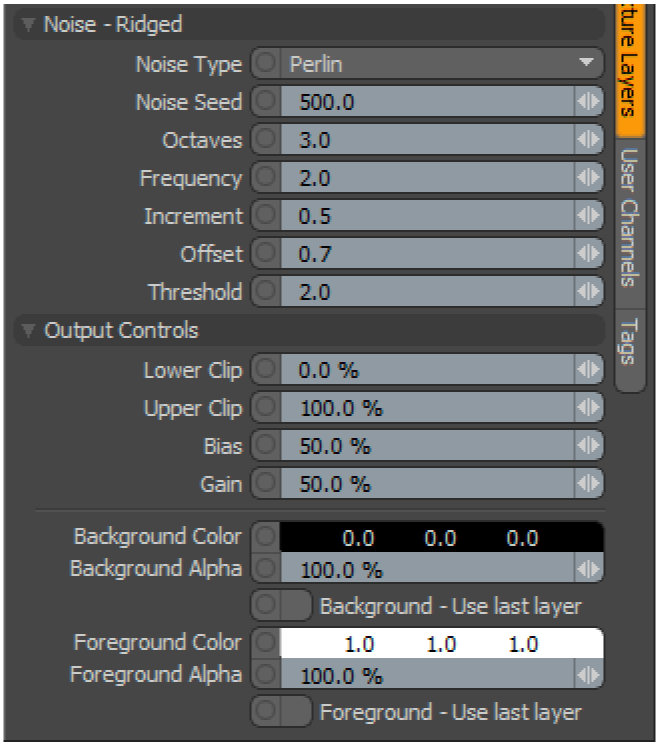


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Ridged Properties



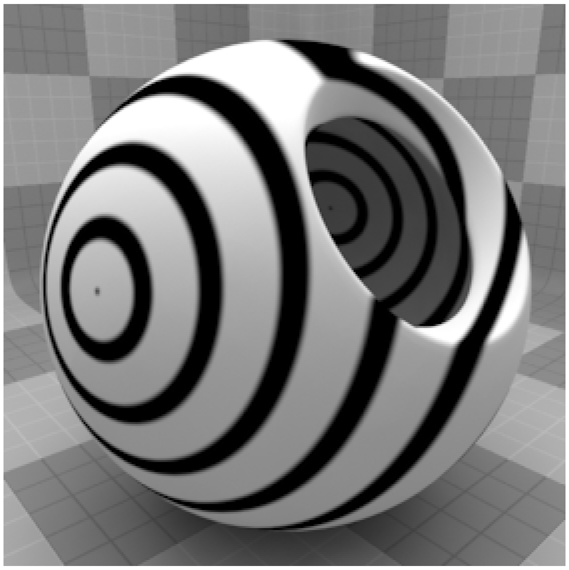
Noise - Ridged
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Frequency: 2.0
Increment: 0.5
Offset: 0.7
Threshold: 2.0
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Ridged	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin**

	• **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Offset	Controls the point at which the texture becomes more detailed.
Threshold	Controls the point at which the valleys of the textures become ridges.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the

	value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Ring

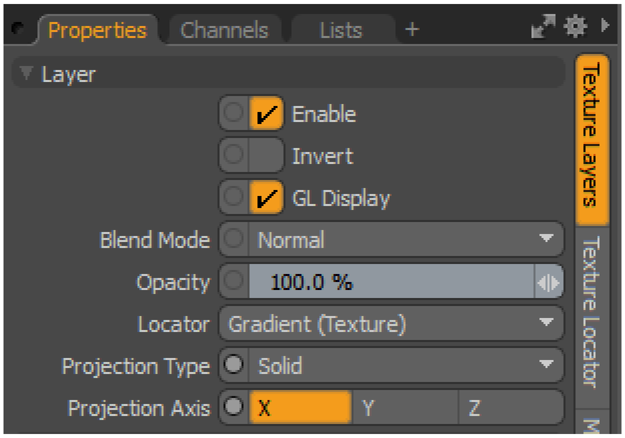


The **Ring** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Ring** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a series of adjustable ring patterns.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

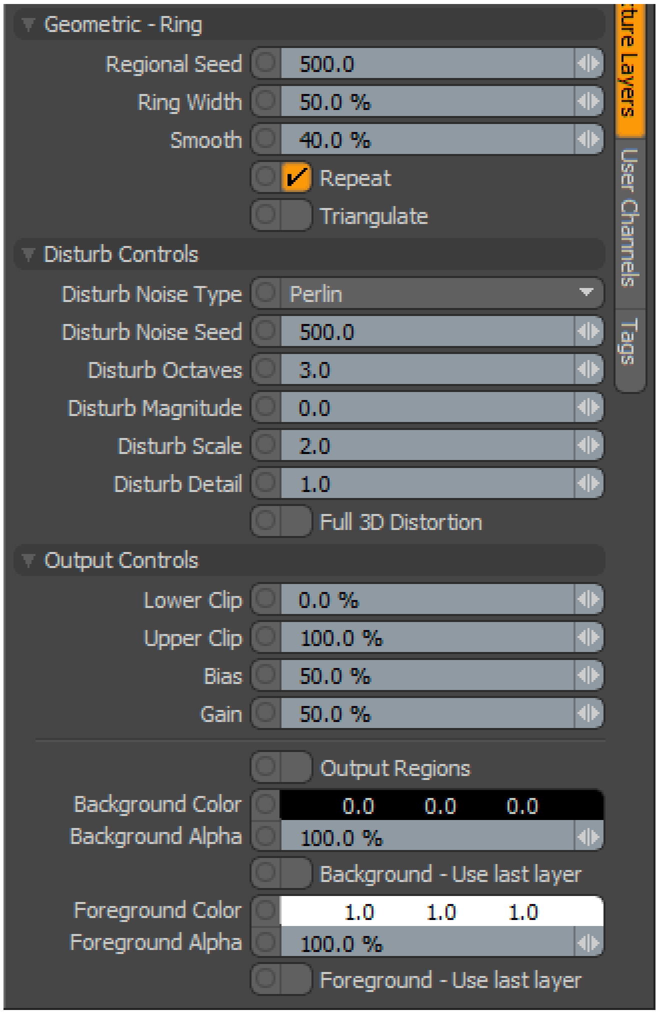
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Ring Properties



Geometric - Ring

Regional Seed 500.0

Ring Width 50.0 %

Smooth 40.0 %

☑ Repeat

☐ Triangulate

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

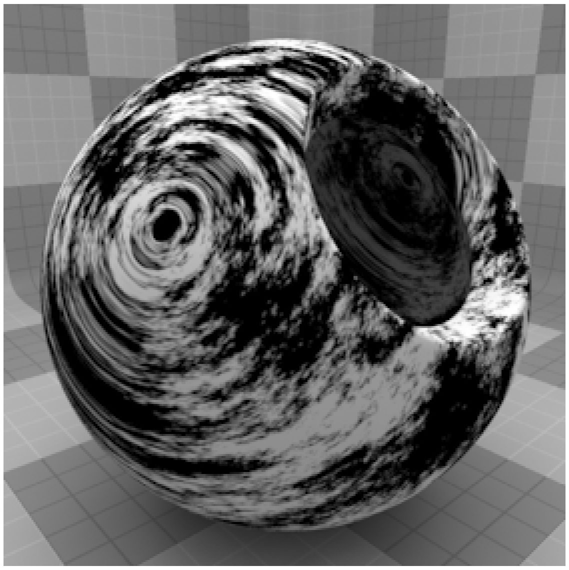
☐ Foreground - Use last layer

Geometric - Ring	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Ring Width	Determines the width of the individual rings.
Smooth	Determines the amount of smoothing around the edges of the rings. Higher values create smoother edges.
Repeat	When enabled, causes the texture to repeat indefinitely.

Triangulate	When enabled, causes the pattern to have a triangle-based edging pattern instead of the default sawtooth blend.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	

Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Rings

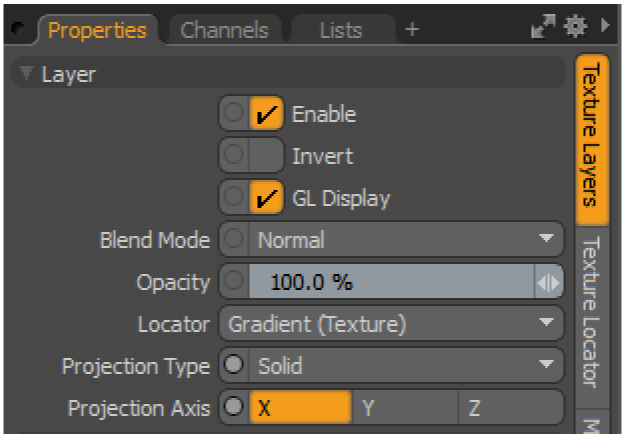


The **Rings** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Rings** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a noisy ring pattern that is similar to the rings around Saturn. You can set a minimum radius for the rings as well as ring scaling and a noise amount.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

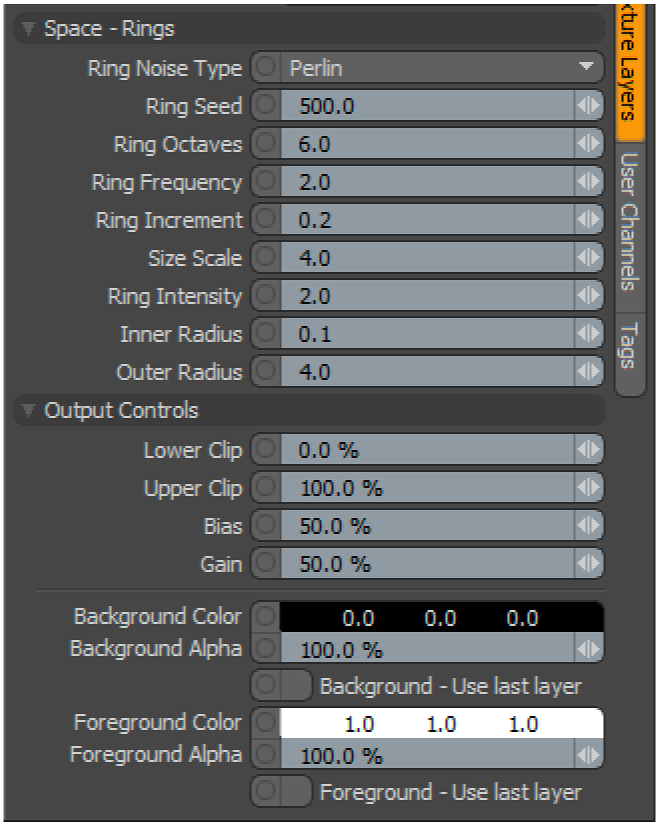
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Rings Properties



Space - Rings

Ring Noise Type: Perlin
Ring Seed: 500.0
Ring Octaves: 6.0
Ring Frequency: 2.0
Ring Increment: 0.2
Size Scale: 4.0
Ring Intensity: 2.0
Inner Radius: 0.1
Outer Radius: 4.0

Output Controls

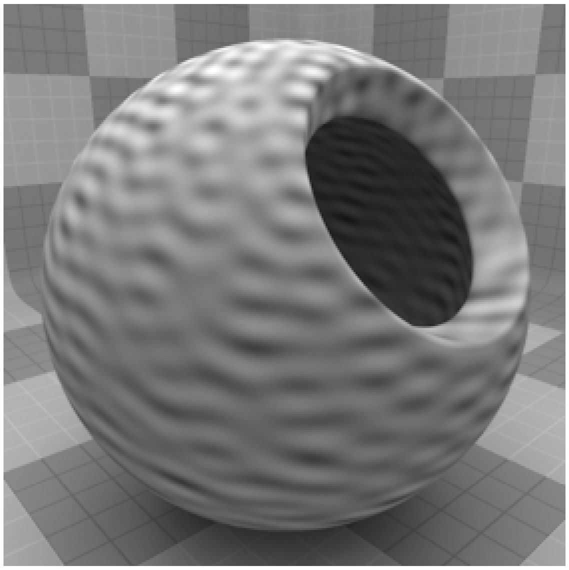
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Space - Rings	
Ring Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Ring Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in

	changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Ring Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Ring Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Ring Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Size Scale	Controls the scaling of the actual rings.
Ring Intensity	Controls the intensity of the rings as they transition to the background color. The higher the value is, the smoother the transition occurs.
Inner Radius	Sets the minimum radius for the inner rings.
Outer Radius	Sets the maximum radius for the outer rings.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.

Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Ripples

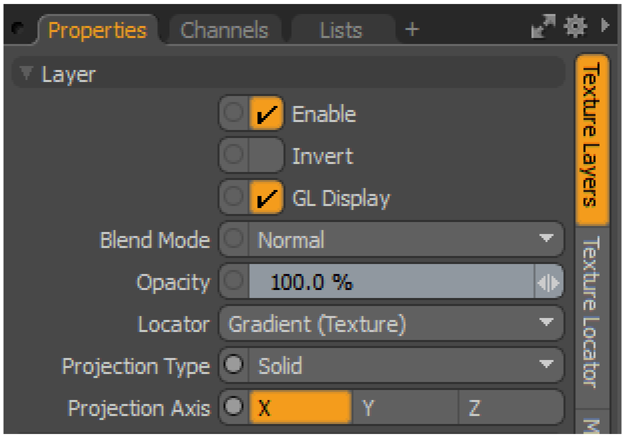


The **Ripples** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Ripples** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a pattern based on ever-expanding ripples that are similar a water's surface disturbed by wind.

> **Note:** For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

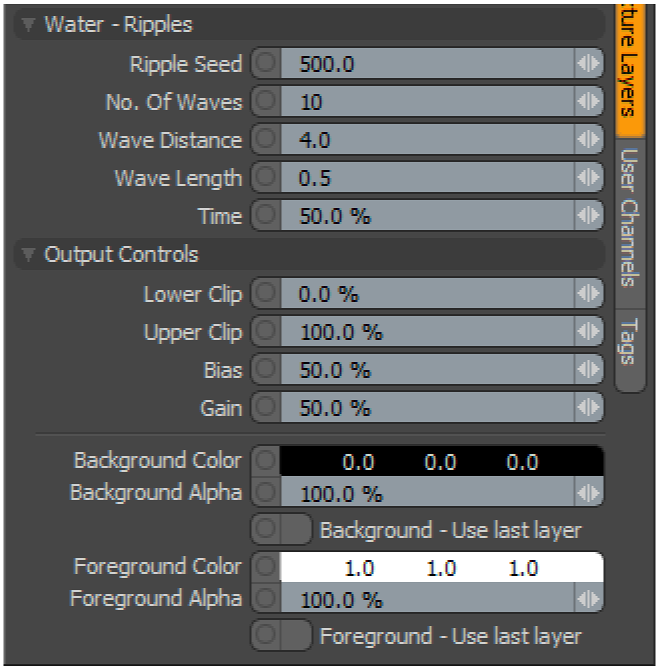
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Ripples Properties



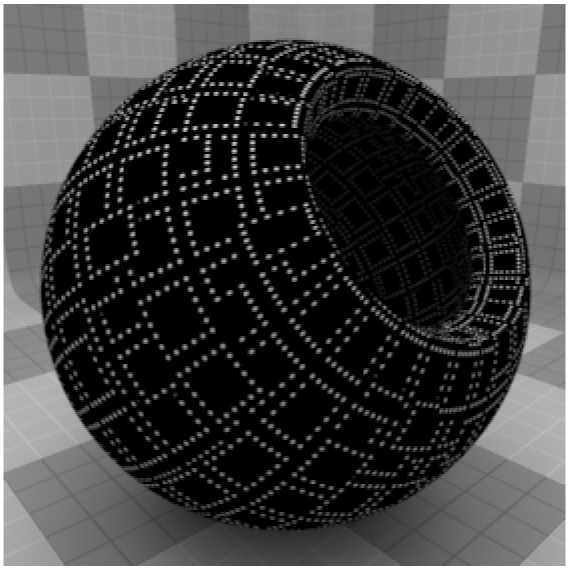
Water - Ripples
Ripple Seed 500.0
No. Of Waves 10
Wave Distance 4.0
Wave Length 0.5
Time 50.0 %
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Water - Ripples	
Ripple Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

No. Of Waves	Specifies the number of waves in the pattern.
Wave Distance	Determines the spread between the ripples.
Wave Length	Determines the wavelength of the ripples throughout the animation.
Time	Controls the progress of the animation (based on the other settings) through time. The animation morphs through multiple waves as the timer goes from **0%** to **100%**.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.

Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Rivets



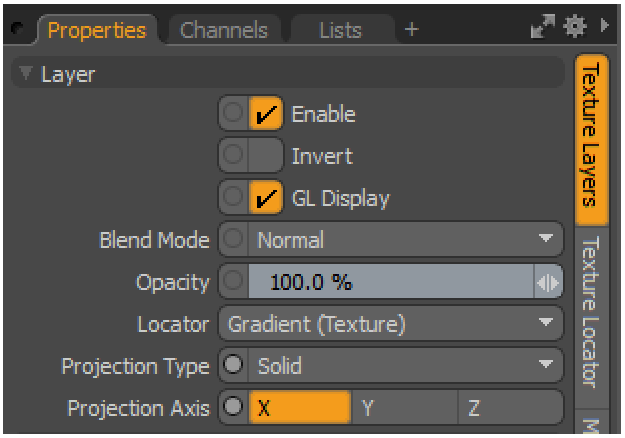
The **Rivets** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Rivets** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of rectangular plates with rivets. You can use this to texture a metal ship's hull or a bridge tower.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

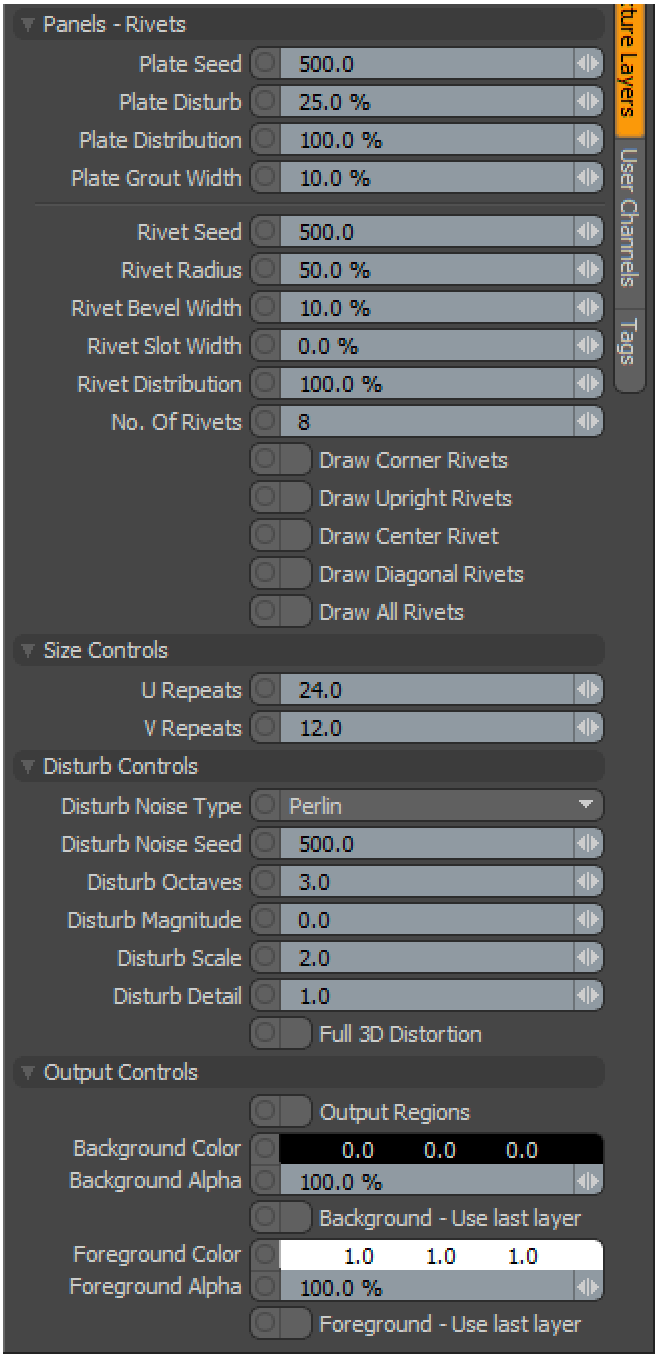
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Rivets Properties



Panels - Rivets

Plate Seed 500.0
Plate Disturb 25.0 %
Plate Distribution 100.0 %
Plate Grout Width 10.0 %
Rivet Seed 500.0
Rivet Radius 50.0 %
Rivet Bevel Width 10.0 %
Rivet Slot Width 0.0 %
Rivet Distribution 100.0 %
No. Of Rivets 8
Draw Corner Rivets
Draw Upright Rivets
Draw Center Rivet
Draw Diagonal Rivets
Draw All Rivets

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

Output Controls

Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

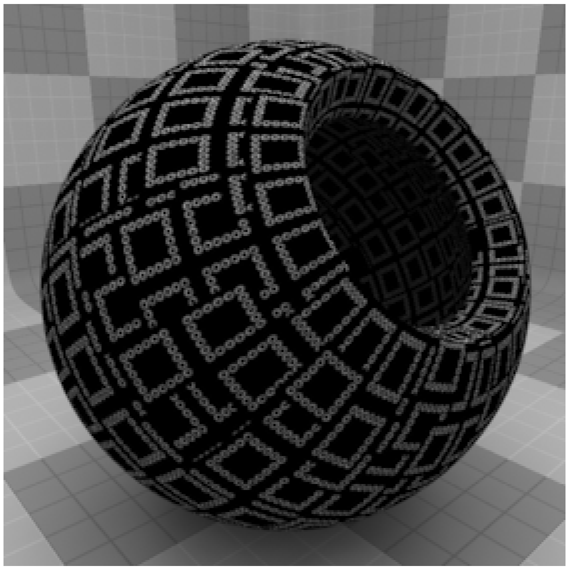
Panels - Rivets	
Plate Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in

	changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Plate Disturb	Determines the amount of distortion among the plates. When you set this to a low value, the plates are very uniform. The plate sizes vary more as you increase this value.
Plate Distribution	Determines the percentage of the surface that is covered by plates.
Plate Grout Width	Determines the amount of grout between the individual plates.
Rivet Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Rivet Radius	Determines the radius of the individual rivets.
Rivet Bevel Width	Determines the width of the bevel on the rivets.
Rivet Slot Width	Determines the width of the slot between the rivets.
Rivet Distribution	Determines the maximum number of rivets that Modo draws.
No. Of Rivets	Specifies the number of rivets that Modo draws on each edge.
Draw Corner Rivets	When enabled, draws the texture with rivets only in the corners of the panels.
Draw Upright Rivets	When enabled, draws the texture with the rivets in a cross pattern within each panel.
Draw Center Rivets	When enabled, draws the texture with one rivet in the center of each panel.
Draw Diagonal Rivets	When enabled, draws the texture with diagonal rivets across each panel.
Draw All Rivets	When enabled, draws the texture with all available rivets in each panel.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).

Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use	When enabled, makes the **Background Color** area completely transparent to reveal

Last Layer	the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Rivet Rust



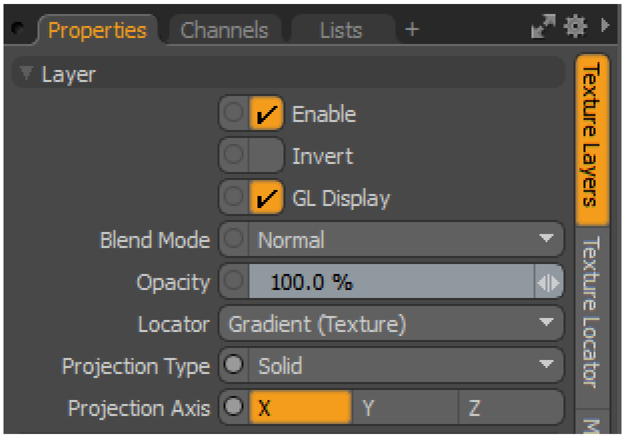
The **Rivet Rust** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Rivet Rust** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of rectangular plates with rusty rivets. You can use this texture for an old metal ship's hull or a rusty bridge tower.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

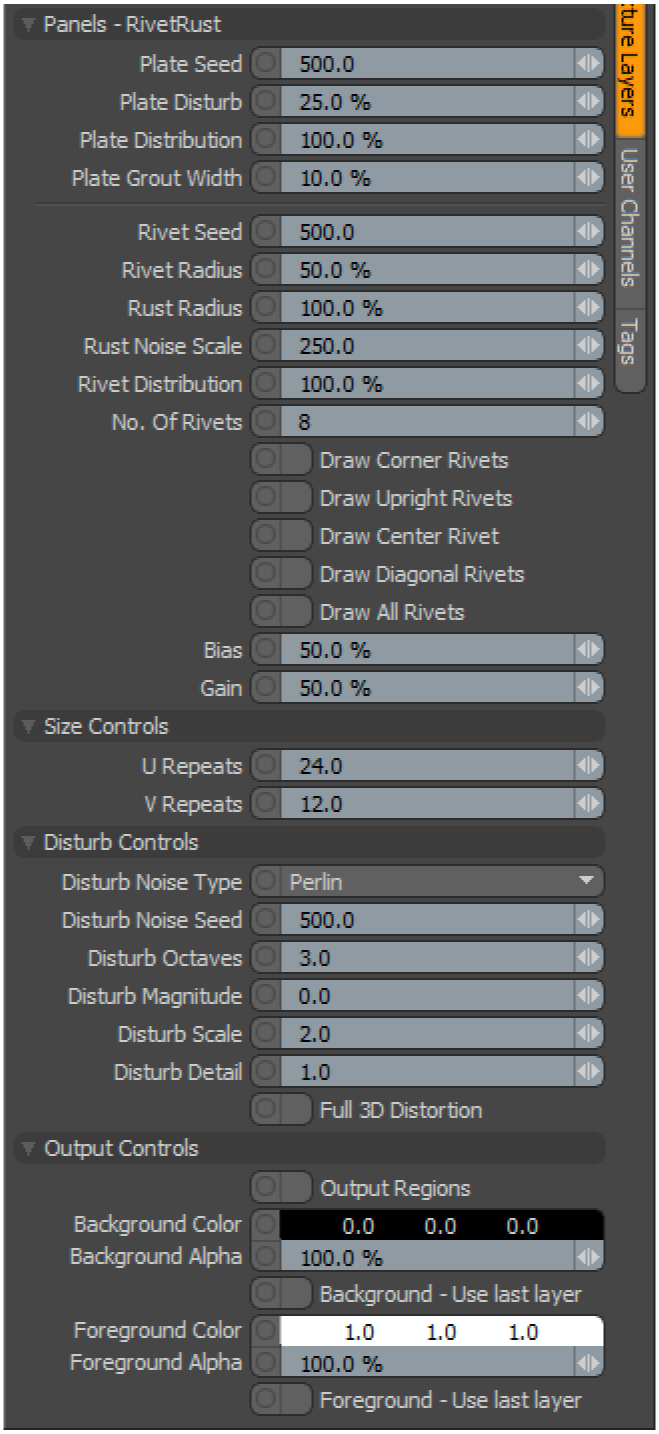
Projection Type: Solid

Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Rivet Rust Properties



Panels - RivetRust

Plate Seed	500.0
Plate Disturb	25.0 %
Plate Distribution	100.0 %
Plate Grout Width	10.0 %
Rivet Seed	500.0
Rivet Radius	50.0 %
Rust Radius	100.0 %
Rust Noise Scale	250.0
Rivet Distribution	100.0 %
No. Of Rivets	8
	☐ Draw Corner Rivets
	☐ Draw Upright Rivets
	☐ Draw Center Rivet
	☐ Draw Diagonal Rivets
	☐ Draw All Rivets
Bias	50.0 %
Gain	50.0 %

Size Controls

U Repeats	24.0
V Repeats	12.0

Disturb Controls

Disturb Noise Type	Perlin
Disturb Noise Seed	500.0
Disturb Octaves	3.0
Disturb Magnitude	0.0
Disturb Scale	2.0
Disturb Detail	1.0
	☐ Full 3D Distortion

Output Controls

	☐ Output Regions
Background Color	0.0 0.0 0.0
Background Alpha	100.0 %
	☐ Background - Use last layer
Foreground Color	1.0 1.0 1.0
Foreground Alpha	100.0 %
	☐ Foreground - Use last layer

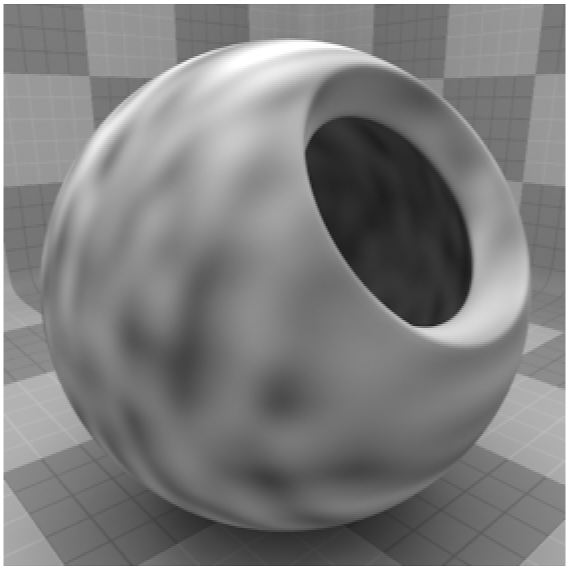
Panels - Rivet Rust

Plate Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Plate Disturb	Determines the amount of distortion among the plates. When you set this to a low value, the plates are very uniform. The plate sizes vary more as you increase this value.
Plate Distribution	Determines the percentage of the surface that is covered by plates.
Plate Grout Width	Determines the amount of grout between the individual plates.
Rivet Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Rivet Radius	Determines the radius of the individual rivets.
Rust Radius	Determines the radius of the rust on the rivets (as a percentage).
Rust Noise Scale	Controls the overall frequency or scale of the rust.
Rivet Distribution	Determines the maximum number of rivets that Modo draws.
No. Of Rivets	Specifies the number of rivets that Modo draws on each edge.
Draw Corner Rivets	When enabled, draws the texture with rivets only in the corners of the panels.
Draw Upright Rivets	When enabled, draws the texture with the rivets in a cross pattern within each panel.
Draw Center Rivets	When enabled, draws the texture with one rivet in the center of each panel.
Draw Diagonal Rivets	When enabled, draws the texture with diagonal rivets across each panel.
Draw All Rivets	When enabled, draws the texture with all available rivets in each panel.
Bias	Determines the brightness of the overall texture.

Gain	Controls the gain (or contrast) of the overall texture.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.

Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

RndLinear (Random Linear)

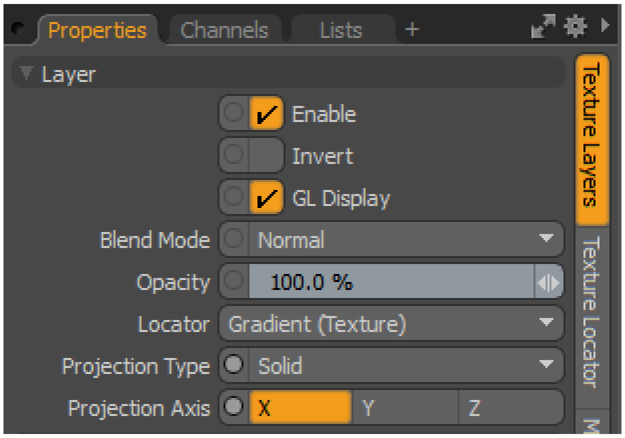


The **RndLinear (Random Linear)** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **RndLinear (Random Linear)** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a randomized linear pattern.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

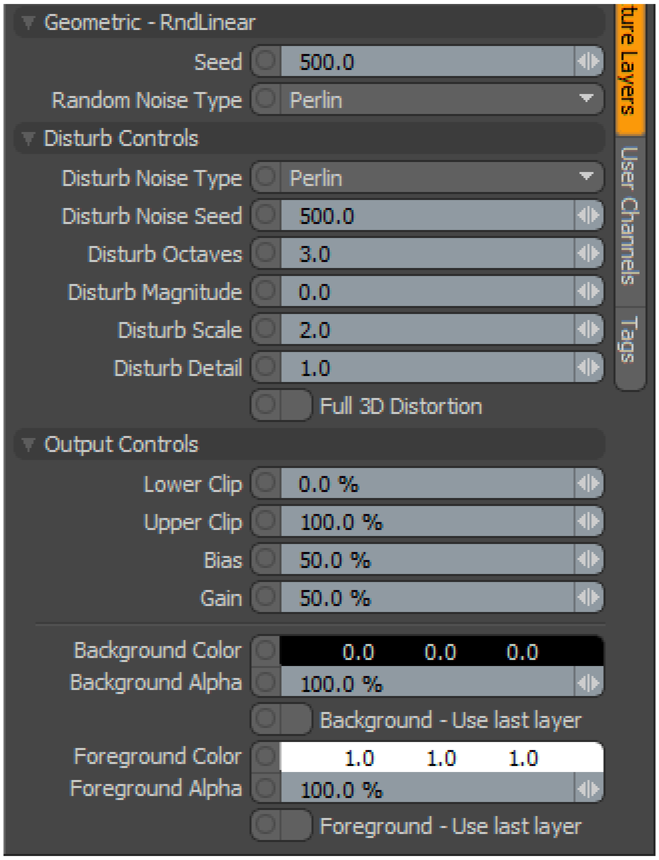


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture**

	Locator that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Random Linear Properties



Geometric - RndLinear
Seed 500.0
Random Noise Type Perlin
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

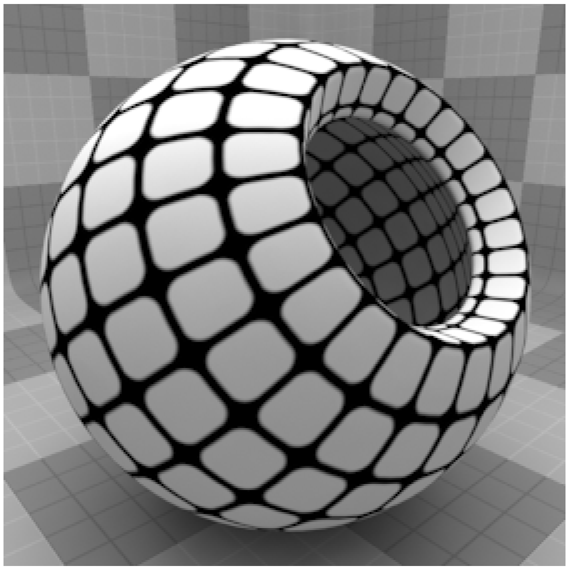
Geometric - Random Linear	
Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in

	changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Random Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the scale of the distortion effect on the base texture. Increasing this value

	intensifies the pattern.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.

Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Rounded Tile



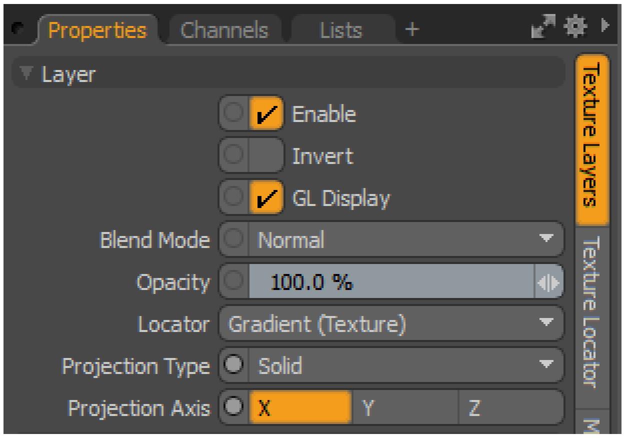
The **Rounded Tile** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Rounded Tile** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a rounded tile texture that is similar to the **Bath Tile** texture with the same options. However, this texture also has an option to specify the amount of roundness for the corners.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

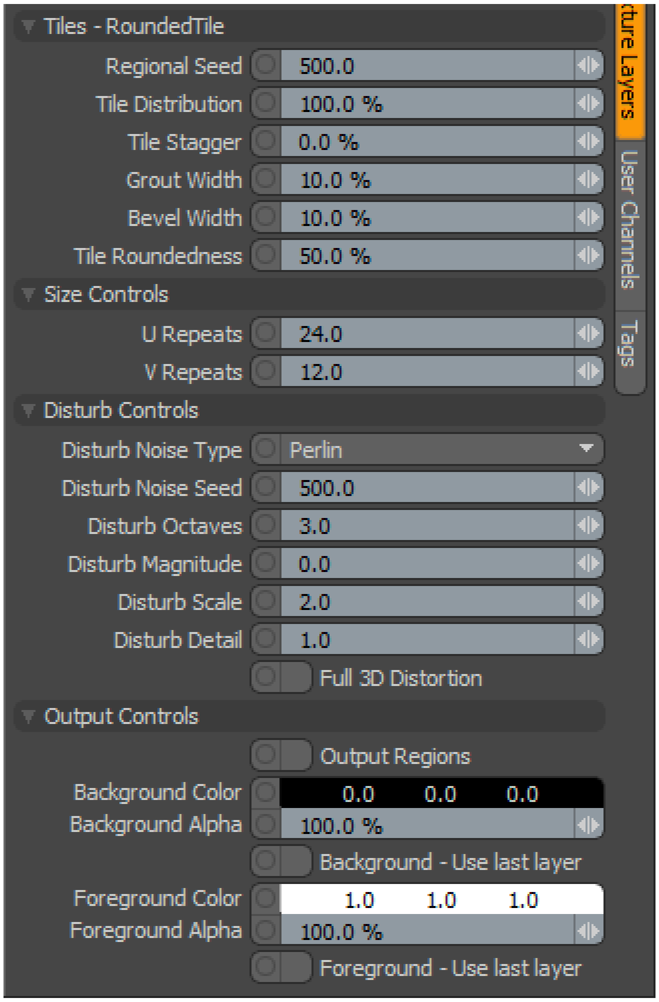
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Rounded Tile Properties



Tiles - RoundedTile
Regional Seed 500.0
Tile Distribution 100.0 %
Tile Stagger 0.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Tile Roundedness 50.0 %

Size Controls
U Repeats 24.0
V Repeats 12.0

Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
○ Full 3D Distortion

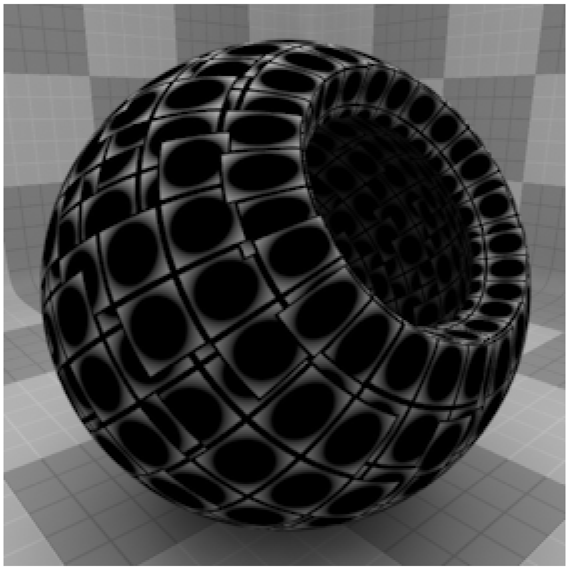
Output Controls
○ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
○ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
○ Foreground - Use last layer

Tiles - Rounded Tile

Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Tile Distribution	Determines the percentage of the surface that is covered by tiles.
Tile Stagger	Determines how staggered the rows of tiles are in the texture. When you set this to a low value, the tiles line up vertically. The tile rows become more staggered as you increase the value.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Tile Roundedness	Controls the radius of each tile's corner. When you set this to a low value, the tiles are square. The corners become more rounded as you increase the value.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb	Specifies the strength of the distortion effect on the base texture.

Magnitude	
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Rust



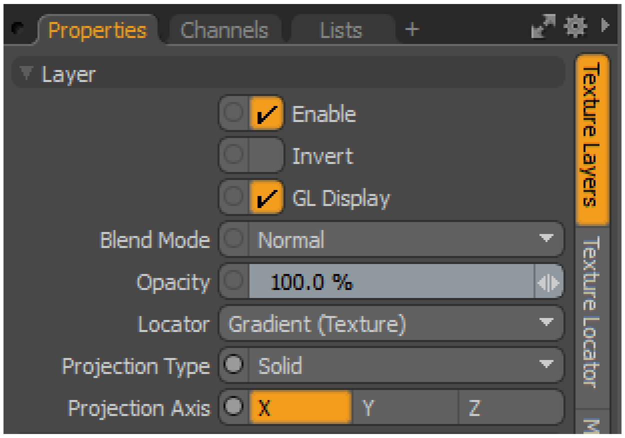
The **Rust** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Rust** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of rectangular plates with a internal frame of rust. You can use this to texture an old water tank or a cobblestone street.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

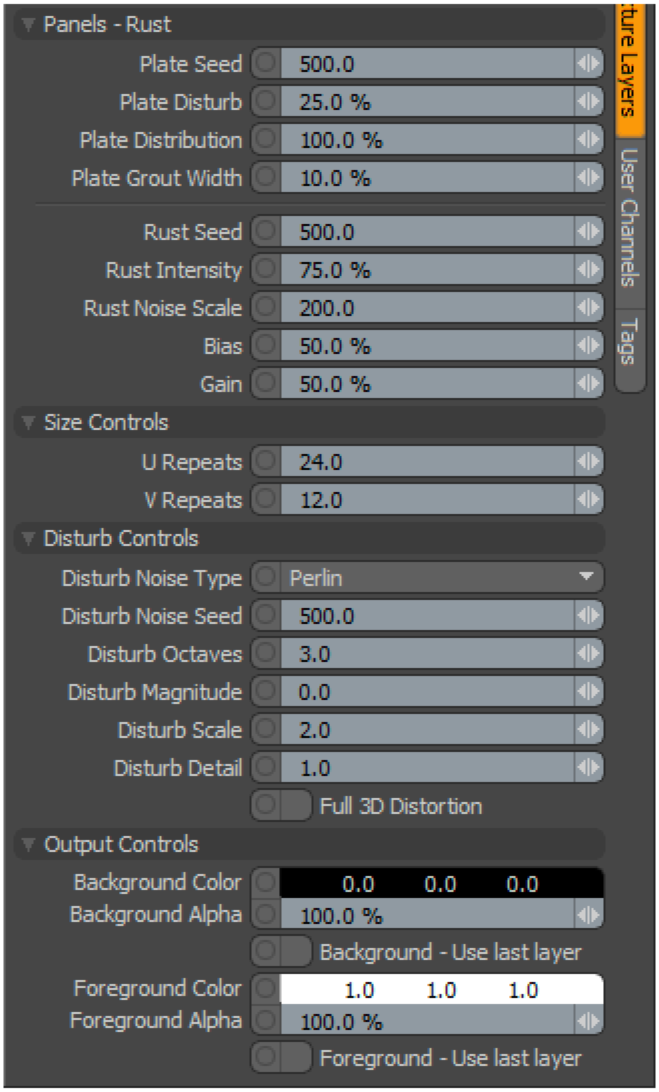
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Rust Properties



Panels - Rust

Plate Seed 500.0
Plate Disturb 25.0 %
Plate Distribution 100.0 %
Plate Grout Width 10.0 %
Rust Seed 500.0
Rust Intensity 75.0 %
Rust Noise Scale 200.0
Bias 50.0 %
Gain 50.0 %

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

Output Controls

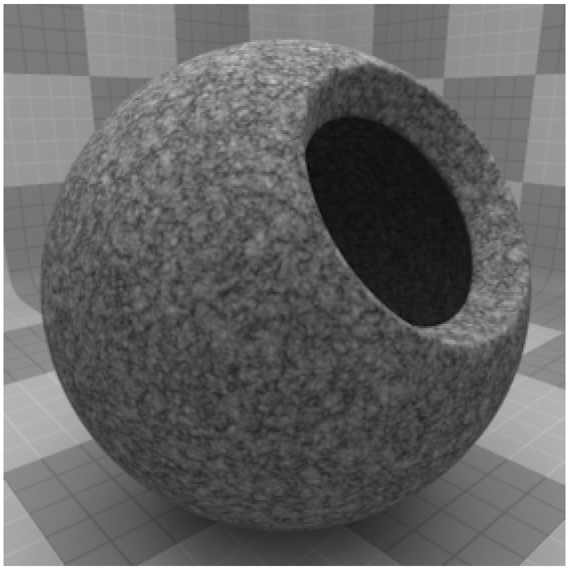
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Panels - Rust	
Plate Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Plate Disturb	Determines the amount of distortion among the plates. When you set this to a low value, the plates are very uniform. The plates sizes vary more as you increase this value.

Plate Distribution	Determines the percentage of the surface that is covered by plates.
Plate Grout Width	Determines the amount of grout between the individual plates.
Rust Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Rust Intensity	Determines the amount of rust inside the individual plates as the rust extends from the corners.
Rust Noise Scale	Controls the overall frequency (or scale) of the rust.
Bias	Determines the brightness of the overall texture.
Gain	Controls the gain (or contrast) of the overall texture.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb	Specifies the strength of the distortion effect on the base texture.

Magnitude	
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Scar

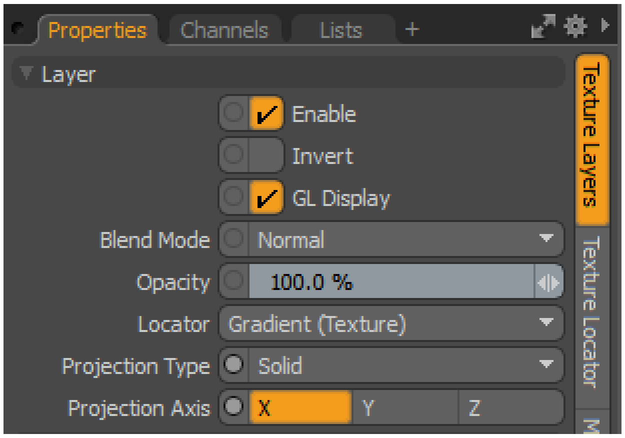


The **Scar** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Scar** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a noisy, ripped pattern rather like scar tissue or lava rock.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

- ☑ Enable
- ☐ Invert
- ☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

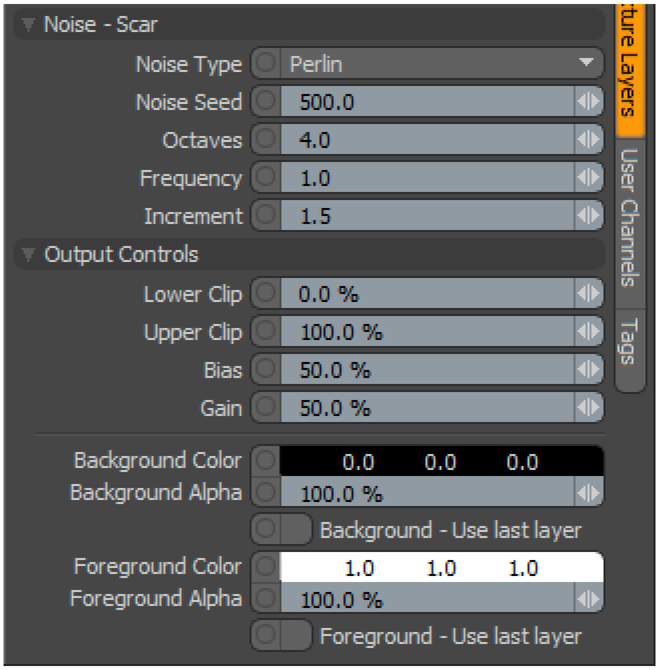
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Scar Properties



Noise - Scar

Noise Type: Perlin

Noise Seed: 500.0

Octaves: 4.0

Frequency: 1.0

Increment: 1.5

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

○ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

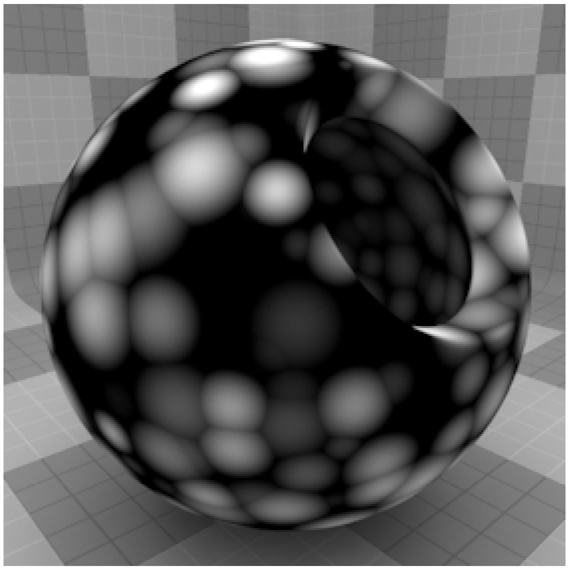
○ Foreground - Use last layer

Noise - Scar	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase

	this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.

Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Scatter

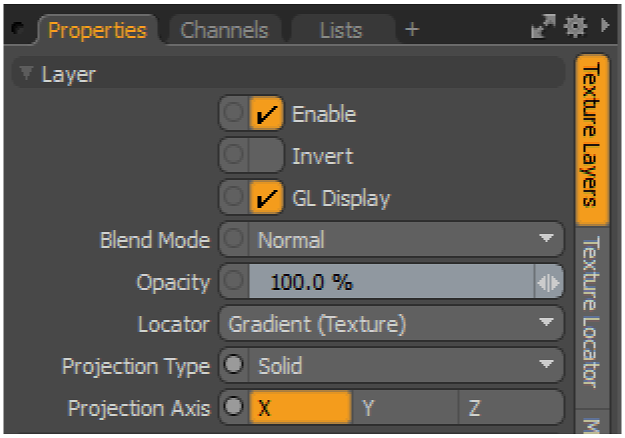


The **Scatter** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Scatter** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This particular texture provides a pseudo-random pattern of spheres, cubes, or diamonds.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

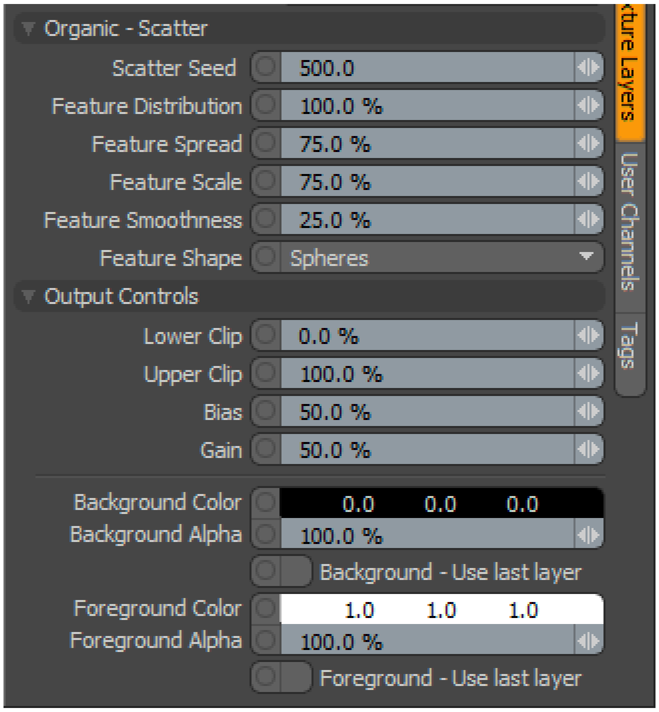


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple

	texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Scatter Properties



Organic - Scatter

Scatter Seed 500.0
Feature Distribution 100.0 %
Feature Spread 75.0 %
Feature Scale 75.0 %
Feature Smoothness 25.0 %
Feature Shape Spheres

Output Controls

Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %

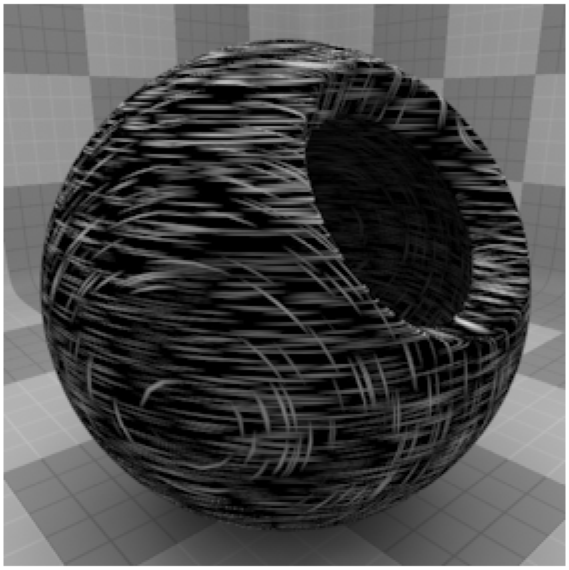
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Organic - Scatter	
Scatter Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Feature Distribution	Defines the number of available features that Modo draws in the texture. Setting lower values creates a less dense texture.
Feature Spread	Controls the distance between the individual features.

Feature Scale	Controls the scaling of the individual features. When you set this to a low value, the features are smaller and more prevalent.
Feature Smoothness	Controls the smoothness between the foreground and background colors of the feature.
Feature Shape	Specifies the shape of the individual features. The options are: • **Spheres** • **Cubes** • **Diamonds**
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.

Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

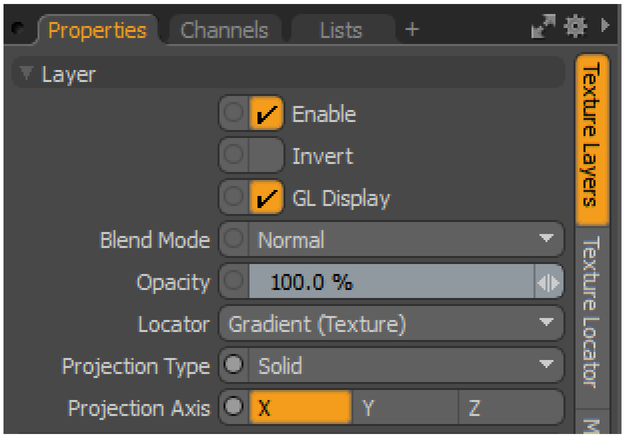
Scratches



The **Scratches** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Scratches** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture applies random scratches, dings, and dents to an object's surface and gives a worn-in, rough look. The long, thin, overlapping cuts and dents can be in up to 16 different layers. Adding color to the scratches can create a texture similar to an antique finish.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

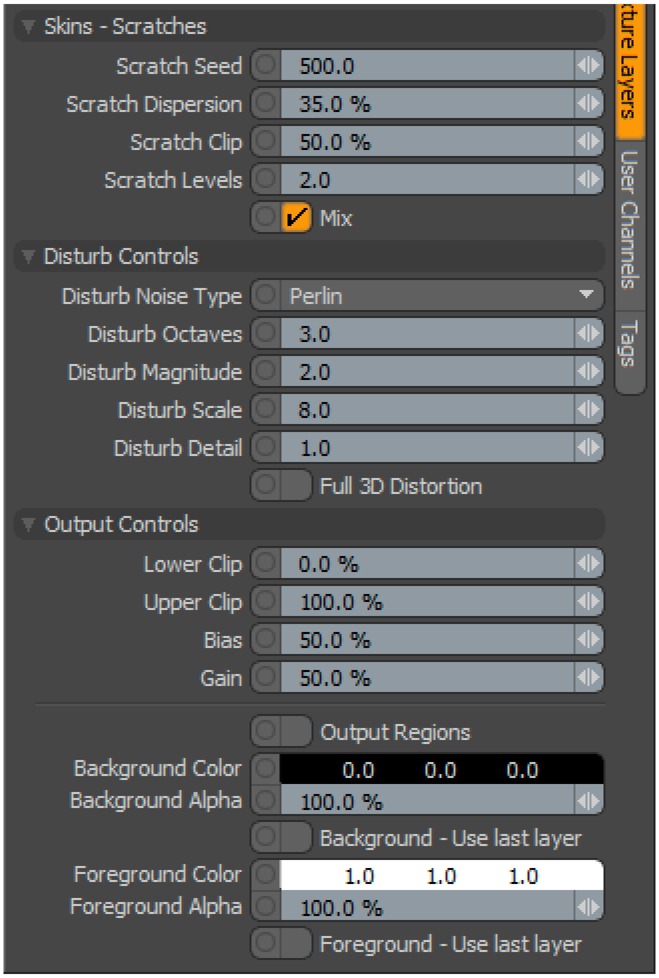
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Scratches Properties



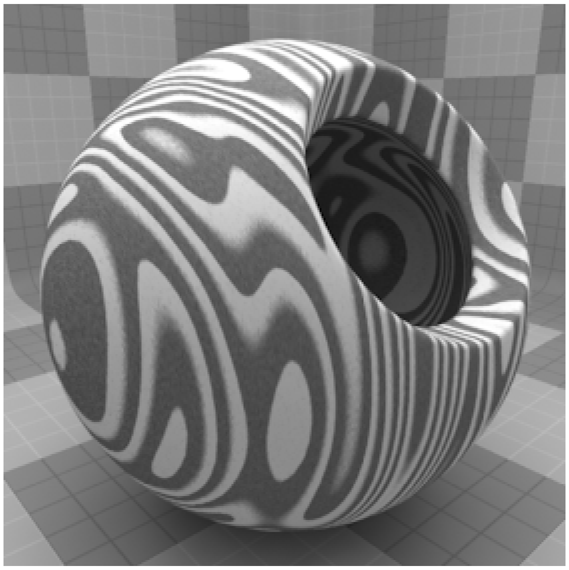
Skins - Scratches
Scratch Seed 500.0
Scratch Dispersion 35.0 %
Scratch Clip 50.0 %
Scratch Levels 2.0
☑ Mix
Disturb Controls
Disturb Noise Type Perlin
Disturb Octaves 3.0
Disturb Magnitude 2.0
Disturb Scale 8.0
Disturb Detail 1.0
☐ Full 3D Distortion
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Skins - Scratches	
Scratch Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Scratch	Determines how many scratches Modo draws.

Dispersion	
Scratch Clip	Determines where the texture's shading transitions to the background color. The higher the value is, the quicker the transition occurs.
Scratch Levels	Determines the number of layers of the texture. Additional levels add additional detail and also lighten the overall look of the texture.
Mix	When enabled, mixes the texture layers. When disabled, Modo stacks the texture layers one upon the other.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	

Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Scruffed

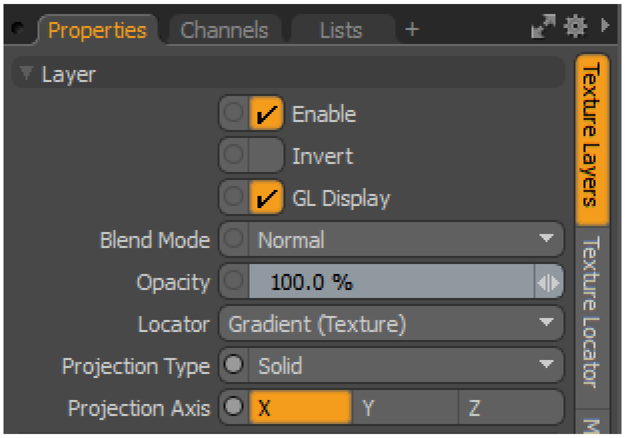


The **Scruffed** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Scruffed** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a surface that looks like the object has been scruffed and scraped along its surface.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

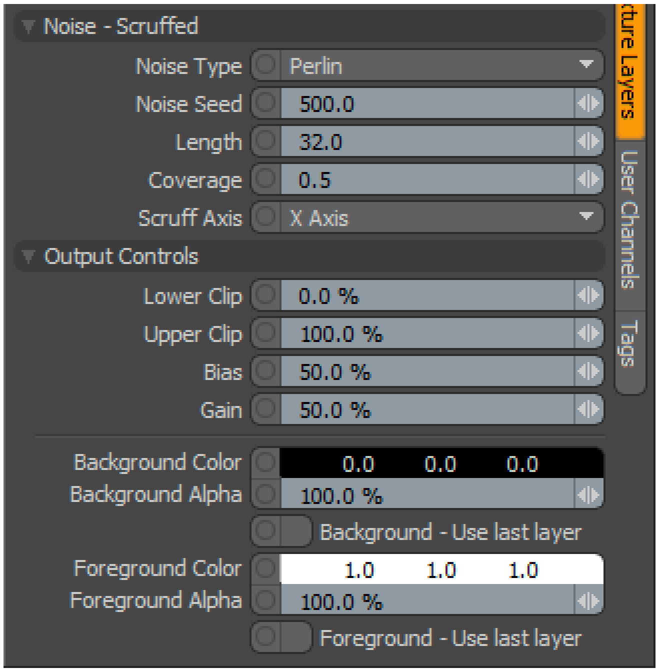
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Scruffed Properties



Noise - Scruffed

Noise Type: Perlin

Noise Seed: 500.0

Length: 32.0

Coverage: 0.5

Scruff Axis: X Axis

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

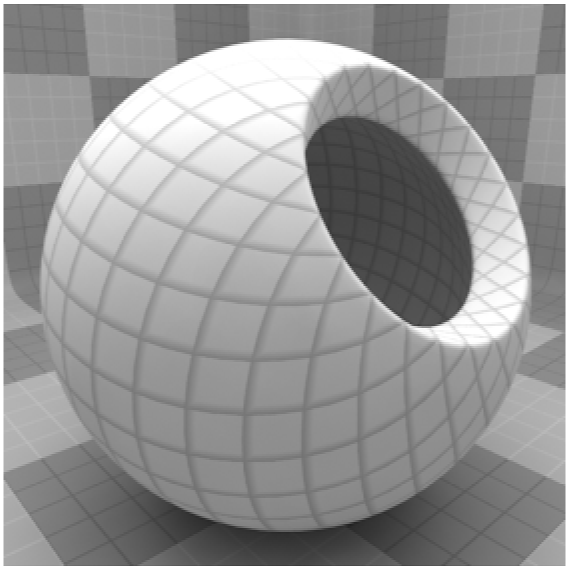
Foreground - Use last layer

Noise - Scruffed	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Length	Controls the amount of the foreground color. Decreasing this value increases the

	overall coverage of the foreground color.
Coverage	Controls the amount of the background color. Decreasing this value increases the overall coverage of the background color.
Scruff Axis	Specifies the axis along which the layers run.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground	Specifies the **Alpha** transparency of the **Foreground Color**.

Alpha	
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Shingles



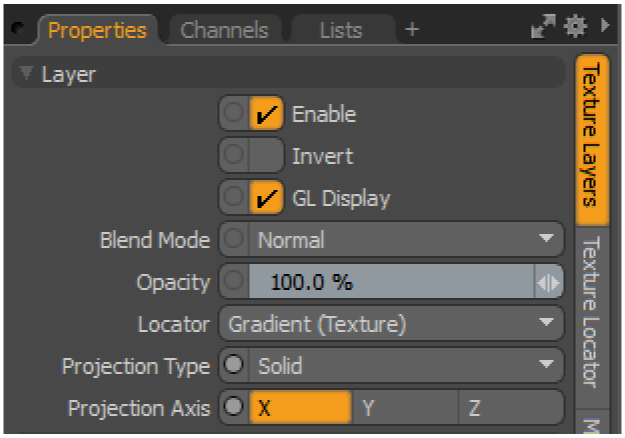
The **Shingles** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Shingles** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a pattern similar to a shingle roof. You can vary the bevel width of each shingle and its respective height.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

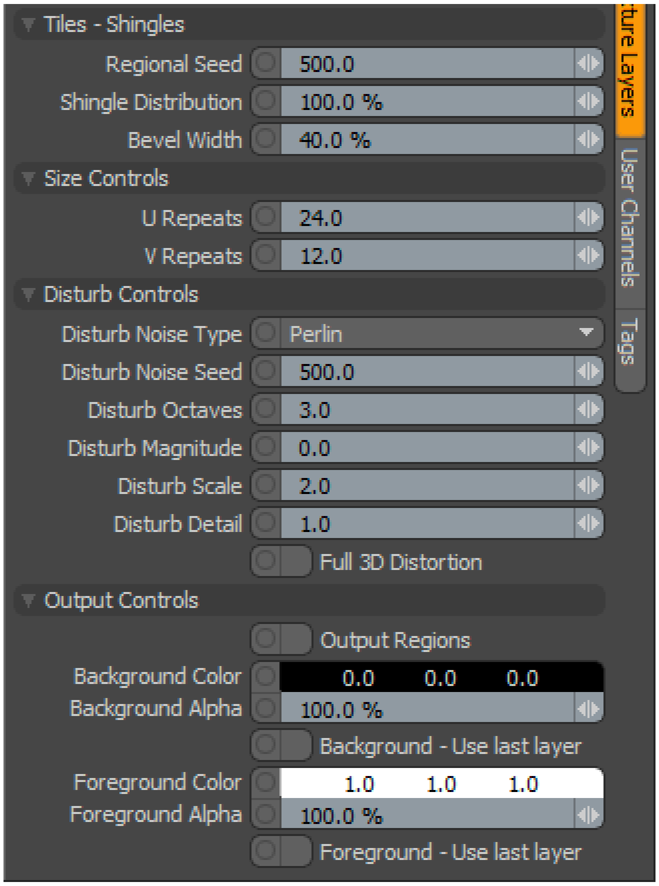
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Shingles Properties



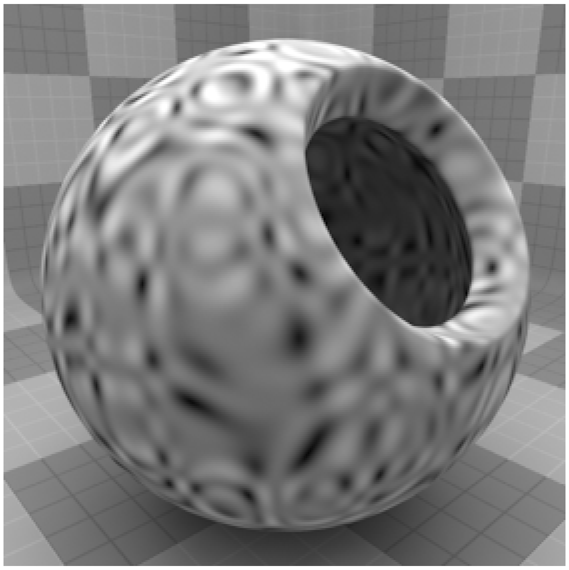
Tiles - Shingles
Regional Seed 500.0
Shingle Distribution 100.0 %
Bevel Width 40.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Shingles	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when

	you want items to retain the same variations.
Shingle Distribution	Determines the percentage of the surface that is covered by shingles.
Bevel Width	Determines the width of the bevel on the shingles.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.

Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Sin Blob

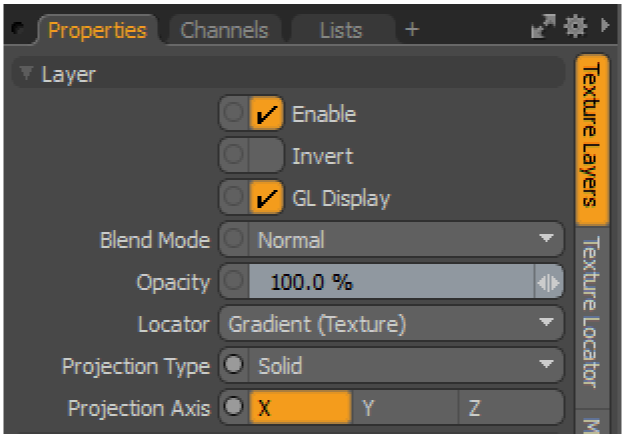


The **Sin Blob** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Sin Blob** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture produces a blob-like sine wave pattern that is based on a pseudo-randomizing process. You can specify a seed to create many different variations.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

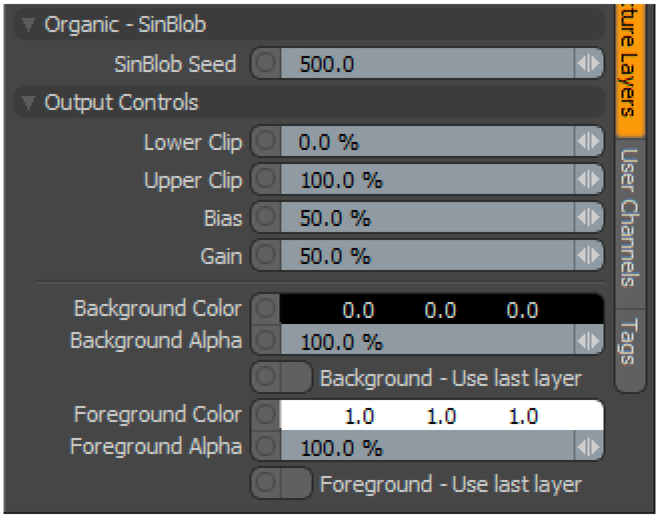
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Sin Blob Properties

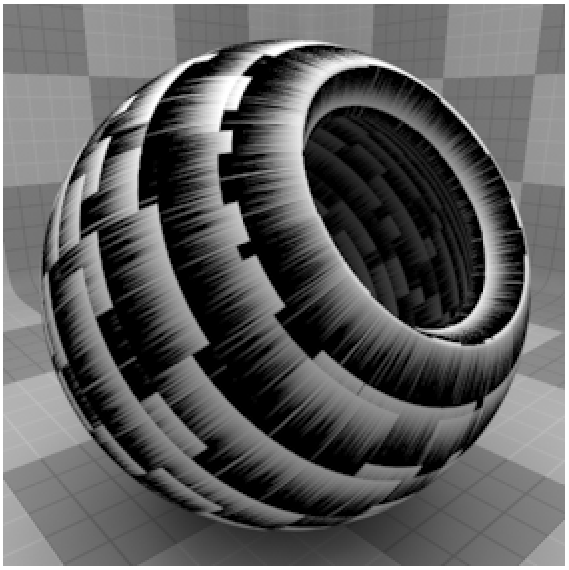


Organic - SinBlob
SinBlob Seed 500.0
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Organic - SinBlob	
SinBlob Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.

Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Smear



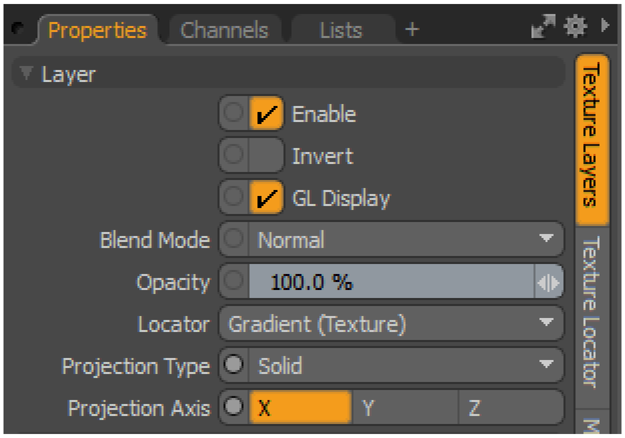
The **Smear** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Smear** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of rectangular plates with a line of dirt or rust smeared in a certain direction. You can use this texture to show a grimy cinder block wall or a burned spaceship.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

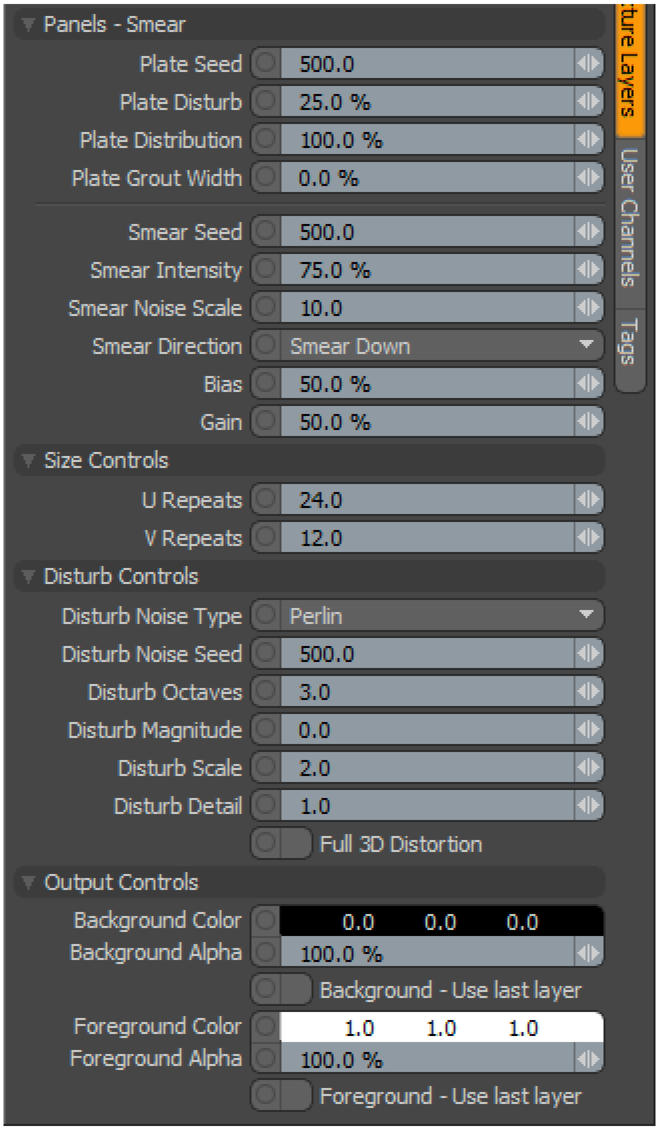
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they

	are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Smear Properties



Panels - Smear

Plate Seed 500.0
Plate Disturb 25.0 %
Plate Distribution 100.0 %
Plate Grout Width 0.0 %
Smear Seed 500.0
Smear Intensity 75.0 %
Smear Noise Scale 10.0
Smear Direction Smear Down
Bias 50.0 %
Gain 50.0 %

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

Output Controls

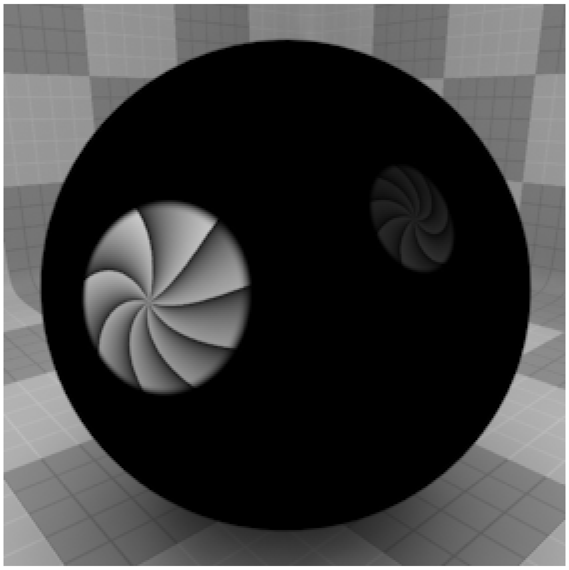
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Panels - Smear	
Plate Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Plate Disturb	Determines the amount of distortion among the plates. When you set this to a low value, the plates are very uniform. The plate sizes vary more as you increase the value.

Plate Distribution	Determines the percentage of the surface that is covered by plates.
Plate Grout Width	Determines the amount of grout between the individual plates.
Smear Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Smear Intensity	Determines the amount of smearing on the surface as it extends from the selected direction (**Smear Up**, **Smear Down**, **Smear Left**, or **Smear Right**).
Smear Noise Scale	Controls the overall frequency (or scale) of the smearing.
Smear Direction	Determines the direction of the smearing. The options are: • **Smear Down** • **Smear Up** • **Smear Right** • **Smear Left**
Bias	Determines the brightness of the overall texture.
Gain	Controls the gain (or contrast) of the overall texture.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value**

	• **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

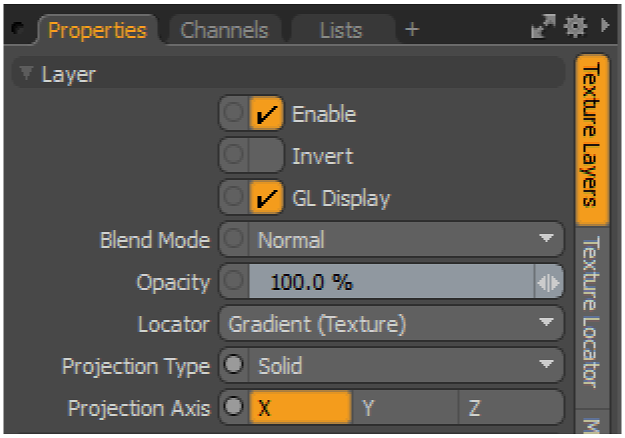
Spiral



The **Spiral** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Spiral** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a customizable spiral pattern.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

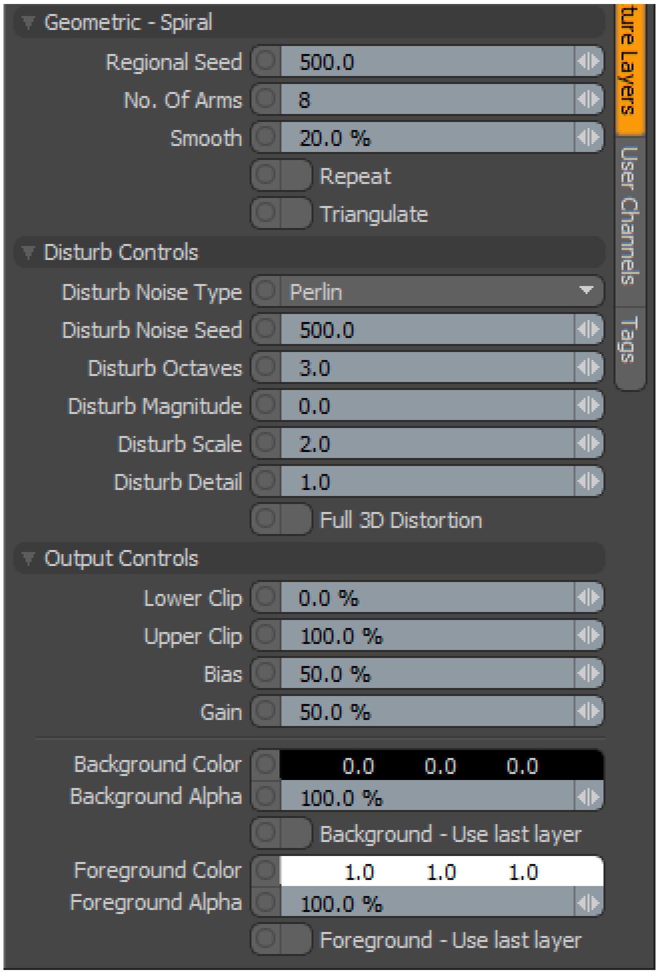
Layer Properties


Properties
Channels
Lists
+
Texture Layers
Texture Locator
Layer
Enable
Invert
GL Display
Blend Mode
Normal
Opacity
100.0 %
Locator
Gradient (Texture)
Projection Type
Solid
Projection Axis
X
Y
Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Spiral Properties



Geometric - Spiral

Regional Seed 500.0

No. Of Arms 8

Smooth 20.0 %

Repeat

Triangulate

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

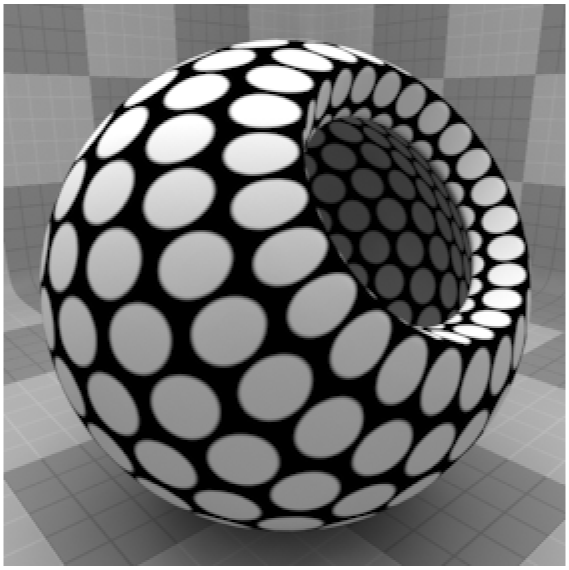
Foreground - Use last layer

Geometric - Spiral	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
No. Of Arms	Determines the number of arms in the spiral pattern.
Smooth	Determines the amount of smoothing around the edges of the spiral. Higher values create smoother edges.
Repeat	When enabled, causes the texture to repeat indefinitely.

Triangulate	When enabled, causes the pattern to have a triangle-based edging pattern instead of the default sawtooth blend.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Spots



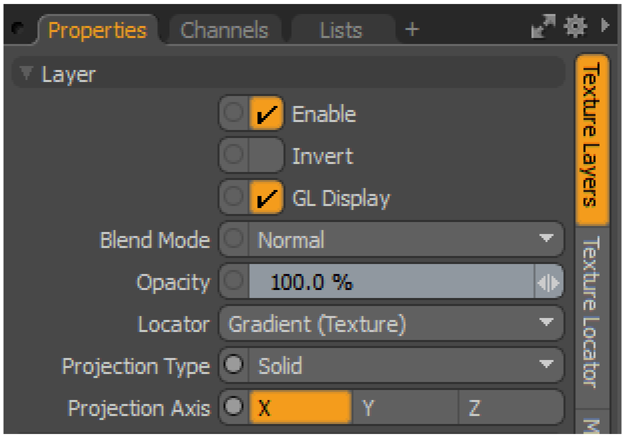
The **Spots** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Spots** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a series of spotted, circular tiles. This texture has the same options as the **Bath Tile** shader.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

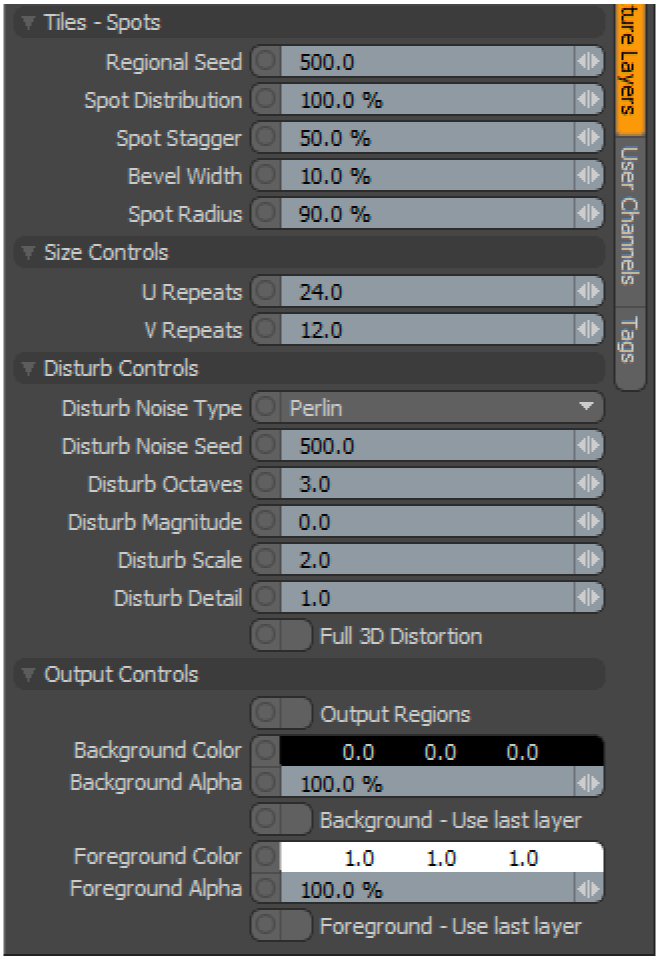
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Spots Properties



Tiles - Spots

Regional Seed 500.0

Spot Distribution 100.0 %

Spot Stagger 50.0 %

Bevel Width 10.0 %

Spot Radius 90.0 %

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

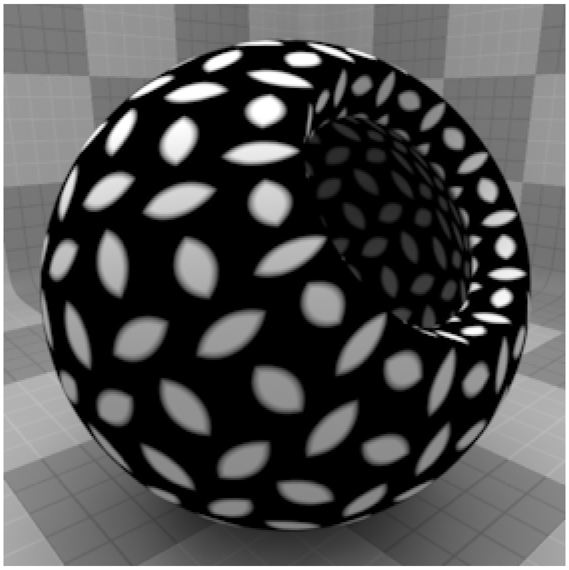
Foreground - Use last layer

Tiles - Spots	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Spot Distribution	Determines the percentage of the surface that is covered by spots.
Spot Stagger	Determines how staggered the rows of spots are in the texture. When you set the value to a low value, the spots line up vertically. The spots become more staggered as you increase the value.
Bevel Width	Determines the width of the bevel on the tiles.

Spot Radius	Determines how much Modo fills in the individual tile area with a spot.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.

Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Stamped



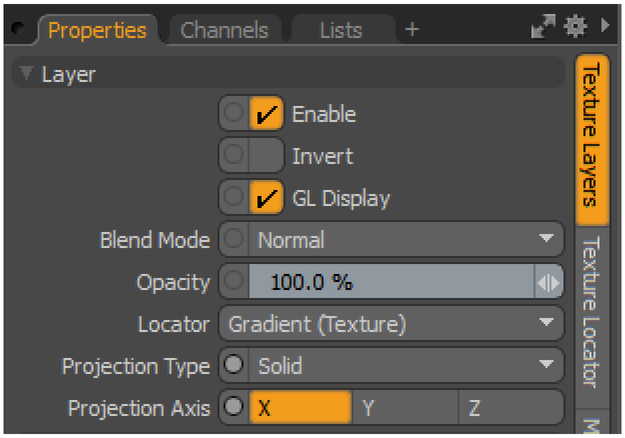
The **Stamped** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Stamped** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a pattern similar to a stamped metal plate. You can adjust the number of stamps per cell and the shape, grout width, bevel width, and height.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

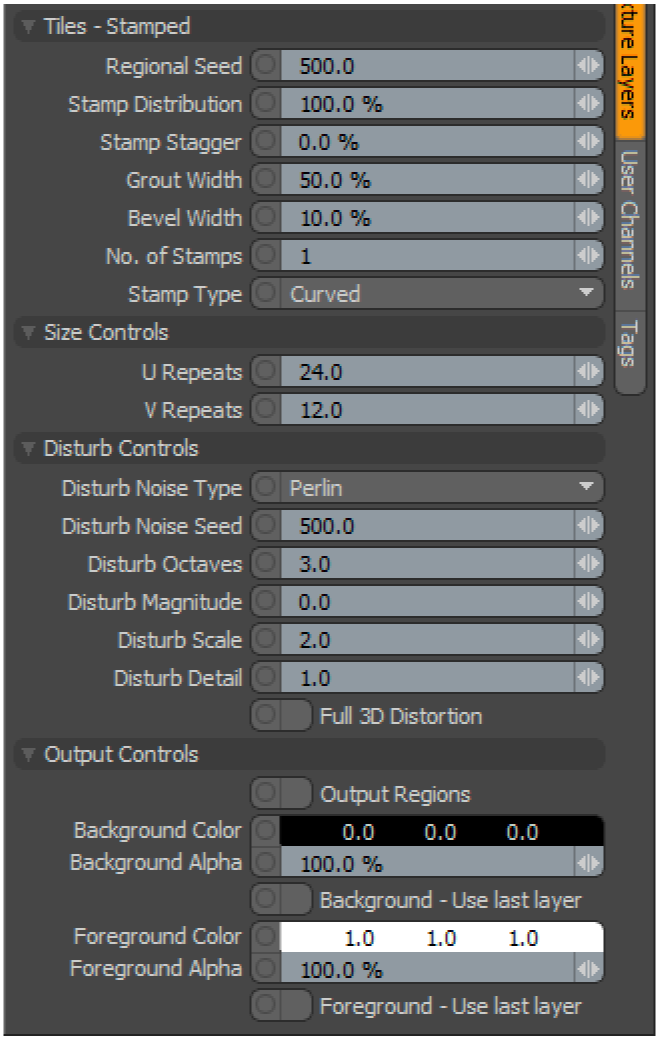
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.

Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Stamped Properties



Tiles - Stamped

Regional Seed 500.0
Stamp Distribution 100.0 %
Stamp Stagger 0.0 %
Grout Width 50.0 %
Bevel Width 10.0 %
No. of Stamps 1
Stamp Type Curved

Size Controls

U Repeats 24.0
V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion

Output Controls

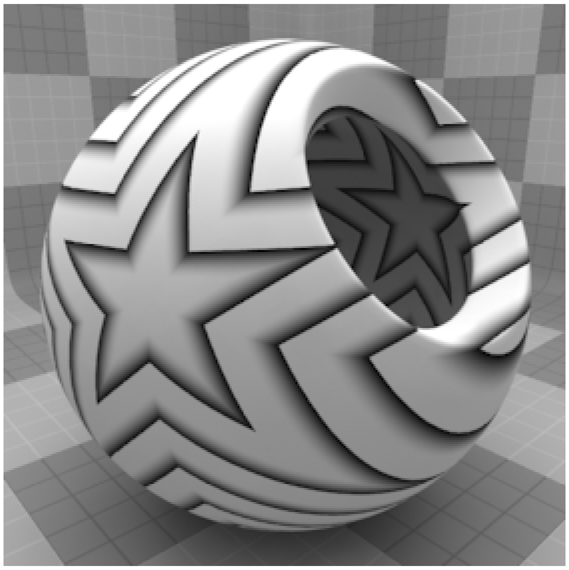
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Option	**Description**
Tiles - Stamped	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Stamp Distribution	Determines the percentage of the surface that is covered by the stamps.

Option	Description
Stamp Stagger	Determines how staggered the rows of stamps are in the texture. When you set this to a low value, the stamps line up vertically. The stamps become more staggered as you increase the value.
Grout Width	Determines the amount of grout between the stamps.
Bevel Width	Determines the width of the bevel on the stamps.
No. Of Stamps	Specifies the number of stamps in each block.
Stamp Type	Determines the shape of the stamp used. The available options are: • **Straight** • **Curved** • **Capsule**
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb	Specifies the strength of the distortion effect on the base texture.

Option	Description
Magnitude	
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Star

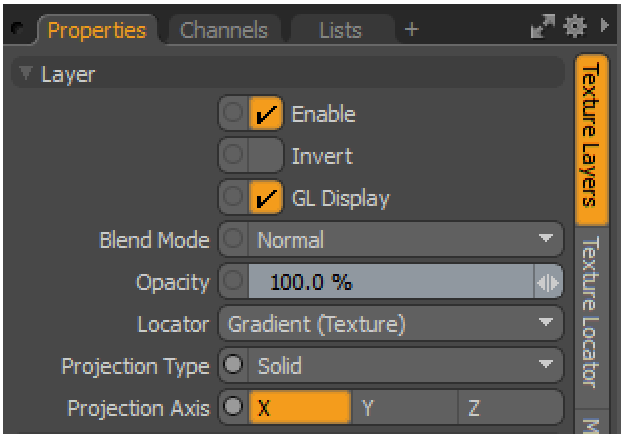


The **Star** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Star** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a customizable star pattern.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

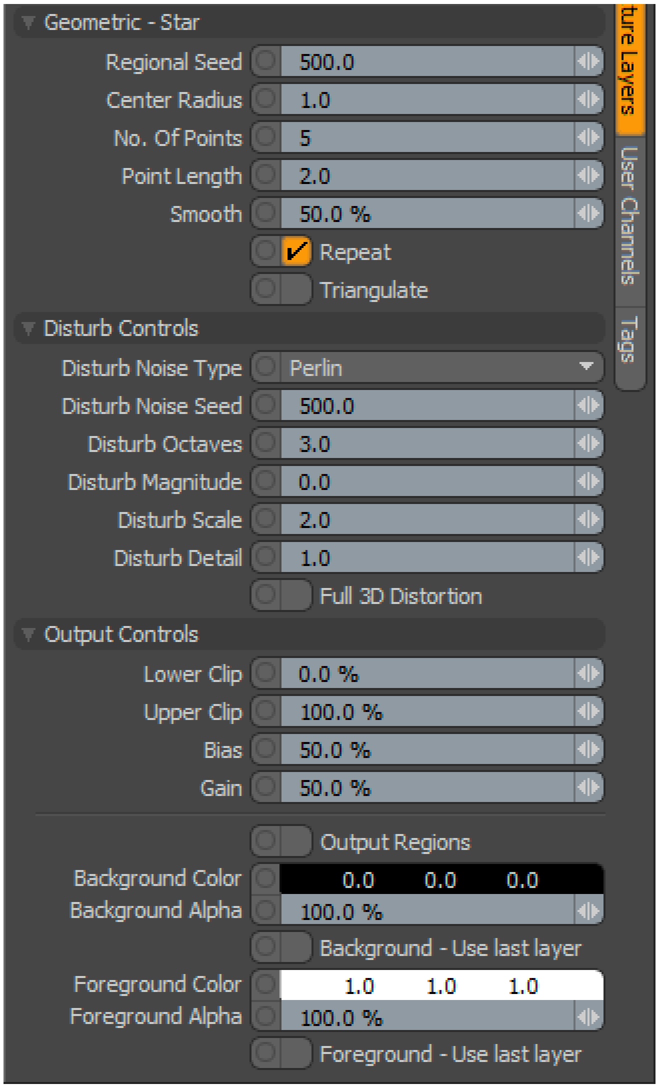
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Star Properties



Geometric - Star

Regional Seed 500.0
Center Radius 1.0
No. Of Points 5
Point Length 2.0
Smooth 50.0 %
☑ Repeat
☐ Triangulate

Disturb Controls

Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion

Output Controls

Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

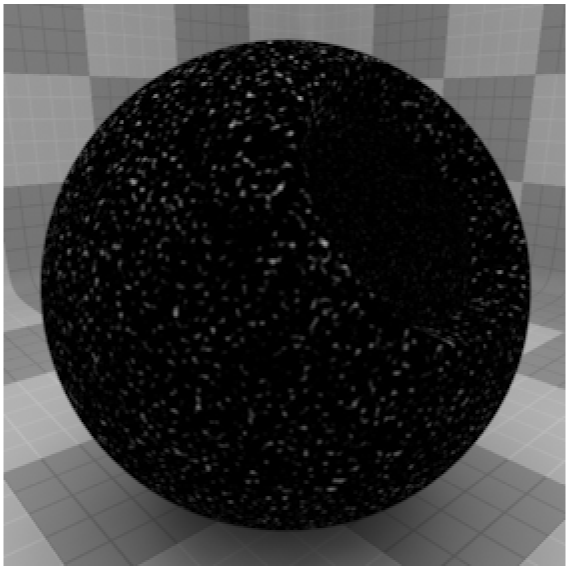
Geometric - Star	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Center Radius	Specifies the radius at the center of the star.
No. Of Points	Determines the number of points in the star pattern.

Point Length	Determines the length of the individual points on the star.
Smooth	Determines the amount of smoothing around the edges of the star. Higher values create smoother edges.
Repeat	When enabled, causes the texture to repeat indefinitely.
Triangulate	When enabled, causes the pattern to have a triangle-based edging pattern instead of the default sawtooth blend.
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.

Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.

Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Star Field

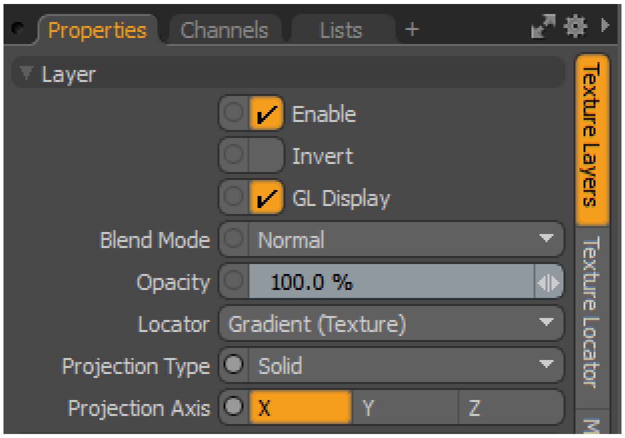


The **Star Field** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Star Field** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a star field similar to what you would see outside on a starry night or in a planetarium.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

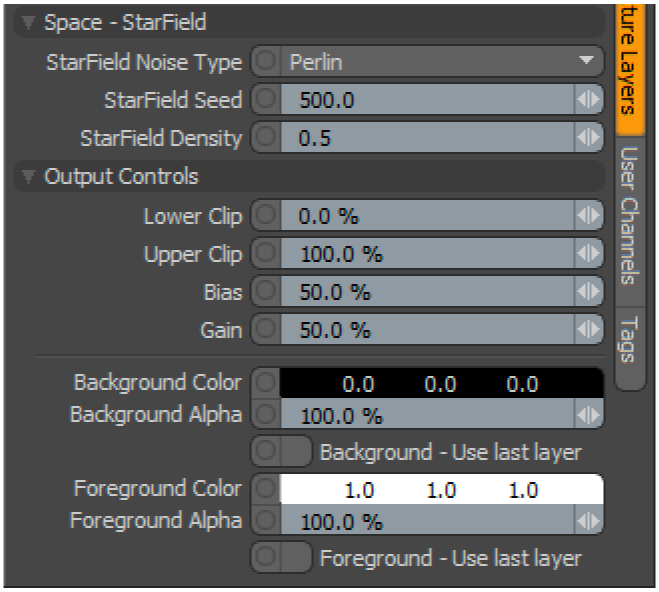
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Star Field Properties



Space - StarField

StarField Noise Type: Perlin

StarField Seed: 500.0

StarField Density: 0.5

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

○ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

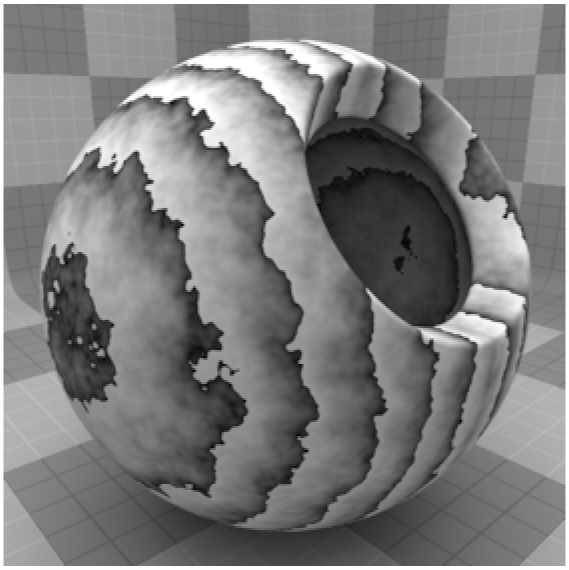
Foreground Alpha: 100.0 %

○ Foreground - Use last layer

Space - Star Field	
Star Field Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Star Field Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Star Field Density	Controls the number of stars present in the star field. Increasing this value creates a more star-filled sky.

Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Strata

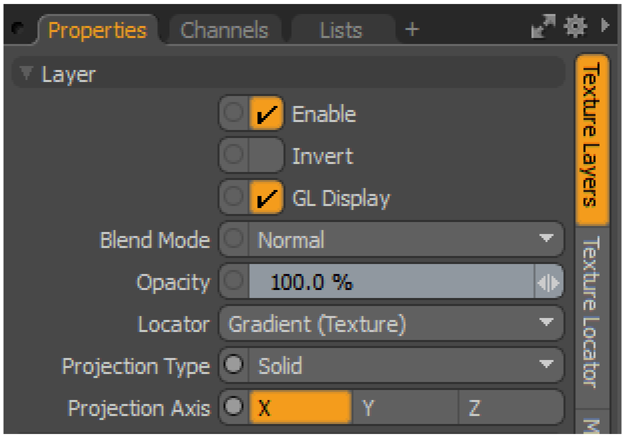


The **Strata** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Strata** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a pattern similar to the layers (or strata) of sedimentary rock.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

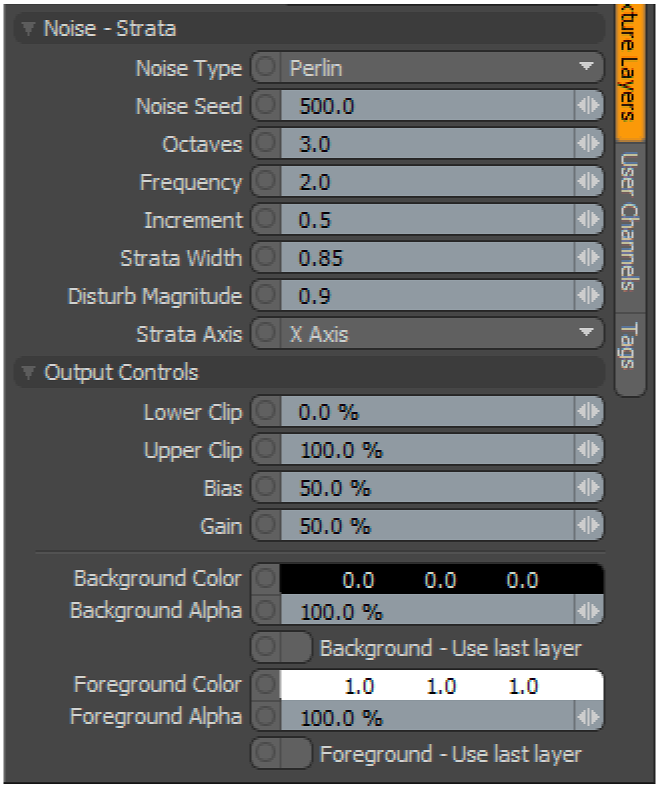
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Strata Properties



Noise - Strata

Noise Type: Perlin
Noise Seed: 500.0
Octaves: 3.0
Frequency: 2.0
Increment: 0.5
Strata Width: 0.85
Disturb Magnitude: 0.9
Strata Axis: X Axis

Output Controls

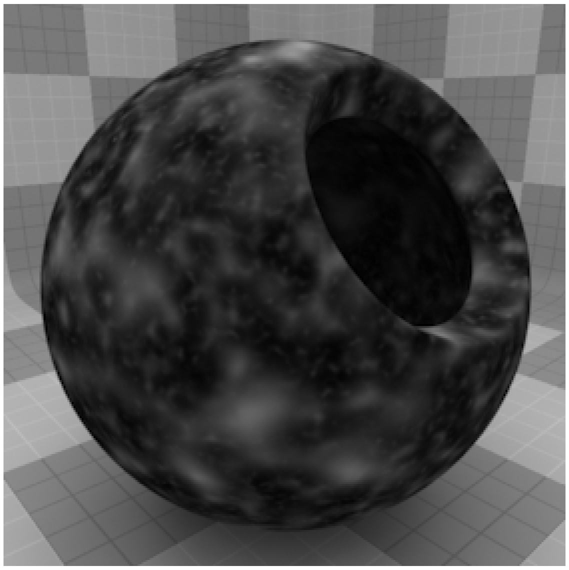
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Strata	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when

	you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Strata Width	Controls the amount of the foreground color. Decreasing this value increases the overall coverage of the foreground color.
Disturb Magnitude	Determines the strength of the distortion effect on the base texture.
Strata Axis	Specifies the axis along which the layers run.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles.

	This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Stucco

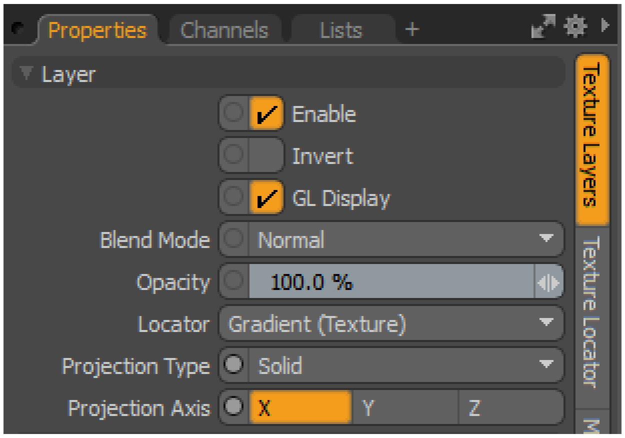


The **Stucco** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Stucco** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides stucco-type bumps that are similar to those found on rendered walls or stippled ceilings.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

- ☑ Enable
- ☐ Invert
- ☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

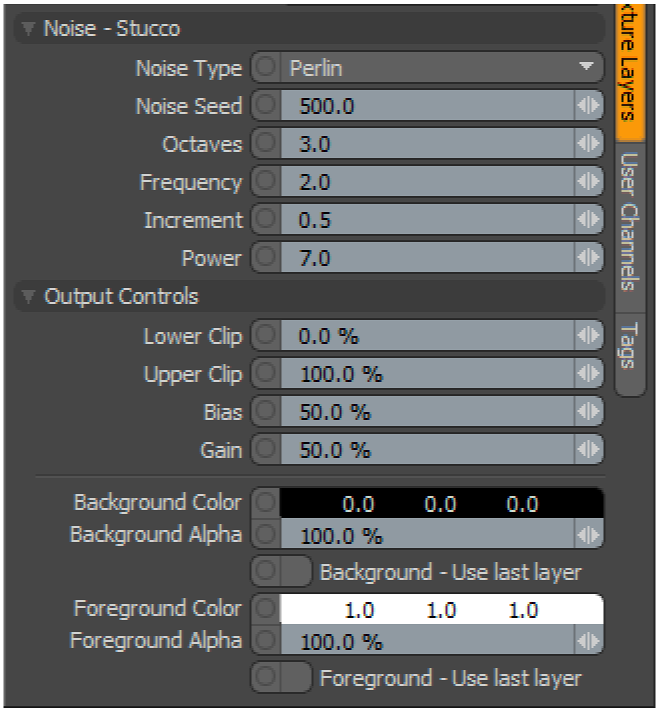
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.

Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Stucco Software



Noise - Stucco

Noise Type Perlin

Noise Seed 500.0

Octaves 3.0

Frequency 2.0

Increment 0.5

Power 7.0

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

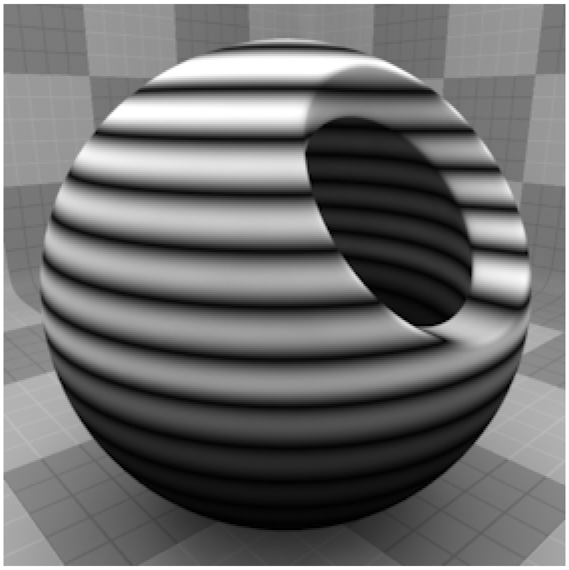
Foreground - Use last layer

Noise - Stucco	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin**

	• **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Power	Controls the intensity of the stipples. As you increase this number, the background color becomes more prevalent as does the delineation between colors.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.

Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Surf

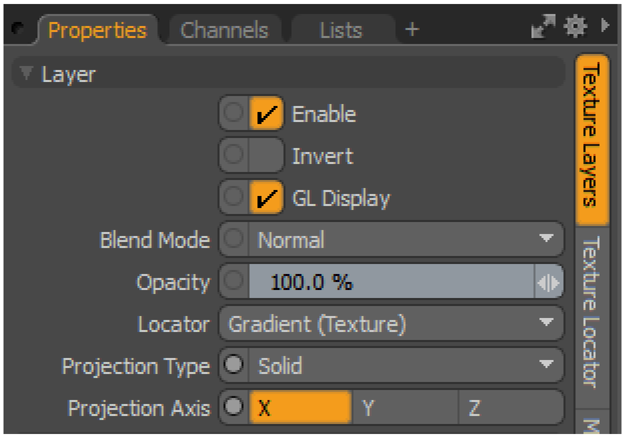


The **Surf** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Surf** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a standard wave pattern emanating from the texture's center. The amplitude of the wave decreases as the wave moves away from the wave source.

> **Note:** For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

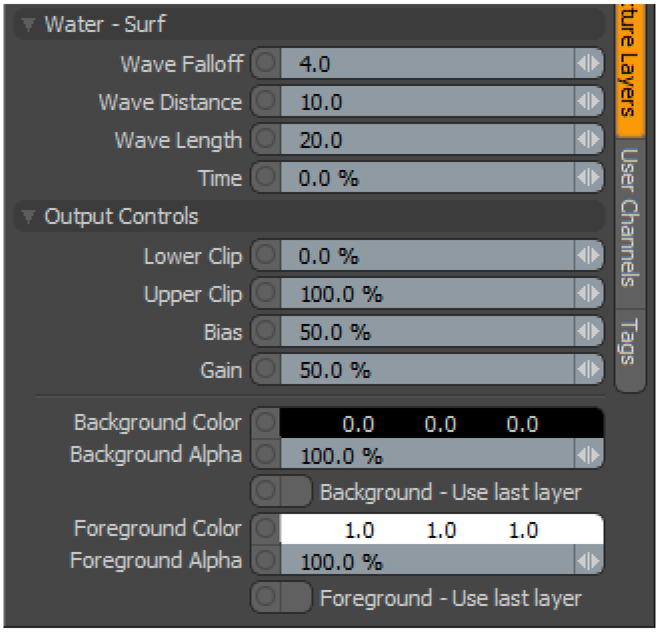
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Surf Properties

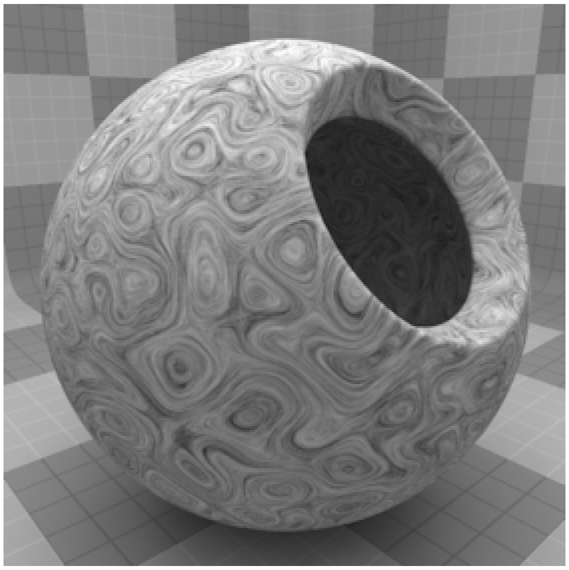


Wave Falloff 4.0
Wave Distance 10.0
Wave Length 20.0
Time 0.0 %
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Water - Surf	
Wave Falloff	Controls the falloff (or fade) of the waves as they emanate from the center. When you specify a low value, the foreground and background transition is similar across the texture. As you increase the value, the transition between colors becomes smoother at the opposite end of the wave source.
Wave Distance	Controls the number of waves that Modo draws on the surface.
Wave Length	Determines the distance moved by the waves.
Time	Controls the progress of the animation (based on the other settings) through time. The animation morphs through multiple batches of rain drops as the timer goes from **0%** to **100%**.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to

	extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Swirl

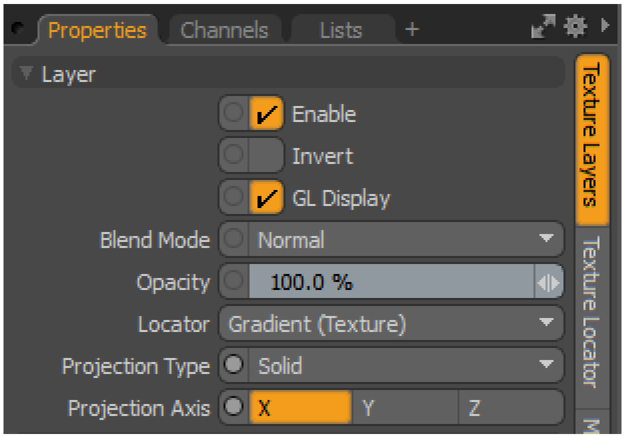


The **Swirl** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Swirl** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture is similar to the **FBM** (Fractal Brownian Motion) texture except that this texture has an additional swirl option. This texture can be highly stirred, which makes it ideal for creating smoke, fog, or a planet similar to Jupiter.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

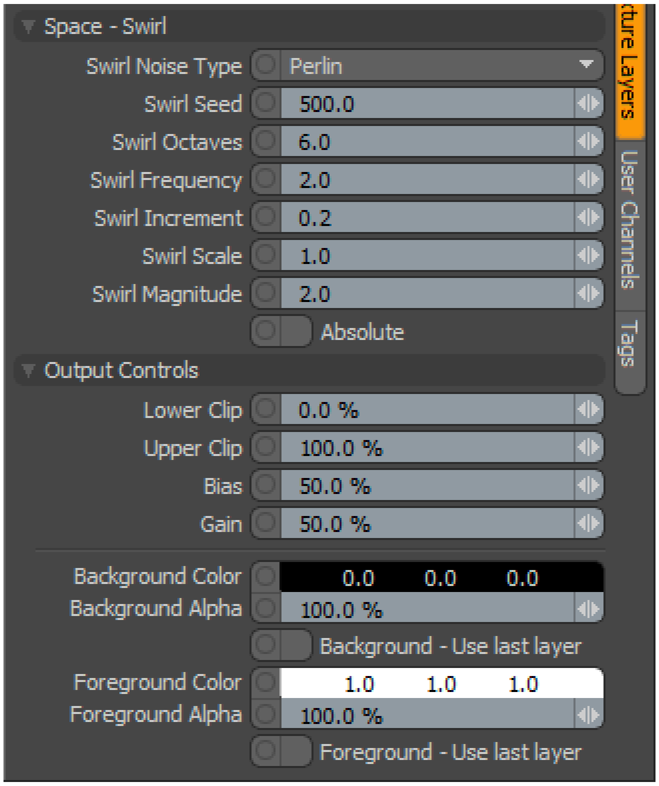
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X | Y | Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Swirl Properties



Space - Swirl

Swirl Noise Type Perlin
Swirl Seed 500.0
Swirl Octaves 6.0
Swirl Frequency 2.0
Swirl Increment 0.2
Swirl Scale 1.0
Swirl Magnitude 2.0
Absolute

Output Controls

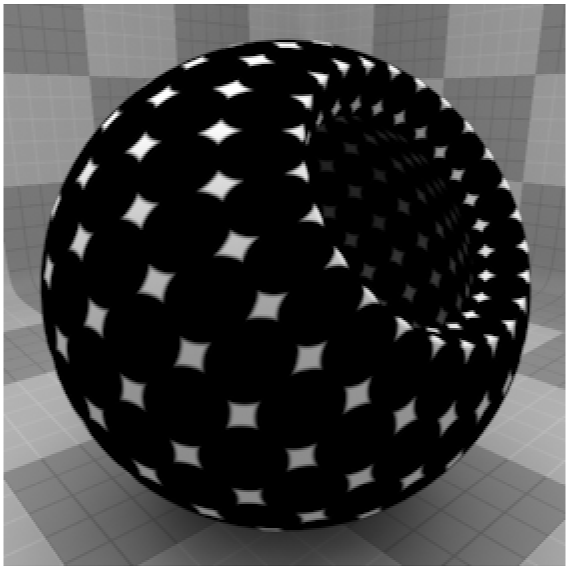
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Space - Swirl	
Swirl Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Swirl Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when

	you want items to retain the same variations.
Swirl Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Swirl Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir the pattern.
Swirl Increment	Controls the scaling of each successive iteration. When you set the value low, the turbulence is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Swirl Scale	Controls the scaling of swirls. When you set this to a low value, the swirls are loose. They become increasingly tighter as you increase the value.
Swirl Magnitude	Controls how much the swirls affect the overall texture. The higher the value is, the more prominent the swirls are.
Absolute	When enabled, uses the absolute value for the noise to provide a stronger and more defined variation on the default texture.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.

Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Tacos



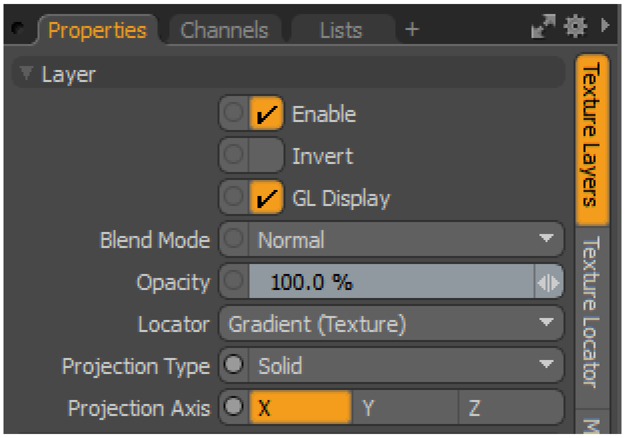
The **Tacos** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Tacos** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides small, diamond-shaped tiles (tacos) and uses the same set of options as the **Bath Tile** texture. You can use these diamond-shaped tiles either independently or to fill the gaps in the **Cornerless** texture.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

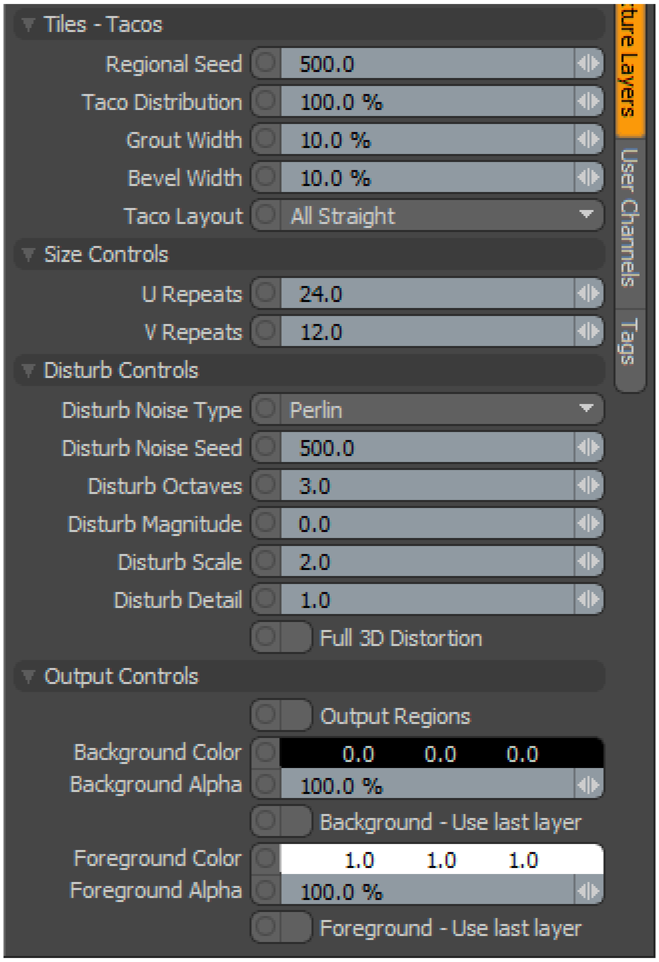
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they

	are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Tacos Properties



Tiles - Tacos
Regional Seed 500.0
Taco Distribution 100.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Taco Layout All Straight

Size Controls
U Repeats 24.0
V Repeats 12.0

Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion

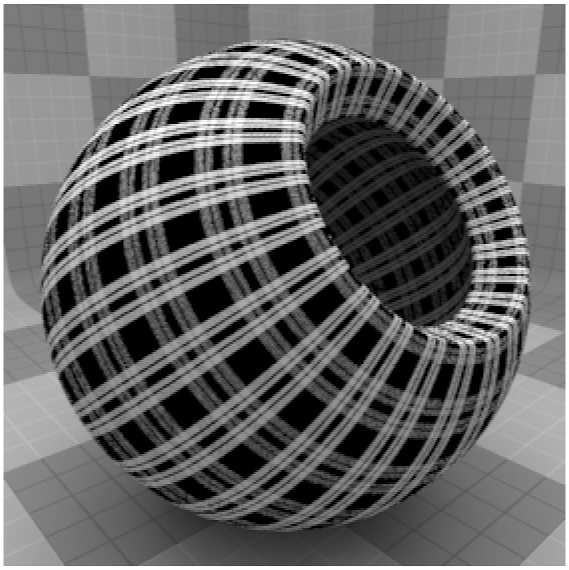
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Tacos	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Taco Distribution	Determines the percentage of the surface that is covered by the tacos (diamond shapes).
Grout Width	Determines the amount of grout between the diamond shapes.
Bevel Width	Determines the width of the bevel on the diamond shapes.
Taco Layout	Determines the layout of the diamond shapes. The options are:

	• **All Straight** • **Alternate Straight** • **All Curved** • **Alternate Curved**
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.

Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Tartan



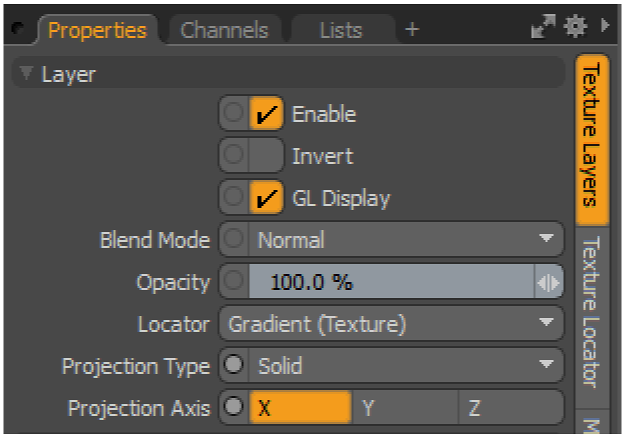
The **Tartan** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Tartan** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a tartan pattern that is similar to the **Plaid** texture except that the weave is visible.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

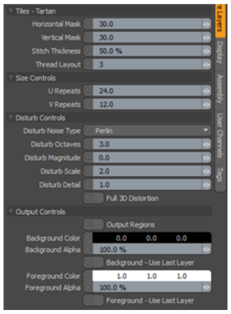
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: [✓ X] Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Tartan Properties



Tiles - Tartan

Horizontal Mask 30.0

Vertical Mask 30.0

Stitch Thickness 50.0 %

Thread Layout 3

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use Last Layer

Foreground Color 1.0 1.0 1.0

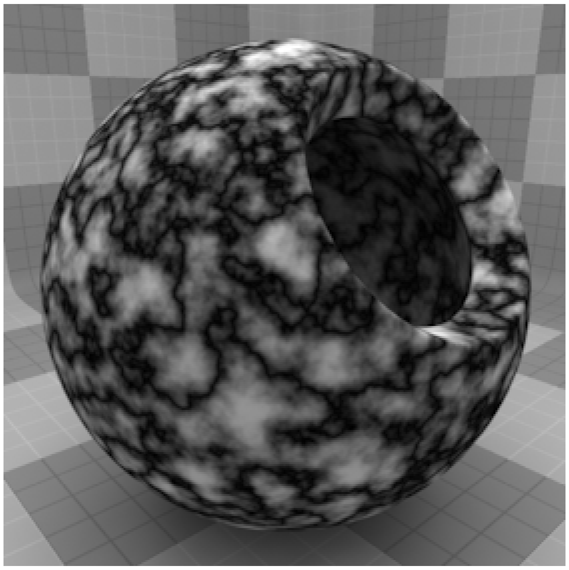
Foreground Alpha 100.0 %

☐ Foreground - Use Last Layer

Tiles - Tartan	
Horizontal Mask	Controls the mask for the horizontal threads.
Vertical Mask	Controls the mask for the vertical threads.
Stitch Thickness	Determines the thickness of the stitching.
Thread Layout	Specifies options for thread layout and provides five different options.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This,

	in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Terra

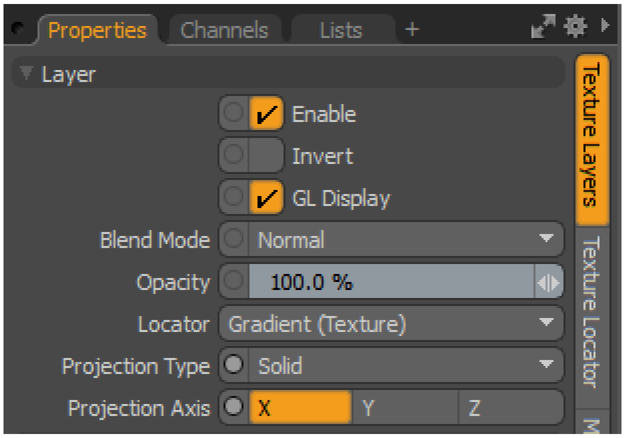


The **Terra** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Terra** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides two sets of noisy bumps with independent height controls. This is ideal for creating the hills and valleys of a surface for a landscape.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

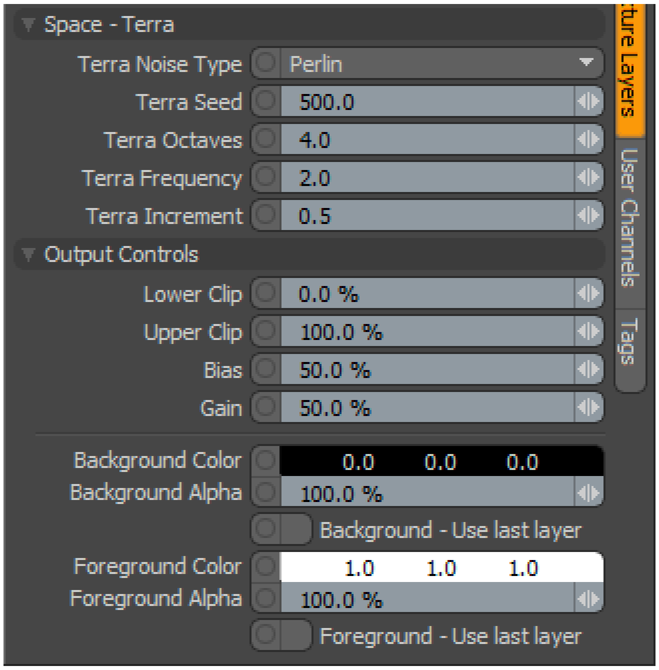
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Terra Properties



Space - Terra

Terra Noise Type: Perlin

Terra Seed: 500.0

Terra Octaves: 4.0

Terra Frequency: 2.0

Terra Increment: 0.5

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

○ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

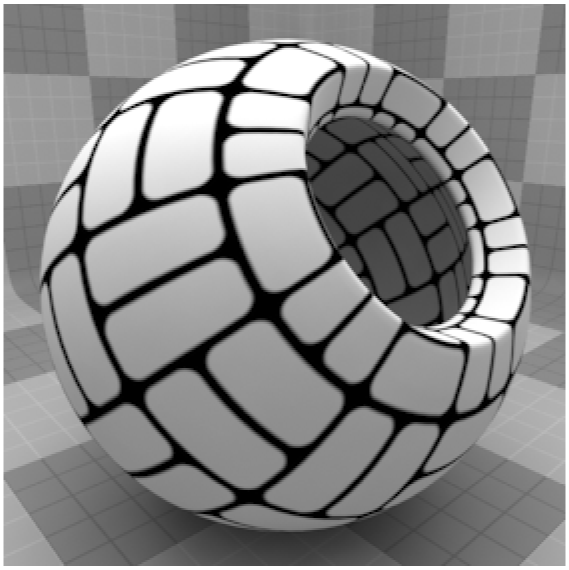
○ Foreground - Use last layer

Space - Terra	
Terra Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Terra Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Terra Octaves	Defines the number of iterations Modo uses to create the texture. As you increase

	this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Terra Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Terra Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV proces layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.

Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Tiler



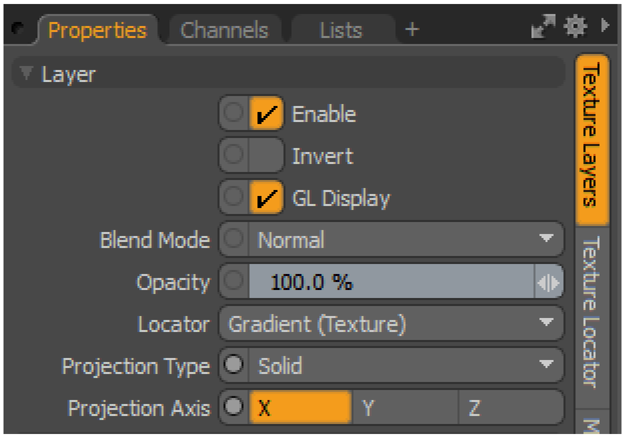
The **Tiler** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Tiler** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of brick (or tile) patterns where the tiles have rounded corners. The **Tile Layout** list provides 28 different layouts.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

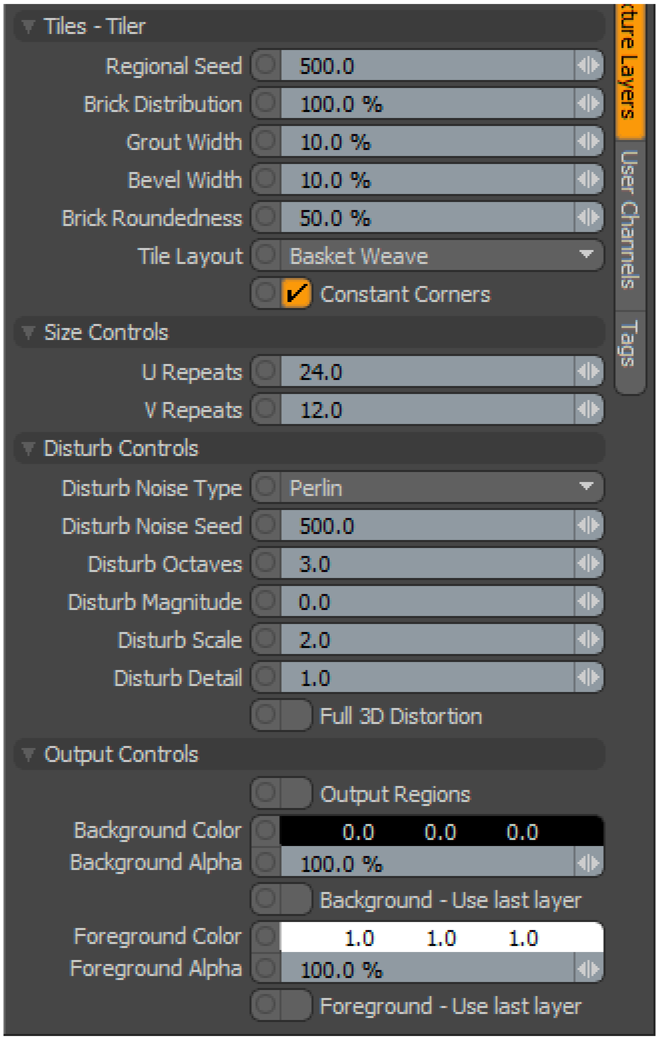
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Tiler Properties



Tiles - Tiler
Regional Seed 500.0
Brick Distribution 100.0 %
Grout Width 10.0 %
Bevel Width 10.0 %
Brick Roundedness 50.0 %
Tile Layout Basket Weave
☑ Constant Corners

Size Controls
U Repeats 24.0
V Repeats 12.0

Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
☐ Full 3D Distortion

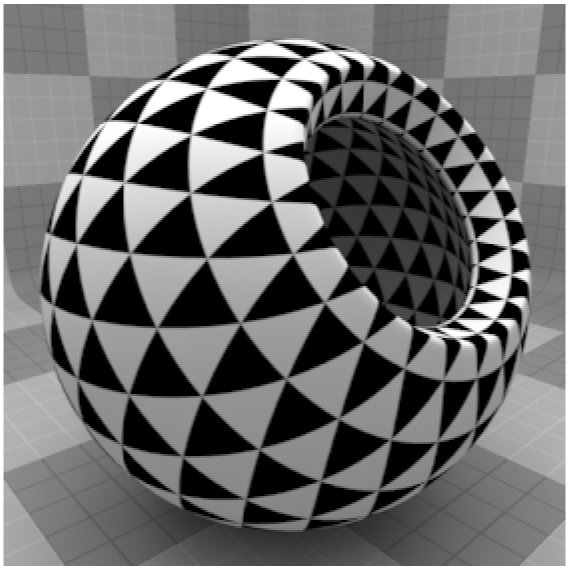
Output Controls
☐ Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Tiles - Tiler	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Brick Distribution	Determines the percentage of the surface that is covered by bricks.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Brick Roundedness	Determines the roundedness of the corners on individual tiles.
Tile Layout	Determines the shape and layout of the tiles. There are 28 options including: • **Basket Weave** • **Windmill** • **Trellis** • **Cobbled** • **Bricks 1**
Constant Corners	When enabled, normalizes the corners so that the larger bricks match the smaller bricks.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value**

	• **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground	Specifies the **Alpha** transparency of the **Foreground Color**.

Alpha	
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Tri Checks



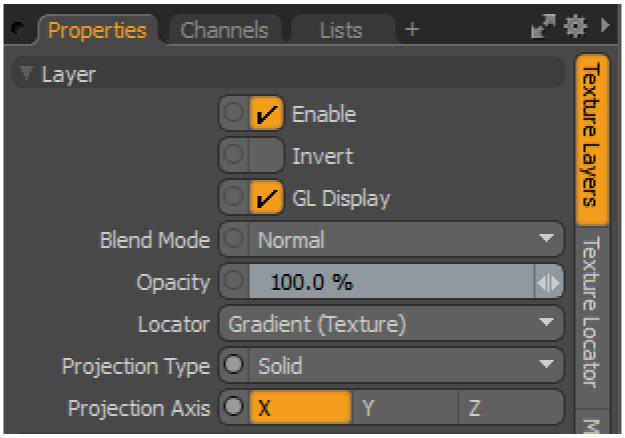
The **Tri Checks** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Tri Checks** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a standard triangular check pattern that is well-suited for flooring or clothing.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

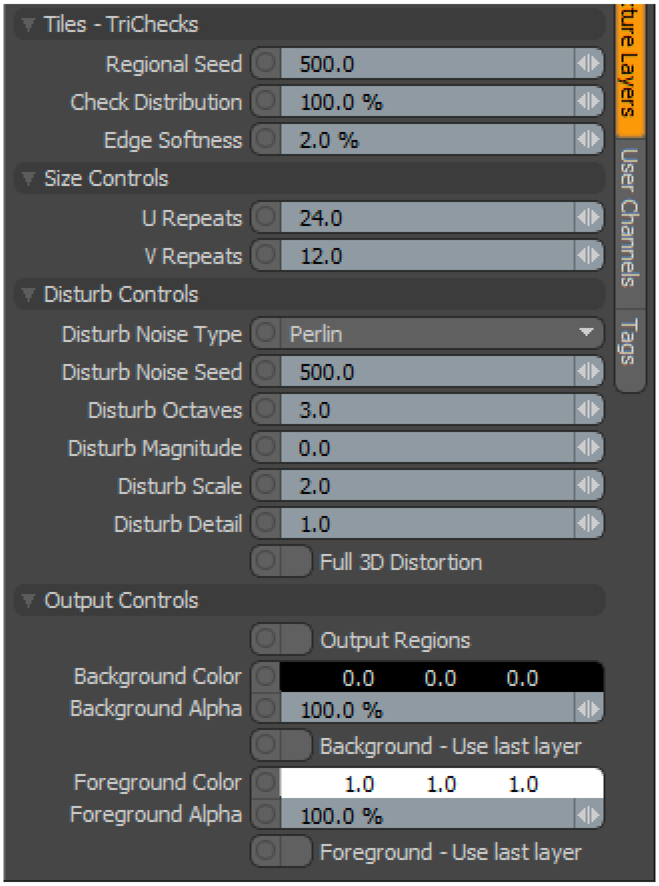
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Tri Checks Properties



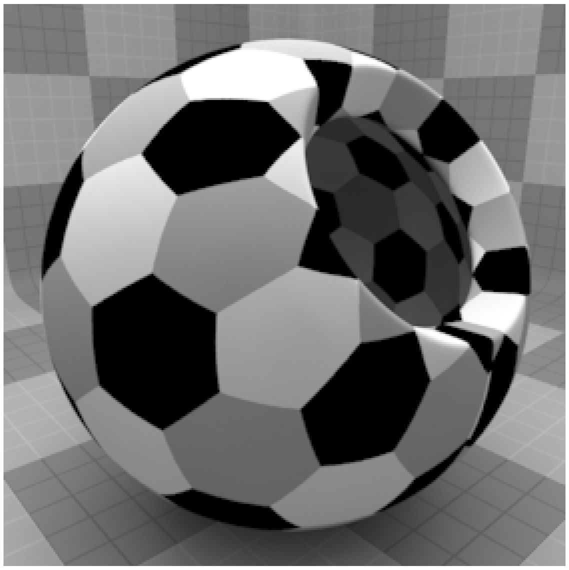
Tiles - TriChecks
Regional Seed 500.0
Check Distribution 100.0 %
Edge Softness 2.0 %
Size Controls
U Repeats 24.0
V Repeats 12.0
Disturb Controls
Disturb Noise Type Perlin
Disturb Noise Seed 500.0
Disturb Octaves 3.0
Disturb Magnitude 0.0
Disturb Scale 2.0
Disturb Detail 1.0
Full 3D Distortion
Output Controls
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Tiles - Tri Checks	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when

	you want items to retain the same variations.
Check Distribution	Determines the percentage of the surface that is covered by checks.
Edge Softness	Controls the blurring or softness of the check edges.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.

Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Tri Hexes



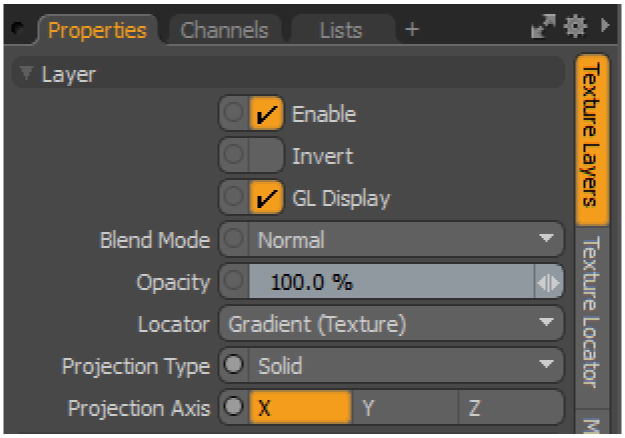
The **Tri Hexes** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Tri Hexes** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a tri-colored hexagonal check pattern that is well-suited for flooring or clothing.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

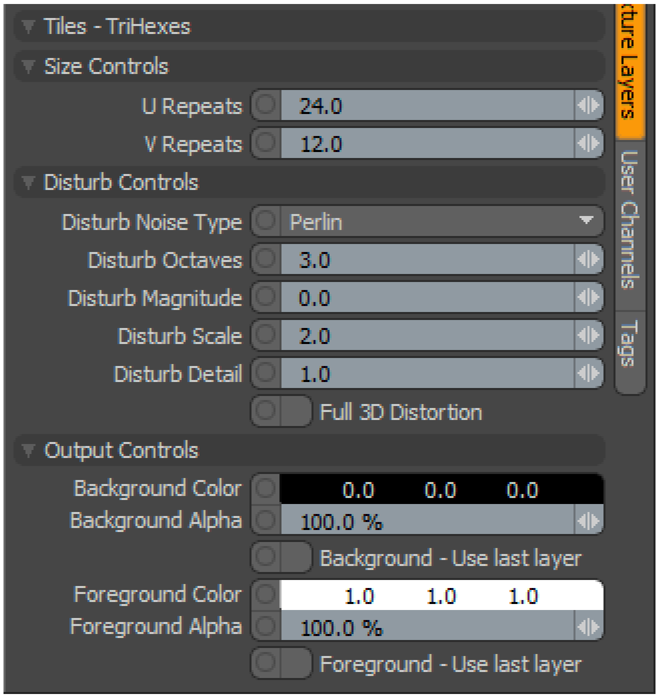
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Tri Hexes Properties



Tiles - TriHexes

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

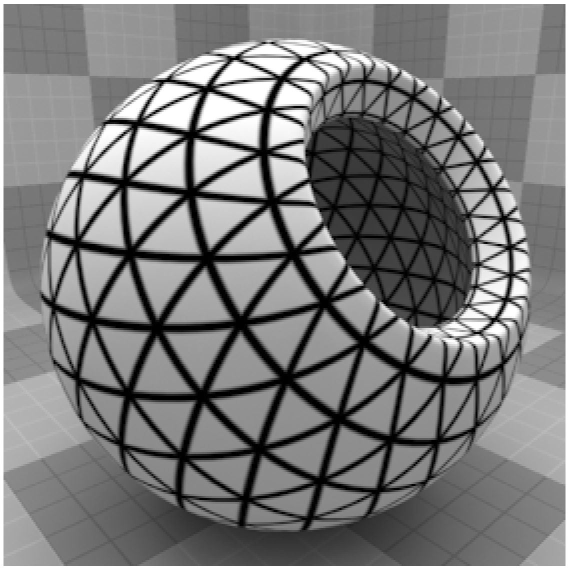
Foreground Alpha 100.0 %

Foreground - Use last layer

Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Tri Tile



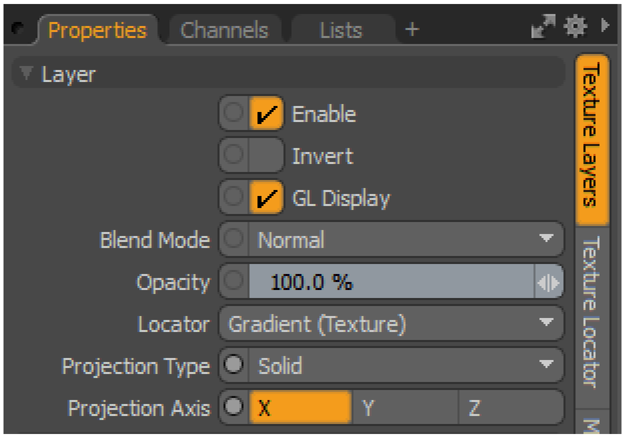
The **Tri Tile** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Tri Tile** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a tile pattern of triangles with variable grout, tile edges, and tile height. You can stagger the tiles and remove random tiles.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

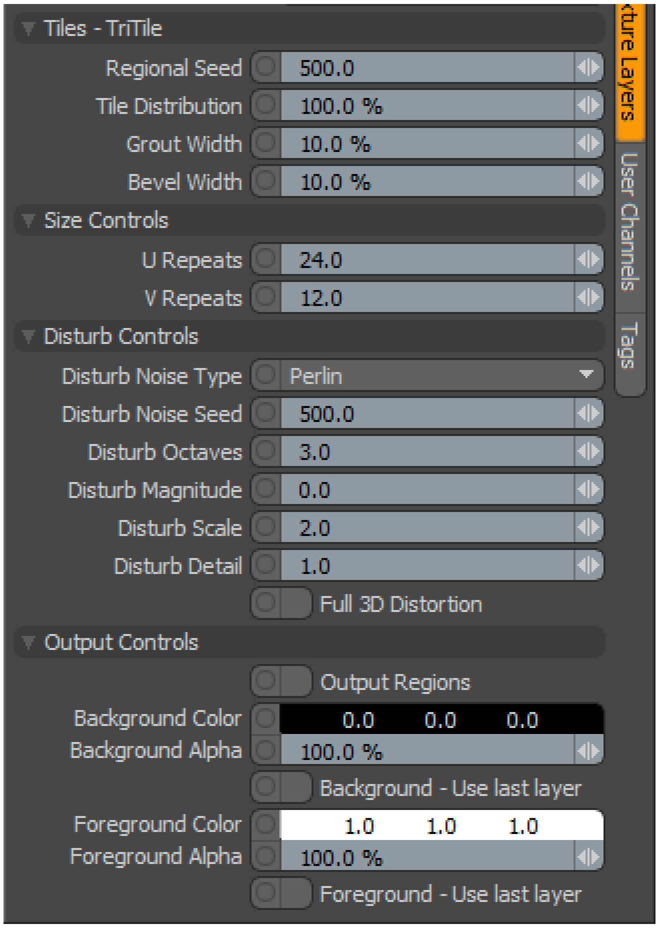
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Tri Tile Properties



Tiles - TriTile

Regional Seed 500.0

Tile Distribution 100.0 %

Grout Width 10.0 %

Bevel Width 10.0 %

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

Full 3D Distortion

Output Controls

Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

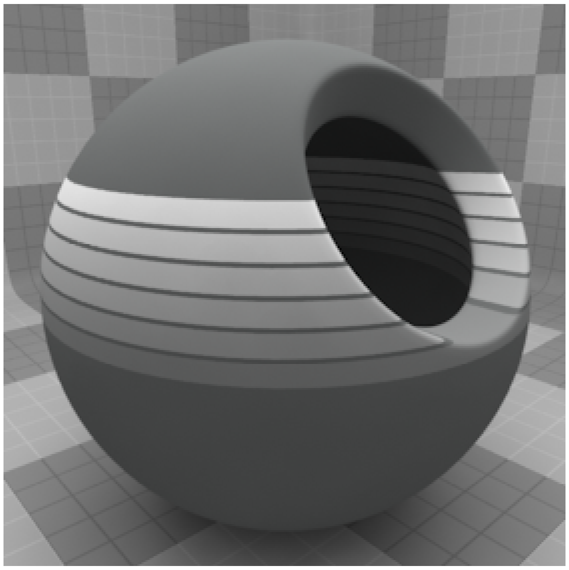
Foreground - Use last layer

Tiles - Tri Tile	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Tile Distribution	Determines the percentage of the surface that is covered by tiles.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).

V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This, in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.

Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

UVLEDs

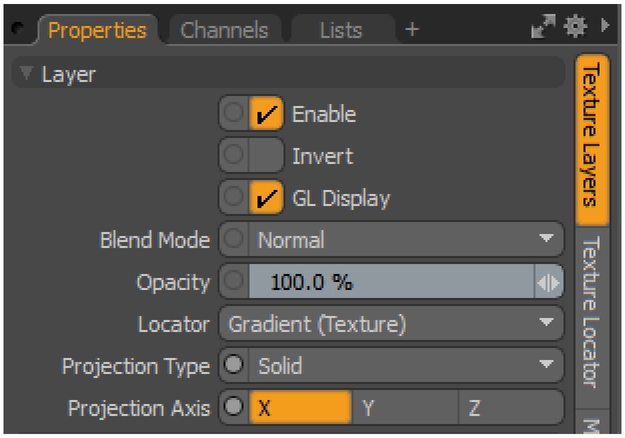


The **UVLEDs** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **UVLEDs** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a row of round or square colored LEDs. You can animate the display to provide a variety of blinking, rotating, or flashing LEDs.

> **Note:** For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

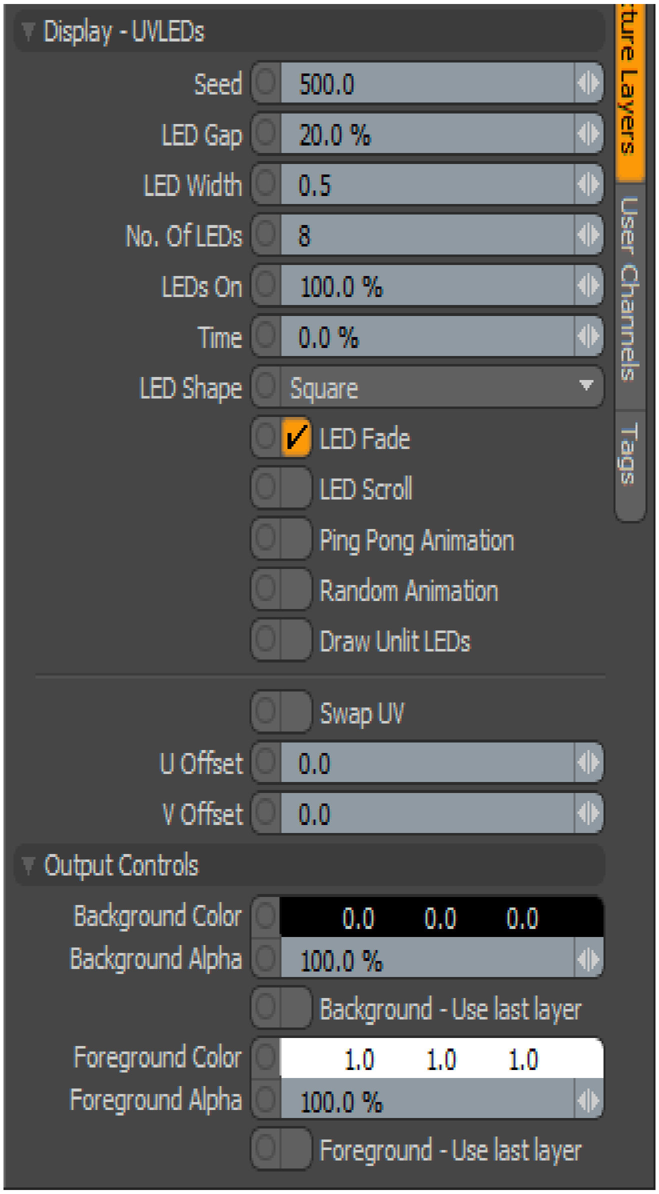
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture**

	Locator that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

UV LEDs Properties



Display - UVLEDs

Seed 500.0
LED Gap 20.0 %
LED Width 0.5
No. Of LEDs 8
LEDs On 100.0 %
Time 0.0 %
LED Shape Square
☑ LED Fade
☐ LED Scroll
☐ Ping Pong Animation
☐ Random Animation
☐ Draw Unlit LEDs

☐ Swap UV
U Offset 0.0
V Offset 0.0

Output Controls

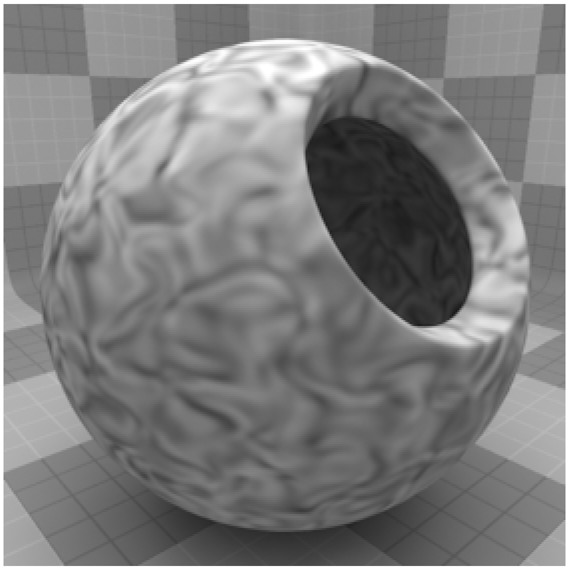
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
☐ Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
☐ Foreground - Use last layer

Option	Description
Layer	
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.

Option	Description
Invert	Inverts the colors (RGB values) for the layer to produce a photonegative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the Shader Tree, if present, or dims the effect of the layer, itself, on the surface.
Locator	Sets the association for the Texture Locator. Most texture layers have a Texture Locator that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but the need to do so is rare. Still, you may want multiple texture items to share a single locator.
Display - UVLEDs	
Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
LED Gap	Determines the spacing between the individual LEDs.
LED Width	Determines the width of the individual LEDs.
No. Of LEDs	Specifies the total number of LEDs in the display.
LEDs On	Specifies the number of LEDs that Modo draws. A value of **100%** shows all LEDs. This setting is useful for controlled animations.
Time	Controls the time value of the LEDS when used with animation (such as the **Ping Pong Animation** or the **Random Animation**). The timer goes from **0%** to **100%**.
LED Shape	Specifies the shape of the LEDs: **Round** or **Square**.
LED Fade	When enabled, fades the LEDs from beginning to end. When disabled, maintains a consistent color based on the foreground color.
LED Scroll	When enabled, creates an animation that cycles through the LEDs from top to

Option	Description
	bottom based on the **Time** value.
Ping Pong Animation	When enabled, creates an animation that cycles the LEDs on and off from the bottom to the top and then back to the bottom based on the **Time** value.
Random Animation	When enabled, creates an animation that randomly cycles through the LEDs based on the **Time** value.
Draw Unlit LEDs	When enabled, draws all the unlit LEDs as determined by the **LEDs On** value.
Swap UV	When enabled, swaps the UV coordinates of the display text.
U Offset	Determines **U** (horizontal) offset of the display text.
V Offset	Determines **V** (vertical) offset of the display text.
Output Controls	
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Vector Bozo

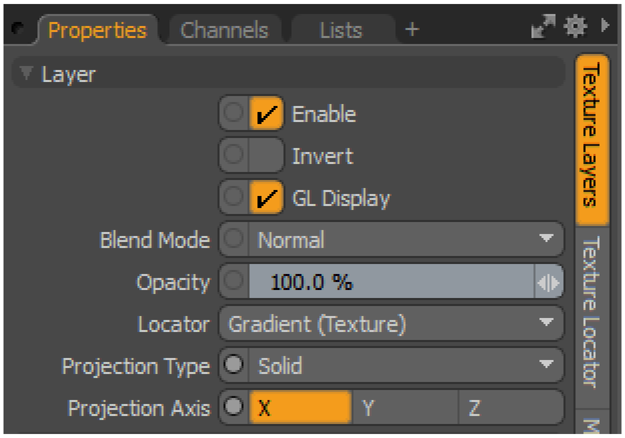


The **Vector Bozo** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Vector Bozo** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a noise pattern with vector-based disturbances.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

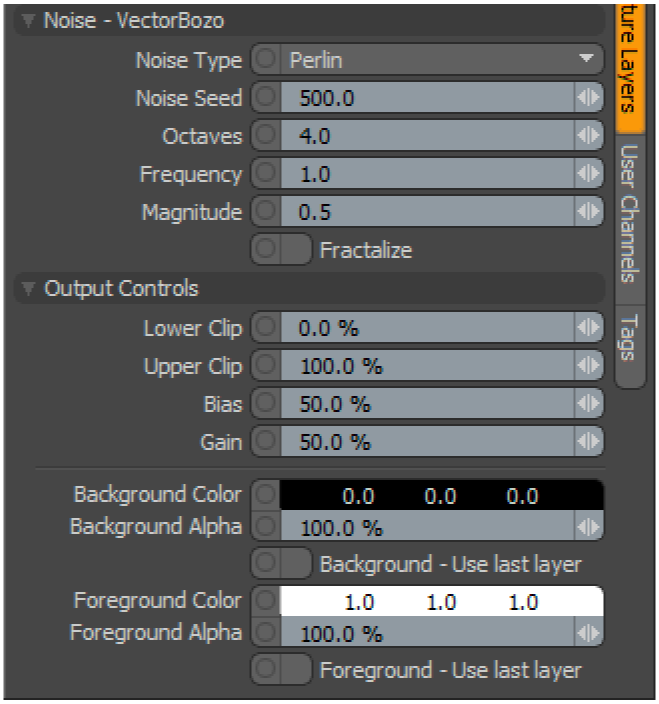
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Vector Bozo Properties

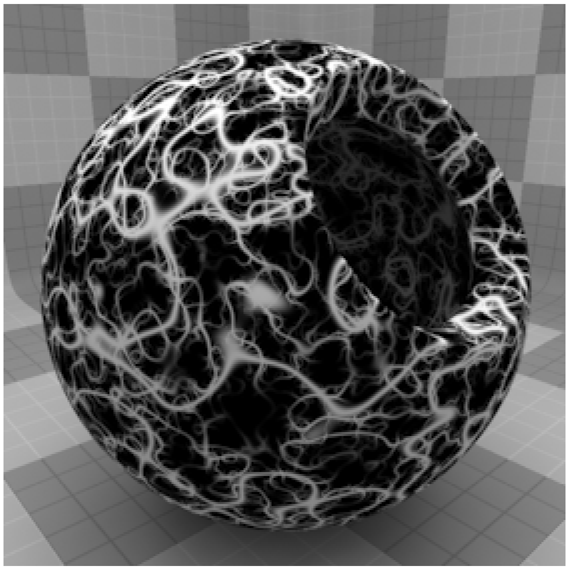

Noise - VectorBozo
Noise Type: Perlin
Noise Seed: 500.0
Octaves: 4.0
Frequency: 1.0
Magnitude: 0.5
Fractalize
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
Foreground - Use last layer

Noise - Vector Bozo	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Magnitude	Controls the strength of each iteration of the noise. Increasing this value tightens the delineation between the background and foreground color.
Fractalize	When enabled, uses the fractalize algorithm that is based on Fractal Brownian Motion (FBM).
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.

Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

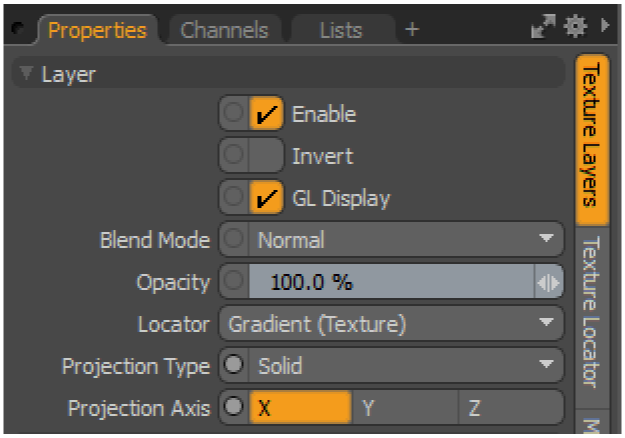
Veins



The **Veins** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Veins** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides vein-like patterns and is ideal for texturing eyeballs or skin.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

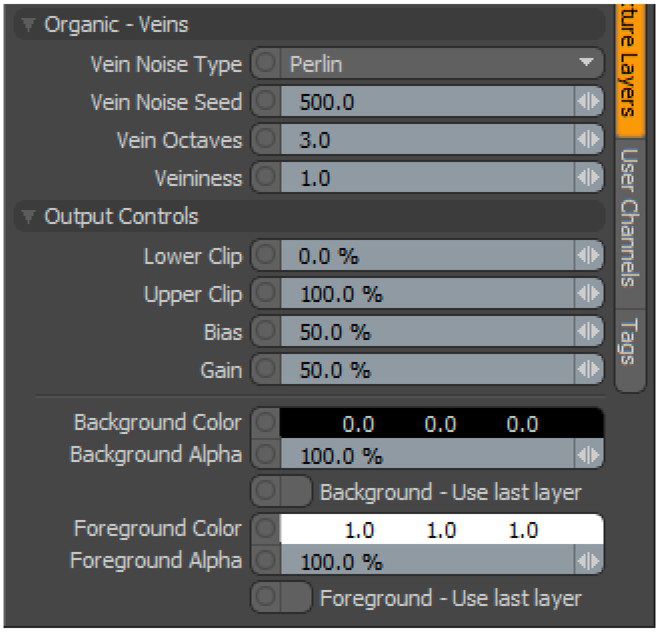
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Veins Properties



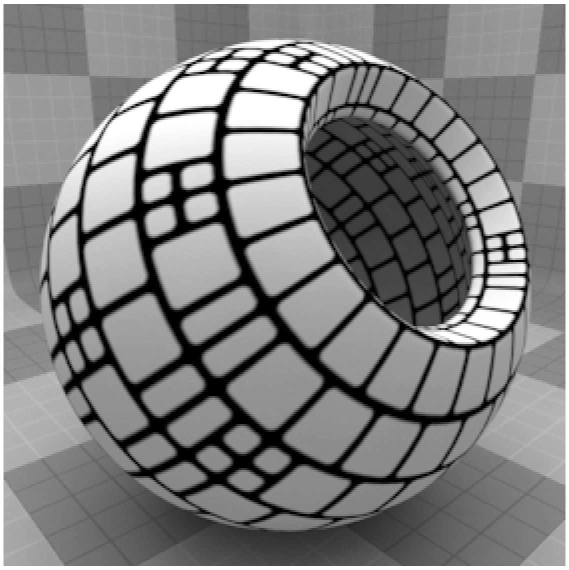
Organic - Veins
Vein Noise Type: Perlin
Vein Noise Seed: 500.0
Vein Octaves: 3.0
Veininess: 1.0
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
○ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
○ Foreground - Use last layer

Organic - Veins	
Vein Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Vein Seed	Specifies the initial number Modo uses when generating the procedural values.

	Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Vein Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Veininess	Controls how thick and layered the veins appear.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.

Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Wall



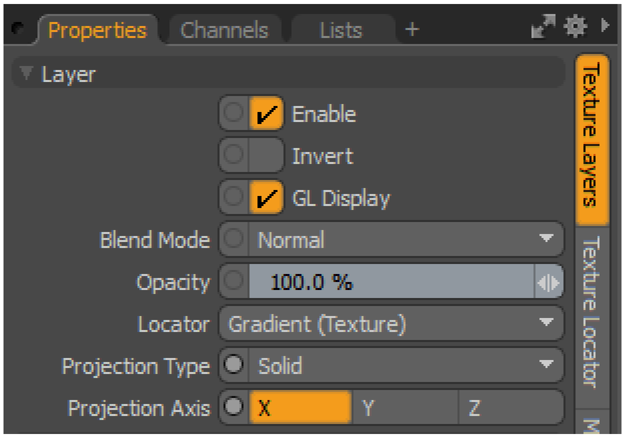
The **Wall** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Wall** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture provides a brick wall-type pattern with randomly sized bricks. You can adjust the number of patterns for a variety of wall types.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>

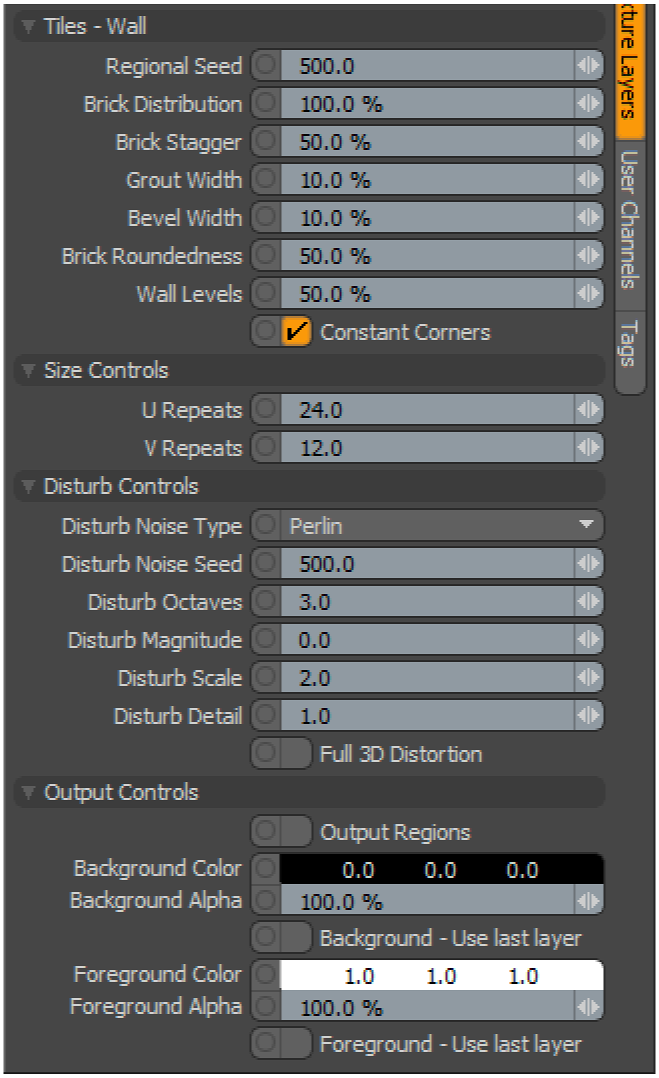


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Wall Properties



Tiles - Wall

Regional Seed 500.0

Brick Distribution 100.0 %

Brick Stagger 50.0 %

Grout Width 10.0 %

Bevel Width 10.0 %

Brick Roundedness 50.0 %

Wall Levels 50.0 %

☑ Constant Corners

Size Controls

U Repeats 24.0

V Repeats 12.0

Disturb Controls

Disturb Noise Type Perlin

Disturb Noise Seed 500.0

Disturb Octaves 3.0

Disturb Magnitude 0.0

Disturb Scale 2.0

Disturb Detail 1.0

☐ Full 3D Distortion

Output Controls

☐ Output Regions

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

☐ Background - Use last layer

Foreground Color 1.0 1.0 1.0

Foreground Alpha 100.0 %

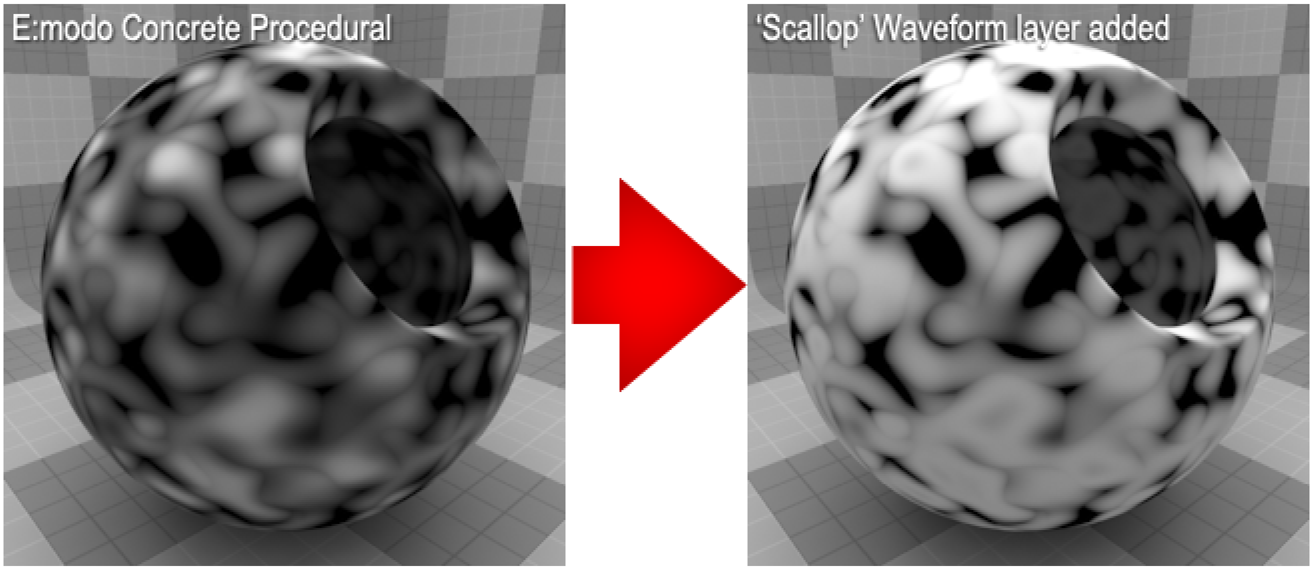
☐ Foreground - Use last layer

Tiles - Wall	
Regional Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Brick Distribution	Determines the percentage of the surface that is covered by bricks.
Brick Stagger	Determines how staggered the rows of bricks are in the texture. When you set this to a low value, the bricks line up vertically. The rows of bricks become more staggered

	as you increase the value.
Grout Width	Determines the amount of grout between the tiles.
Bevel Width	Determines the width of the bevel on the tiles.
Brick Roundedness	Determines the width of the bevel on the tiles.
Wall Levels	Determines how Modo sizes and places the bricks in the wall. At low values the bricks are square in shape and set in even rows and columns. As you increase the value, the brick sizes vary, and Modo places them in offset patterns.
Constant Corners	When enabled, normalizes the corners so that the larger bricks match the smaller bricks.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Disturb Controls - Alters the look of the generated texture. The noise layer distorts the texture based on the **Disturb Magnitude**.	
Disturb Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Disturb Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Disturb Octaves	Specifies the number of layers of noise Modo uses when distorting the texture. This,

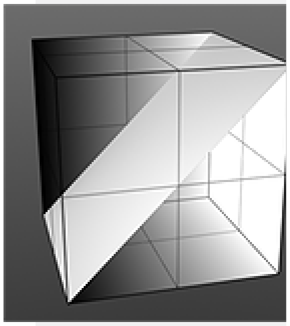
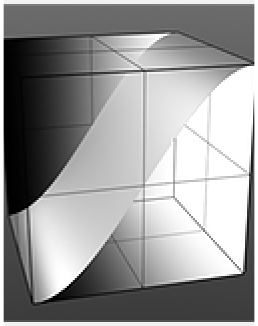
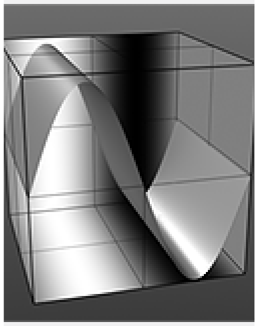
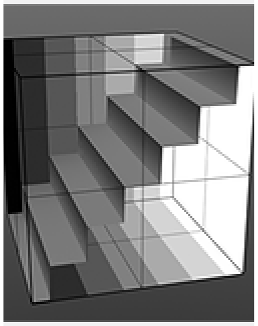
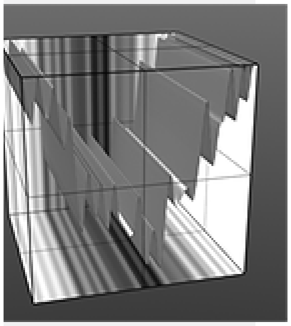
	in effect, produces greater detail and is similar to the **Patch Levels** setting.
Disturb Magnitude	Specifies the strength of the distortion effect on the base texture.
Disturb Scale	Specifies the size (scale) of the distortion effect on the base texture.
Disturb Detail	Specifies the step size that Modo uses between each iteration of the noise to disturb the texture.
Full 3D Distortion	Turns on full 3D distortion, which yields better results, but a full 3D distortion takes longer to render.
Output Controls	
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Waveforms

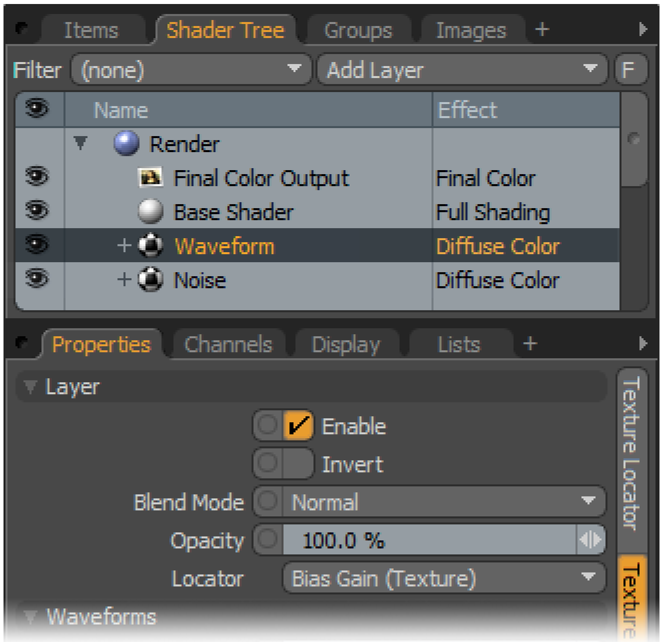


Each Enhance:Modo **Waveform** layer is a special type of Shader Tree layer that produces no results by itself because a **Waveform** layer modifies a similar texture layer below it in the Shader Tree hierarchy. For example, a **Waveform** layer would only affect a texture layer (such as a **Diffuse** layer) if the **Waveform** layer has its effect set to **Diffuse** and is above the texture layer. To affect a **Displacement** layer, you must change the **Waveform** layer's effect to **Displacement** as well. If you have multiple layers of the same type blended together, the **Waveform** layer only affects the nearest visible layer below it. Each **Waveform** type produces its own independent result.

If you are familiar with Photoshop's **Curves** functionality, then you might find it easier to understand the **Waveform** function. The **Waveform** function is a means to manipulate the values of the texture layer (just like **Curves** affect values in an image). If you make a graph of the input values versus the output values of a surface, the direct correlation of the input and output shows as a straight line. As you make adjustments to the graph, the values on the graph change. These adjusted values can modify the shades of gray across a procedural texture or the shape of a displacement map. In the examples, the image on the far left shows a displaced plane with an applied gradient ramp and offset that is the same height as its width to show a ramp at a 45-degree angle. If you apply various **Waveform** layers to the gradient, the other images show how the **Waveform** layers affect the values across the gradient ramp to illustrate the shaping ability of the **Waveform** function.

				
Displaced Plane	S-Curve Waveform	Sine Waveform	Stairs Waveform	Noise Waveform

To work add a **Waveform** layer, select the Material Group item (for the texture) in the Shader Tree. Then, in the **Add Layer** list, select one of the available waveform options (**Add Layer > Enhance:Modo Textures >** ***a waveform***). For additional information about adding layers and working with Shader Tree item layers, see the Shader Tree topic. Each waveform produces its own unique results. (Generally, the waveform's name suggests how it shapes the associated layer.) You can modify the attributes of the **Waveform** layer in its **Properties** panel.


Items
Shader Tree
Groups
Images
+
Filter
Add Layer
F
Properties
Channels
Display
Lists
+
Texture Locator
Texture

Filter (none)

Name	Effect
Render	
Final Color Output	Final Color
Base Shader	Full Shading
Waveform	Diffuse Color
Noise	Diffuse Color

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Locator: Bias Gain (Texture)

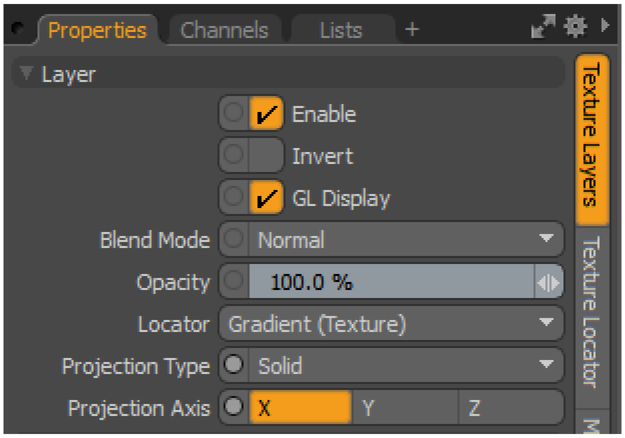
Waveforms

Option	**Description**
Layer	
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling

Option	Description
	visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a photonegative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the Shader Tree, if present, or dims the effect of the layer, itself, on the surface.
Locator	Sets the association for the Texture Locator. Most texture layers have a Texture Locator that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but the need to do so is rare. Still, you may want multiple texture items to share a single locator.
Waveform-Specific Controls	
	See below these tables for waveform-specific options and examples of those settings.

Layer Properties

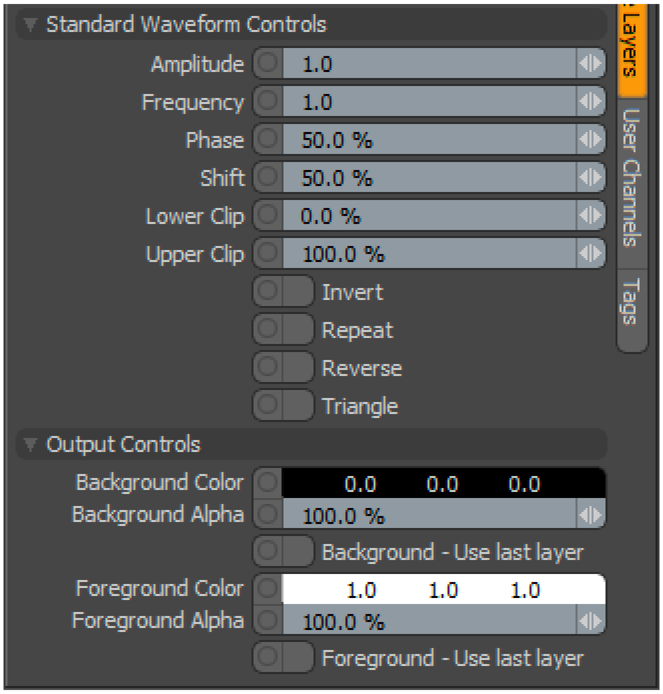
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X (selected) Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Waveform Properties



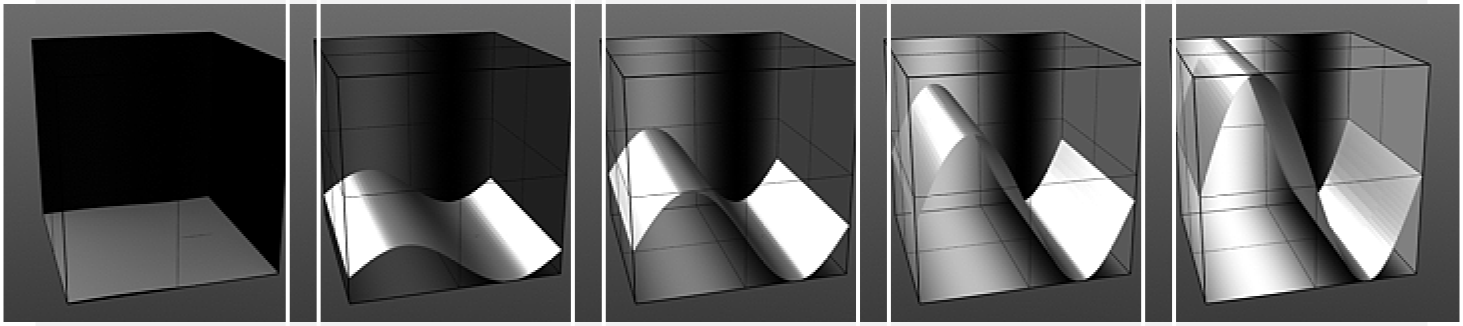
Standard Waveform Controls
Amplitude 1.0
Frequency 1.0
Phase 50.0 %
Shift 50.0 %
Lower Clip 0.0 %
Upper Clip 100.0 %
Invert
Repeat
Reverse
Triangle

Output Controls
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

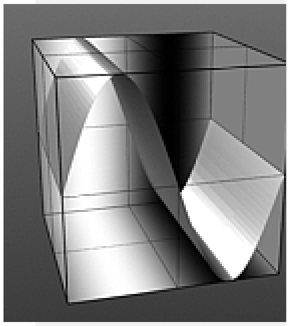
Standard Waveform Controls	
Amplitude	Controls the maximum value of the waveform and scales all other values in relation to this maximum. Click to view examples of various Amplitude settings.
Frequency	Controls the number of waveform occurrences in a single cycle. A waveform only cycles multiple times when you enable **Repeat**. Click to view examples of various Frequency settings.
Phase	Moves the waveform forwards and backwards. Click to view examples of various Phase settings.
Shift	Moves the waveform up and down. Click to view examples of various Shift settings.
Lower Clip	Specifies a clip level for the **Background Color** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture. Click to view examples of various Lower Clip settings.
Upper Clip	Specifies a clip level for the **Foreground Color** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to

	extend or lessen the range of values for the texture. Click to view examples of various Upper Clip settings.
Invert	When enabled, flips the waveform vertically.
Repeat	When enabled, repeats the waveform and useful when if you set the **Frequency** value higher than 1.
Reverse	When enabled, flips the waveform horizontally.
Triangle	When enabled, mirrors the waveform on its own axis.
Output Controls	
Background Color	Specifies the **Color** of the texture's background area, which ramps toward the **Foreground Color**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color	Specifies the **Color** of the texture's foreground area, which ramps toward the **Background Color**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

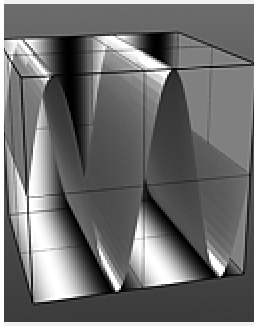
Examples of the Standard Waveform Controls



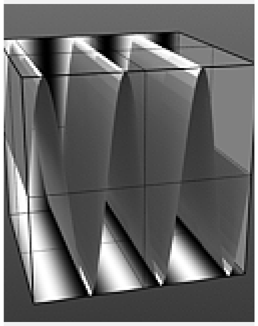
Amplitude 0% | Amplitude 25% | Amplitude 50% | Amplitude 75% | Amplitude 100%



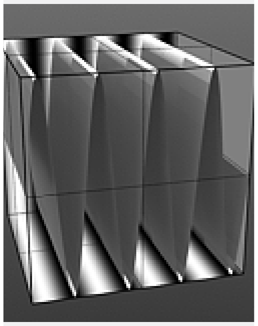
Frequency 1



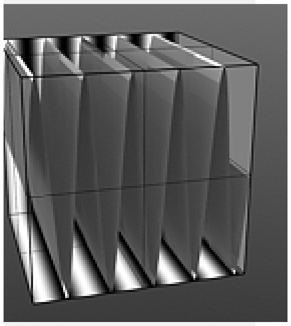
Frequency 2



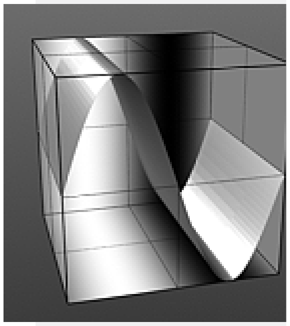
Frequency 3



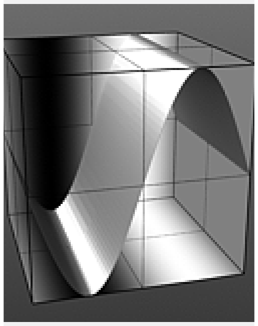
Frequency 4



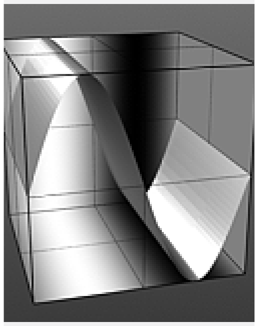
Frequency 5



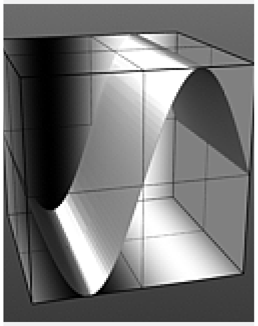
Phase 0%



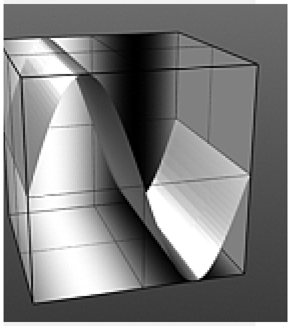
Phase 25%



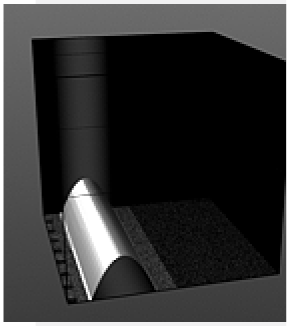
Phase 50%



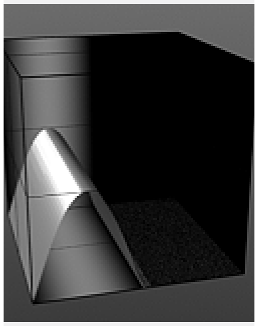
Phase 75%



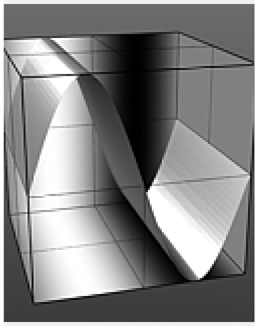
Phase 100%



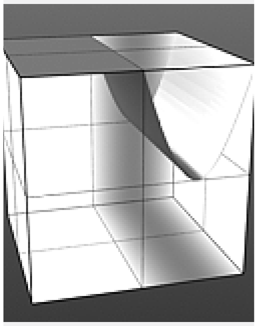
Shift 10%



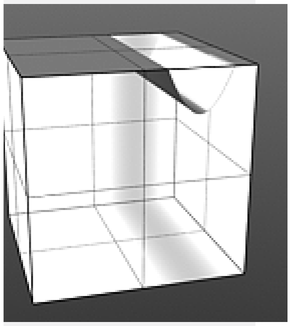
Shift 25%



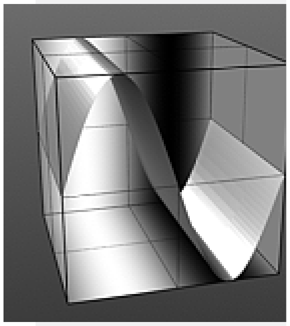
Shift 50%



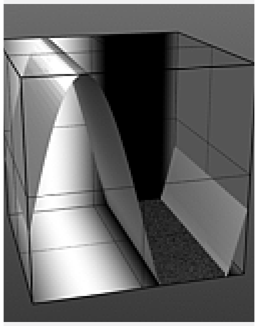
Shift 75%



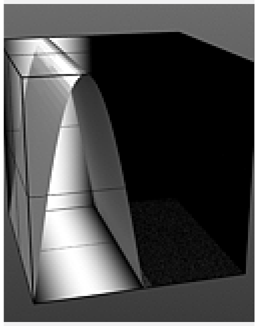
Shift 90%



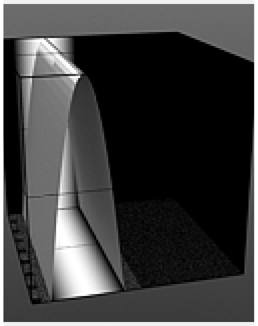
Lower Clip 0%



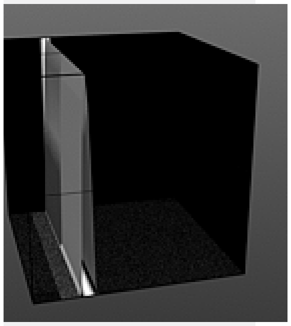
Lower Clip 25%



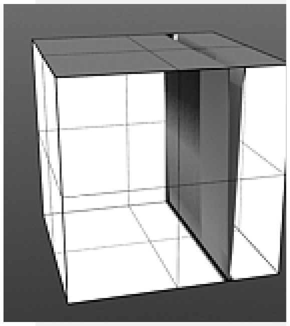
Lower Clip 50%



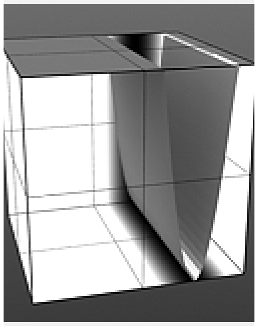
Lower Clip 75%



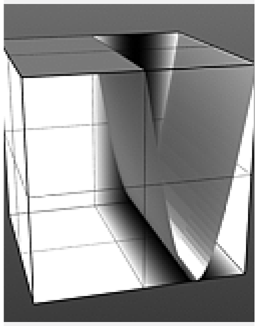
Lower Clip 100%



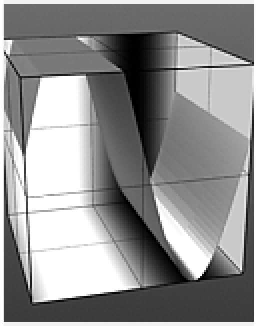
Upper Clip 0%



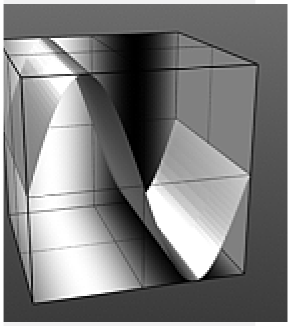
Upper Clip 25%



Upper Clip 50%



Upper Clip 75%

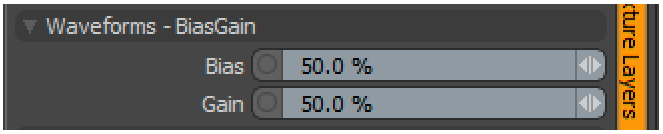


Upper Clip 100%

Waveform-Specific Options

The tables provide explanations for options common to all Enhance:Modo waveforms. This topic explains the options specific to each type of waveform.

Waveforms - Bias Gain

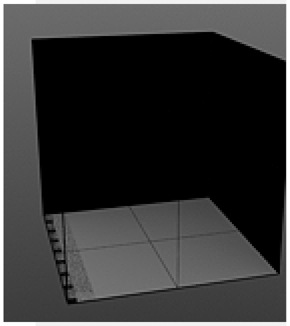


Waveforms - BiasGain

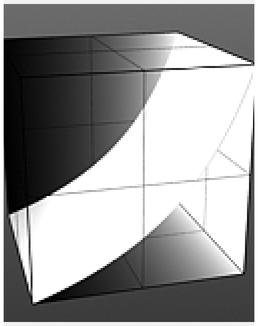
Bias 50.0 %

Gain 50.0 %

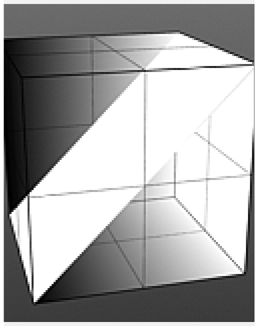
Bias: Affects the attenuation (the gradations) between the primary and secondary colors in a colored texture or the gray values in a grayscale texture. By increasing this value Modo favors the primary color (or value) over the secondary one. By decreasing the value Modo favors the secondary color (or value).



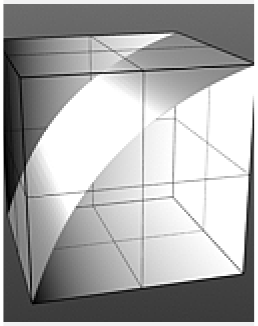
Bias 0%



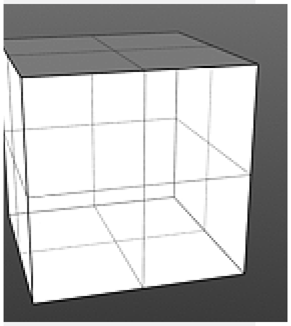
Bias 25%



Bias 50%

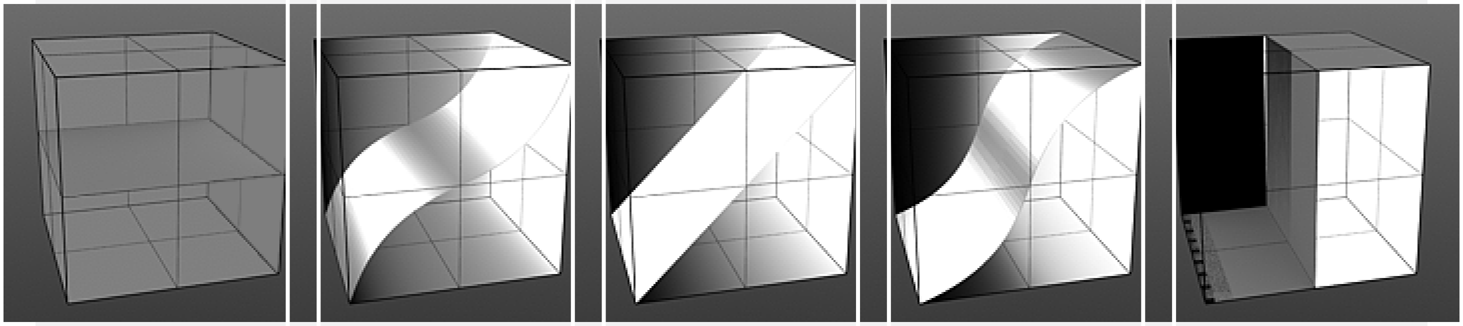


Bias 75%



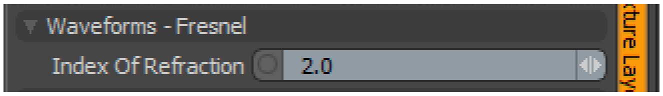
Bias 100%

Gain: Affects the falloff of the gradient ramp between the primary color (or value) and the secondary color (or value). This is similar to a contrast control. Setting the **Gain** to **100%** creates a very sharp contrast. Setting the value to **0%** reduces the contrast between the two to an almost imperceptible amount.



Gain 0% Gain 25% Gain 50% Gain 75% Gain 100%

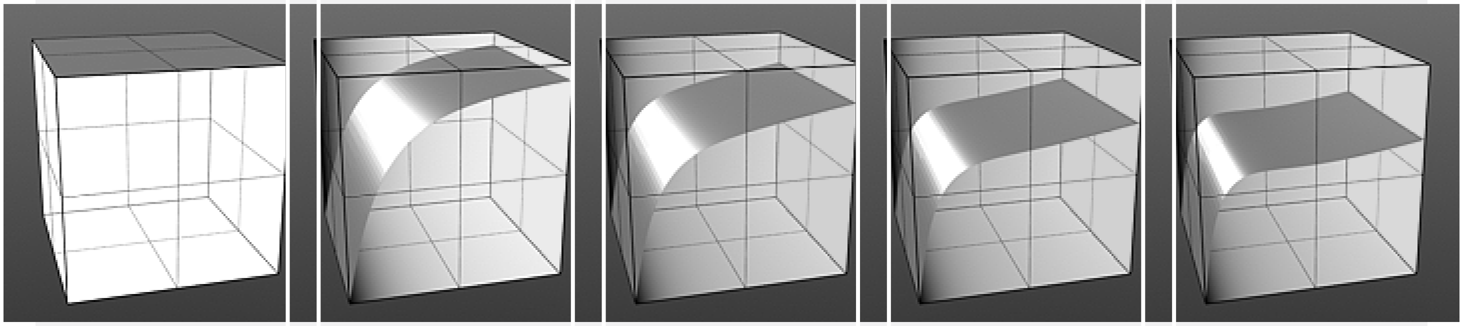
Waveforms - Fresnel



Waveforms - Fresnel

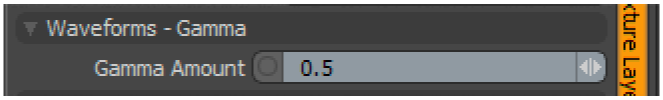
Index Of Refraction 2.0

Index of Refraction: Attenuates the Fresnel falloff rate based on this value. The Fresnel (pronounced Fruh-nell) effect describes refractive and reflective properties of a surface when the viewer's angle relative to the surface changes. The rate at which the change happens is expressed in the Fresnel function. When applied to a gradient (such as one applied to the incidence angle of a surface), the Fresnel function produces the appropriate falloff rate.



IOR 1.0 IOR 2.0 IOR 3.0 IOR 4.0 IOR 5.0

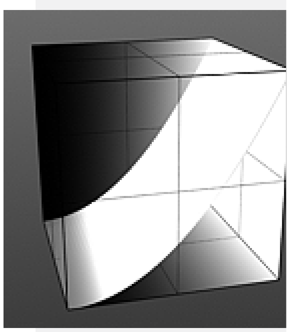
Waveforms - Gamma



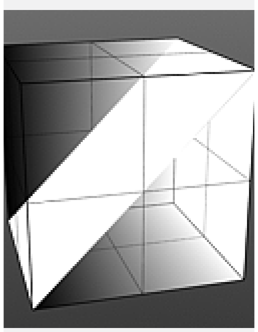
Waveforms - Gamma

Gamma Amount 0.5

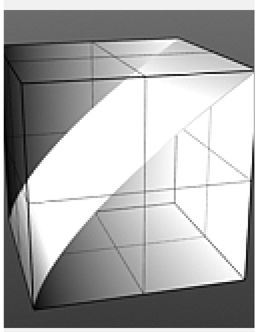
Gamma Amount: Manipulates the values across a spectrum. Like the Fresnel **Index of Refraction** option, Gamma is a nonlinear adjustment function for manipulating the values across a spectrum with the curve is expressed as a mathematical function. When applied to a texture layer, adjustments to the mid-tone value (those between black and white favoring the darker side) match any gamma adjustment values applied to image map layers directly. A value of **1** produces no results because the output matches that of the input. Values **less than 1** darken the gradient; values **greater than 1** lighten it.



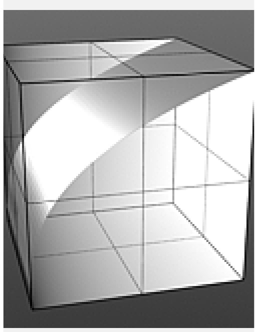
Gamma 0.5



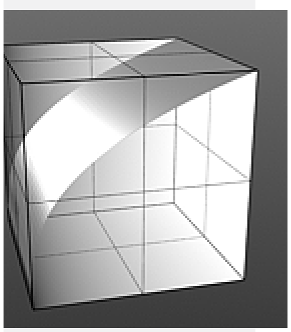
Gamma 1.0



Gamma 1.5

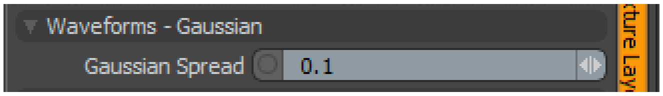


Gamma 2.0



Gamma 3.0

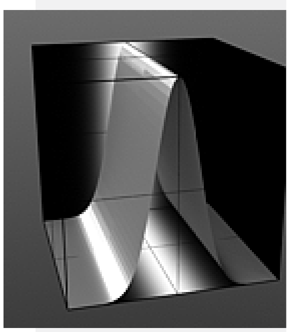
Waveforms - Gaussian



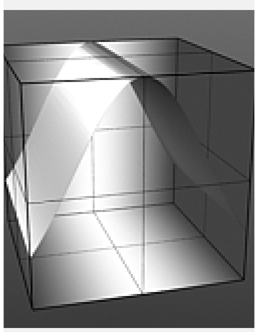
Waveforms - Gaussian

Gaussian Spread 0.1

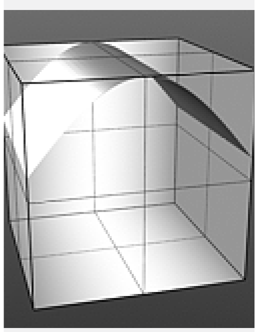
Gaussian Spread: Adjusts the values of a graduated texture layer based on a Gaussian distribution. A Gaussian distribution is a mathematical function that determines the spread of a number of values across a spectrum.



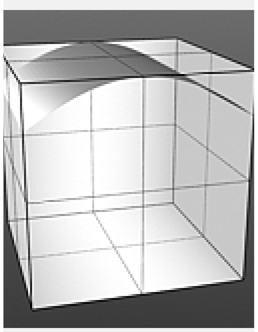
Spread 0.1



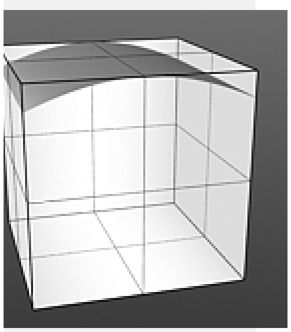
Spread 0.3



Spread 0.5

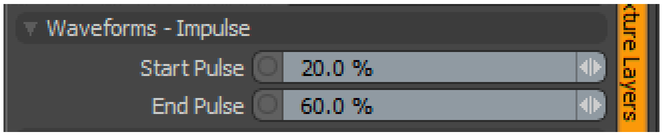


Spread 0.75



Spread 1.0

Waveforms - Impulse



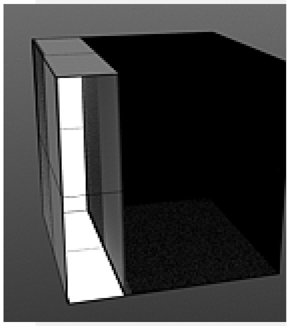
Waveforms - Impulse

Start Pulse 20.0 %

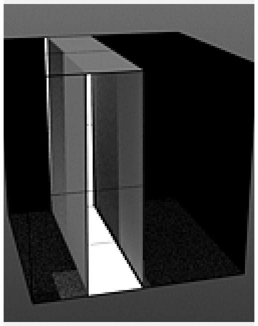
End Pulse 60.0 %

Start Pulse: Clamps the values of the gradient at the user-defined **Start** position and maximizes all values in between based on both the **Start Pulse** and **End Pulse** values.

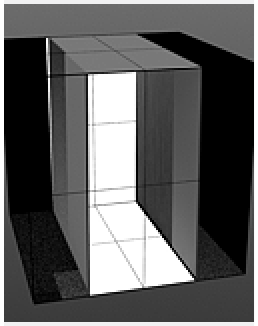
End Pulse: Clamps the values of the gradient at the user-defined **End** position and maximizes all values in between based on both the **Start Pulse** and **End Pulse** values.



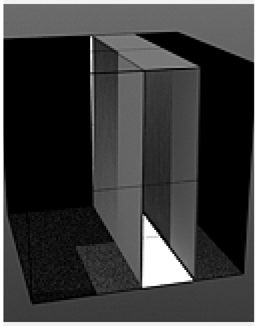
Start 0/End 25



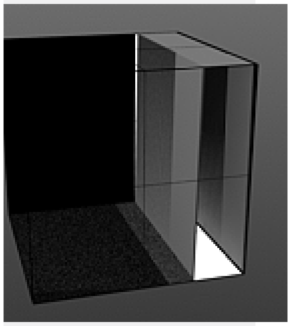
Start 25/End 50



Start 25/End 75

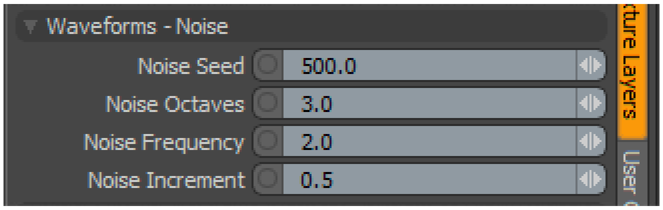


Start 50/End 75



Start 75/End 100

Waveforms - Noise



Waveforms - Noise

Noise Seed 500.0

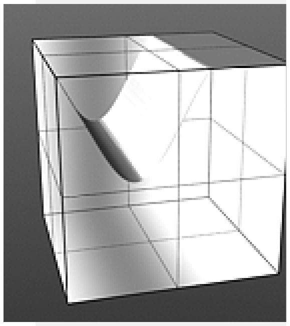
Noise Octaves 3.0

Noise Frequency 2.0

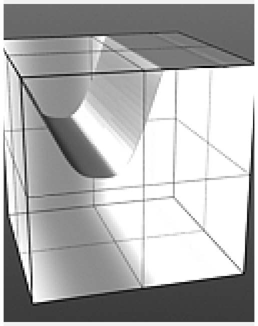
Noise Increment 0.5

Noise Seed: Specifies the initial number Modo uses when generating values. Different **Seed** values produce dramatically or subtly different results. Use the same **Seed** value when you want items to retain the same variations.

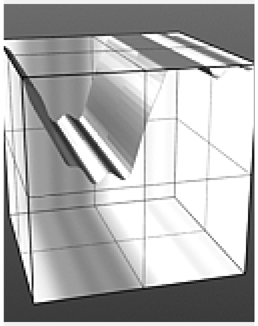
Noise Octaves: Determines the number of layers of noise that Modo combines to produce the final result. Higher values produce more randomized (and detailed) results.



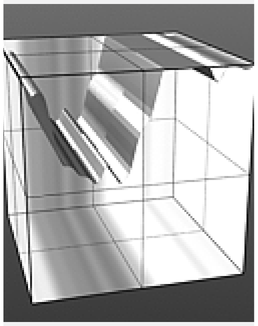
Octaves 1



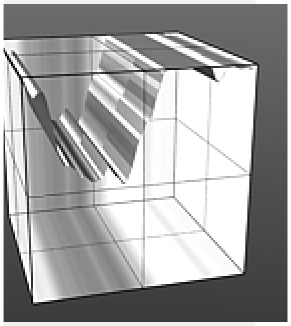
Octaves 2



Octaves 3

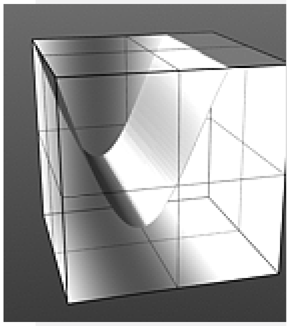


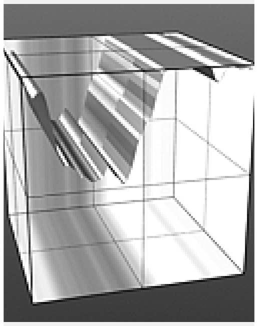
Octaves 4

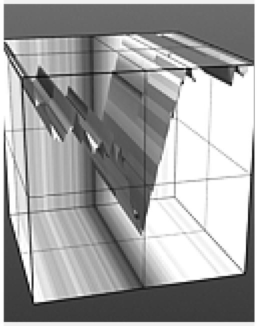


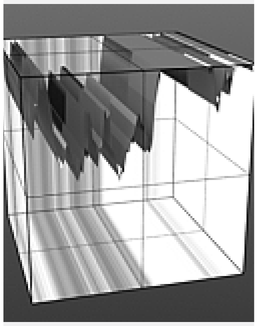
Octaves 5

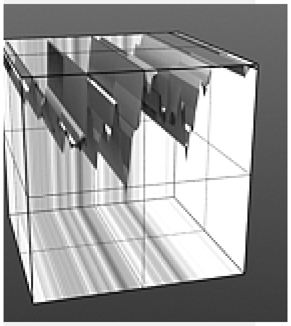
Noise Frequency: Determines the number of occurrences of the noise function when applied to the surface.


Frequency 1

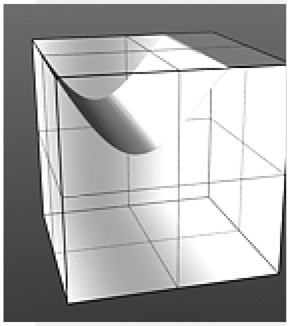

Frequency 2

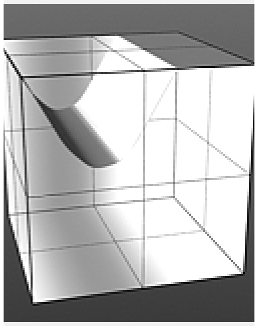

Frequency 3

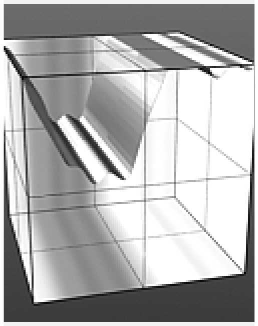

Frequency 4

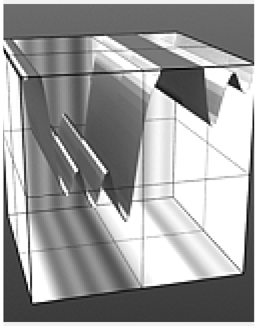

Frequency 5

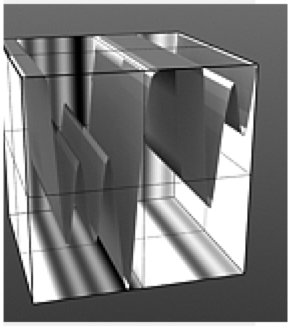
Noise Increment: Determines the number of noise bands across each Frequency occurrence.


Increment 0.0


Increment 0.25

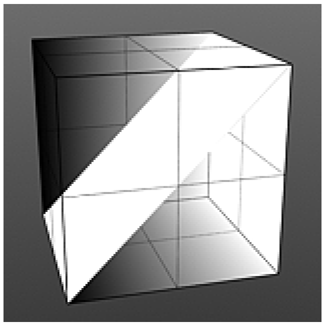

Increment 0.5


Increment 0.75


Increment 1.0

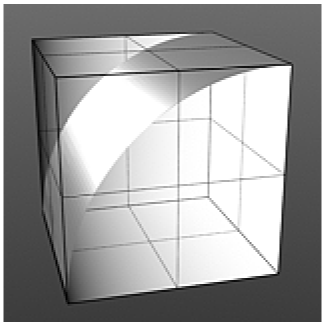
Waveforms - Ramp

Ramp has no user-controlled settings.

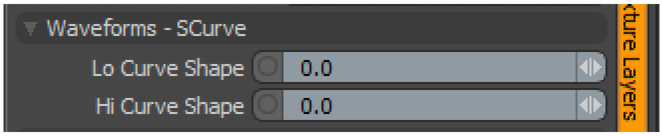


Waveforms - Rounded

Rounded has no user-controlled settings.



Waveforms - S-Curve

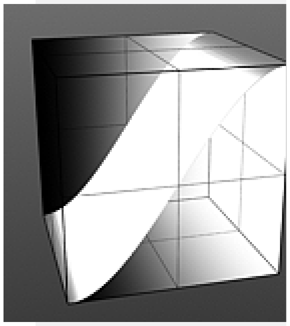


Waveforms - SCurve

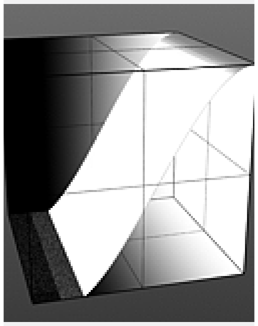
Lo Curve Shape 0.0

Hi Curve Shape 0.0

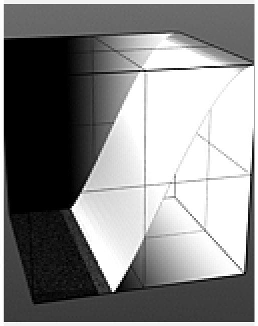
Lo Curve Shape: Determines the low-cutoff area for the gradient that the S-Curve waveform function produces.



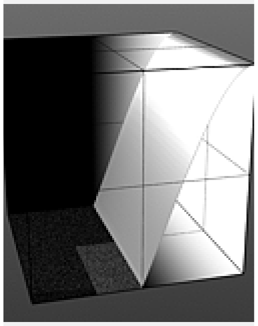
Lo Curve 0



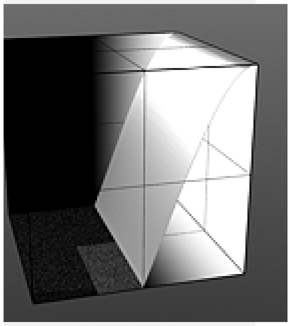
Lo Curve 1



Lo Curve 2

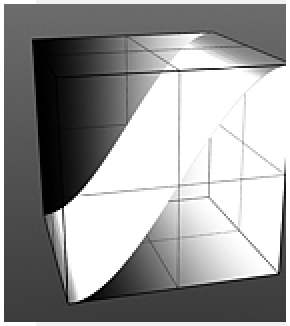


Lo Curve 3

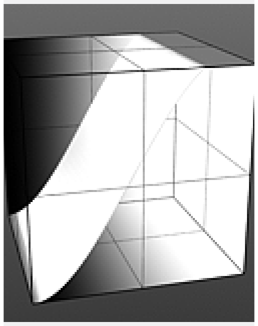


Lo Curve 4

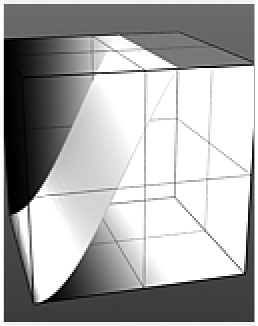
Hi Curve Shape: Determines the high-cutoff area for the gradient that the S-Curve waveform function produces.



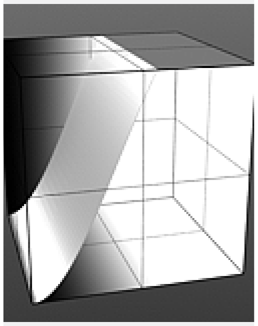
Hi Curve 0



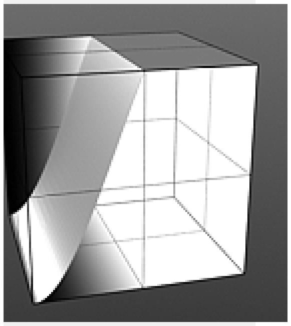
Hi Curve 1



Hi Curve 2

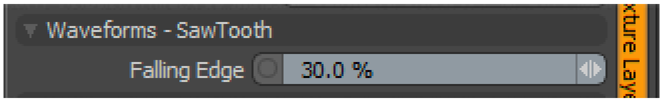


Hi Curve 3



Hi Curve 4

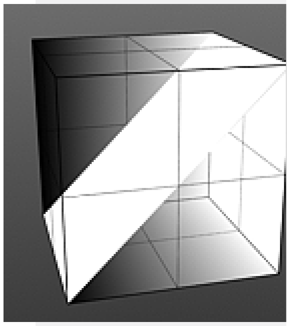
Waveforms - Sawtooth



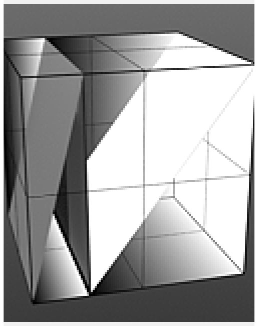
Waveforms - SawTooth

Falling Edge 30.0 %

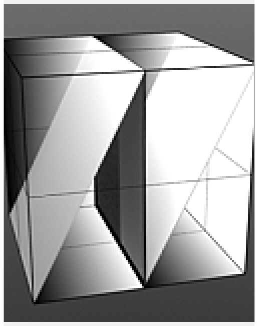
Falling Edge: Determines the position for the sudden minimum value drop in the graduated spectrum.



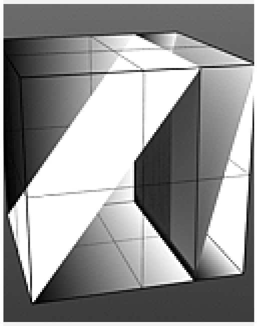
Falling Edge 0



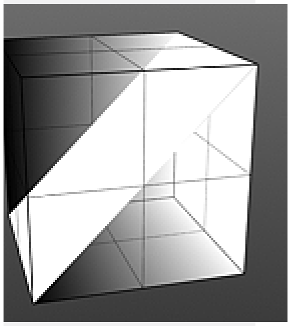
Falling Edge 25



Falling Edge 50



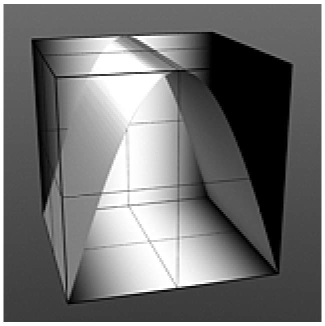
Falling Edge 75



Falling Edge 100

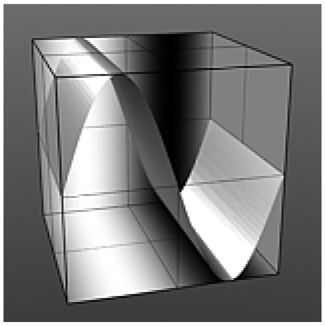
Waveforms - Scallop

Scallop has no user-controlled settings.

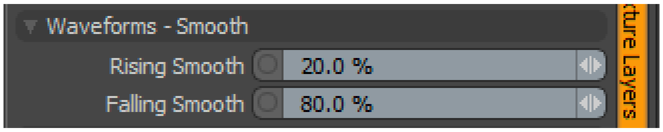


Waveforms - Sine

Sine has no user-controlled settings.



Waveforms - Smooth



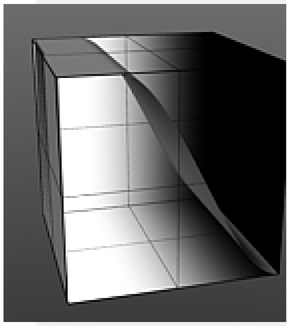
Waveforms - Smooth

Rising Smooth 20.0 %

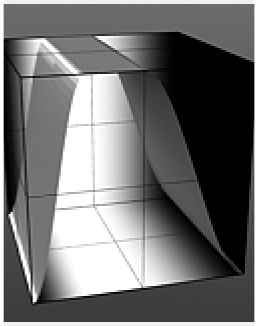
Falling Smooth 80.0 %

Rising Smooth: Specifies the minimum value for the a smooth S-shaped curve. The **Smooth** waveform produces a result similar to the **Impulse** waveform by maximizing values between a low and high end, but between the minimum value and the maximum value it produces a smooth S-shaped curve.

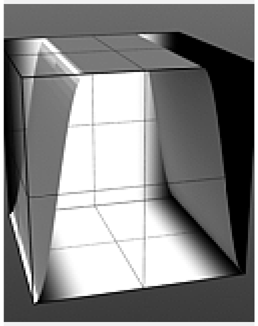
Falling Smooth: Specifies the maximum value for the a smooth S-shaped curve. The **Smooth** waveform produces a result similar to the **Impulse** waveform by maximizing values between a low and high end, but between the minimum value and the maximum value it produces a smooth S-shaped curve.



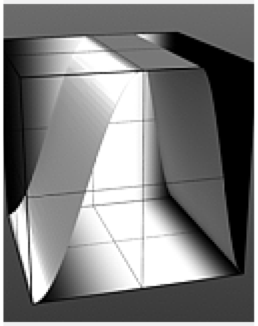
Rise 0/Fall 25



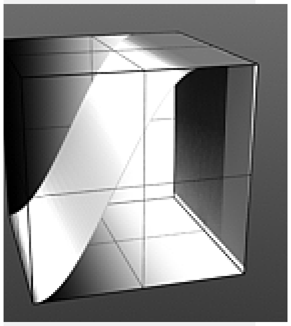
Rise 25/Fall 50



Rise 25/Fall 75

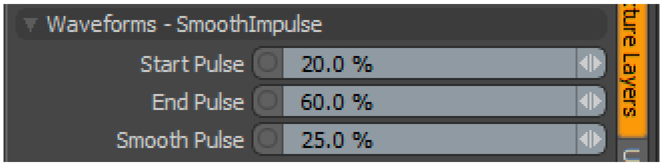


Rise 50/Fall 75



Rise 75/Fall 100

Waveforms - Smooth Impulse



Waveforms - SmoothImpulse

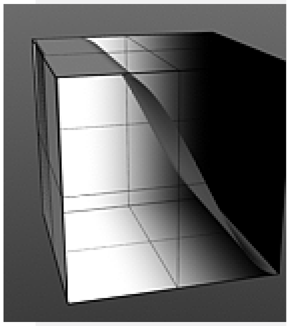
Start Pulse 20.0 %

End Pulse 60.0 %

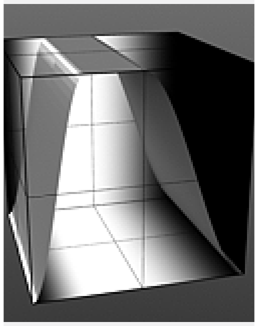
Smooth Pulse 25.0 %

Start Pulse: Specifies the beginning of the maximal part of the **Smooth Impulse** waveform, which is a cross between the **Impulse** waveform and the **Smooth** waveform. The **Start Pulse** option reveals the **Impulse** waveform aspect of this combined waveform.

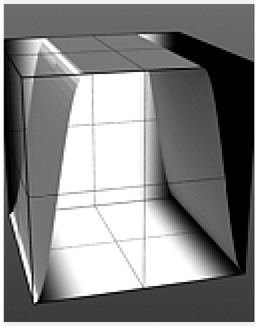
End Pulse: Specifies the end of the maximal part of the **Smooth Impulse** function, which is a cross between the **Impulse** function and the **Smooth** function. The **End Pulse** option reveals the **Impulse** waveform aspect of this combined waveform.



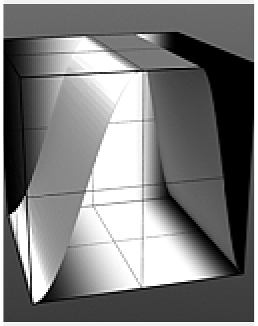
Start 0/End 25



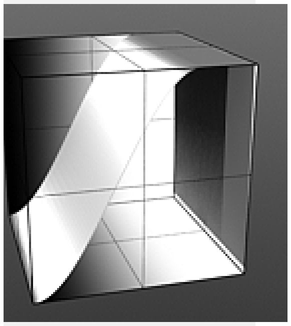
Start 25/End 50



Start 25/End 75

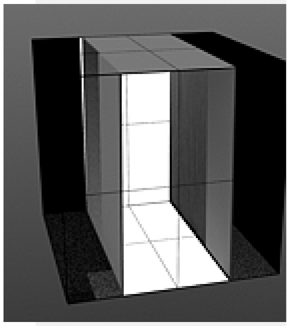


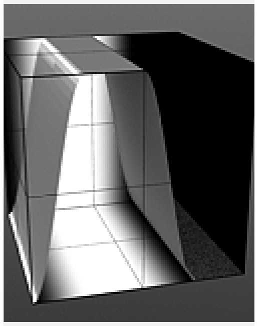
Start 50/End 75

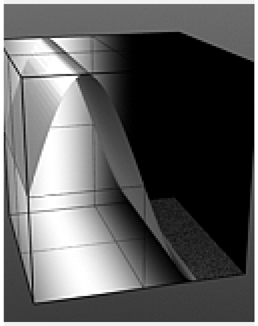


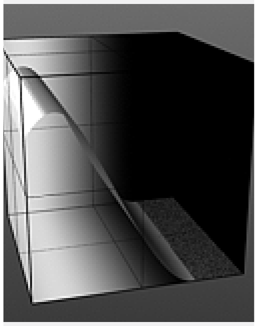
Start 75/End 100

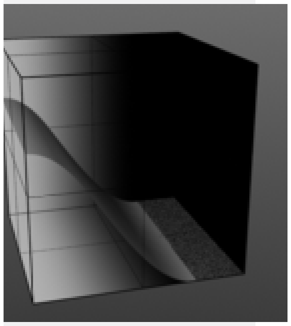
Smooth Pulse: Determines the amount of smoothing applied to the **Smooth Impulse** waveform. The **Smooth Pulse** option reveals the **Smooth** waveform aspect of this combined waveform.


Smooth 0%

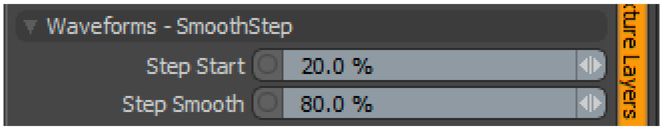

Smooth 25%


Smooth 50%


Smooth 75%


Smooth 100%

Waveforms - Smooth Step

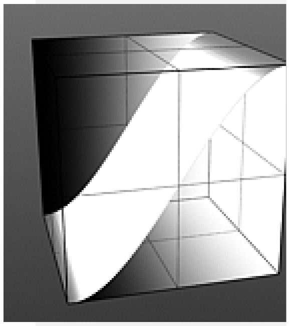


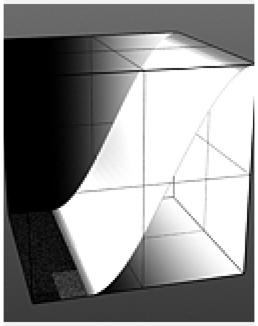
Waveforms - SmoothStep

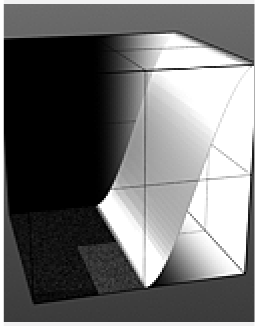
Step Start 20.0 %

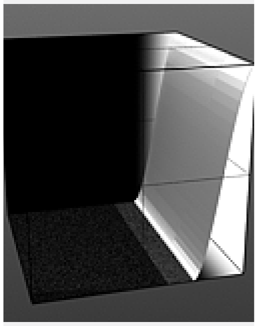
Step Smooth 80.0 %

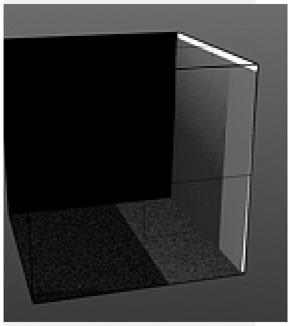
Step Start: Determines the start position for the smoothed step ramp.


Start 0%

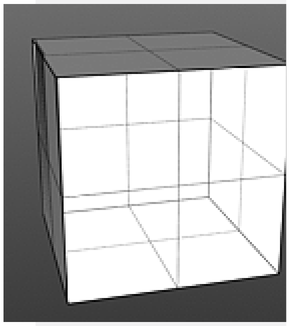

Start 25%

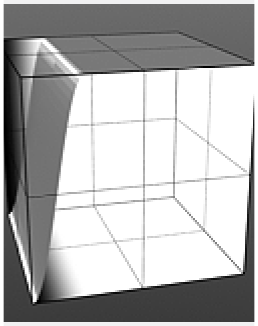

Start 50%

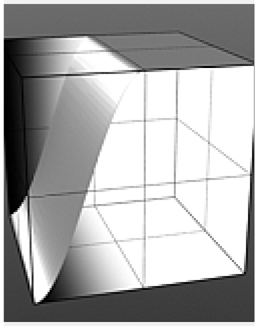

Start 75%

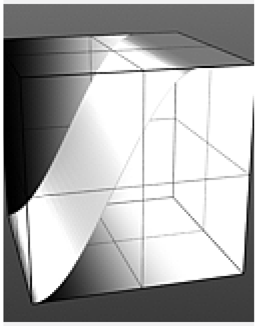

Start 100%

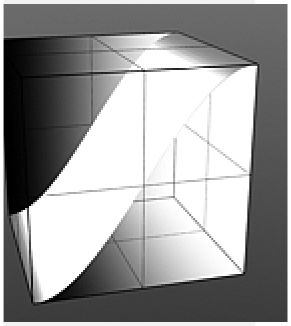
Step Smooth: Determines the amount of smoothing applied to the step position.


Smooth 0%

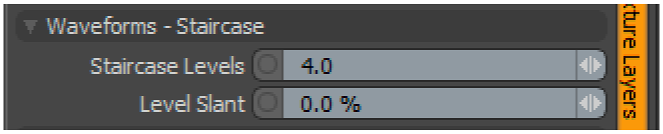

Smooth 25%


Smooth 50%


Smooth 75%


Smooth 100%

Waveforms - Staircase

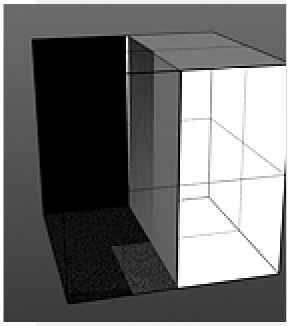


Waveforms - Staircase

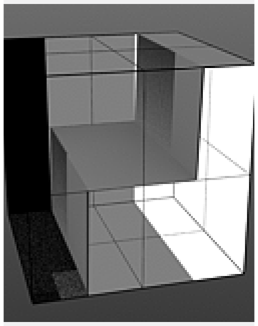
Staircase Levels 4.0

Level Slant 0.0 %

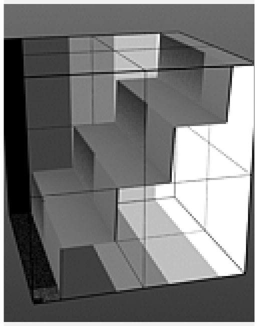
Staircase Levels: Determines number of discreet levels that the Staircase waveform breaks the smooth gradations into to produce a posterized effect.



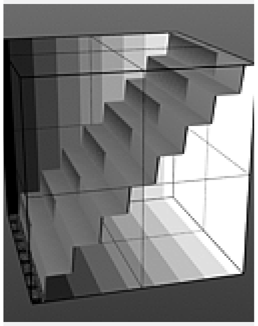
Levels 1



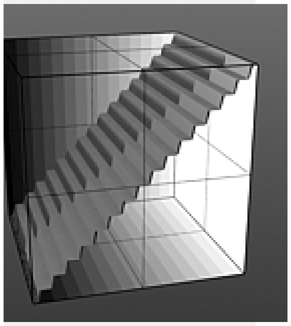
Levels 2



Levels 4

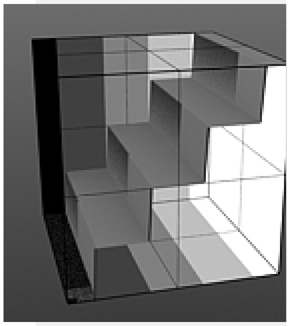


Levels 8

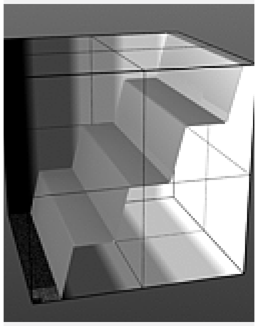


Levels 16

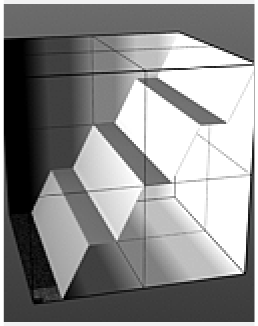
Level Slant: Interpolates between the step levels to produce a less abrupt change from one step level to the next.



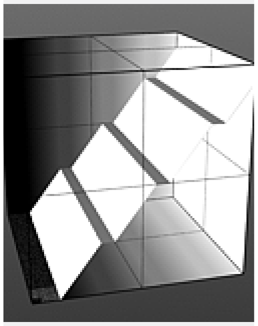
Slant 0%



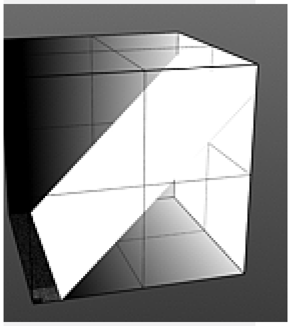
Slant 25%



Slant 50%



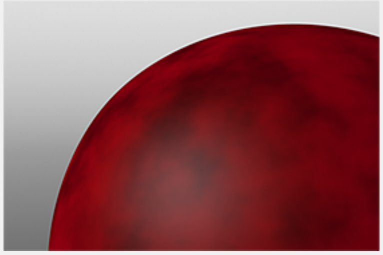
Slant 75%



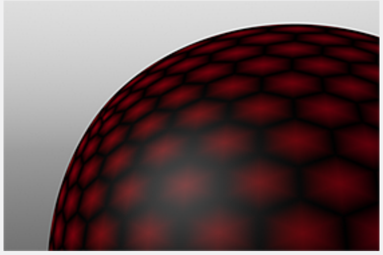
Slant 100%

Examples of Applied Waveforms

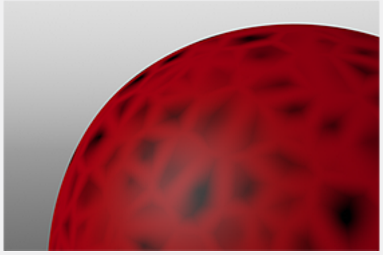
This gallery of rendered images illustrates the **Waveform** functionality.



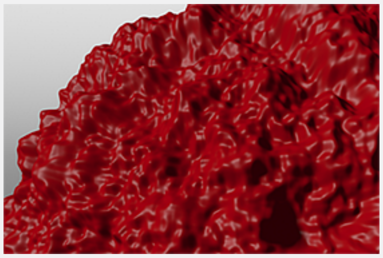
Noise as Diffuse Color Base



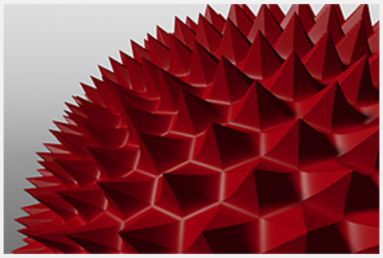
Grid as Diffuse Color Base



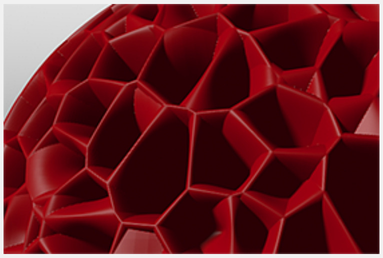
Cellular as Diffuse Color Base



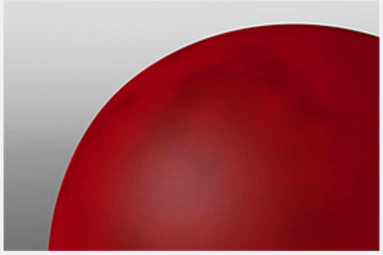
Noise as Displacement Base



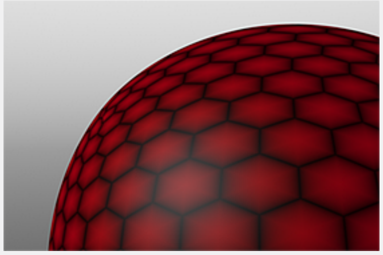
Grid as Displacement Base



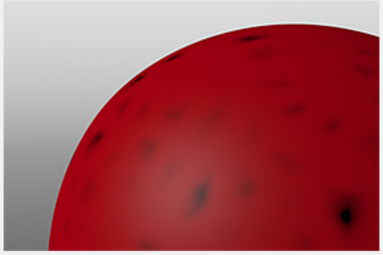
Cellular as Displacement Base



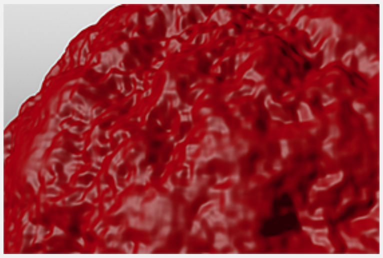
Rounded Waveform Applied to Base



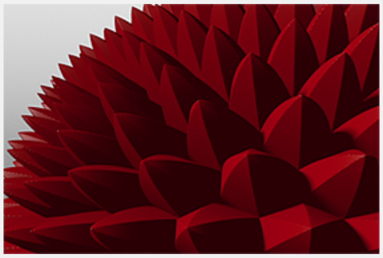
Rounded Waveform Applied to Base



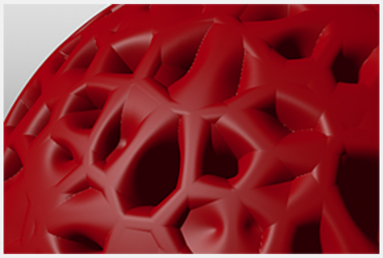
Rounded Waveform Applied to Base



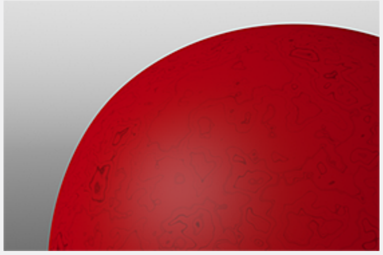
Rounded Waveform Applied to Base



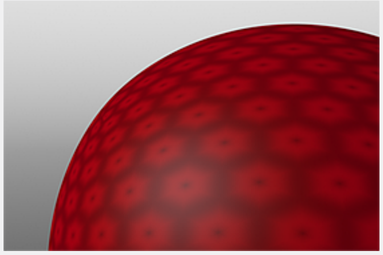
Rounded Waveform Applied to Base



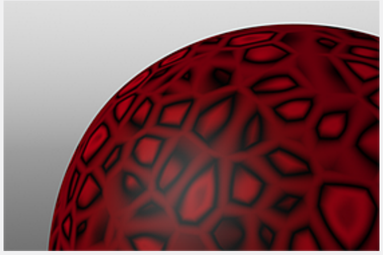
Rounded Waveform Applied to Base



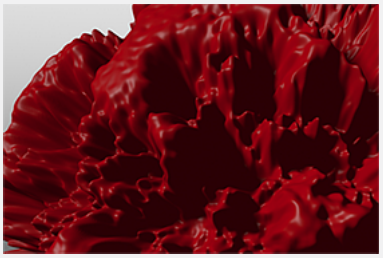
Sine Waveform Applied to Base



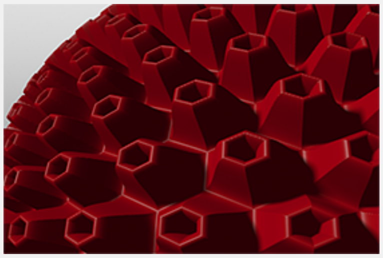
Sine Waveform Applied to Base



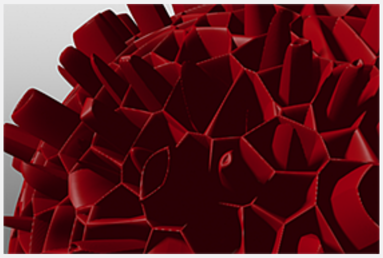
Sine Waveform Applied to Base



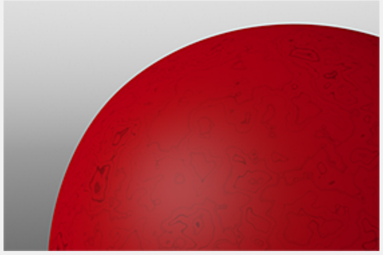
Sine Waveform Applied to Base



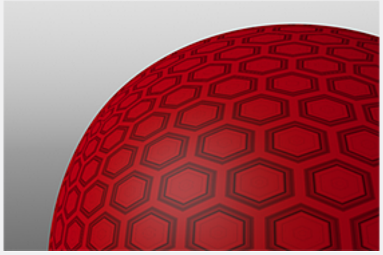
Sine Waveform Applied to Base



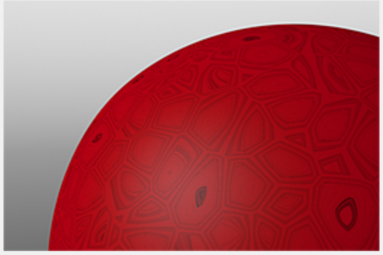
Sine Waveform Applied to Base



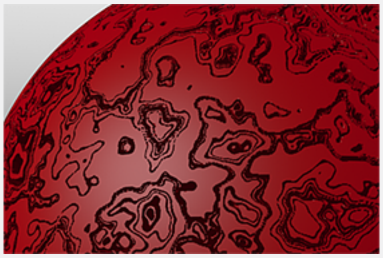
Noise Waveform Applied to Base



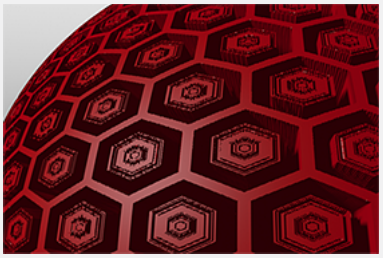
Noise Waveform Applied to Base



Noise Waveform Applied to Base



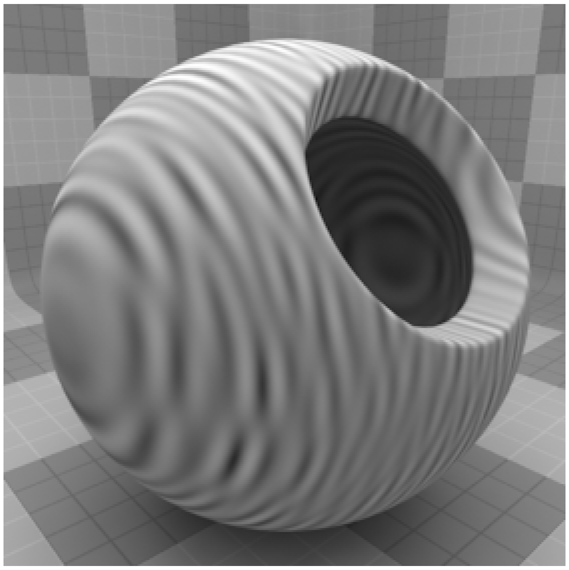
Noise Waveform Applied to Base



Noise Waveform Applied to Base

Noise Waveform Applied to Base

Waves

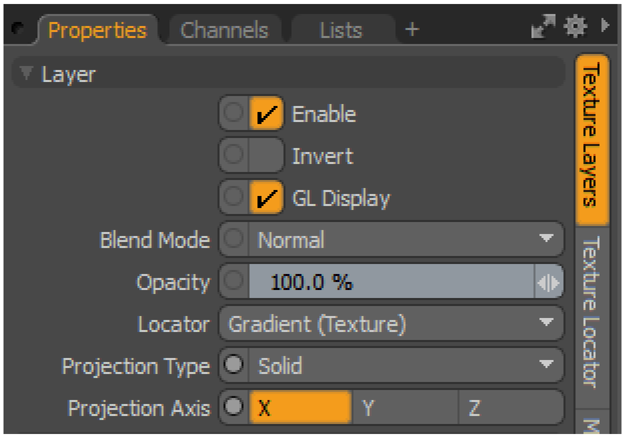


The **Waves** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Waves** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a loose, random wave pattern emanating from the texture's center. The amplitude of the wave decreases as the wave moves away from the wave source.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

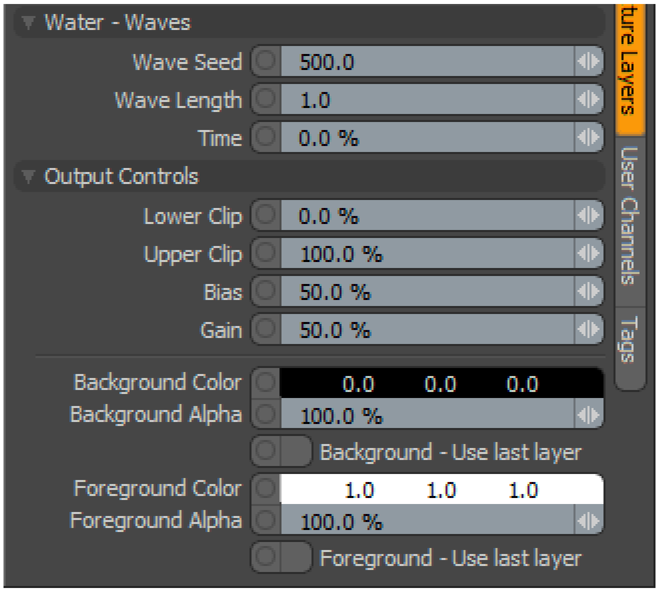
Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Waves Properties



Water - Waves

Wave Seed 500.0

Wave Length 1.0

Time 0.0 %

Output Controls

Lower Clip 0.0 %

Upper Clip 100.0 %

Bias 50.0 %

Gain 50.0 %

Background Color 0.0 0.0 0.0

Background Alpha 100.0 %

○ Background - Use last layer

Foreground Color 1.0 1.0 1.0

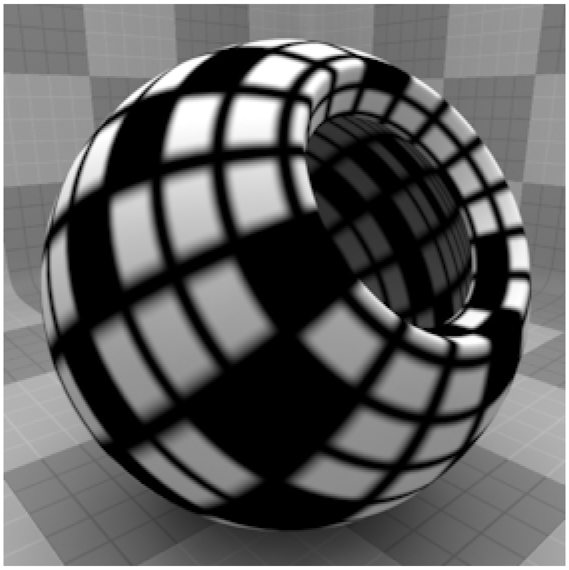
Foreground Alpha 100.0 %

○ Foreground - Use last layer

Water - Waves	
Wave Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Wave Length	Determines the distance moved by the waves.
Time	Controls the progress of the wave texture over the surface. The animation morphs through a complete wave animation cycle as the timer goes from **0%** to **100%**.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the

	value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Windows



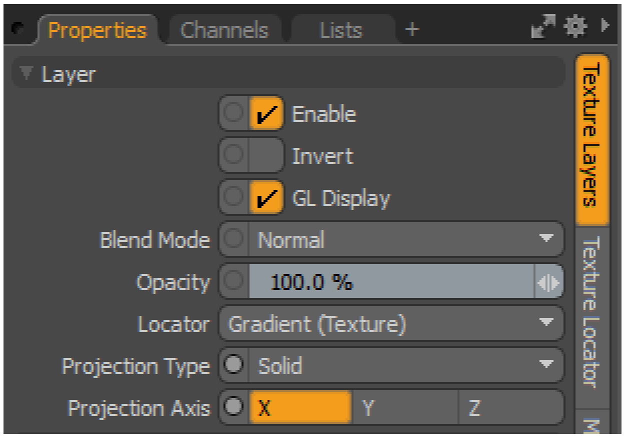
The **Windows** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Windows** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a series of rectangular windows that can be smoothed and turned on or off. You can also illuminate the windows to create banks of lit and unlit lights.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Warning: Due to the nature of this texture, the default **Solid** projection method produces no result. When applied, you need to modify the projection in the associated Texture Locator to a different type. To select the Texture Locator, click the **+** icon preceding the layer's name in the Shader Tree and click the revealed layer. In the Properties Panel, change the **Projection Type** to a UV Map (see Working with UV Maps) or select a geometric projection (such as **Planar**, **Cylindrical**, or **Spherical**) and size the texture accordingly.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

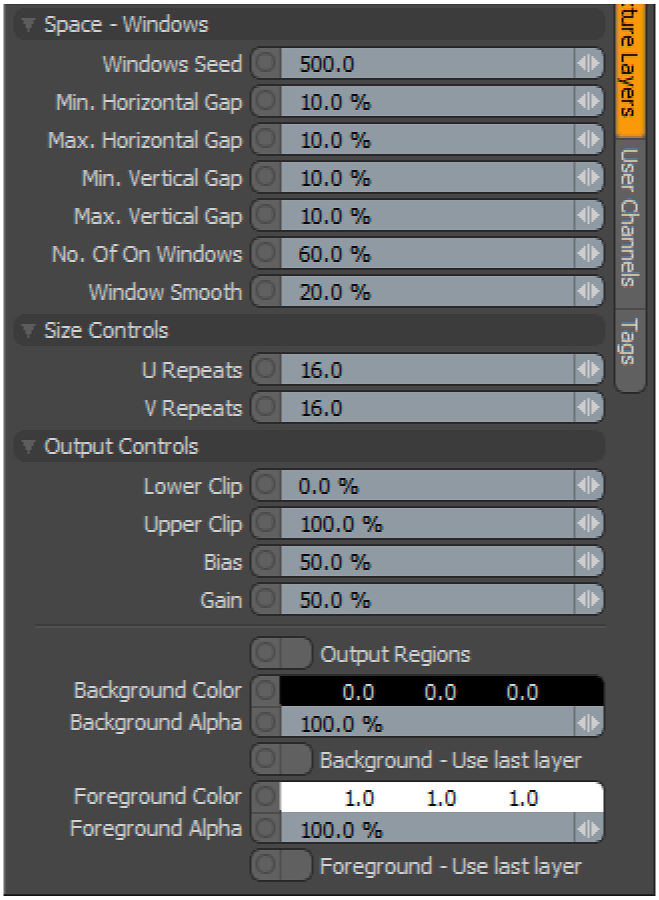
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping

	of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Windows Properties



Space - Windows
Windows Seed 500.0
Min. Horizontal Gap 10.0 %
Max. Horizontal Gap 10.0 %
Min. Vertical Gap 10.0 %
Max. Vertical Gap 10.0 %
No. Of On Windows 60.0 %
Window Smooth 20.0 %

Size Controls
U Repeats 16.0
V Repeats 16.0

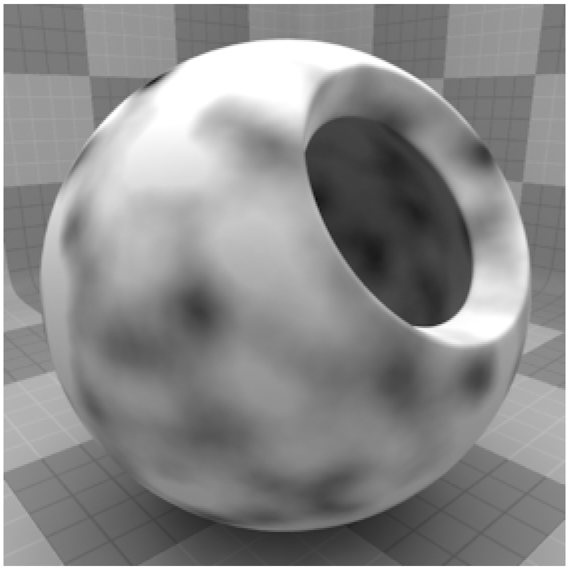
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Output Regions
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Space - Windows	
Windows Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in

	changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Min. Horizontal Gap	Sets the minimum distance between the windows in the horizontal direction.
Max. Horizontal Gap	Sets the maximum distance between the windows in the horizontal direction.
Min. Vertical Gap	Sets the minimum distance between the windows in the vertical direction.
Max. Vertical Gap	Sets the maximum distance between the windows in the vertical direction.
No. Of On Windows	Specifies the number of windows that are 'on' and, therefore, visible.
Window Smooth	Controls the edge smoothness of the windows. When you set this to a low value, the transition between the windows and the background color is very strong. The transition becomes smoother as you increase the value.
Size Controls	
U Repeats	Determines the number of times the pattern repeats in the **U** direction (horizontally).
V Repeats	Determines the number of times the pattern repeats in the **V** direction (vertically).
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.

Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Windy Waves

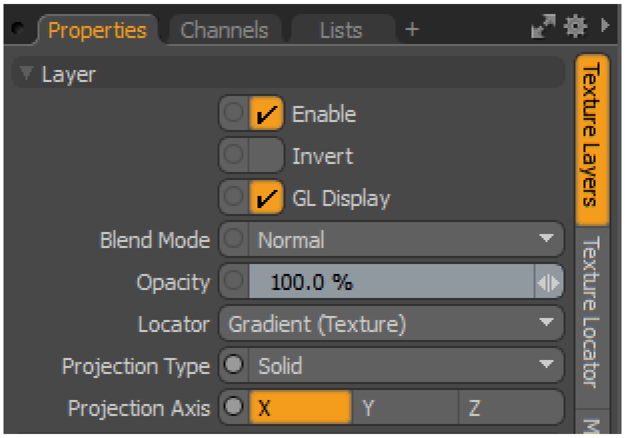


The **Windy Waves** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Windy Waves** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a very loose, random wave pattern that is similar to the deep ocean.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>



Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

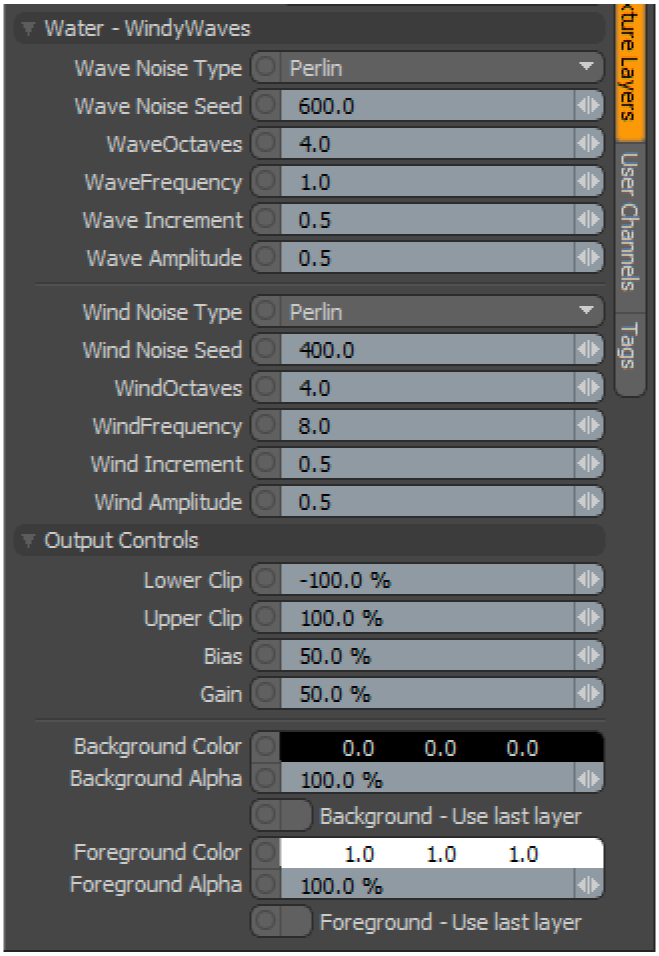
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Windy Waves Properties



Water - WindyWaves

Wave Noise Type: Perlin
Wave Noise Seed: 600.0
WaveOctaves: 4.0
WaveFrequency: 1.0
Wave Increment: 0.5
Wave Amplitude: 0.5

Wind Noise Type: Perlin
Wind Noise Seed: 400.0
WindOctaves: 4.0
WindFrequency: 8.0
Wind Increment: 0.5
Wind Amplitude: 0.5

Output Controls

Lower Clip: -100.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %

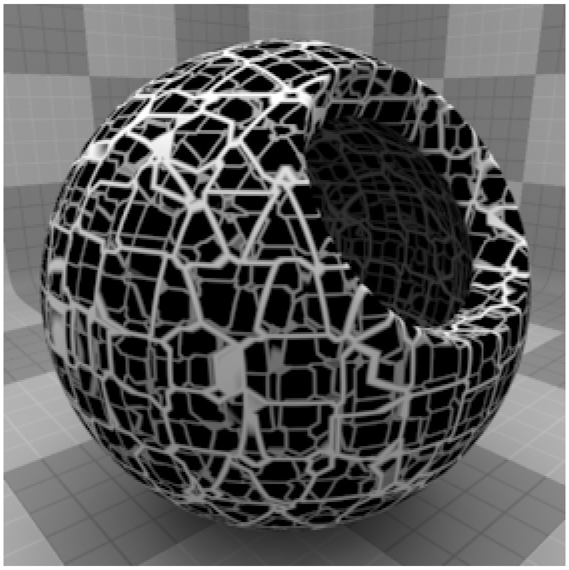
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Water - Windy Waves	
Wave Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
WaveOctaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates turbulence by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is less to create a smooth texture, but the turbulence increases as you increase the number of octaves.
WaveFrequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir up the pattern.
Wave Increment	Controls the scaling of each successive iteration. When you set the value low, the turbulence is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Wave Amplitude	Determines the height of the waves.
Wind Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value**

	• **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Wind Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
WindOctaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates turbulence by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the turbulence is less to create a smooth texture, but the turbulence increases as you increase the number of octaves.
WindFrequency	Controls the overall frequency (or scale) of the texture. Increasing this value creates more turbulence to stir up the pattern.
Wind Increment	Controls the scaling of each successive iteration. When you set the value low, the turbulence is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Wind Amplitude	Determines the amount of wind.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau

	around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Wires

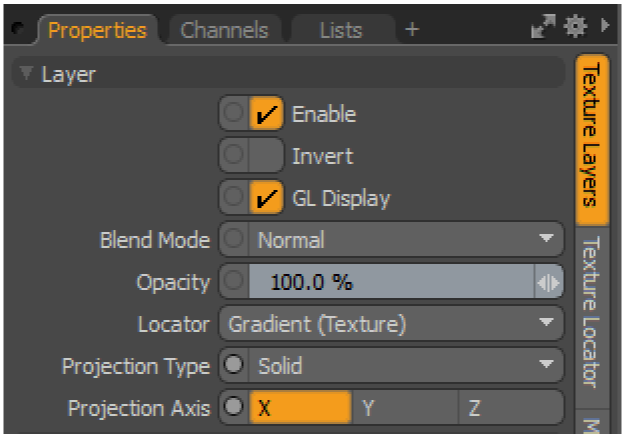


The **Wires** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Wires** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a random, intersecting series of wires.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

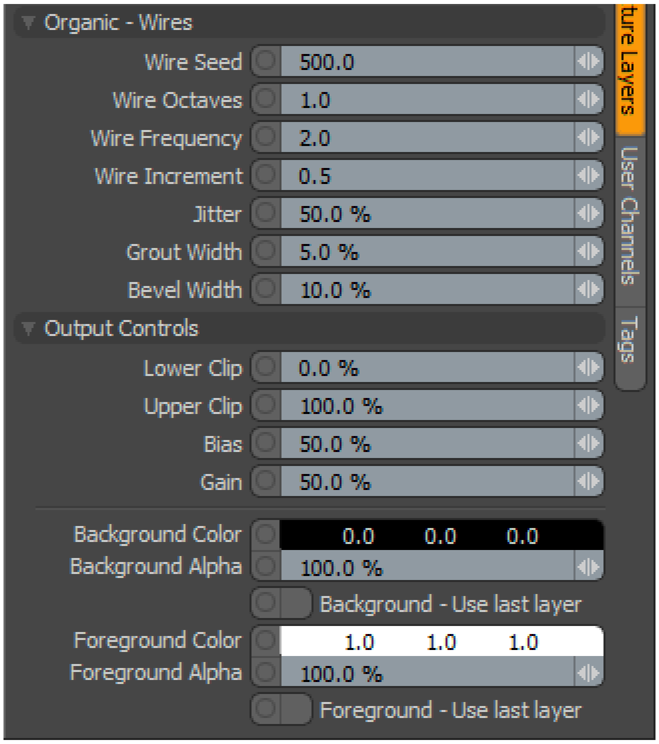


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Wires Properties



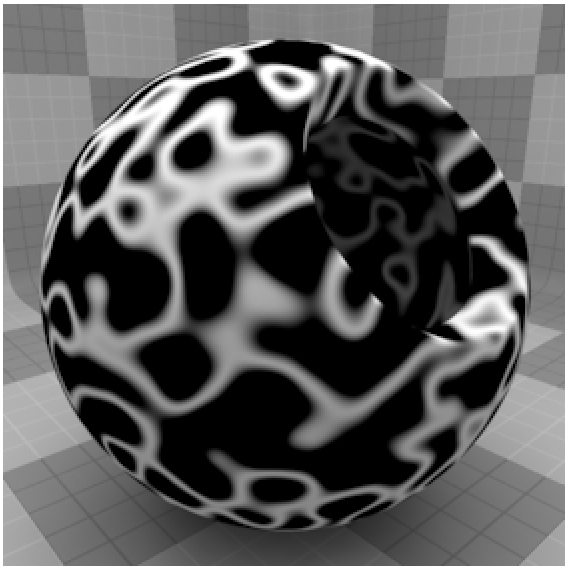
Organic - Wires
Wire Seed 500.0
Wire Octaves 1.0
Wire Frequency 2.0
Wire Increment 0.5
Jitter 50.0 %
Grout Width 5.0 %
Bevel Width 10.0 %
Output Controls
Lower Clip 0.0 %
Upper Clip 100.0 %
Bias 50.0 %
Gain 50.0 %
Background Color 0.0 0.0 0.0
Background Alpha 100.0 %
Background - Use last layer
Foreground Color 1.0 1.0 1.0
Foreground Alpha 100.0 %
Foreground - Use last layer

Organic - Wires

Wire Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Wire Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Wire Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Wire Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Jitter	Controls the jitter. Higher values cause more jitter in the texture.
Grout Width	Controls the grout width between wires.
Bevel Width	Controls the bevel between the edges of the wires.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.

Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Worm Vein

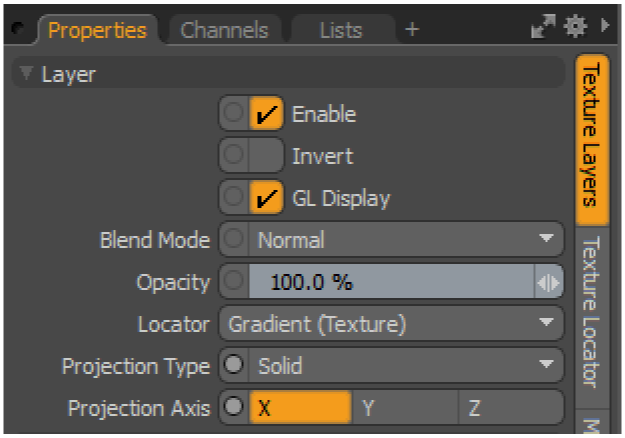


The **Worm Vein** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Worm Vein** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture creates a wormy vein-type of pattern that is similar to veiny skin or electrical arcs.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

>

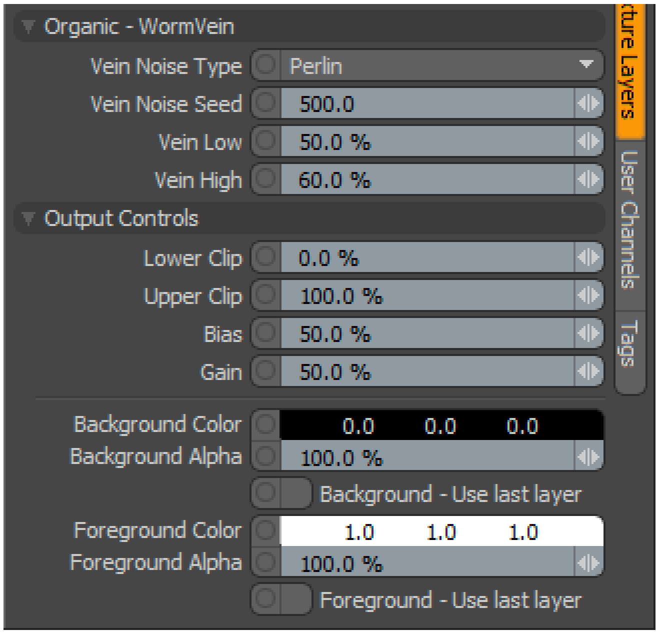


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing

	the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Worm Vein Properties



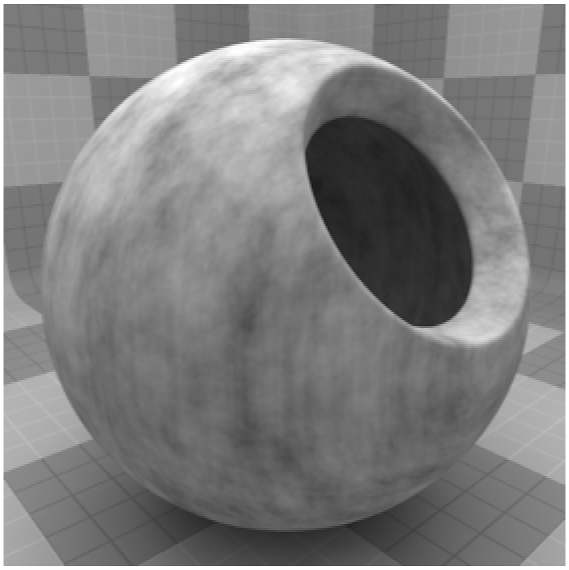
Organic - WormVein
Vein Noise Type: Perlin
Vein Noise Seed: 500.0
Vein Low: 50.0 %
Vein High: 60.0 %
Output Controls
Lower Clip: 0.0 %
Upper Clip: 100.0 %
Bias: 50.0 %
Gain: 50.0 %
Background Color: 0.0 0.0 0.0
Background Alpha: 100.0 %
☐ Background - Use last layer
Foreground Color: 1.0 1.0 1.0
Foreground Alpha: 100.0 %
☐ Foreground - Use last layer

Organic - Worm Vein	
Vein Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient**

	• **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**
Vein Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Vein Low	Controls how thick and layered the veins appear.
Vein High	Controls how thick and layered the veins appear.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background Color/Value	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward the **Foreground Color/Value**.
Background	Specifies the **Alpha** transparency of the **Background Color**.

Alpha	
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

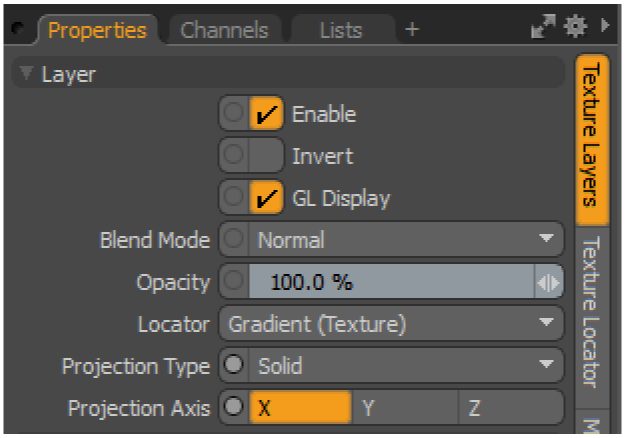
Wrapped fBm



The **Wrapped fBm** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Wrapped fBm** texture can be addressed by its two zones: the **Background** and **Foreground** colors. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Background** and **Foreground**. This texture is the same as the **FBM** (Fractal Brownian Motion) texture texture except that the noise is wrapped radially.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

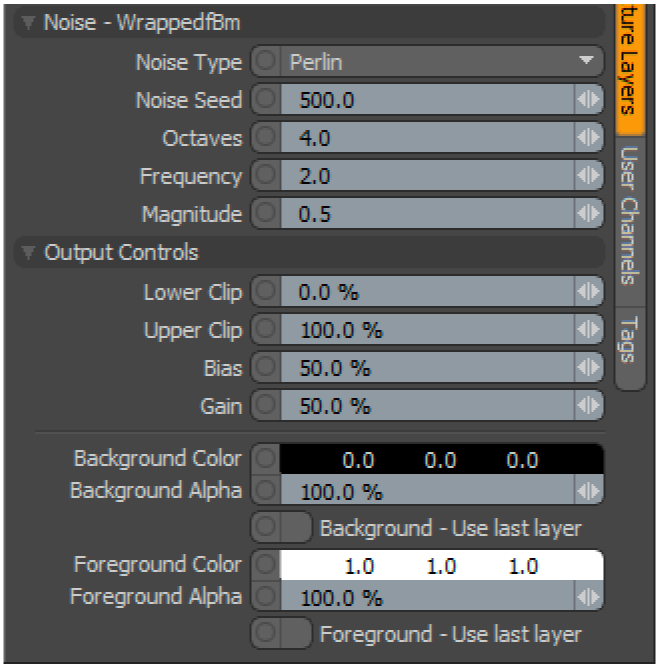
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in

	their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Wrapped fBm Properties



Noise - WrappedfBm

Noise Type: Perlin

Noise Seed: 500.0

Octaves: 4.0

Frequency: 2.0

Magnitude: 0.5

Output Controls

Lower Clip: 0.0 %

Upper Clip: 100.0 %

Bias: 50.0 %

Gain: 50.0 %

Background Color: 0.0 0.0 0.0

Background Alpha: 100.0 %

☐ Background - Use last layer

Foreground Color: 1.0 1.0 1.0

Foreground Alpha: 100.0 %

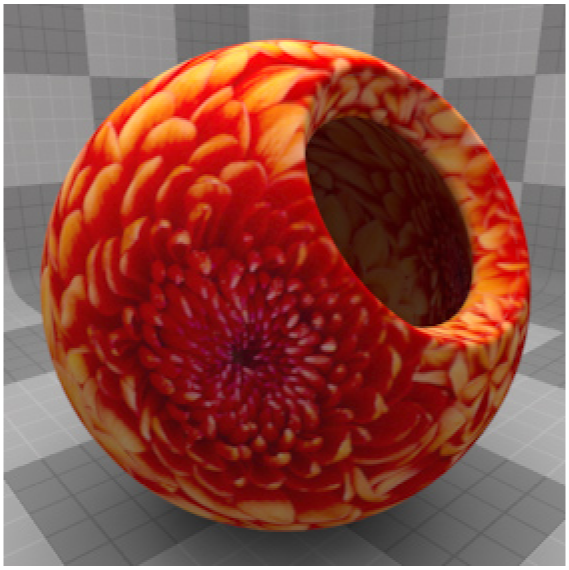
☐ Foreground - Use last layer

Noise - WrappedfBm	
Noise Type	Specifies the look of the texture's distortion and provides several noise function types: • **Perlin** • **Enhanced Perlin** • **Gradient** • **Value** • **Gradient Value** • **Impulse** • **Lattice** • **Bubble**

Noise Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Octaves	Defines the number of iterations Modo uses to create the texture. As you increase this value, Modo generates distortion by summing noise with each summed noise being half the magnitude of the previously summed noise. At low values, the distortion is less to create a smooth texture, but the distortion increases as you increase the number of octaves.
Frequency	Controls the overall frequency (or scale) of the texture. Increasing this value intensifies the pattern.
Increment	Controls the scaling of each successive iteration. When you set the value low, the texture is tight and crinkled, and it becomes increasingly smoother as you increase the value.
Output Controls	
Lower Clip	Specifies a clip level for the **Background Color/Value** to truncate values beyond the defined setting. Combined with the **Upper Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Upper Clip	Specifies a clip level for the **Foreground Color/Value** to truncate values beyond the defined setting. Combined with the **Lower Clip** value, you can apply this option to extend or lessen the range of values for the texture.
Bias	Determines whether the texture favors the foreground or background color. Increasing this value causes the texture to favor the foreground color; decreasing the value favors the background color.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two color values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme.
Output Regions	When enabled, outputs random gray shades per region rather than outlines for tiles. This adds random variety to the procedurally created texture. You can further control the amount of variation using the Regional HSV process layer.
Background	Specifies the **Color** (or **Value**) of the texture's background area, which ramps toward

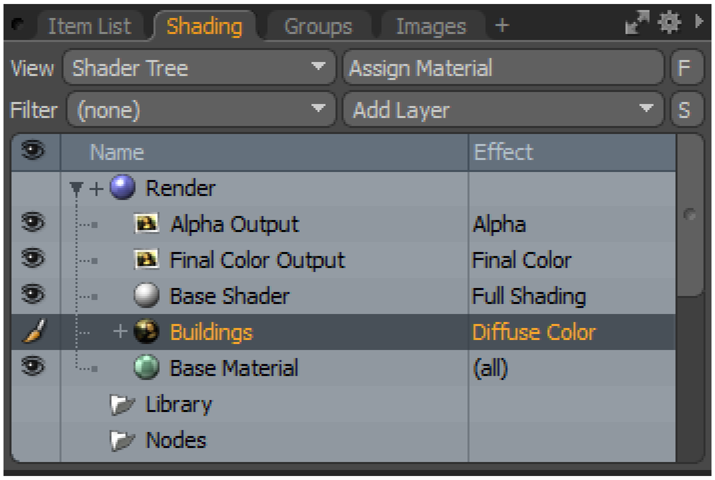
Color/Value	the **Foreground Color/Value**.
Background Alpha	Specifies the **Alpha** transparency of the **Background Color**.
Background - Use Last Layer	When enabled, makes the **Background Color** area completely transparent to reveal the shading results of lower layers.
Foreground Color/Value	Specifies the **Color** (or **Value**) of the texture's foreground area, which ramps toward the **Background Color/Value**.
Foreground Alpha	Specifies the **Alpha** transparency of the **Foreground Color**.
Foreground - Use Last Layer	When enabled, makes the **Foreground Color** area completely transparent to reveal the shading results of lower layers.

Image Map



The image map item could be considered the most important texture layer as it allows you to apply bitmap images to surfaces in Modo. When originating from a photographic source, bitmaps are a means of getting the richness of the real world into Modo, producing effects that would be difficult, if not impossible to produce otherwise. Want a surface to look like asphalt or dirt? Take a photograph and apply it as an image map; photographic textures are essential to photo-real rendering.

Images created in editing/painting and illustration/typographic programs are extremely useful as well; labels, posters, and other signage are great examples of this application. Image maps shouldn't be strictly limited to representing textured surfaces either, as they are also useful in many other cases; they can represent multiple complex procedural layers baked together, reducing the calculation overhead. They can be displacement maps created by Modo or other packages that apply high frequency details to lower resolution subdivision surface models, providing a better level of interactivity to items among other benefits.

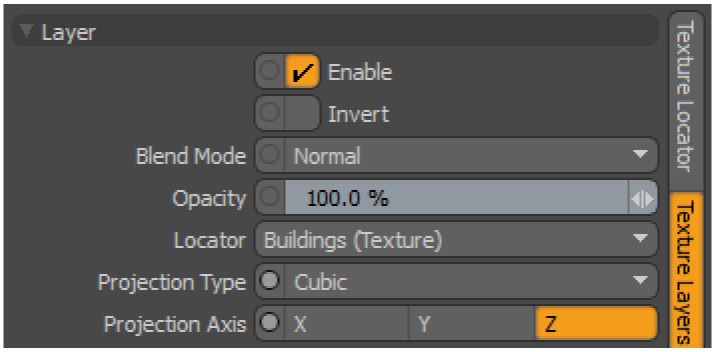


Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Buildings	Diffuse Color
Base Material	(all)
Library	
Nodes	

You can add an image map in the menu bar, by clicking **Texture** > **Add Shader Tree Layer** > **Image Map** or using the **Add Layer** function at the top of the full Shader Tree. The image map item contains all the attributes directly related to Modo's usage of the bitmap and are available in the **Properties** viewport when the item layer is selected in the **Shader Tree**. Scale, Position, Projection and other mapping-specific settings are available in the associated Texture Locator.

Image Map Properties

Click the **Texture Layers** tab to open the image properties.



☑ Enable
☐ Invert
Blend Mode: Normal
Opacity: 100.0 %
Locator: Buildings (Texture)
Projection Type: Cubic
Projection Axis: X Y Z (Z selected)

Layer	
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a photonegative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the **Shader Tree**, if present, or dims the effect of the layer, itself, on the surface.
Locator	Most texture layers also have an associated Texture Locator that is automatically created in the **Items** list. This defines the mapping of the texture to the surface. The **Locator** option sets that association. You can choose alternate locators, however, the need to do so is very rare; still, there are some possible instances where you may want multiple texture items to share a single locator.
Projection Type	Defines how a the image is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The imagel is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

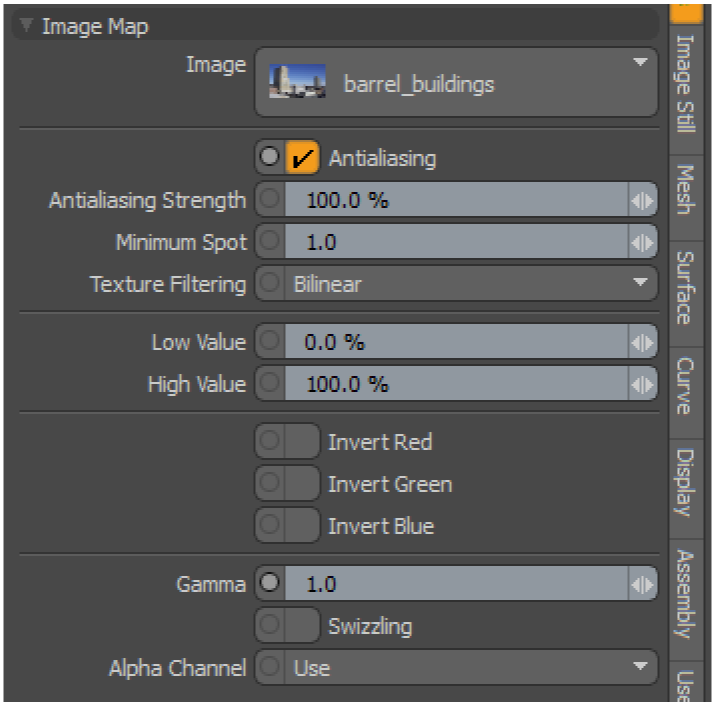


Image Map

Image: barrel_buildings

☑ Antialiasing

Antialiasing Strength: 100.0 %

Minimum Spot: 1.0

Texture Filtering: Bilinear

Low Value: 0.0 %

High Value: 100.0 %

☐ Invert Red

☐ Invert Green

☐ Invert Blue

Gamma: 1.0

☐ Swizzling

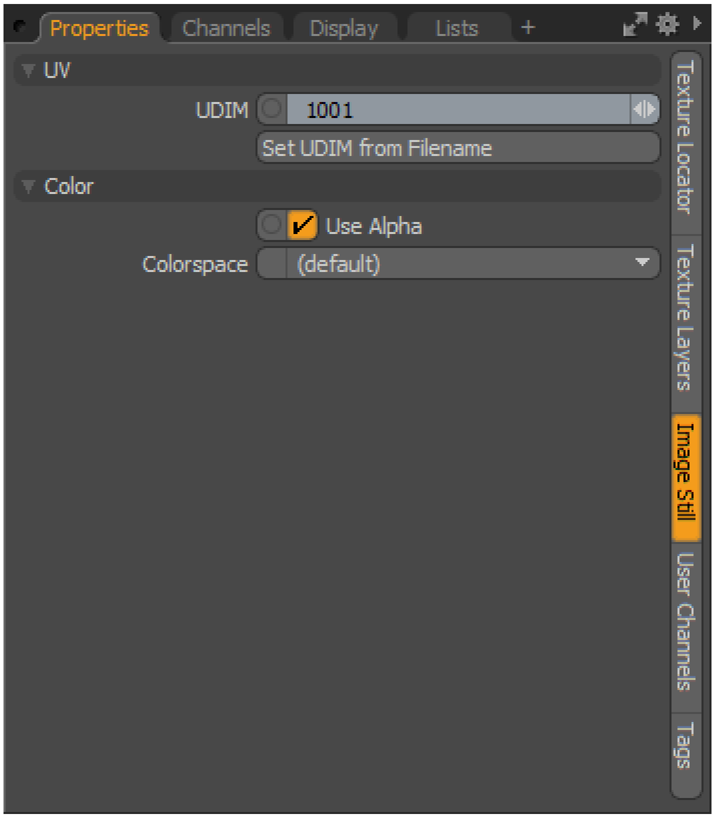
Alpha Channel: Use

Image Map	
Image	This pop-up allows you to select or load the specific image to be applied. Additionally, there are options for creating new blank images which can be useful for painting, sculpting and image baking.
Antialiasing	This checkbox toggles the **Antialiasing** function for the selected image layer, a means of filtering the texture to produce smoother rendered results. Antialiasing samples the image's pixels in texture space (as opposed to screen or render space) and averages them based on spot size and strength. When enabled, the **Antialiasing Strength** and **Minimum Spot** options become available.
Antialiasing Strength	Active only when **Antialiasing** is enabled, this setting allows you to control the amount of antialiasing on the image. Setting this value above 100% creates a blurring effect that smoothes the texture image.
Minimum Spot	The spot size relates to the spot created on the surface of a mesh when intersected by a ray. Rays are actually conical; think of the beam of light from a flashlight. The further the ray's origin is from the surface that it intersects the larger the spot is. Further, if the ray hits the surface at an glancing angle it can be even larger. When there is an image applied to the surface in question the render engine needs to know how many 2D texture pixels should be sampled within that spot. This becomes more important the closer the camera is to the surface. If the minimum spot value is 1 or

	smaller the spot size is not clamped and accurately samples the texture image at the sub pixel level, if necessary. When the spot size is set to a value of 2 or higher, the number of pixels sampled is always greater than 1. This results in a slight blurring of the image. You rarely need to adjust this value. The default of **1** means that the spot size is accurate to the sub-pixel level. Increasing the minimum spot size is mostly useful for blurring an image or reducing details when using an image for an environment map.
Texture Filtering	When spot size is set to 1 and the camera is so close to a surface that it samples the image at the sub-pixel level (that is, an image pixel is bigger than a rendered pixel), Texture filtering averages across those samples. The result is a smooth interpolation of pixels rather than sharp angular pixel boundaries. You can choose from several options. • **Nearest** - Disables filtering of image, the actual pixel of the image is sampled. • **Bilinear** - Samples neighboring pixels vertically and horizontally, producing smooth, or slightly blurry results. • **Bicubic** - Samples all adjacent pixels providing the smoothest result.
Low/High Value	These values provide a means to re-map the total range of tones in an image, especially useful for controlling value setting. For instance, if you are using an image as a displacement map, you might want the darkest pixels to push the surface in and the bright pixels to push them out. By default, the pixels of an image go from 0 to 100% which means that none of the pixels yield a value to push the pixels inward. By setting the value range from -100% to 100%, the image is effectively remapped, so that the black pixels create a negative displacement and the white pixels a positive displacement with mid-tones being neutral. These settings are also useful in extending the range of standard (low dynamic range) images, making it possible to use standard non-HDR images to light a scene (although the results don't look as good as using a real HDRI). • **Low Value** - Determines the effective value of pixels that have a true value of 0,0,0 (r,g,b) • **High Value** - Determines the effective value of pixels that have a true value of 255,255,255 (r,g,b).
Brightness	Adjusts the brightness of the image. **Note:** The effect is only visible in the **Advanced** viewport or a render.

Contrast	Adjusts the contrast of the image. **Note:** The effect is only visible in the **Advanced** viewport or a render.
Clamp	Enable **Clamp** to limit the brightest and darkest values so no pixels can have a value higher than 1.0 and or lower than 0.0.
Invert Red, **Green**, **Blue**	These three checkboxes allow you to invert any of the three color channels of an image map individually. While you rarely need to use this option, especially on color images, it can be helpful to save normal maps where certain channels are mapped differently to what Modo expects them to be.
Gamma	This can be used as an image adjustment control to lighten or darken the image without the need to externally adjust it in an external bitmap editor. The **Gamma** adjustment is non-linear and affects the mid-tones of the image greater than the shadow and highlight areas of the image. For best results, leave the **Gamma** value at the default 1.0.
Swizzling	In Unreal Engine 4 and Unity 5, it is possible to assign a specific channel of RGBA (Red, Green, Blue, Alpha) to a material parameter and set another channel to a different material parameter from the same image. In Modo, this feature can be achieved by using multiple texture layers with channel swizzling. The following options are available: • **RGB** - Sets the channel as RGB1 (alpha to 1). • **RGBA** - The default setting. Uses all components of the color for RGBA channel. • **Alpha** - Sets the channel as AAA1. • **Red** - Sets the channel as RRR1. • **Green** - Sets the channel as GGG1. • **Blue** - Sets the channel as BBB1.
Alpha Channel	Allows you to choose how Modo handles alpha channel data (if present) contained in the image file. • **Use Alpha** - Masks the color pixels based on the embedded alpha channel data. • **Ignore** - Discards the embedded alpha channel. • **Alpha Only** - Ignores any RGB information in the file, treating it strictly as a single (alpha) channel image.

Image Still Properties

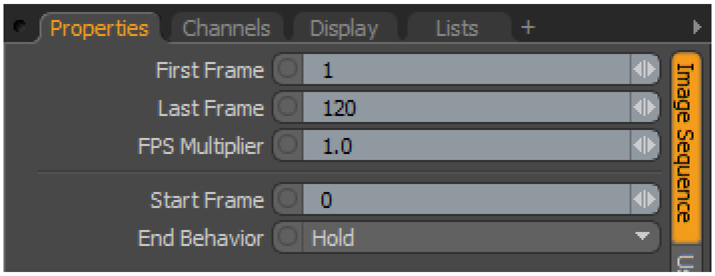


Properties
Channels
Display
Lists
+
UV
UDIM
1001
Set UDIM from Filename
Color
Use Alpha
Colorspace
(default)
Texture Locator
Texture Layers
Image Still
User Channels
Tags

Option	Description
UDIM	The UDIM value is a notation that designates an offset to a specific UV tile location. It is designed to allow you to assign many textures to a surface at once. For more information on using and working with UDIM textures, see the UDIM Workflow topic.
Set UDIM from Filename	Clicking this button causes Modo to read the filename for the selected texture layer (s) and automatically set the above UDIM value based on the filename.
Color	
Use Alpha	In certain game asset workflows, an embedded alpha channel on an image map might represent an effect other than the image's transparency. When set as such, it makes it difficult to view the image in the 3D viewports. In these cases the **Use Alpha** option is a global way to disable any embedded alpha channels from an image, allowing its color components to still be visible.

Colorspace	This setting defines the color space of the associated image layer, which controls how Modo interprets the colors of the image itself for rendering. The settings for the **Colorspace** option can have a dramatic effect over how the colors look in the final rendered image, so setting it properly is important. For more information on a color managed workflow in Modo, see the Color Management topic.

Image Sequence



First Frame: 1
Last Frame: 120
FPS Multiplier: 1.0
Start Frame: 0
End Behavior: Hold

Additional options appear when you apply an image sequence as a texture layer. To access these options, select the image sequence item, then select the **Image Sequence** sub-tab from the **Shader Tree Properties** panel.

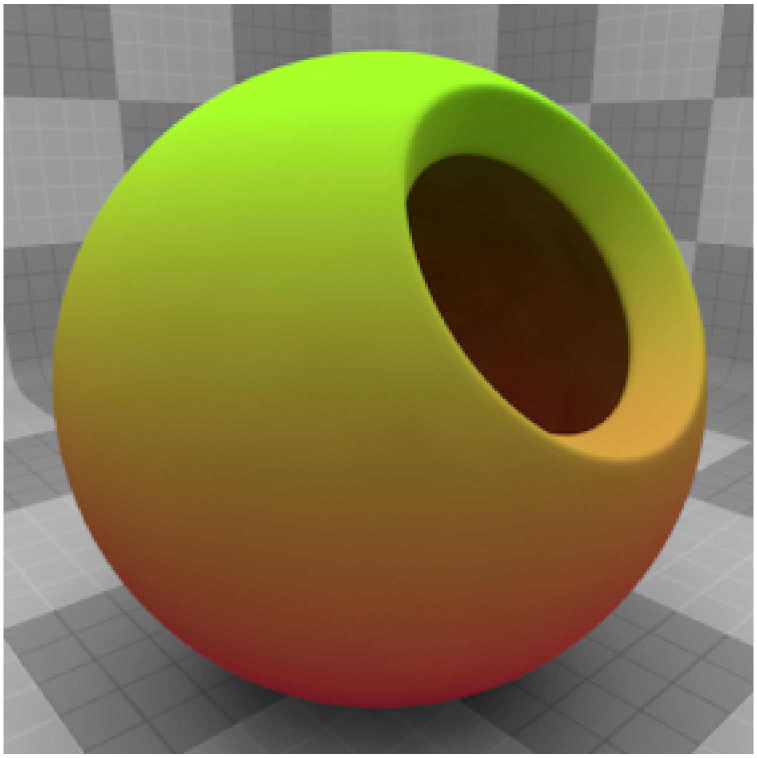
Option	Description
First/Last Frame	These options determine the range from the sequence to display/render in Modo. Frames outside of this range are ignored/truncated.
FPS Multiplier	This value determines the speed at which the sequence plays in relation to the FPS rendered in the Modo scene, allowing you to adjust the timing of frames. So a 30fps image sequence applied to a 30fps plays at 1:1 frames when the value is 1.0 (one sequence frame for every rendered frame). A 2.0 value is twice as fast as the scene, effectively skipping every other frame, conversely a 0.5 value plays back half as fast, doubling the frames. **Note:** This speed affects the **Start Frame** value below. A **Start Frame** of 48 starts at 24 when the speed is 2.0.
Start Frame	This value determines the first rendered frame in Modo in which to begin the playback of the image sequence, allowing you to delay the start of the sequence. Frames rendered prior to this value hold the **First Frame** as specified above, duplicating the **First Frame** in subsequent rendered frames until reaching the **Start**

	Frame value.
End Behavior	These options determine how Modo handles the image sequence when the **Last Frame** value is reached. • **Hold** - Duplicates the **Last Frame** on all subsequent rendered frames. • **Repeat** - Loops the sequence between the **First** and **Last Frame** values. • **Mirror** - Ping-pongs the sequence forwards and backwards between the **First** and **Last Frame** values.

Tip: Modo automatically uses the first available or last available frame when corresponding frames don't exist in the specified sequence. Take, for example, a sequence from frame 45 to 100 applied as a background plate. If the **First** and **Last Frame** values are 1 and 100 respectively, and the **Start Frame** value is 0, when rendering out the sequence, frame 45 would be duplicated across rendered frames 1-44 as no frames existed in the specified sequence - essentially the same behavior as the **First /Last** option. Interim missing frames are duplicated until a proper sequence frame is encountered.

Gradient Fill Image

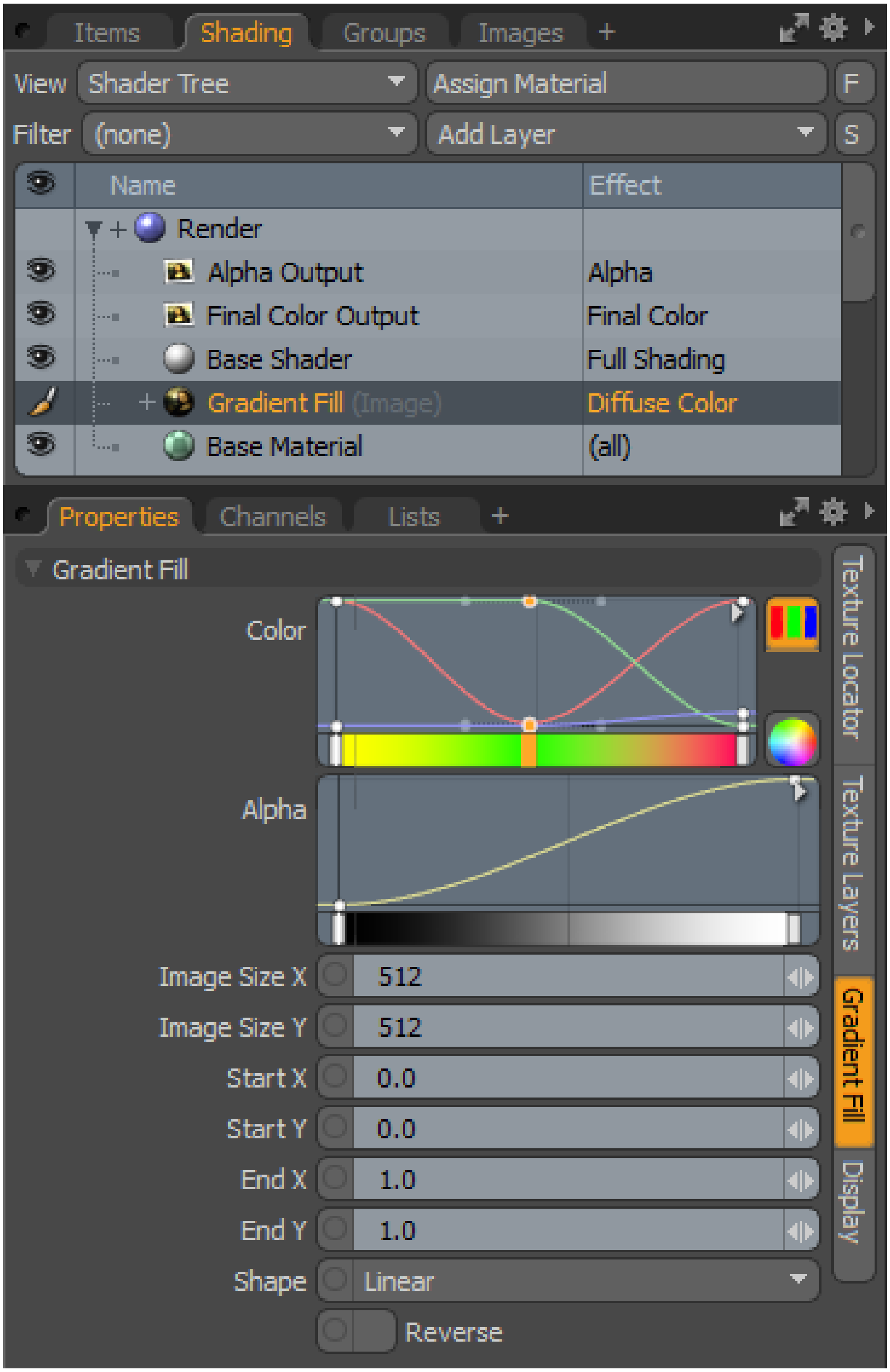
The **Gradient Fill Image** layer is a hybrid texture, as it is procedurally-generated gradient but it is based on a dynamically-created **bitmap** image. As a procedurally-created image, it doesn't have to be saved independent of the scene, as it is generated automatically at run-time as needed. As a bitmap, all the different ways a texture layer can be applied can also be assigned to the gradient layer including projection types and UV maps, modified in its associated **Texture Locator**. There are several types of **Gradient Styles** that can be generated: Linear, Radial, Angular, and Diamond. These basic styles specify the appearance on the image tile and the **Start** and **End** positions its position and orientation. The gradient itself is user-defined in the **Gradient Editor**.



The **Gradient Fill Image** is added to the **Shader Tree** using the **Add Layer** option found under **Add Layer** > **Image Map** > **image** > **Gradient Fill**. Once created, the layer itself can be positioned, and its **Effect** defined in the **Shader Tree**.

Note: For information regarding adding and working with **Shader Tree** item layers, please refer to the Shader Tree Item Layers topic.

With the layer selected, all its various properties are available in the **Properties** viewport.



Items Shading Groups Images +

View Shader Tree

Assign Material

Filter (none)

Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Gradient Fill (Image)	Diffuse Color
Base Material	(all)

Properties Channels Lists +

Gradient Fill

Image Size X 512

Image Size Y 512

Start X 0.0

Start Y 0.0

End X 1.0

End Y 1.0

Shape Linear

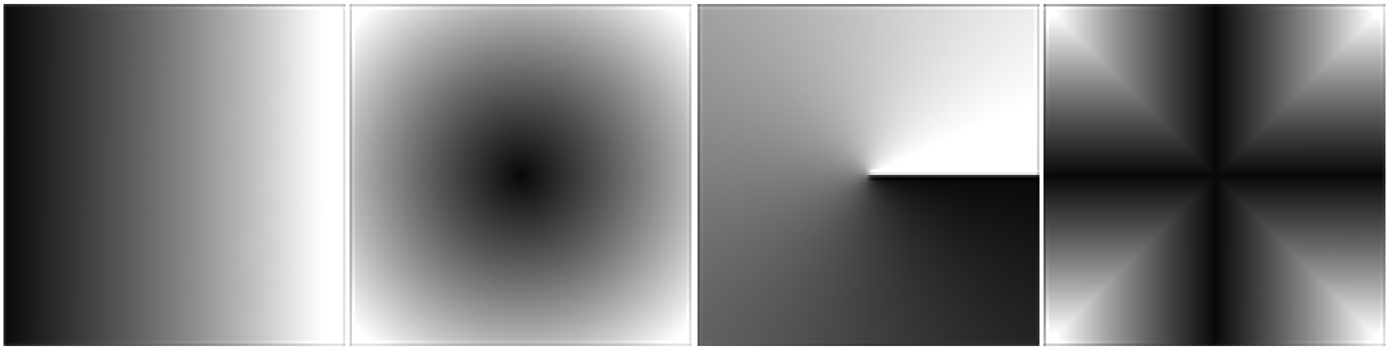
Reverse

The following **Gradient Fill** options are available for **Gradient Fill Image**:

- **Color** - Allows you to quickly define basic gradients directly within the main interface. The gradient values are defined by keys, like keyframes in a Timeline. Gradient keys define specific color values at the defined locations that smoothly attenuate between the keys. Keys are defined in the editor by middle-clicking within the gradient window or, for Mac users, **Ctrl**+**Alt**+clicking.

This sets a key location, then you can click on the color bars or wheel, and then define a **Color** for that location. New key positions are added and updated in the same way. Defined keys can also be edited by clicking and dragging the points up and down or left and right. Below the gradient key window is a graduated bar that represents the gradient itself. The window is limited to just the 0%-100% range on either axis. For further refinement, it may be desirable to edit the gradient in the full sized **Gradient Editor** available in the menu found under the small arrow in the upper-right hand corner of the window.

- **Alpha** - This works in exactly the same way as the **Color** mini gradient editor, but defines a single value that determines the transparency amount for the **Color** gradient.
- **Image Size X/Y** - Determines the vertical and horizontal resolution of the resulting canvas that the gradient is generated into.
- **Start/End X/Y** - Determine both the position and, indirectly, the direction of the resulting gradient, where **start** represents the starting position and **end** represents the termination point. The gradient is then generated between these two locations. The values are represented as a floating point value across the generated canvas (acting as a percentage), with the '0, 0' origin at the lower-left corner, and '1,1' position in the upper-right corner (identical to how a normalized UV field value is defined). A value of '0.5' would put the position in the center of the canvas, regardless of its resolution and aspect ratio.
- **Shape** - controls the appearance of the resulting gradient. There are four possible options (pictured below): **Linear**, **Radial**, **Angular**, and **Diamond** respectively.



- **Reverse** - swaps the **Start** and **End** positions, resulting in a mirrored gradient.

Projection Shader

The **Projection Shader** simplifies the workflow of using projected bitmap images. When applied, it replaces all surface shading with a photographic image. This type of projection is useful when working with digital matte paintings or when you need to simply add a photographic texture to a surface. In the sample below, the example sphere has had a tennis ball image projected onto it so all of the surface shading is now gone, including the interior hollow section. Now, it appears as a solid sphere, represented only by its silhouette.

A small amount of noise has been added to displace the underlying surface, providing for a more convincing effect. This was done literally in a matter of seconds (which is why it doesn't look all that great), but it is much better than having to model a highly detailed tennis ball using fur and other complex surfacing attributes. Or imagine instead of modeling an entire city block, giant cubes could be used as stand-in geometry, with photos of actual buildings projected onto them, providing a more realistic result in a fraction of the time. These are just a couple examples of how the **Projection Shader** can save loads of time on a project.

Using the Projection Shader

To add a Projection Shader, do the following:

1. In the **Shader Tree**, click **Add Layer** > **Special** > **Projection Shader**.

   The Projection Shader appears in the **Shader Tree**.

2. For proper results, add the Projection Shader to the target material Group, then drag the entire group above the Base Shader.

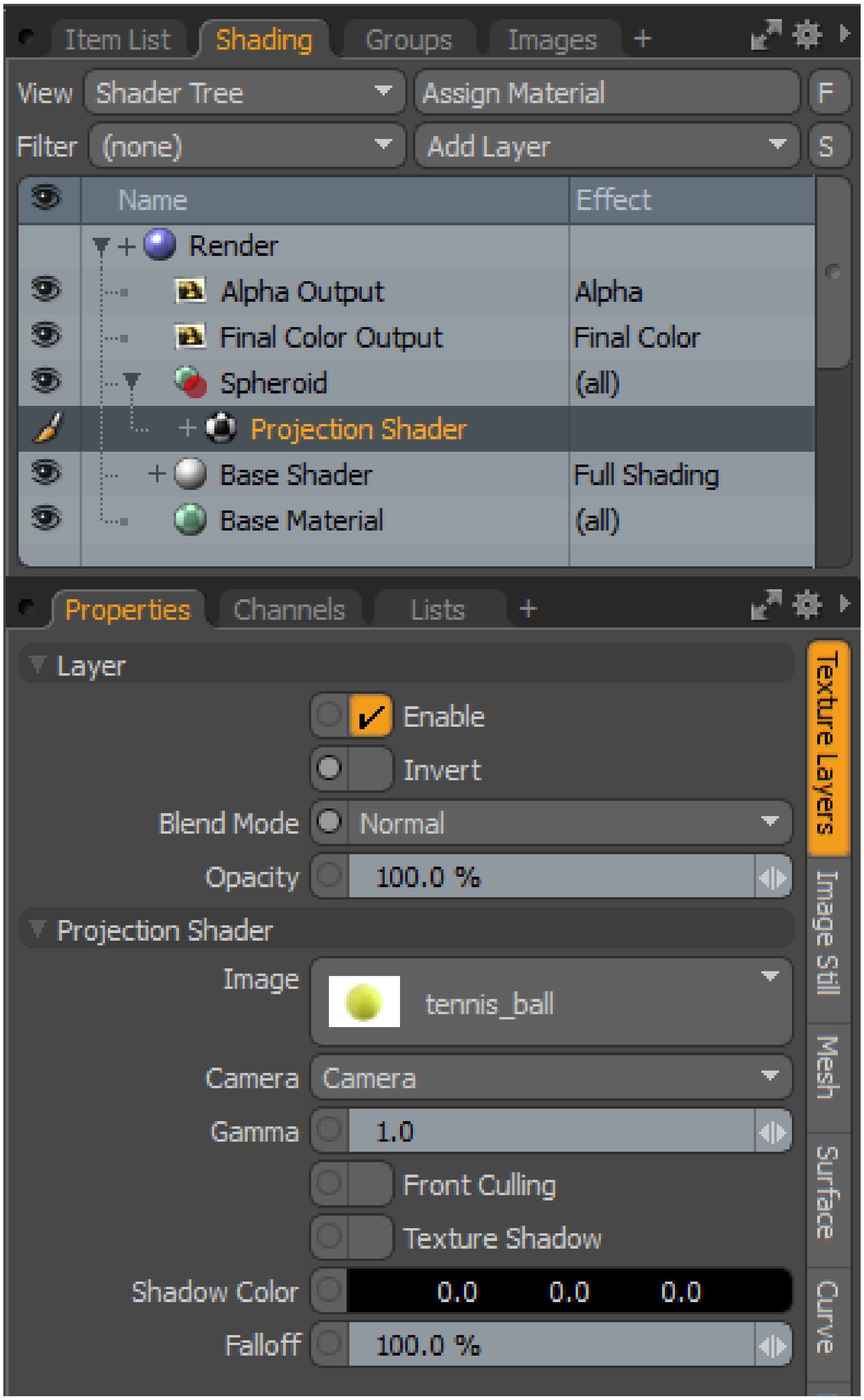
Note: A different approach would be to simply add a new material **Group** item above the **Base Shader** and set its **Item** attribute to the target mesh layer, working as an item mask. Then add the **Projection Shader** to this new material group and it's applied to the entire target object.

When the **Projection Shader** is added, the **Clip Browser** window opens.

3. Select the desired projection image by clicking the target image. If it's not already loaded, click the **Add Clip** button, and click **(load image)**.

Note: The image is automatically stretched to fill the entire rendered frame, so the projected image needs to be the same aspect ratio as the final render **Frame** in order to avoid looking stretched.

It should be noted that as a standard **Shader Tree** layer, the **Projection Shader** layer can be masked like any other **Shader Tree** layer. For more information regarding adding and working with **Shader Tree** item layers, please refer to the Shader Tree topic. When the **Projection Shader** layer is selected, the following attributes appear in the **Properties** viewport.


Item List
Shading
Groups
Images
+
Assign Material
F
Add Layer
S
Properties
Channels
Lists
+
Layer
Projection Shader
Texture Layers
Image Still
Mesh
Surface
Curve

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Spheroid	(all)
Projection Shader	
Base Shader	Full Shading
Base Material	(all)

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Image: tennis_ball

Camera: Camera

Gamma: 1.0

☐ Front Culling

☐ Texture Shadow

Shadow Color: 0.0 0.0 0.0

Falloff: 100.0 %

The following **Layer** options are available for **Projection Shader**:

- **Enable** - toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
- **Invert** - inverts the RGB values for the layer, producing a negative effect.

- **Blend Mode** - affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, refer to the Layer Blend Modes topic.
- **Opacity** - changes the transparency of the current layer. Reducing these values increasingly reveals lower layers in the **Shader Tree** if present, or dims the effect of the layer itself on the surface.

The following **Projection Shader** options are available for **Projection Shader**:

- **Image** - uses the **Clip Browser** window to define the projected image. Once loaded, a projected image is stretched to fill the full frame, so make sure that the selected image is the same aspect ratio to avoid any distortion or stretching.
- **Gamma** - allows you to adjust the mid-tone values of the image, should a gamma correction be necessary. For most cases, the default 1.0 value should suffice, where Gamma is handled by the Color Management system.
- **Front Culling** - creates a texture shadow, that means it only maps the parts of the geometry directly seen by the camera, the parts hidden from the camera POV uses the texturing below.
- **Shadow Color** - instead of using the texturing of a material in the front-culled areas, you can specify a diffuse color.
- **Falloff** - sets the amount of falloff from the projection on surfaces not facing the camera.

Projection Texture

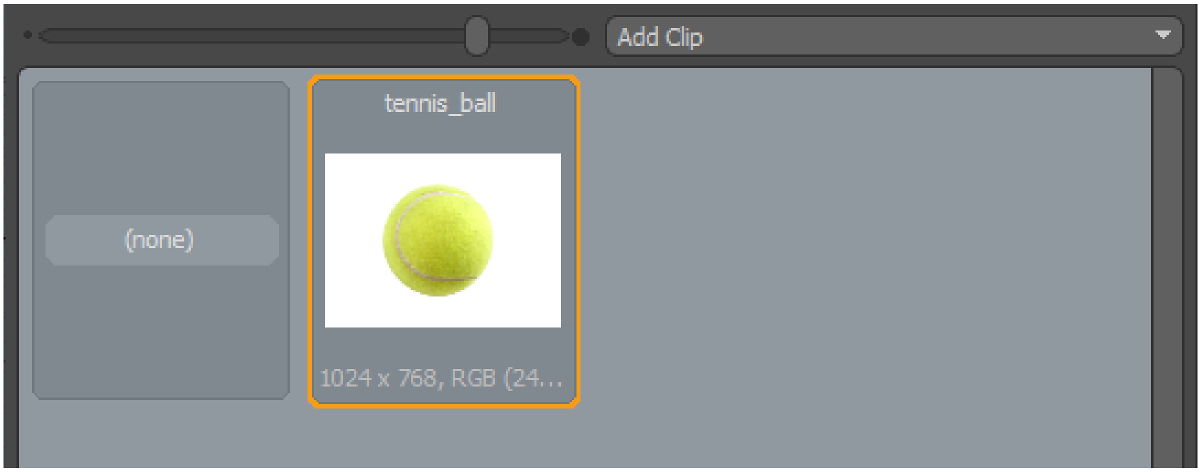
You can use Modo's Projection Texture to project a texture onto an object for a given value, like diffuse color, stencil, and so on. It works the same way as the Projection Shader, but while the Projection Shader is like a material and a shader in one, the Projection Texture is a material, requiring a shader.

Using the Projection Texture

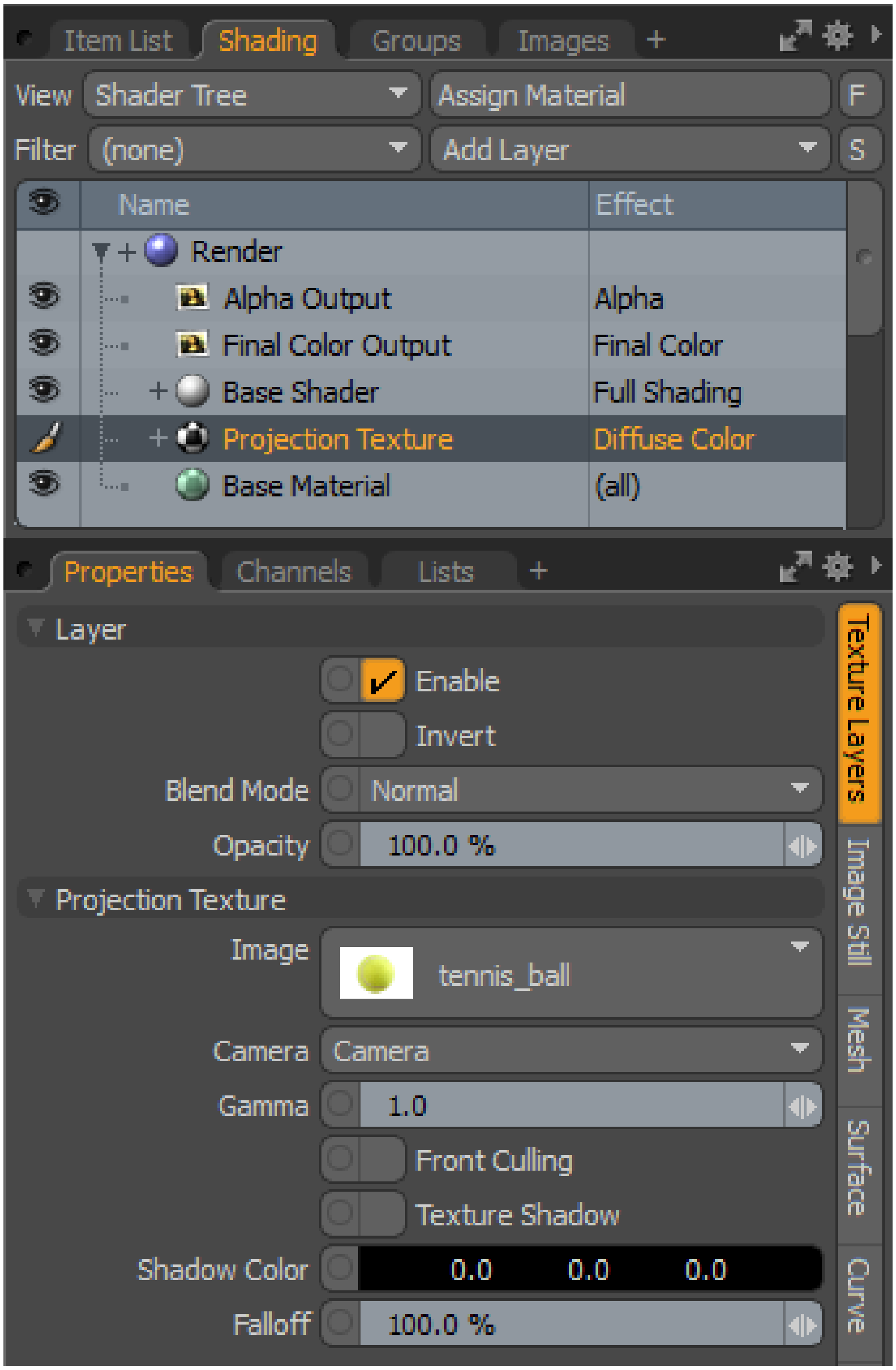
You can add a Projection Texture the following way:

1. In the **Shader Tree**, click **Add Layer** > **Special** > **Projection Texture**.

   The **Projection Texture** appears in the **Shader Tree**, and the **Clip Browser** window opens.


Add Clip
(none)
tennis_ball
1024 x 768, RGB (24...

2. Select the desired projection image by clicking the target image. If it's not already loaded, click the **Add Clip** button, and click **(load image)**.


Item List
Shading
Groups
Images
+
Assign Material
F
Add Layer
S
Properties
Channels
Lists
+
Texture Layers
Image Still
Mesh
Surface
Curve
Layer
Projection Texture

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Projection Texture	Diffuse Color
Base Material	(all)

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Image: tennis_ball

Camera: Camera

Gamma: 1.0

☐ Front Culling

☐ Texture Shadow

Shadow Color: 0.0 0.0 0.0

Falloff: 100.0 %

In the Projection Texture's **Properties** panel, the following **Layer** options are available:

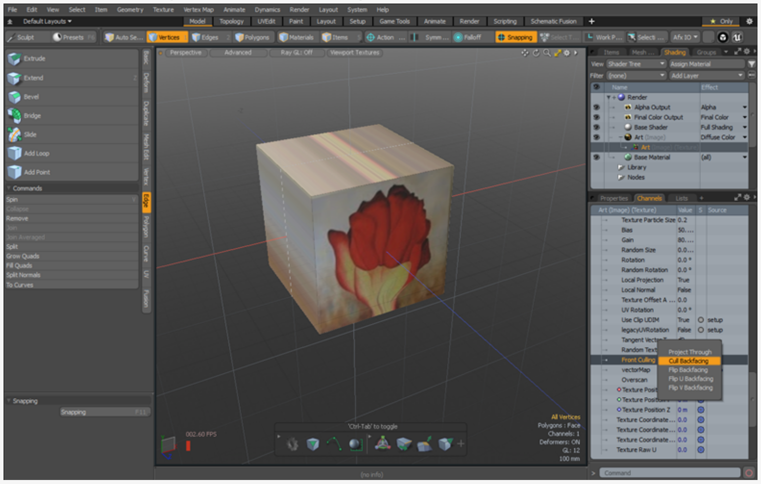
- **Enable** - toggles the effect of the layer on and off, duplicating the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, disabled layers are saved with the scene and are persistent across Modo sessions.
- **Invert** - inverts the RGB values for the layer, producing a negative effect.
- **Blend Mode** - affects blending between different layers of the same effect type, allowing you to stack several layers for different effects. For more on blending, refer to the Layer Blend Modes topic.

- **Opacity** - changes the transparency of the current layer. Reducing these values increasingly reveals lower layers in the **Shader Tree** if present, or dims the effect of the layer itself on the surface.

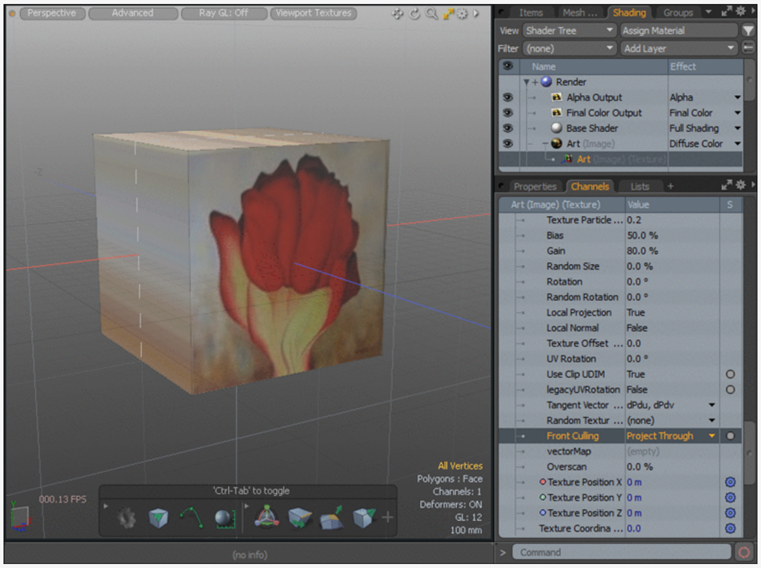
The following **Projection Texture** options are available:

- **Image** - uses the **Clip Browser** window to define the projected image. Once loaded, a projected image is stretched to fill the full frame, so make sure that the selected image is the same aspect ratio to avoid any distortion or stretching.
- **Gamma** - allows you to adjust the mid-tone values of the image, should a gamma correction be necessary. For most cases, the default 1.0 value should suffice, where Gamma is handled by the Color Management system.
- **Front Culling** - creates a texture shadow, that means it only maps the parts of the geometry directly seen by the camera, the parts hidden from the camera POV use the texturing below.

Tip: You can set the type of projection mapping for the **Front Culling** option in the **Channels** tab. Open the **Shading** tab, select the project image you've added, and then click the **Value** field for **Front Culling**. The following options are available:

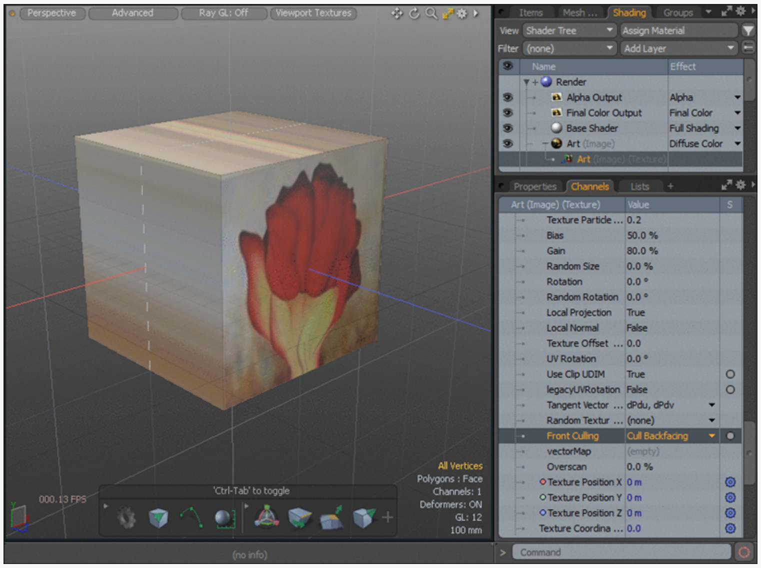

Front Culling
Project Through
Cull Backfacing
Flip Backfacing
Flip U Backfacing
Flip V Backfacing

- **Project Through**: Projects the texture image onto the back face of the mesh, along negative Z direction, and is mirrored to appear the same as the front face image. Click on the image below to view an animation.


Perspective
Advanced
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Art (Image)
Diffuse Color
Art (Image) (Texture)
Properties
Channels
Lists
All Vertices
Polygons : Face
Channels: 1
Deformers: ON
GL: 12
100 mm
000.13 FPS
'Ctrl-Tab' to toggle
(no info)
Command

Art (Image) (Texture)	Value
Texture Particle ...	0.2
Bias	50.0 %
Gain	80.0 %
Random Size	0.0 %
Rotation	0.0 °
Random Rotation	0.0 °
Local Projection	True
Local Normal	False
Texture Offset ...	0.0
UV Rotation	0.0 °
Use Clip UDIM	True
legacyUVRotation	False
Tangent Vector ...	dPdu, dPdv
Random Textur ...	(none)
Front Culling	Project Through
vectorMap	(empty)
Overscan	0.0 %
Texture Position X	0 m
Texture Position Y	0 m
Texture Position Z	0 m
Texture Coordina ...	0.0

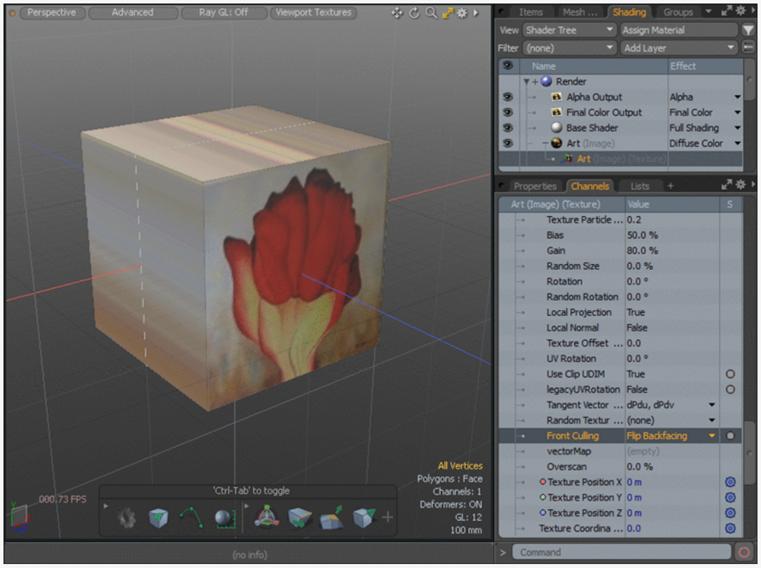
- **Cull Backfacing**: Projects the texture image only on the front of the mesh. Click on the image below to view an animation.


Perspective
Advanced
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Properties
Channels
Lists
All Vertices
Polygons : Face
Channels: 1
Deformers: ON
GL: 12
100 mm
000.13 FPS
'Ctrl-Tab' to toggle
(no info)
Command

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Art (Image)	Diffuse Color
Art (Image) (Texture)	

Art (Image) (Texture)	Value	S
Texture Particle ...	0.2	
Bias	50.0 %	
Gain	80.0 %	
Random Size	0.0 %	
Rotation	0.0 °	
Random Rotation	0.0 °	
Local Projection	True	
Local Normal	False	
Texture Offset ...	0.0	
UV Rotation	0.0 °	
Use Clip UDIM	True	○
legacyUVRotation	False	○
Tangent Vector ...	dPdu, dPdv	
Random Textur ...	(none)	
Front Culling	Cull Backfacing	●
vectorMap	(empty)	
Overscan	0.0 %	
Texture Position X	0 m	
Texture Position Y	0 m	
Texture Position Z	0 m	
Texture Coordina ...	0.0	

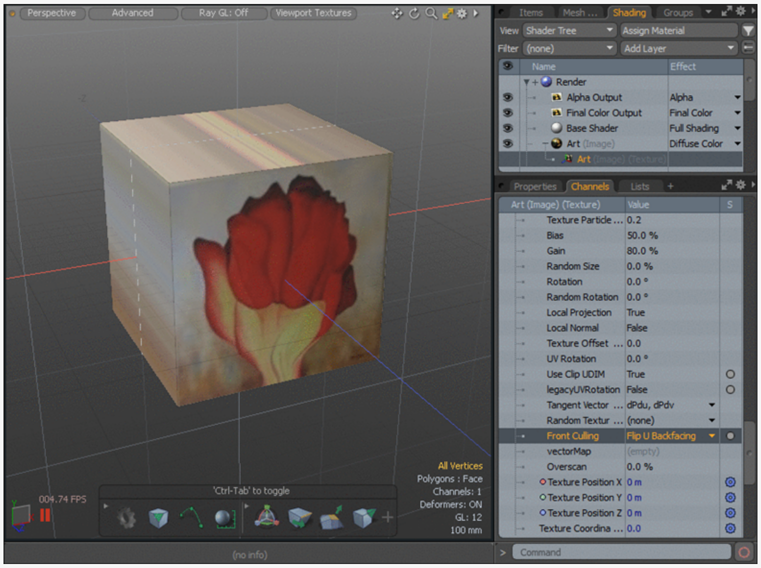
- **Flip Backfacing**: Projects and flips a texture image on the back of the mesh along the U and V directions. It appears up-side-down. Click on the image below to view an animation.


Perspective
Advanced
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Properties
Channels
Lists
All Vertices
Polygons : Face
Channels: 1
Deformers: ON
GL: 12
100 mm
000.73 FPS
'Ctrl-Tab' to toggle
(no info)
Command

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Art (Image)	Diffuse Color
Art (Image) (Texture)	

Art (Image) (Texture)	Value	S
Texture Particle ...	0.2	
Bias	50.0 %	
Gain	80.0 %	
Random Size	0.0 %	
Rotation	0.0 °	
Random Rotation	0.0 °	
Local Projection	True	
Local Normal	False	
Texture Offset ...	0.0	
UV Rotation	0.0 °	
Use Clip UDIM	True	○
legacyUVRotation	False	○
Tangent Vector ...	dPdu, dPdv	
Random Textur ...	(none)	
Front Culling	Flip Backfacing	●
vectorMap	(empty)	
Overscan	0.0 %	
Texture Position X	0 m	
Texture Position Y	0 m	
Texture Position Z	0 m	
Texture Coordina ...	0.0	

- **Flip U Backfacing**: Projects the texture image on a back face of the mesh, flipped along **U** direction, and it is oriented the same way as front face. Click the image below to view the animation.

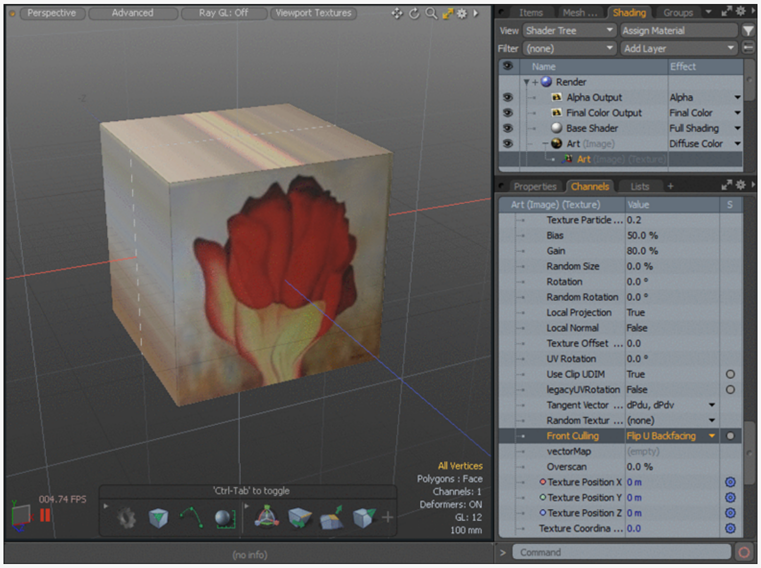

Perspective
Advanced
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Groups
Properties
Channels
Lists
'Ctrl-Tab' to toggle
004.74 FPS
All Vertices
Polygons : Face
Channels: 1
Deformers: ON
GL: 12
100 mm
(no info)
Command

View: Shader Tree
Assign Material
Filter: (none)
Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Art (Image)	Diffuse Color
Art (Image) (Texture)	

Art (Image) (Texture)	Value
Texture Particle ...	0.2
Bias	50.0 %
Gain	80.0 %
Random Size	0.0 %
Rotation	0.0 °
Random Rotation	0.0 °
Local Projection	True
Local Normal	False
Texture Offset ...	0.0
UV Rotation	0.0 °
Use Clip UDIM	True
legacyUVRotation	False
Tangent Vector ...	dPdu, dPdv
Random Textur ...	(none)
Front Culling	Flip U Backfacing
vectorMap	(empty)
Overscan	0.0 %
Texture Position X	0 m
Texture Position Y	0 m
Texture Position Z	0 m
Texture Coordina ...	0.0

- **Flip V Backfacing**: Projects the texture image on a back face of the mesh, flipped along **V** direction, and it is oriented the same way as front face. Click on the image below to view an animation.


Perspective
Advanced
Ray GL: Off
Viewport Textures
Items
Mesh ...
Shading
Groups
Properties
Channels
Lists
'Ctrl-Tab' to toggle
004.74 FPS
All Vertices
Polygons : Face
Channels: 1
Deformers: ON
GL: 12
100 mm
(no info)
Command

View: Shader Tree
Assign Material
Filter: (none)
Add Layer

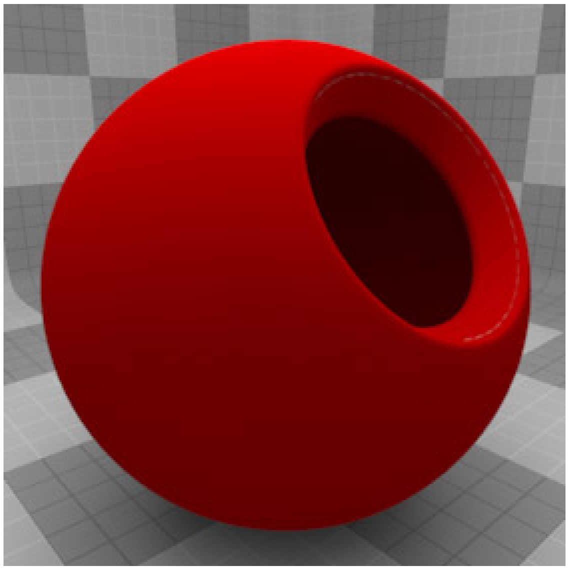
Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Art (Image)	Diffuse Color
Art (Image) (Texture)	

Art (Image) (Texture)	Value
Texture Particle ...	0.2
Bias	50.0 %
Gain	80.0 %
Random Size	0.0 %
Rotation	0.0 °
Random Rotation	0.0 °
Local Projection	True
Local Normal	False
Texture Offset ...	0.0
UV Rotation	0.0 °
Use Clip UDIM	True
legacyUVRotation	False
Tangent Vector ...	dPdu, dPdv
Random Textur ...	(none)
Front Culling	Flip U Backfacing
vectorMap	(empty)
Overscan	0.0 %
Texture Position X	0 m
Texture Position Y	0 m
Texture Position Z	0 m
Texture Coordina ...	0.0

- **Shadow Color** - instead of using the texturing of a material in the front-culled areas, you can specify a diffuse color.
- **Falloff** - sets the amount of falloff from the projection on surfaces not facing the camera.

Processing

Constant

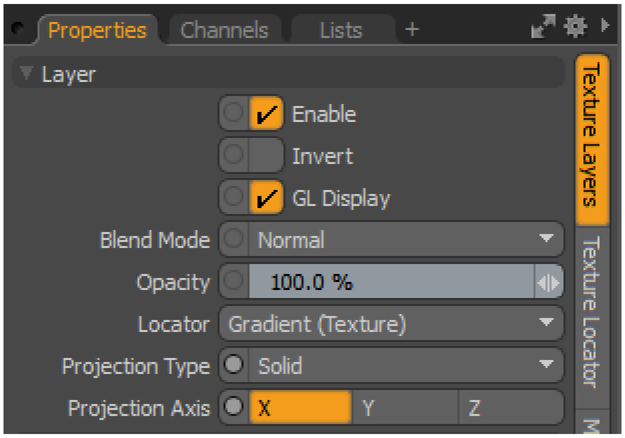


The Constant texture generates a layer with a constant color (or value). Use this to set a single color or value for a specific effect without needing to create a material. For example, with a Constant texture set to **Diffuse Color** and applied inside an item or polygon mask, the masked elements all use the material settings from the last material beneath the mask but each can have its own **Diffuse Color** setting.

> **Note:** For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

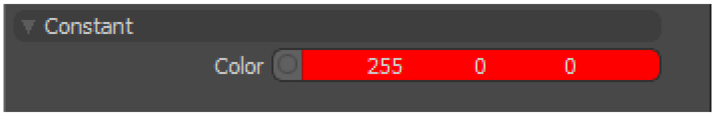
>



☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X (selected), Y, Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

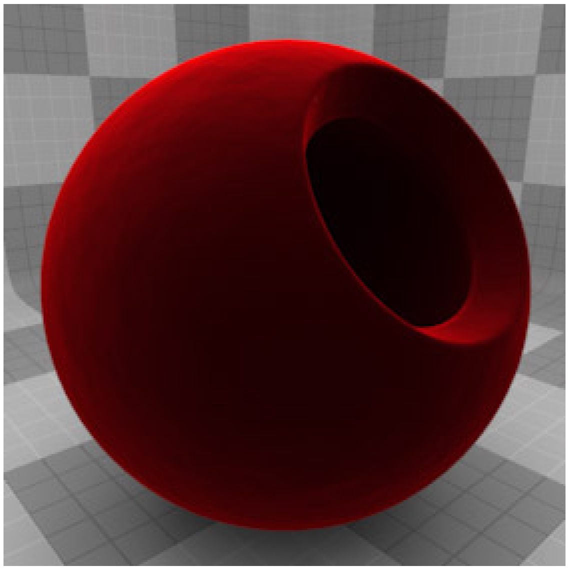
Constant Properties



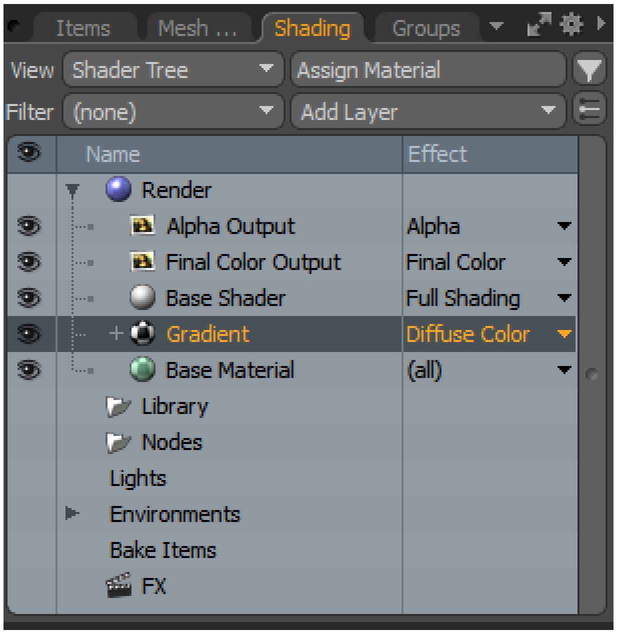
Color 255 0 0

Constant	
Color	Specifies the RGB color (or value) of the layer depending on the effect.

Gradient



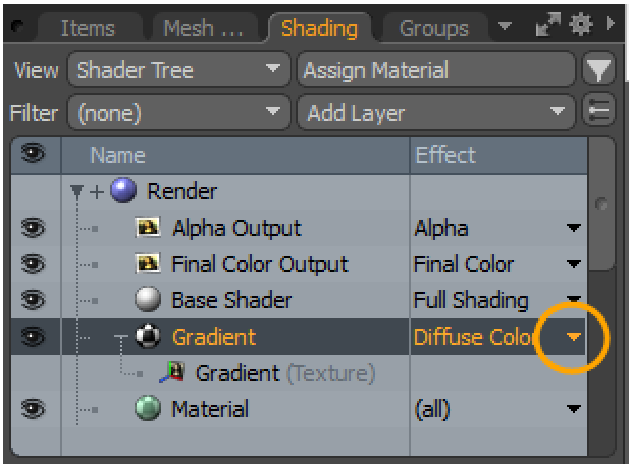
The Gradient texture layer is an incredibly powerful asset in the shading arsenal. Gradients provide a method for creating parametric materials. Put simply, gradients remap colors or value-based input parameters you define, and produce an output based on the values in the gradient itself. For instance, with a gradient texture you can tell a surface to ramp from purple to light blue based on the incidence angle to the camera or purple to green based on the slope of the surface. You can control the number of clones in a Replicator based on height and angle of a surface. Gradients are also useful in controlling the color of fur and Replicators. The following samples help provide more information about a particular gradient's use.


Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Gradient
Diffuse Color
Base Material
(all)
Library
Nodes
Lights
Environments
Bake Items
FX

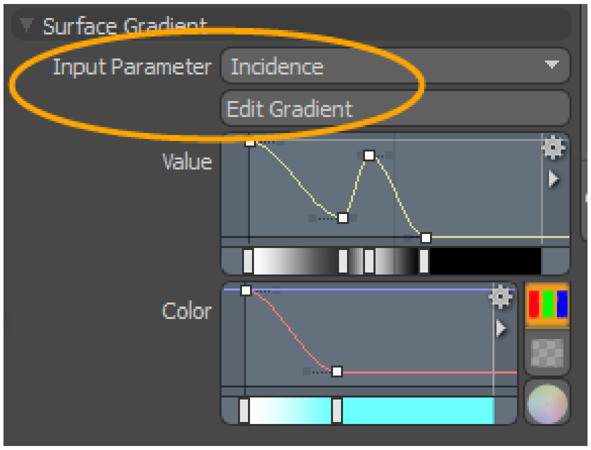
Applying a Gradient

Gradients are added to the **Shader Tree** and the effect is shown in the **Effect** column, such as Basic Channels > Transparency Color or Diffuse Color.

1. On the right panel, open the **Shading** tab and click **Add Layer**.
2. Select **Processing** > **Gradient**.
3. Select the parent **Gradient** item, click the dropdown menu under the **Effects** column, and select an option.


Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Gradient
Diffuse Color
Gradient (Texture)
Material
(all)

Next, you need to select the Input Parameters on the gradient texture's **Properties** panel to define how the gradient modulates the surface it is affecting.


Surface Gradient
Edit Gradient
Value
Color

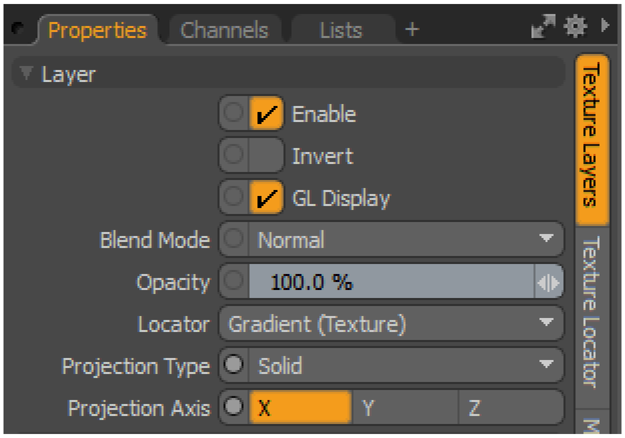
Input Parameter: Incidence

By clicking the **Edit Gradient**, you can open the Gradient Editor, where the actual gradient is defined. Its usage is covered below.

For additional information on adding and using layers, see the Shader Tree topic.

Layer Properties

>

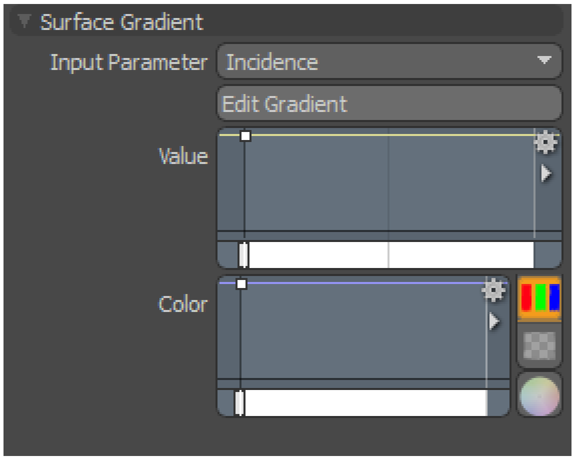

Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator
M

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects.

	For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

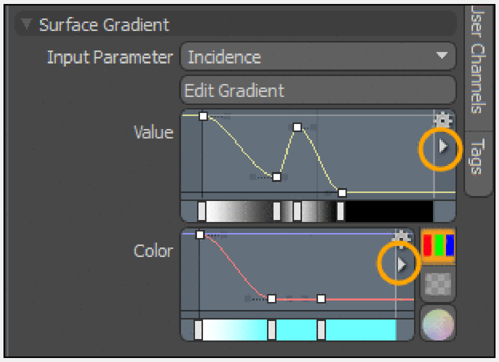
Surface Gradient Properties



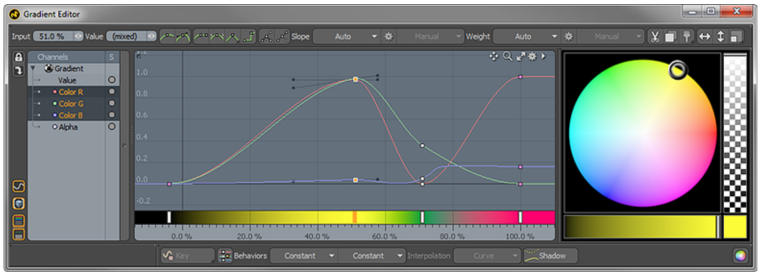
Surface Gradient

Input Parameter: Incidence

Input Parameters	The input parameters define the way in which the gradient is applied to its target. There are many different input parameter types to select from, organized by category. Demonstrated further below, you can select from any of the various categories to see the rendered examples. For the various input parameter types, see Input Samples.
Edit Gradient	Opens the Gradient Editor. User can create the gradient in this viewport or refine a

	gradient from the mini gradient editors.
Value/Color	The **Value** and **Color** mini gradient editors allow you to quickly define basic gradients directly within the main interface. • **Value** - This input is a black and white or grayscale type texture used for effects that typically look for a percentage amount. The gradient values are defined by keys, like keyframes in a timeline. Gradient keys define specific values at a defined location and then smoothly attenuate the values between the keys. Keys are created by middle-clicking within the mini-gradient editor, or on a Mac, by **Ctrl**+**Alt**+clicking. Defined keys can be edited by clicking and dragging the points up and down or left and right. • **Color** - This input is desirable for effects that look for RGB values such as where an image map might be used. Below the gradient key window is a graduated bar that represents the gradient itself. The window is limited to the 0%-100% range on either axis. Clicking on a key opens the **Gradient Editor**, which is used to control gradient textures in Modo. For more information, see **Gradient Editor**. You can save your **Value** and **Color** as presets for later use by clicking the down arrow. The preset is saved to the **Preset Browser** **Assets** > **Envelopes** directory. Once done, **Color Gradients** and **Percentage Gradients** directories are created.  Input Parameter: Incidence Surface Gradient Edit Gradient Value Color User Channels Tags You can also drag-and-drop saved presets into the Surface Gradient **Value** and **Color** graphs. For more information, see Saving Presets.

Using the Gradient Editor



Input 51.0 %
Value (mixed)
Slope Auto
Weight Auto
Behaviors Constant Constant
Interpolation Curve

The **Gradient Editor** is where you can define the gradient values. Though visually and functionally similar to the Graph Editor, the **Gradient Editor** assigns values over a range instead of over time as the **Graph Editor** does. Mostly, you need to edit the **R**, **G**, and **B** color, **Alpha** and **Value** channels. Modifying any other attributes reverts the bar to a Timeline where values are modified over time.

To open the Gradient Editor:

1. On the right panel, open the **Shading** tab, and select the **Gradient** item.
2. Under the **Surface Gradient** properties, click **Edit Gradient**.

   Alternatively, click **Layouts** > **Palettes** > **Gradient Editor**.

The window itself is divided into several sections. Across the top are the numeric input text fields followed by key controllers. These allow you to apply specific values to a key, as well as control the slope of incoming and outgoing curve handles. The left side of the window displays the channels available for editing. Modo generally selects the appropriate channel based on the layer's effect setting, but you can select **Value** or **RGB Colors** based on the layer's intended use. The center window is where you define the keys and modify their positions. New keys are added by middle-clicking in the graph area at the position where the key should be created, or with **Ctrl**+**Alt**+clicking on a Mac.

Once set, you can modify the key by clicking in the color palette on the right side. Positions of keys can be modified interactively by clicking on the individual handle bases (the white dots along the curves) and dragging them around in the viewport. Key positions can also be modified by dragging the handles that appear in the gradient bar below the graph area.

Video: This video explains the basic functionality of the **Gradient Editor**.

Using Channels

From within the channel section, you can select the value you wish to modulate. For diffuse color this is the **Color R**,**G** and **B** channels, for specular amount, this is the **Value** channel. Once selected, a line representing the value appears in the graphing window.

Setting Keys

Keys can be added to the currently selected curve(s) by middle-clicking either on the curve itself or anywhere on the graph area background.

Setting Colors for Keys

The color of selected keys can be set or adjusted by selecting the color key and either clicking and dragging on the **Color Picker** viewport or from the right-click context menu. Selecting **Key Color** from this menu opens the **Color Picker** from which you can select a color. In both cases additional keys can be created on associated curves at the key **Input** value.

Deleting Keys

Keys can be deleted from curves in two ways, to delete only the currently selected keys press the **Delete** key. To delete the currently selected keys and keys at the same **Input** value on associated curves, select **Delete Key** from the right-click context menu.

Making Selections

Clicking selects a key under the cursor, holding the **Shift** key while selecting adds to the selection, and holding **Ctrl** removes a key from the selection set. Clicking on the background deselects all keys.

Click and drag on the background starts box selection, drag out a box to enclose the keys you want to select and release the mouse button. All keys within the box become selected, any outside the box are deselected. Holding **Shift** adds the keys within the box to the selection, while **Ctrl** removes them.

Click on a curve in the background (dimmed) to select the curve. Curves can be added to the selection by holding **Shift** and clicking on a curve and can be removed from the selection with the **Ctrl** key.

Curves may also be selected from the menu at the top left of the viewport, this menu lists all the curves associated with the currently selected gradient item(s).

Editing Keys

Click and drag on a key selects the key if it's not already selected and allows quick editing of both input and output values at the same time. If there are several keys already selected and you click and drag one of them, all the selected keys are modified. Middle-click and drag on a key works in a similar way except that editing is constrained to the **Output** value. Similarly, right-click and drag constrains the edits just to the **Input Parameter** value.

Middle-clicking and dragging vertically on the background adjusts the **Output** value of all selected keys. Right-clicking and dragging horizontally on the background adjusts the **Input** value of all selected keys.

Ctrl+middle-clicking and dragging on the background provides constrained editing of **Input** and **Output** values for selected keys based on the initial direction of movement.

Double-clicking on a key selects all the keys on the curve.

Using the Context Menu

You can open the **Gradient Editor** context menu by right-clicking on the editor. If the cursor is over a key that is not selected, when the menu is opened the key is added to the selection. Operations performed from the context menu affect all the currently selected keys.

Navigating in the Gradient Editor

Panning and zooming behavior in the **Gradient Editor** is similar to the 3D Viewports with a few additions.

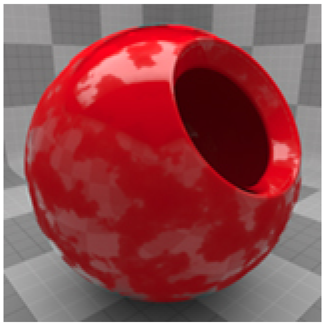
- **Alt**+middle-click and drag adjusts the start value of the **Input** (horizontal) range displayed in the editor, while **Alt**+right-click and drag adjusts the end value of the **Input Range**.
- **Alt**+**Shift**+middle-click zooms the **Output Range** displayed in the editor, while **Alt**+**Shift**+right-click zooms the **Input Range**. In both cases the zoom is centered on the initial cursor location.
- **Alt**+**Ctrl**+right-click provides box zooming.
- The mouse wheel zooms in and out around the cursor location.

Input Samples

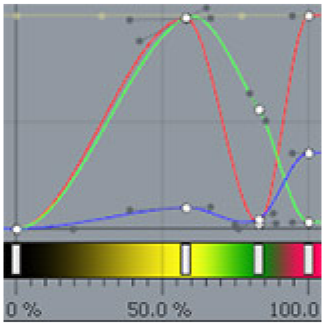
Modo has many different input parameter types organized into categories. The input parameters specify the way in which the gradient is applied to its target. This topic provides examples of each type. Click the **+** icon to reveal the description of a category.

Input Parameter - Basic

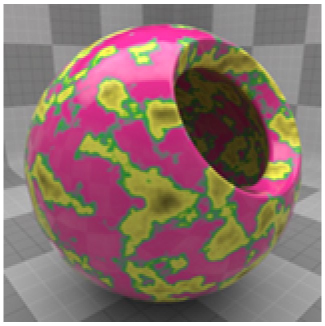
Clearcoat Amount - When the **Input Parameter** is set to **Clearcoat Amount**, the gradient keys off of any clearcoat texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the Gradient only registers the corresponding single gradient color/value.



Clearcoat

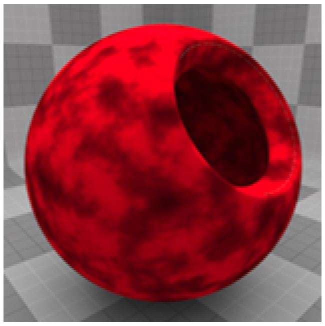


Gradient

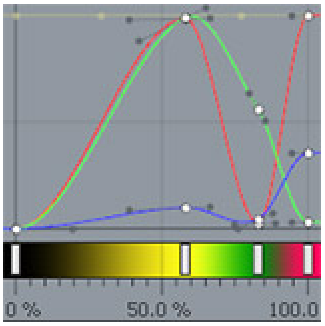


Clearcoat Gradient

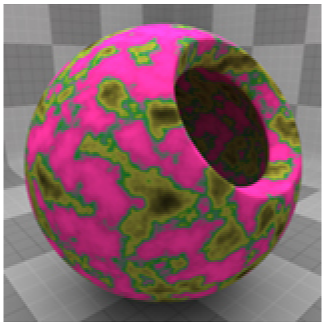
Diffuse Amount - When the **Input Parameter** is set to **Diffuse Amount**, the gradient keys off of any diffuse amount texture layer present on the surface. Keep in mind the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Diffuse Amount

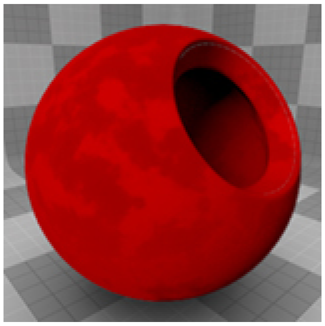


Gradient

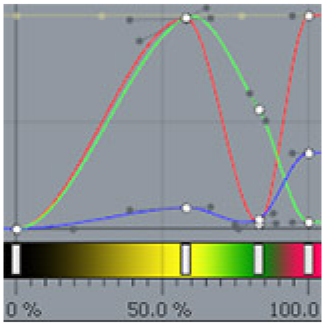


Diff. Amt. Gradient

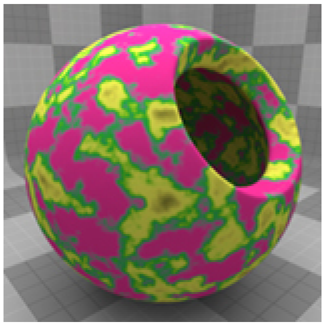
Diffuse Roughness - The gradient keys off of any diffuse roughness texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Diffuse Roughness

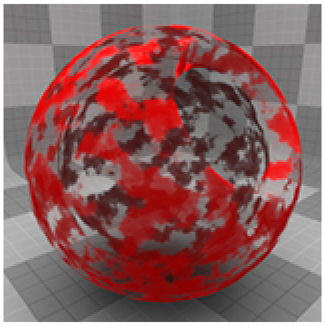


Gradient

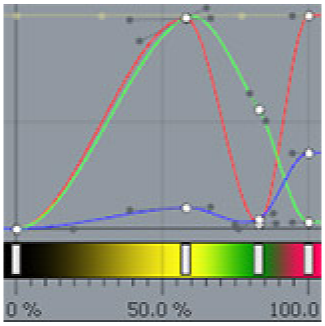


Diff. Roughness Gradient

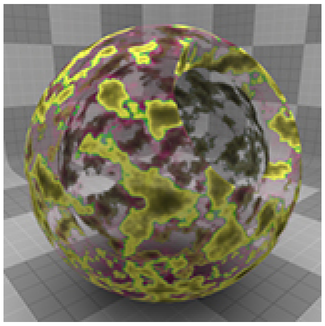
Dissolve - The gradient keys off of any dissolve texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Dissolve

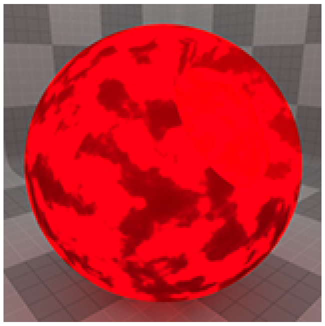


Gradient

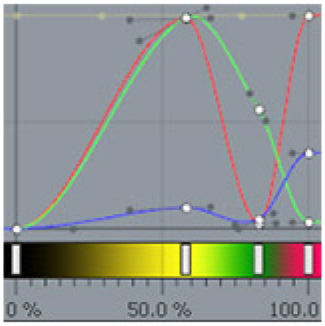


Dissolve Gradient

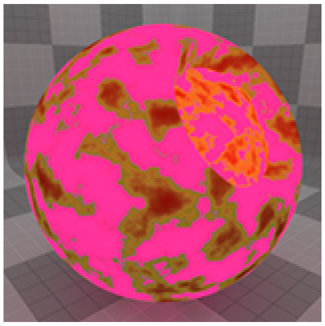
Luminous Amount - The gradient keys off of any luminous amount texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Luminous Amount

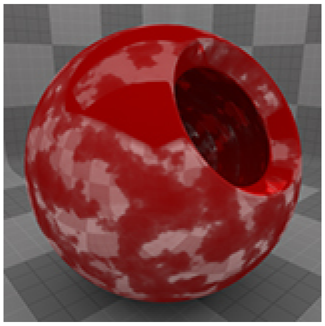


Gradient

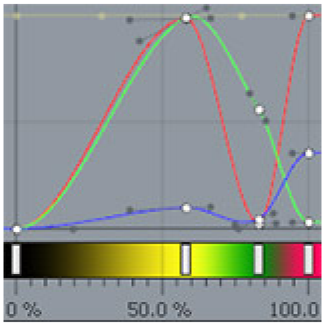


Luminous Amt. Gradient

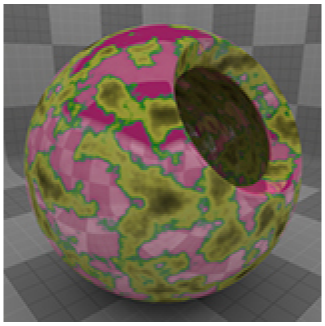
Reflection Amount - The gradient keys off of any reflection amount texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Reflection Amount

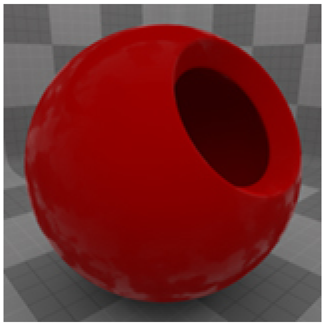


Gradient

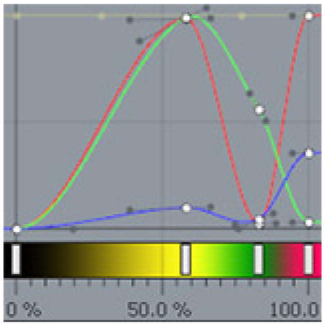


Reflection Amt. Gradient

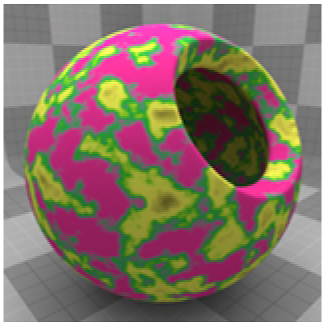
Reflection Fresnel - The gradient keys off of any Fresnel texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Reflection Fresnel

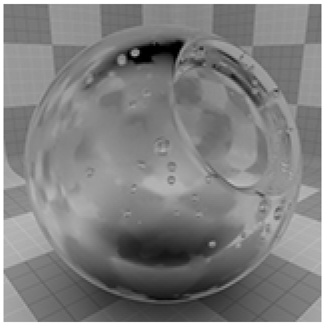


Gradient

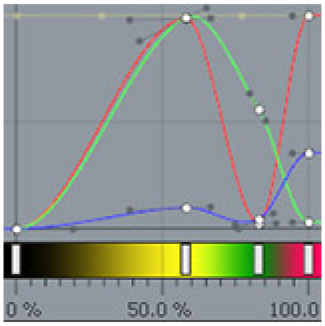


Ref. Fresnel Gradient

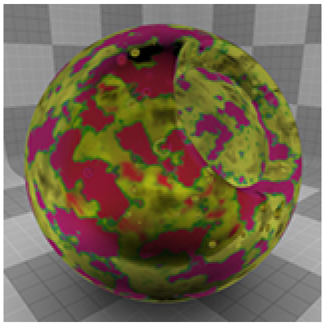
Refraction Roughness - The gradient keys off of any refraction roughness texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Refraction Roughness

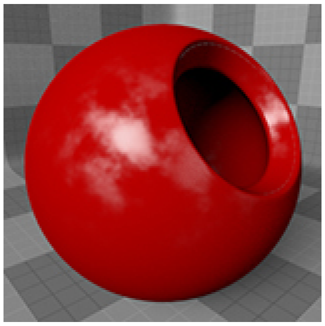


Gradient

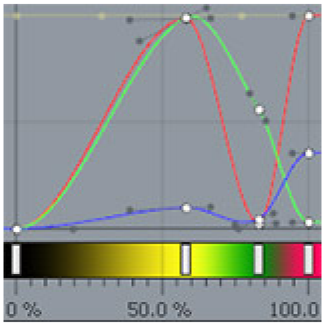


Clearcoat Gradient

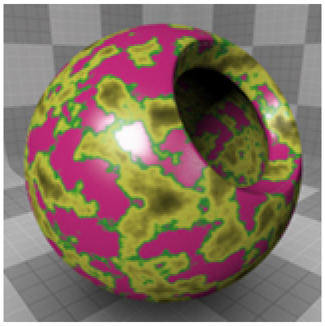
Roughness - The gradient keys off of any roughness texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Roughness

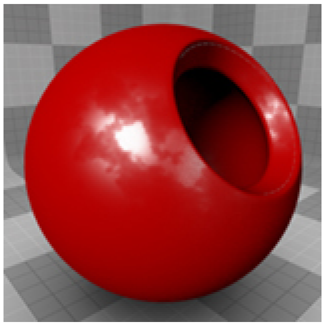


Gradient

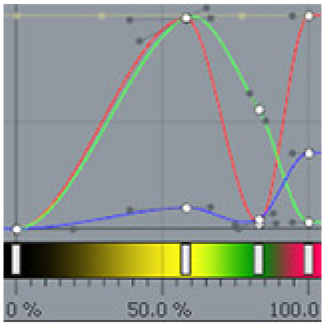


Roughness Gradient

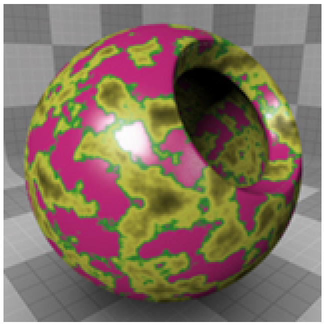
Specular Amount - The gradient keys off of any specular amount texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Specular Amount

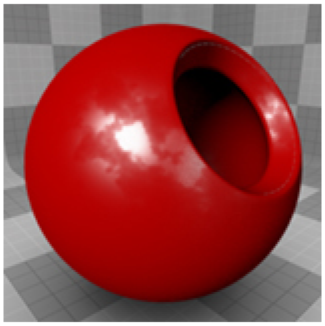


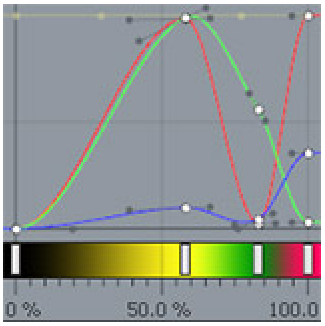
Gradient

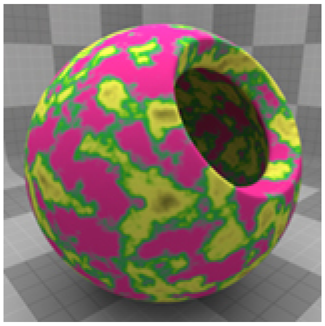


Specular Gradient

Specular Fresnel - The gradient keys off of any specular Fresnel texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.





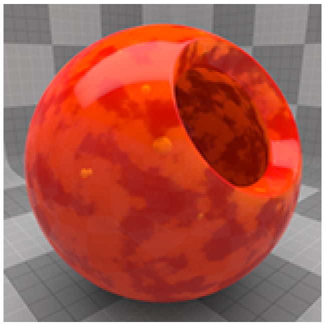


Specular Fresnel

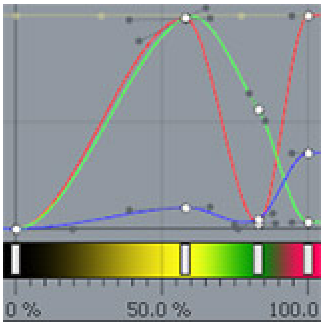
Gradient

Spec. Fres. Gradient

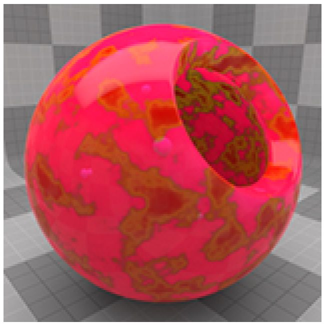
Subsurface Amount - The gradient keys off of any subsurface amount texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Subsurface Amount

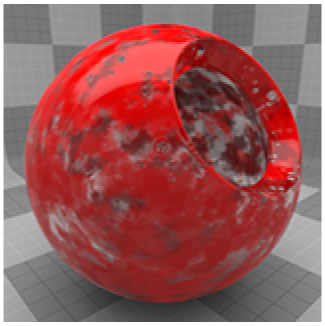


Gradient

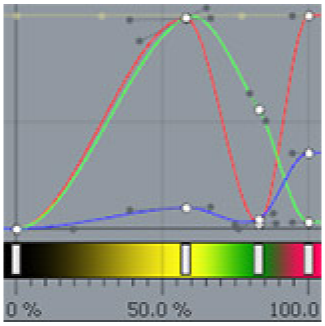


Subsurface Gradient

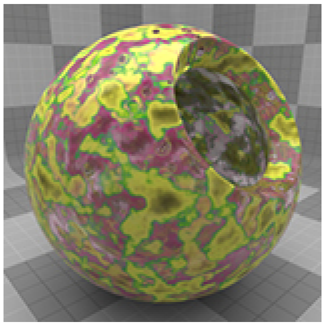
Transparent Amount - The gradient keys off of any transparent amount texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.



Transparent Amount



Gradient



Transparent Gradient

Input Parameter - Custom Material

Cel Edge Width - When the **Input Parameter** is set to **Cel Edge Width**, as related to a Cel Edge material, the gradient keys off of any Cel Edge width texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Halftone Channels - When the **Input Parameter** is set to any of the **Halftone Channels**, as related to a Halftone Material, the gradient keys off of any of the Halftone texture layer settings present on the surface.

Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Skin Material Channels - When the **Input Parameter** is set to any of the **Skin Material Channels**, as related to a Skin Material, the gradient keys off of any of the Skin material layer settings present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Input Parameter - Fur

Fur Bump - When the **Input Parameter** is set to **Fur Bump**, as related to a Fur Material, the gradient keys off of any fur bump texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Clump Density - The gradient keys off of any Clump Density texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Clumps - The gradient keys off of any Fur Clumps texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Curls - The gradient keys off of any Fur Curls texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

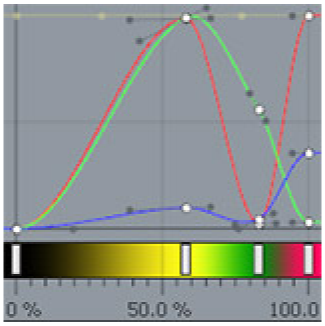
Fur Density -The gradient keys off of any Fur Density texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Flex - The gradient keys off of any Fur Flex texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

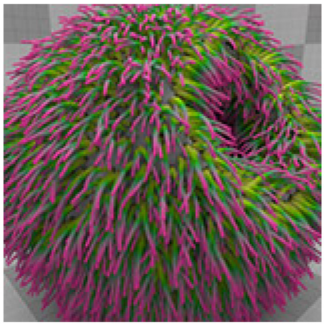
Fur Growth Jitter - The gradient keys off of any Growth Jitter texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Length - The gradient keys off of any Fur Length texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Parametric Length - The gradient applies its colors/values along the length of individual fur fibers, 0% at the root, to 100% at the tip.



Gradient



Fur Length

Fur Root Bend - The gradient keys off of any Root Bend texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Root Frizz - The gradient keys off of any Root Frizz texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Root Kink - The gradient keys off of any Root Kink texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

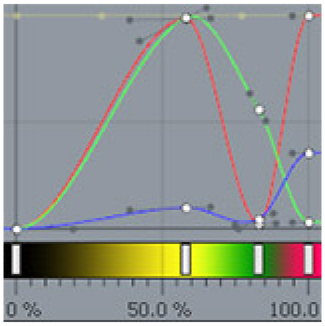
Fur Stray Hairs - The gradient keys off of any Stray Hairs texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Fur Tip Frizz - The gradient keys off of any Tip Frizz texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

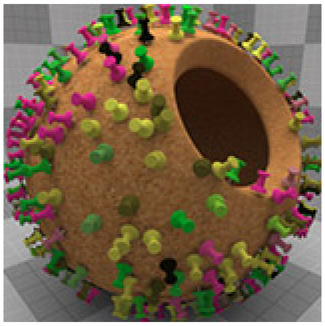
Fur Tip Kink - The gradient keys off of any Tip Kink texture layer present on the surface. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Input Parameter - Particle Effects

Particle Age - When the **Input Parameter** is set to **Particle Age**, the gradient keys off Particle Age values present in a particle simulation and imported RealFlow particle animation, providing additional controls over the animation of Blobs, Sprites, and Volumes.
Particle ID - When fur fibers, surface generators, point clouds and particle simulations are created, each is assigned a random **Particle ID** value between 0 and 1. When a gradient is applied with the **Particle ID** input parameter, each fiber or replica pulls a value from the gradient based on that randomly generated ID, allowing you to create controlled randomness. This makes it easy to generate, for instance, 'salt and pepper' fibers for a fur material.



Gradient



Replicator Particle ID

Fur Particle ID

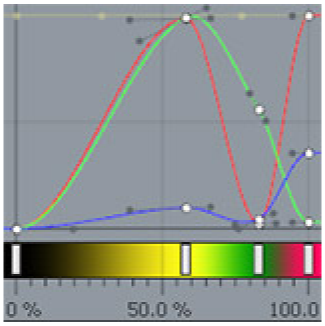
Particle Velocity - When the **Input Parameter** is set to **Particle Velocity**, the gradient keys off the speed the particle is traveling for the current frame of animation in a particle simulation and imported RealFlow particle animation, providing additional controls over the animation of blobs, sprites and volumes.

Texture Particle ID - A Particle ID for Texture Replicators. When you distribute textures with the Texture Replicator, each texture gets a Texture Particle ID. When a gradient is applied with the **Texture Particle ID** input parameter, each replica pulls a value from the gradient based on that randomly generated ID, allowing you to create controlled randomness.

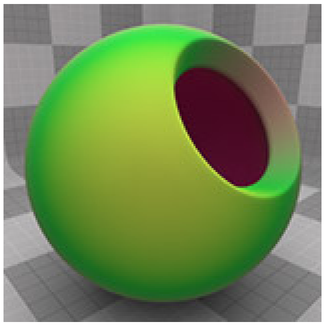
Input Parameter - Sample Parameters

Distance to Camera - Modulates the value as a function of the physical distance to the camera measured from its center point. This opens a lot of interesting possibilities, such as adding fog-like effects or fading luminosity as the surface gets further from the camera.

Note: A gradients layer isn't strictly limited to affecting diffuse color.

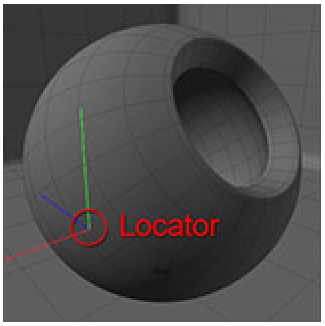


Gradient

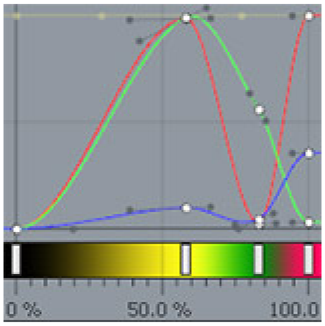


Distance to Camera

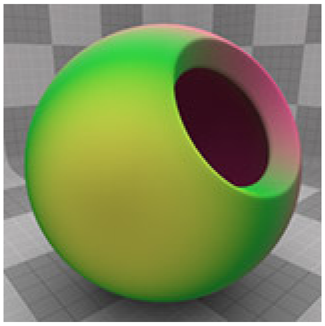
Distance to Locator - Very similar to the **Distance to Camera** parameter, this parameter modulates the gradient based on the position of the item's texture locator, specified by the **Locator** option above.



Item Locator Position

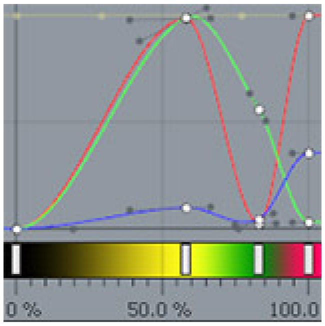


Gradient

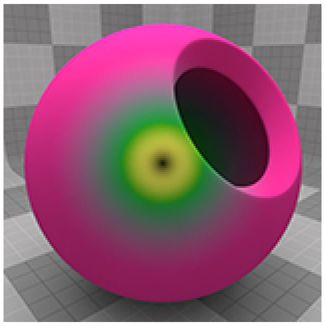


Dist. to Locator

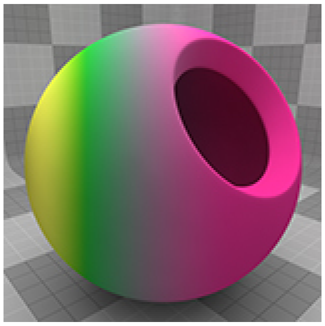
Screen R/X/Y - These are screen space gradients that apply the gradient to the surface spanning the rendered frame, either radially (**R**), horizontally (**X**), or vertically (**Y**).



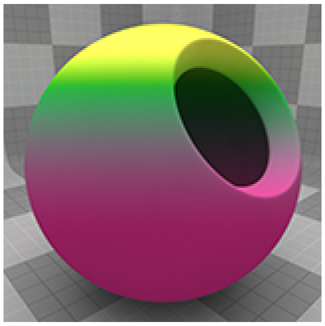
Gradient



Screen R

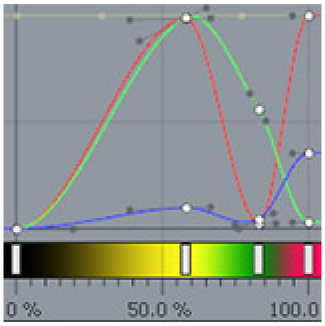


Screen X

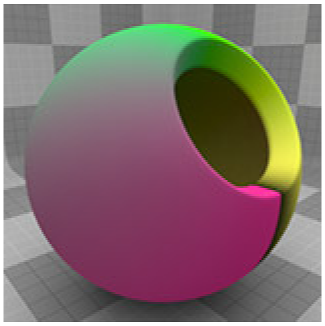


Screen Y

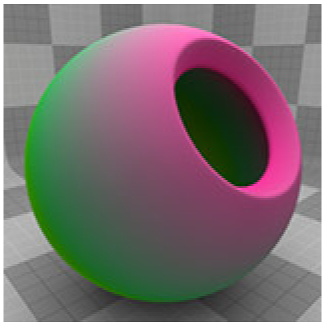
Texture U/V - The gradient reads the texture coordinates from the UV map and modulate the value based on the **U** (horizontal) or **V** (vertical) value of the map.



Gradient



Texture U



Texture V

Texture Value - This setting applies the gradient across the values of any texture layer directly beneath it, regardless of that texture's **Effect**, **Blending** or **Opacity** settings.

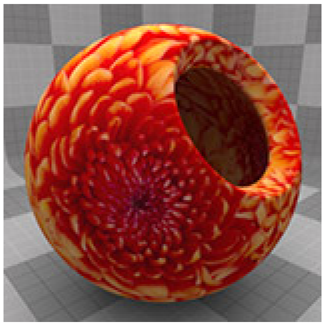
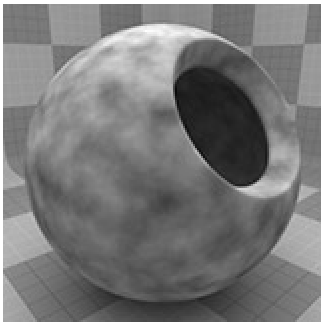
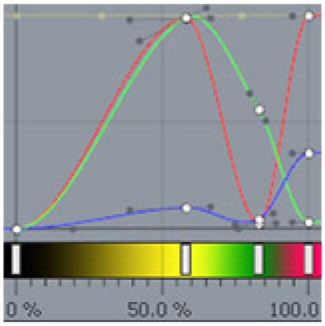


Image Map Texture



Procedural Texture



Gradient

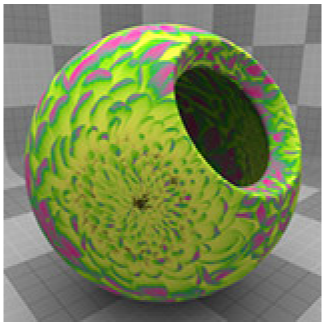
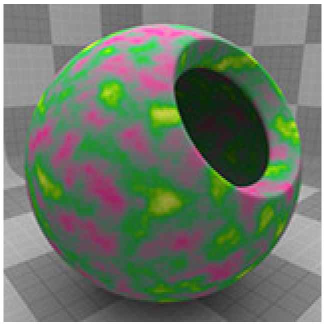
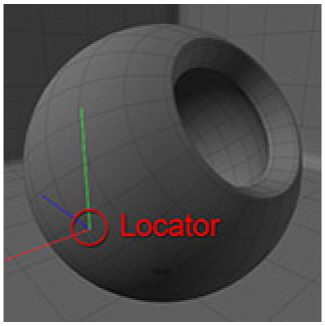


Image Map Grad. result

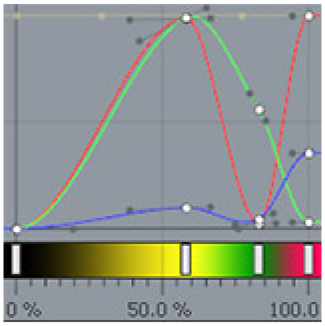


Procedural Grad. result

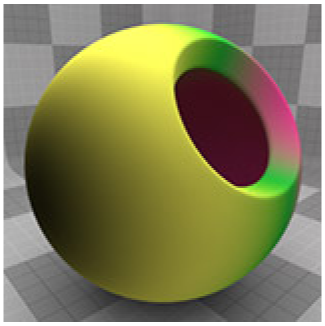
XYZ Distance to Locator - Creates a distance-based linear gradient that aligns itself along either the X, Y or Z axes as specified, with the gradient's origin controlled by the position of the item's texture locator, specified by the **Locator** option above.



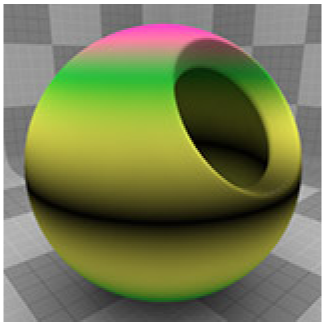
Item Locator Position



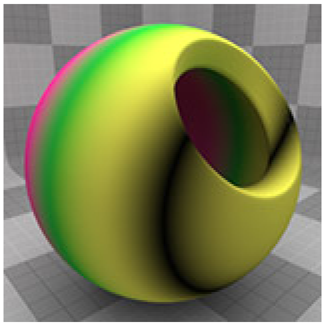
Gradient



X Distance



Y Distance

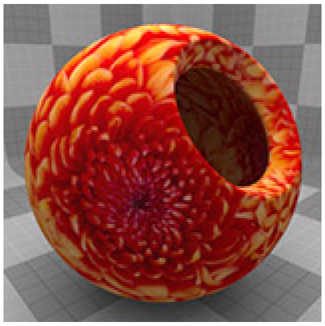


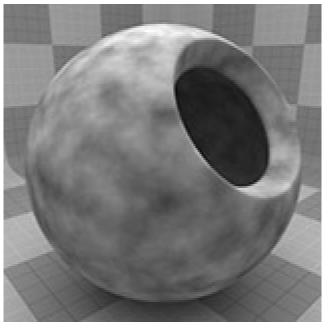
Z Distance

Input Parameter - Shader Control

Driver A, B, C, D - This settings read any texture item or combination thereof, whose **Effect** is set to the same **Driver** letter (**A**, **B**, **C** or **D**). This allows you to have texture layers that can drive gradients very easily. It also allows you to remap the tones of a texture layer by using the gradient.

Note: When using colored bitmap textures to drive the gradient, the actual driving property is the brightness or luminosity of the image. The gradient needs to be above the driving texture in the **Shader Tree** and that the letter value of the texture items **Effect** needs to match the gradient's **Input Parameter** value (for example, both need to be set to **Driver A** for it to work).





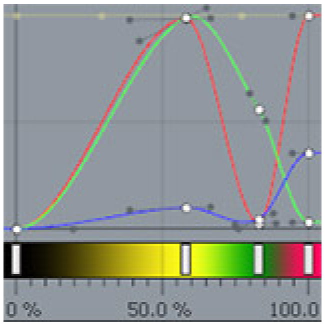


Image Map Driver

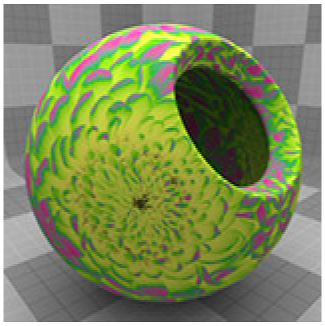
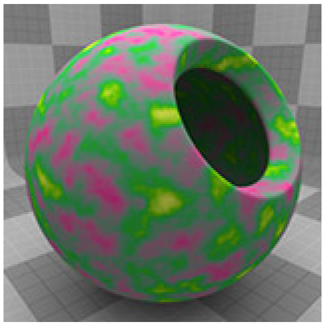


Image Map Grad. result

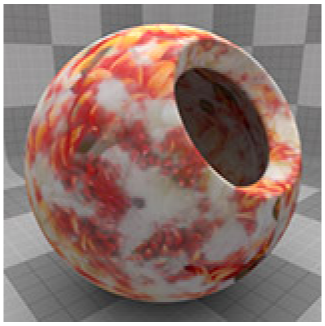
Procedural Driver



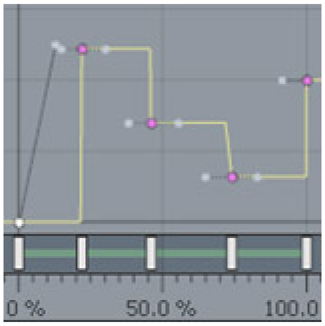
Procedural Grad. result

Gradient

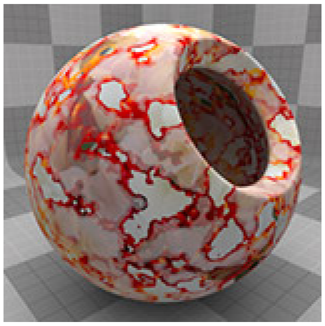
Group Mask - When the gradient layer item's **Input Parameter** is set to **Group Mask** and the **Effect** is set to the same, a gradient can modulate any texture layers themselves set as **Group Mask** (within the same group, of course). When the **Effect** is set to anything else, the gradient is driven by the group mask items, much like the **Driver** input parameter, overriding the effect of the mask on the group.



Base Group Mask

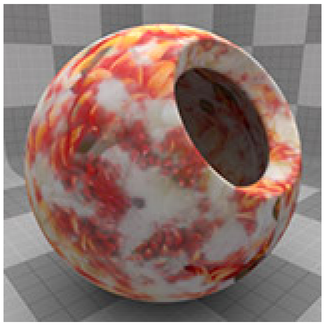


Stepped Gradient

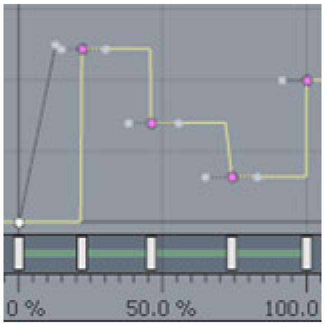


Modulated Mask

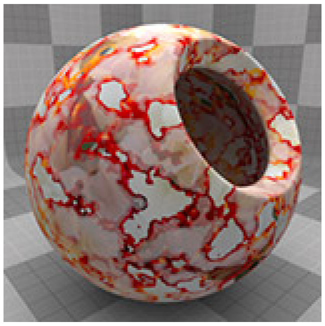
Layer Mask - This works identically to the **Group Mask** input parameter, but where a **Group Mask** affects an entire group, a layer mask only affects a single layer. Setting the gradient layer's **Effect** to **Layer Mask**, the gradient modulates the texture layer directly below. When the gradient's **Effect** is set to anything else, the gradient is driven by the layer mask items, much like the **Driver** input parameter, overriding the effect of the mask.



Base Layer Mask



Stepped Gradient



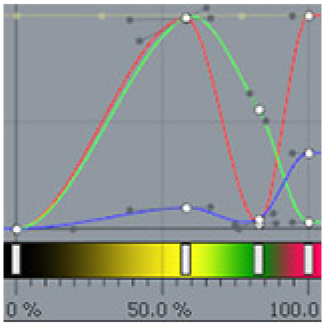
Modulated Mask

Input Parameter - Special Effects

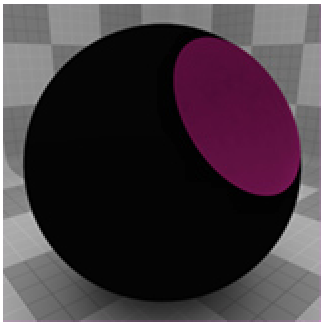
Stencil - When the **Input Parameter** is set to **Stencil**, the gradient keys off of any texture layers with their effect designated as Stencil. Keep in mind that the gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Input Parameter - Surface Parameters

Back Facing - The gradient colors both the front and rear sides of a polygon. The **Color** or **Value** at the 0% position shades the front of the polygon, while the **Color** or **Value** at the 100% position shades the opposite side of the polygon. You first need to enable the **Double Sided** option in the Material item to see the results of this gradient parameter.

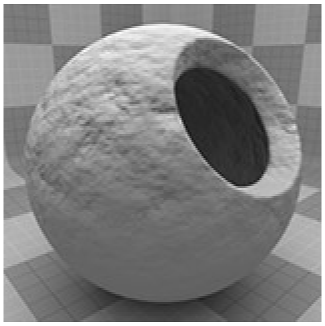


Gradient

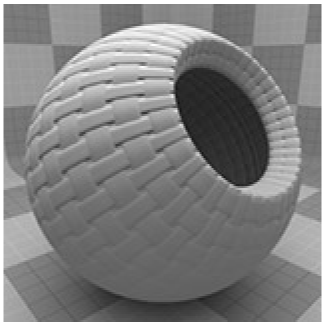


Back Facing

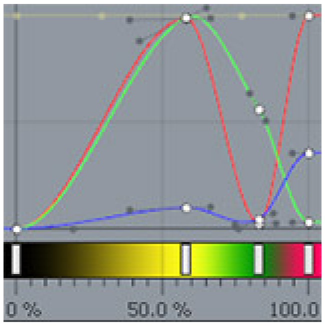
Bump Height - The gradient is driven by any bump map textures that are affecting the same material, and the input value becomes a percentage where 0% is the lowest area of the bump texture and 100% is the highest peak. If there are no texture layers driving the bump map effect the gradient is disabled.



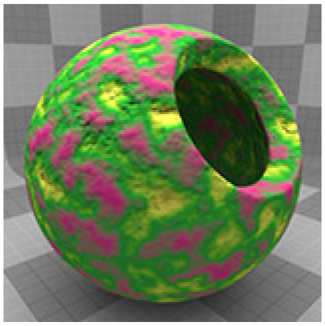
Noise Bump Map



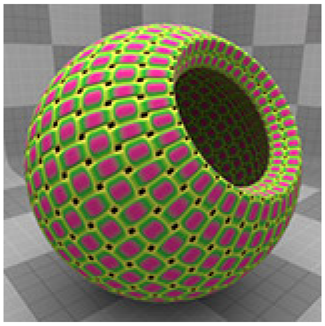
Weave Bump Map



Gradient

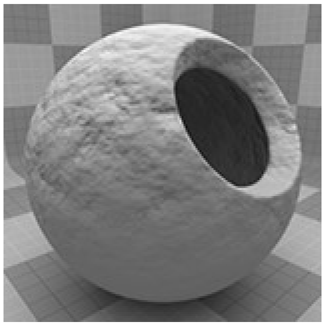


Bump Height Gradient

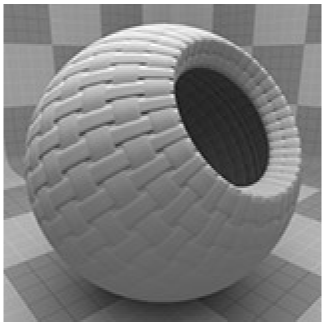


Bump Height Gradient

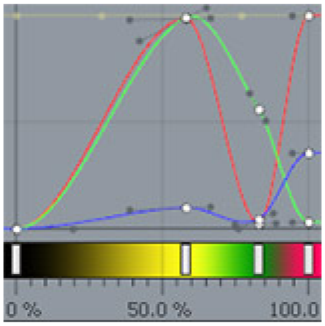
Bumpless Incident - When set to **Bumpless Incident** the Gradient ignores any Bump Map textures that are effecting the same material and apply the gradient as if it weren't present providing in easy way to produce faked transparency shading or thickness effects.



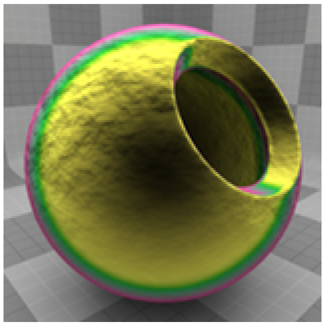
Noise Bump Map



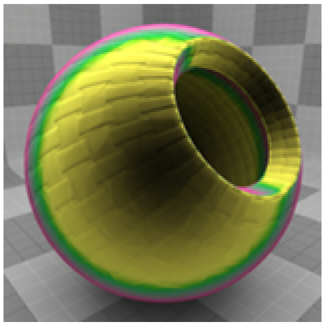
Weave Bump Map



Gradient

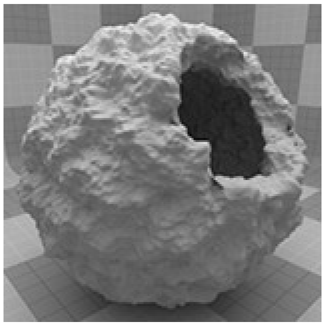


Bumpless Incidence

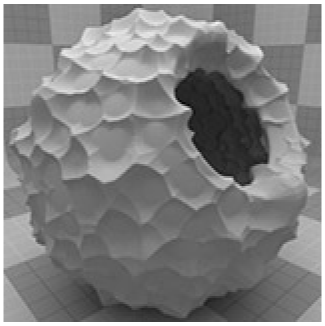


Bumpless Incidence

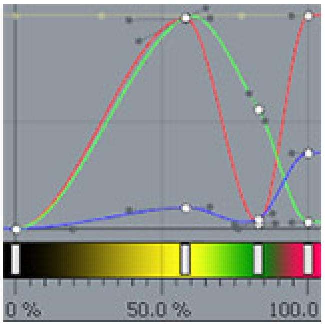
Displacement Height - Similar to **Bump Height**, this input parameter is driven by a displacement texture layer directly below the gradient in the **Shader Tree** and maps itself along the displacement height amount. 0% at the lowest displacement area, and 100% at the maximum displacement amount. Negative gradient values may also be necessary for remapped displacement values (that is, changing the low/high values in the texture item to -100%, 100% respectively).



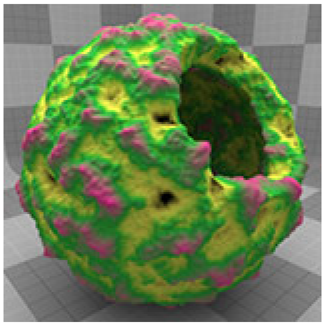
Displacement Map



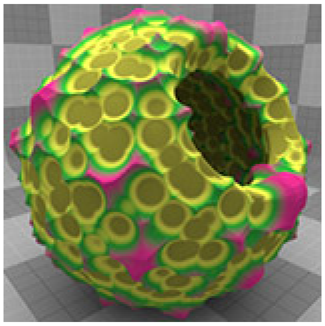
Displacement Map



Gradient

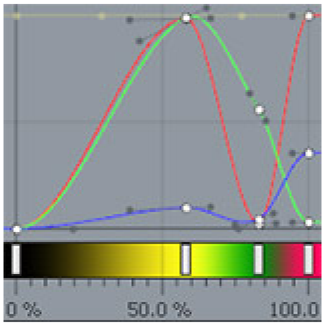


Disp. Height Gradient

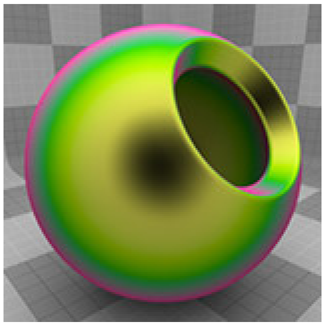


Disp. Height Gradient

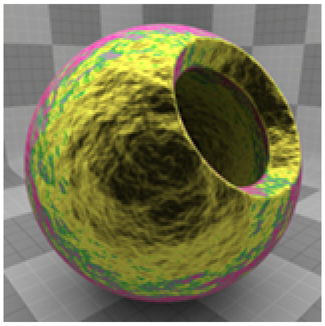
Incidence Angle - This is the relative angle between the camera and the surface being evaluated. Surfaces that face the camera head on have an incidence value of 0% while the edges that face away (perpendicular or glancing) to the camera have a value of 100%. This setting is often used to simulate the Fresnel effect as well as simulate the look of velvet surfaces.



Gradient

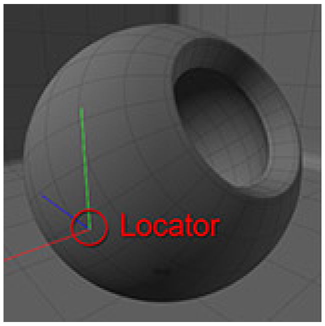


Incidence

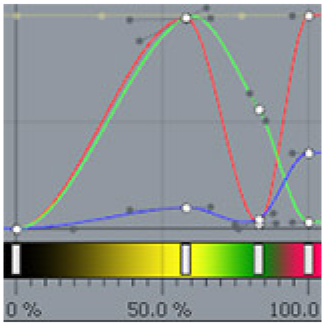


Incidence w. Bump

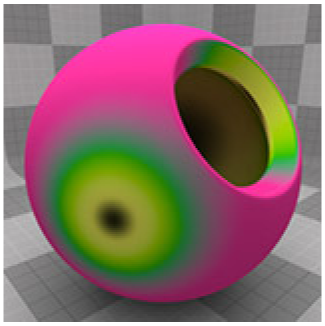
Locator Incidence - This parameter is identical in effect to **Incidence Angle**, but can be controlled by a locator's position instead of fixed to the camera's point of view.



Item Locator Position

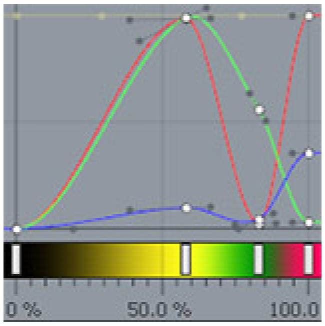


Gradient

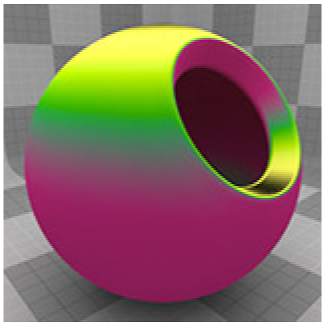


Locator Incidence

Slope - This is the angle of the surface normal as compared to the World axes. 0° straight up is 0%, and 90° perpendicular is 100%.

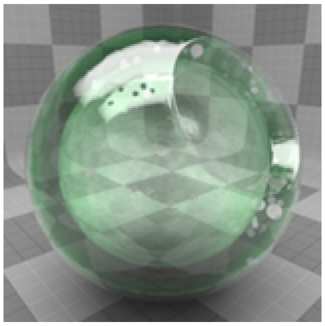


Gradient

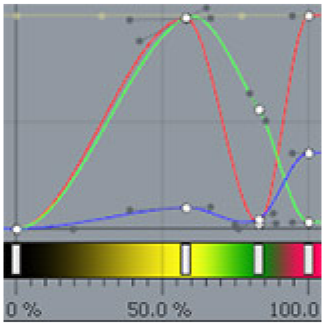


Slope

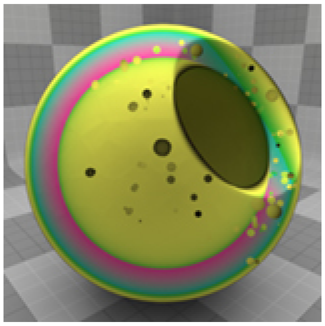
Thickness - Determines how thick or deep an area is and applies the gradient based on the measured depth. The **Thickness** is calculated by a ray being fired inward from the first surface, until the next outward facing surface is encountered (the backside) or another surface is encountered in the case of surface intersections.



Transparent Surface



Gradient



Thickness Gradient

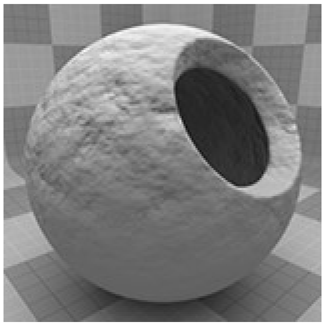
Input Parameter - Surface Particle Generation

Surface Particle Density - The gradient keys off of the **Surface Particle Density** texture layer present on the surface. Keep in mind that the gradient maps across multitudes values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

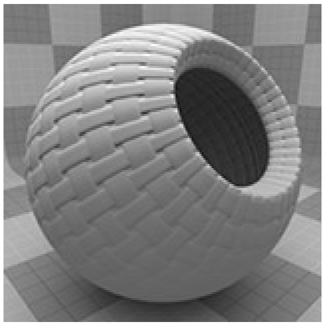
Surface Particle Size - The gradient keys off of the **Surface Particle Size** texture layer present on the surface. Keep in mind the gradient maps across values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

Input Parameter - Surface Shading

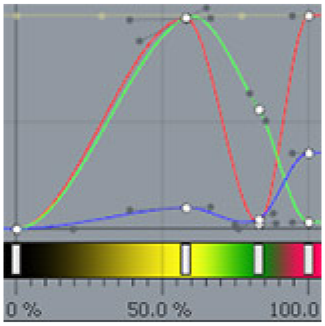
Bump - The gradient is driven by any bump map textures that are affecting the same material and the input value becomes a percentage where 0% is the lowest area of the bump texture and 100% is the highest peak. If there are no texture layers driving the bump map effect, the gradient is disabled.



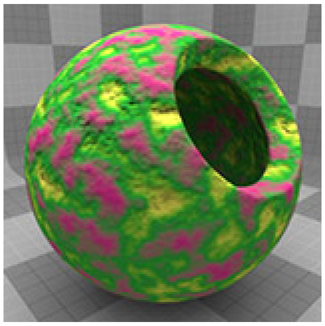
Noise Bump Map



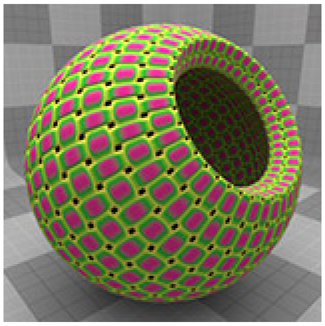
Weave Bump Map



Gradient

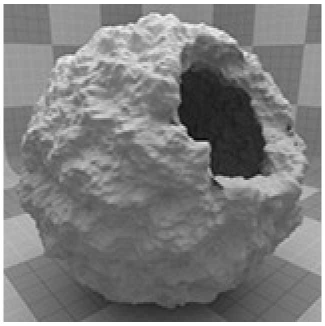


Bump Height Gradient

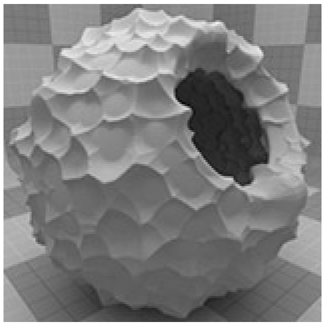


Bump Height Gradient

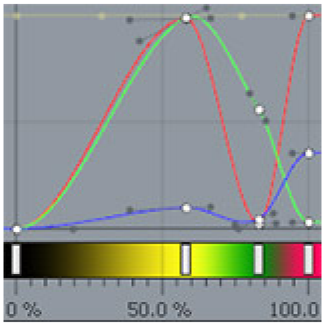
Displacement - Similar to **Bump Height**, this input parameter is driven by a displacement texture layer directly below the gradient in the **Shader Tree** and maps itself along the displacement height amount. 0% at the lowest displacement area, and 100% at the maximum displacement amount. Negative gradient values may also be necessary for remapped displacement values (that is, changing the low/high values in the texture item to -100%, 100% respectively).



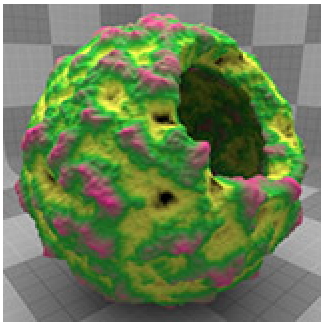
Displacement Map



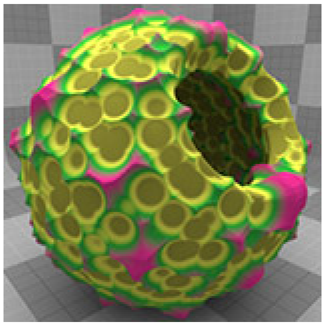
Displacement Map



Gradient



Disp. Height Gradient



Disp. Height Gradient

Input Parameter - Volumetric Effects

Fluid Fuel - When the **Input Parameter** is set to **Fluid Fuel**, the gradient keys off of a VDB Voxel's "fuel" feature.

Fluid Pressure - When the **Input Parameter** is set to **Fluid Fuel**, the gradient keys off of a VDB Voxel's "pressure" feature.

Fluid Temperature - When the **Input Parameter** is set to **Fluid Fuel**, the gradient keys off of a VDB Voxel's "temperature" feature.

Volume Sample Density - When the **Input Parameter** is set to **Fluid Fuel**, as related to volumes, the gradient keys off of any Volume Sample Density texture layer present.

Volumetric Absorption Amount - When the **Input Parameter** is set to **Volumetric Absorption Amount**, as related to volumes, the gradient keys off of any Volumetric Absorption Amount texture layer present.

Volumetric Density - When the **Input Parameter** is set to **Volumetric Density**, as related to volumes, the gradient keys off of any Volumetric Density texture layer present.

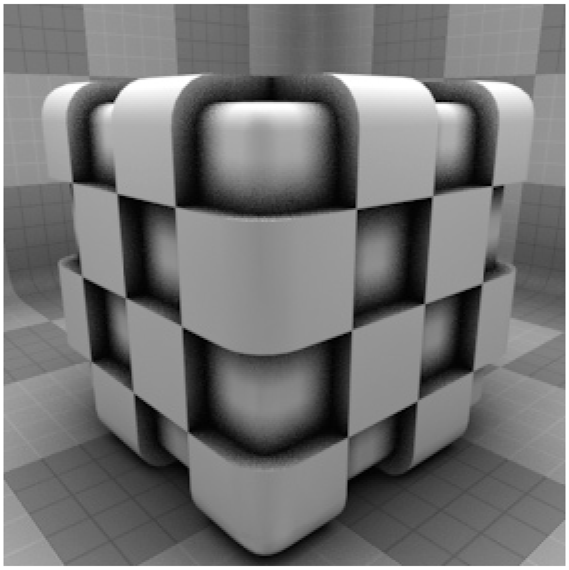
Volumetric Level - When the **Input Parameter** is set to **Volumetric Level**, as related to volumes, the gradient keys off of any Volumetric Level texture layer present.

Volumetric Luminosity Amount - When the **Input Parameter** is set to **Volumetric Level**, as related to volumes, the gradient keys off of any Volumetric Luminosity Amount texture layer present.

Volumetric Scattering Amount - When the **Input Parameter** is set to **Volumetric Scattering Amount**, as related to volumes, the gradient keys off of any Volumetric Scattering Amount texture layer present.

Note: The gradient maps across multitudes of values, such as those of a procedural texture layer, so if you have only a single continuous value, the gradient only registers the corresponding single gradient color/value.

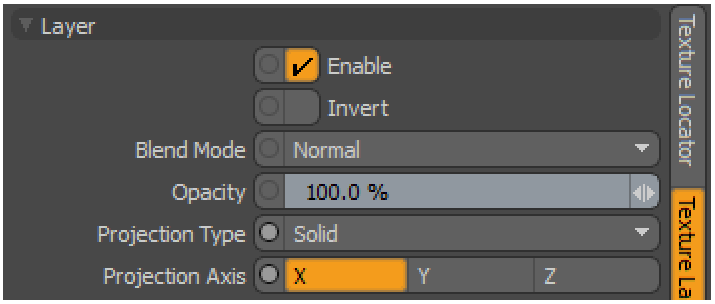
Occlusion



Occlusion is a technique for calculating the shading of surfaces based on how occluded they are. Depending on the settings, any receding or recessed areas are shaded darker (or lighter). Modo also has a render output called Ambient Occlusion, that produces similar results, but does not take surface attributes into account. Occlusion is a simple and fast way to simulate global illumination without the overhead of calculating any scene lighting, as it only takes into account surfaces and how they adjoin to each other. Occlusion makes its calculations by firing a number of somewhat random rays and calculating shading for a pixel based on how many rays intersect with neighboring surfaces. There are a number of settings with which you can control how the occlusion layer shades the surface, producing a wide variety of effects. It's also useful to change the layer effect where the contributions of other layers can be controlled by the occlusion layer.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Layer Properties

> 

☑ Enable

☐ Invert

Blend Mode: Normal

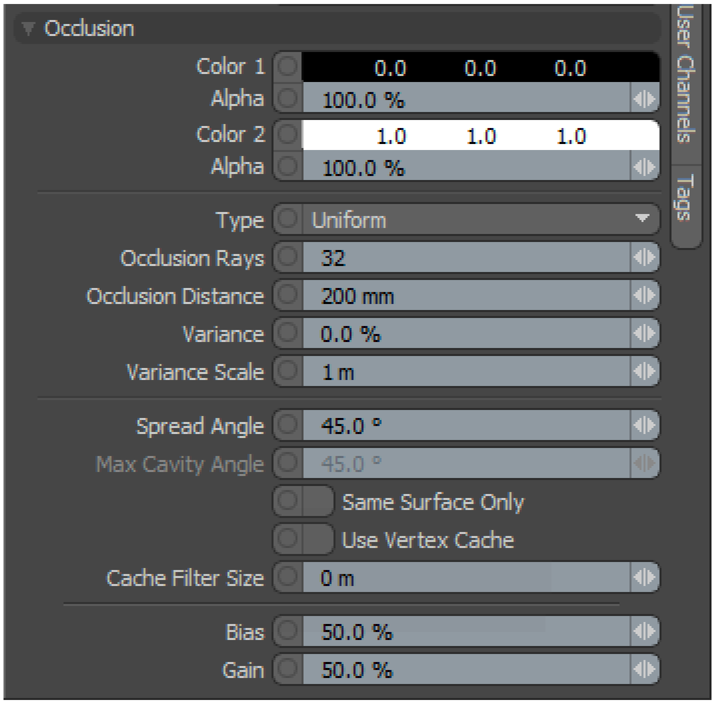
Opacity: 100.0 %

Projection Type: Solid

Projection Axis: X Y Z

Layer	
Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Projection Type	Defines how a material us applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Occlusion Properties



Occlusion

Color 1: 0.0 0.0 0.0

Alpha: 100.0 %

Color 2: 1.0 1.0 1.0

Alpha: 100.0 %

Type: Uniform

Occlusion Rays: 32

Occlusion Distance: 200 mm

Variance: 0.0 %

Variance Scale: 1 m

Spread Angle: 45.0 °

Max Cavity Angle: 45.0 °

☐ Same Surface Only

☐ Use Vertex Cache

Cache Filter Size: 0 m

Bias: 50.0 %

Gain: 50.0 %

Occlusion

Color/Value 1: Specifies the color of the shading in the occluded areas. This color ramps toward **Color 2**.

Alpha: Specifies the level of transparency for **Color 1**.

> **Note:** You can adjust the **Alpha** value using the **Color Picker**.

Color/Value 2: Specifies the color of the shading in the non-occluded areas. This color ramps toward **Color 1**.

Alpha: Specifies the level of transparency for **Color 2**.

Type: Determines the occlusion calculation method, controlling the way rays are fired from the surface at each pixel location. Each type produces different results.

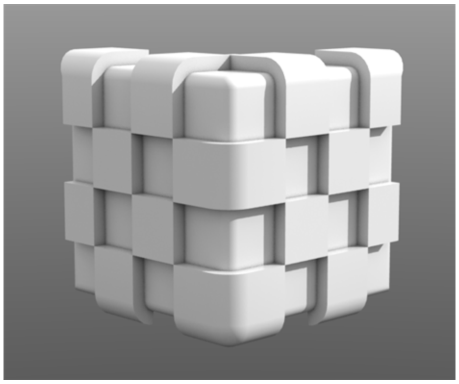
- **Uniform**: Fires rays in all directions in the hemisphere.
- **Up**: Looks for occlusions above the shading point.
- **Down Slope**: Looks for occlusions above the shading point and in the surface plane.
- **Reflection**: Looks for occlusions in the direction of the reflection. This is determined by the camera's position (when rendering) where a ray is fired and when it hits a surface, it bounces off at the angle of

incidence and if it hits another surface after bouncing, it's considered occluded. Useful for faking soft reflection among other uses.

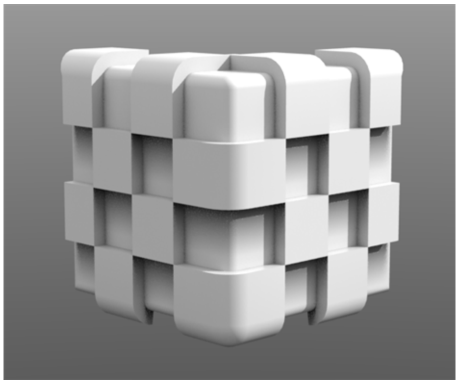
- **Concavity**: Occludes where the surface is concave, as determined by the **Max Cavity Angle** value.
- **Convexity**: Occludes where the surface is convex, as determined by the **Max Cavity Angle** value.
- **Concavity & Convexity**: Occludes in both the convex and concave areas of a surface as determined by the **Max Cavity Angle** value.
- **Thickness**: This is effectively an inverse Ambient Occlusion: Ambient Occlusion fires rays outwards from a point on the surface and records the average distance from there to the surface it hits. **Thickness**, on the other hand, fires rays inwards from a point on the surface and records the average distance from there to the back-facing surfaces it hits.

> **Note:** For the **Thickness** occlusion type, we recommend using a **Spread Angle** of 180 degrees and a **Cutoff Angle** of 0-10 degrees.

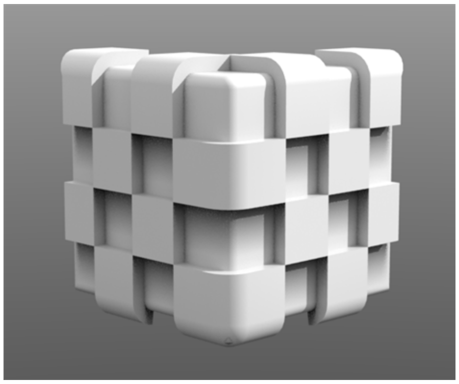
- **Curvature**: Produces occlusion where there's curvature in the geometry.



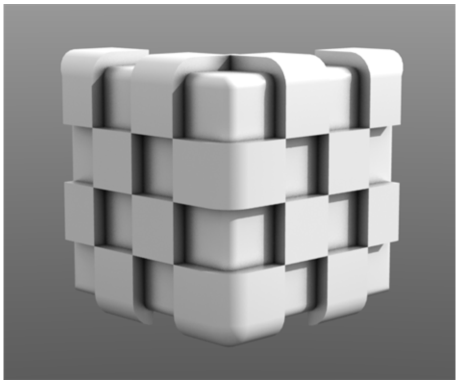
Uniform



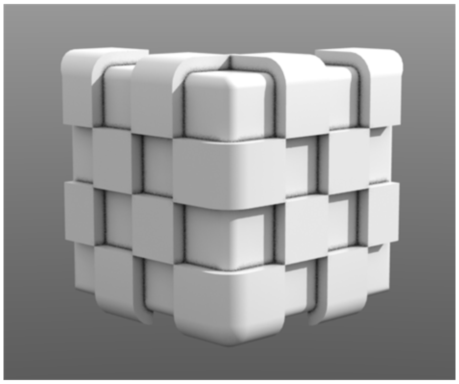
Up



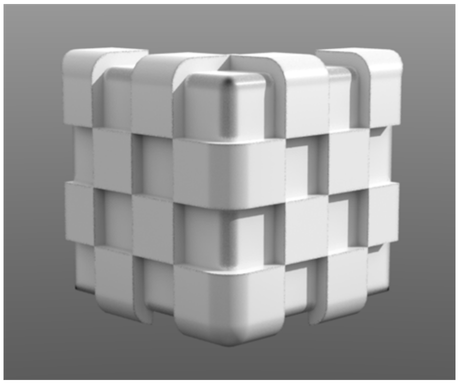
Down Slope



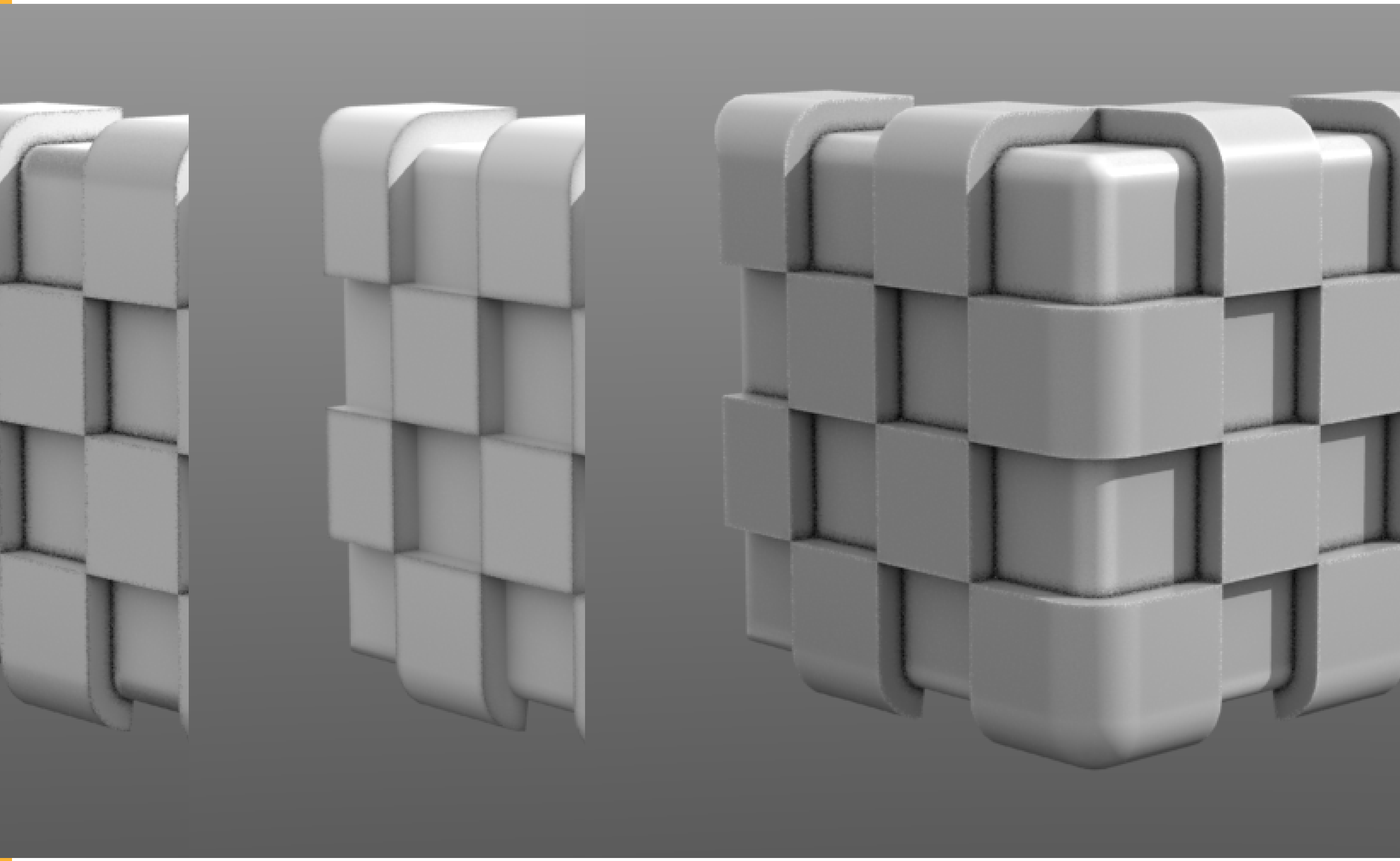
Reflection



Concavity



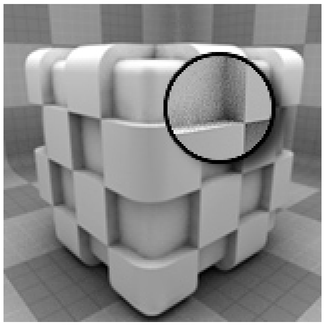
Convexity



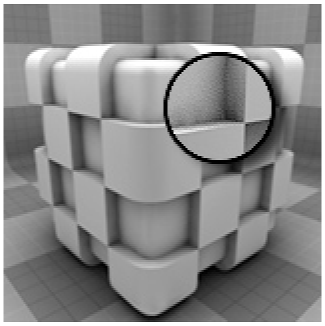
Concavity & Convexity Thickness Curvature

Occlusion Rays: This setting specifies the number of semi-random rays Modo shoots out from a surface when shading a pixel for Ambient Occlusion. Generally, the higher the number of rays, the more accurate (and smoother) the Occlusion calculations are, low numbers introduce noise as variations in neighboring pixels become more evident.

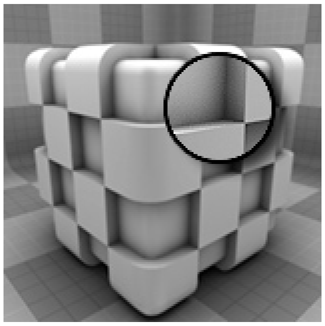
Note: Occlusion rays fired from the base of the source surface don't hit the back side of polygons by default. In order for occlusion rays to hit the back side of surfaces, their materials need to be double sided.



Occlusion Rays 32

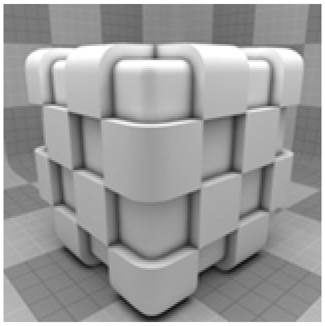


Occlusion Rays 128

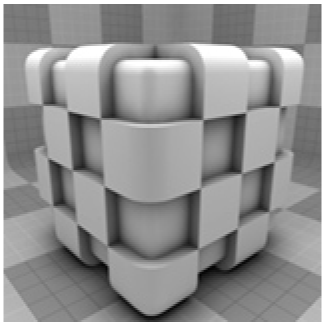


Occlusion Rays 512

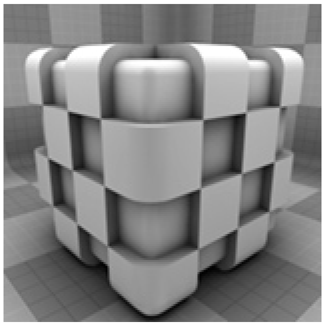
Occlusion Distance: Specifies the length of the rays that are fired from a surface. If you have a complex object with overlapping elements, limiting the ray range is a good way to control to what degree surfaces are shaded, it also helps to push the shading further into or away from recessed areas.



Distance 5 mm

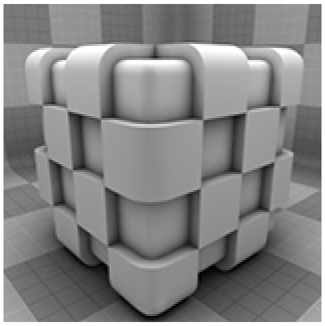


Distance 20 mm

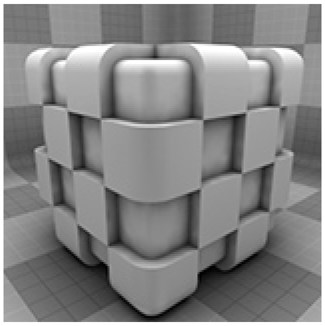


Distance 80 mm

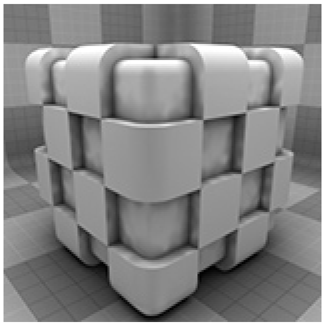
Variance: Adds a perlin noise function modulating the **Occlusion Distance**, meant to vary the strength of the occlusion effect. Value determines the amplitude of the function where greater values produce stronger, more obvious results (try values above 100%).



Variance 0%

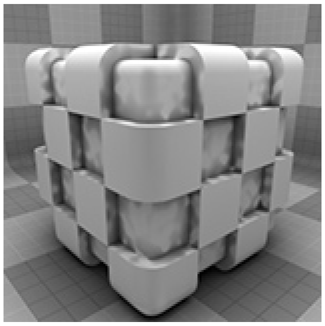


Variance 50%

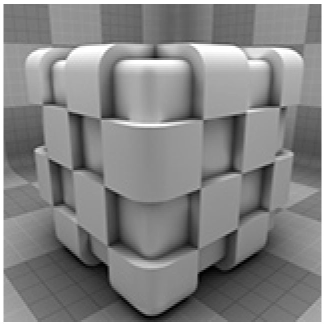


Variance 100%

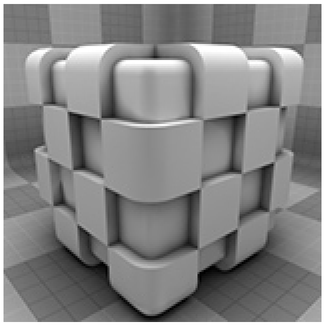
Variance Scale: This value determines the scale (size) of the noise function applied to the variance operation.



Scale 5 mm



Scale 15 mm



Scale 25 mm

Spread Angle: Determines the maximum amount of spread for each shading position in the direction determined by the **Type** setting. At each shading position, a number of rays (determined by the **Occlusion Rays** option) are fired in a semi-random pattern, starting in the normal direction from the shading surface and angling out the amount determined here (half each side away from the normal), so at 180° **Spread Angle**, a full hemisphere of rays is fired from each shading position. The wider the **Spread Angle**, the more likely a recessed area is considered occluded, also a higher number of rays is necessary to produce a smooth result.

Max Cavity Angle: Used with only the **Concavity**/**Convexity** options, this value determines the angle of difference between adjoining surfaces that determines if the surface is concave or convex.

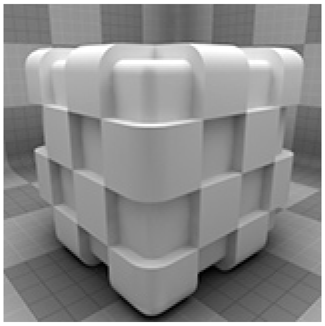
Cutoff Angle: This parameter is used to discard occlusion hits that are too close to the original orientation. So if the occlusion sample hits lie in the same plane as the original sample (plus or minus the cutoff angle) it ignores them. We don't recommend changing this number.

Same Surface Only: When this option is enabled, occlusion is only calculated for the surface that the shader is applied to, ignoring all other surrounding polygons. When disabled (the default), surrounding polygons are taken into account in shading calculations.

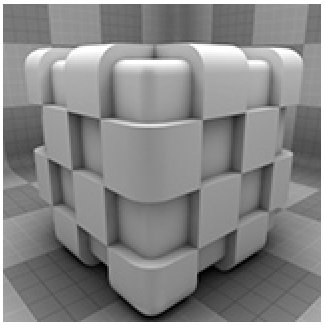
Use Vertex Cache: When this option is enabled, the occlusion sample values are stored in the geometry's vertex, producing a faster but somewhat softer result; as values are interpolation between the vertices. The **Cache Filter Size** option can be adjusted to control the softness of the result.

Cache Filter Size: This option works with the **Use Vertex Cache** option to control the number of vertex values that are interpolated to produce the final result. For each vertex, all the adjoining vertex values are interpolated that are within the user-defined range. Larger filter sizes produce smoother results.

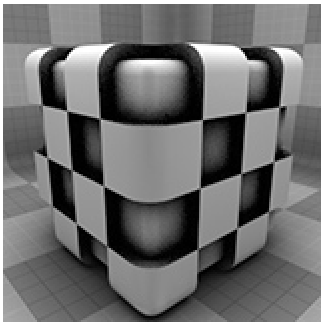
Bias: Increasing this value causes the texture to favor the primary color or value over the secondary, whereas decreasing the value causes the secondary color or value to be favored.



Bias 0%

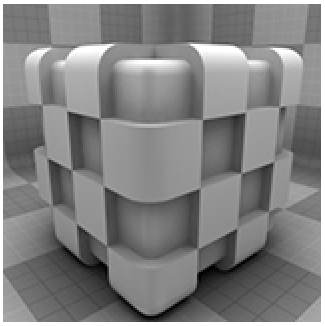


Bias 50%

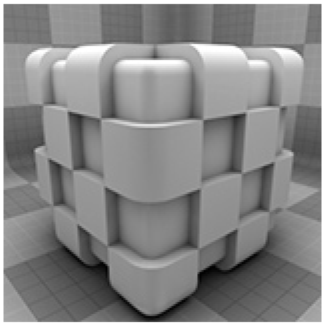


Bias 100%

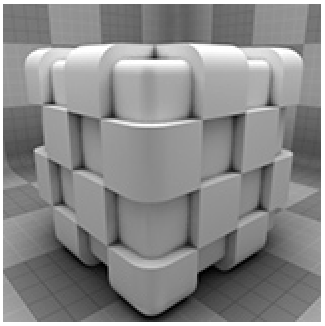
Gain: This setting is similar to a contrast control that affects the falloff of the gradient ramp between the **Color/Value 1** and **Color/Value 2**. Setting the **Gain** to 100% creates a very sharp contrast whereas setting the value to 0% reduces the contrast between the two to almost imperceptible amount.



Gain 0%



Gain 50%



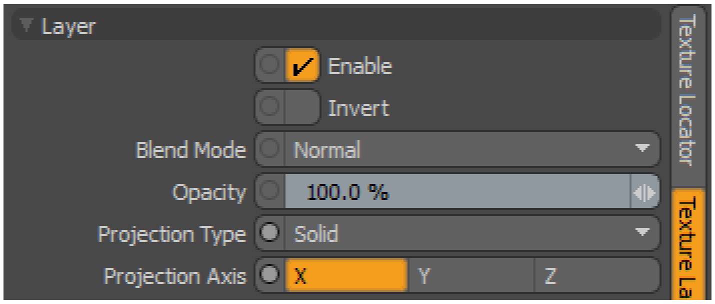
Gain 100%

Process

The Process layer offers some basic image adjustment options that affect any texture layer directly below it in the **Shader Tree**, including image map, procedural and gradient texture layers.

Layer Properties

>



☑ Enable

☐ Invert

Blend Mode: Normal

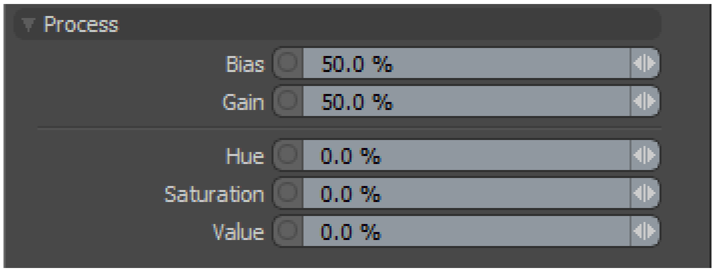
Opacity: 100.0 %

Projection Type: Solid

Projection Axis: X (selected), Y, Z

Layer	
Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Projection Type	Defines how a material us applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Process Properties



Process

Bias 50.0 %

Gain 50.0 %

Hue 0.0 %

Saturation 0.0 %

Value 0.0 %

Option	Description
Layer	
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.

Option	Description
Invert	Inverts the colors (RGB values) for the layer to produce a photonegative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the **Shader Tree**, if present, or dims the effect of the layer, itself, on the surface.
Process	
Bias	This amount, when adjusting a texture layer with two colors/values, works identically to the texture layer's **Bias** function where high values favor secondary colors or value settings, and lower values favor primary color or value settings. When adjusting full color RGB texture layers, the **Bias** adjusts the color spectrum based on its luminosity amount.
Gain	Produces similar results to adjusting gamma in a texture layer having its greatest affect on the contrast of the layer. Higher values increase contrast while lower values decrease contrast.
Hue	Manipulates the hue or color the texture layer.
Saturation	Adjusts the colors' apparent saturation. A setting of -100% turns a colored texture layer to grayscale.
Value	Adjusts the brightness of the texture layer.

Tension Texture

The Tension texture is a special case gradient that changes a surface's colorization or value based on the amount of deformation applied to it. This is done by either calculating the difference of angle between each polygon's deformed and undeformed states, or its difference in area. The color or value amount that is applied depends on the degree of change, correlating the computed value to a position along the defined gradient. You can add a number of interesting effects to deformed meshes, most obviously applying wrinkles to the interior areas of a deformed joint. You can add the texture to the surface in the **Shader Tree** under **Add Layer** > **Processing** > **Tension Texture** menu. Once added, you can modify its effect, directing how the tension shading is applied.

> **Note:** For information regarding adding and working with **Shader Tree** item layers, see the Shader Tree topic.

Usage

It's not possible to cover every conceivable use case for the Tension texture, however there are a couple of compelling and common uses that may not be directly obvious as to how to set them up. The first case controlling displacements using the Tension texture in the **Shader Tree** to mask off a displacement layer. This is accomplished by following these steps:

1. First add a new Material Group Item and place it inside the same material group as the surface it is meant to displace.
2. Within that new material group, place the Tension texture and the displacement texture. The displacement layer can be a procedural texture, or a sculpted displacement or a vector displacement map.
3. Next, change the **Effect** column of the Tension texture to **Group Mask**, meaning that any and all layers inside that specific material group are further limited by the masking of the Tension layer.
4. The next step is setting keys on the Tension texture layer. Since the **Group Mask** effect is looking for a value, and not a color, the input gradient that needs to be modified. Middle-click the **Value** input gradient to define a key, then drag the resulting key up or down with the left mouse button to set its value.

   Dragging all the way downward sets its value to zero. Now deforming the target surface with a deformer (it can't be a modeling deformation applied in one of the component modes) shows the displacement effect only in the created areas of the deformation. You may need to adjust the defined key position to produce the desired results.

> **Tip:** A useful troubleshooting tip is to change the Tension layer's **Effect** to something more easily visible in the **Preview** window to adjust its values more easily, such as **Diffuse Amount**, and then switch back to **Group Mask** for the finalized effect.

Another compelling use is to mask a multiresolution sculpting layer, but this requires a little more Modo knowledge as it is setup using the **Schematic** viewport. Follow the steps below:

1. In the **Sculpt** viewport (found in the **Paint** workspace layout), select the sculpting layer that is masked by the Tension texture and right-click to open the contextual menu and then select the **Add Modifier**

option. This adds a Channel Modifier item to the Mesh Item that can be used to drive its application using falloffs, or in this case a **Shader Tree** layer.

2. Next the Tension texture is added to the **Shader Tree** within the material group for the surface it is going to be affecting.
3. Both of these items, the resulting Channel Modifier (found under the Item in the **Items** list) named after the sculpting layer it originated from, and the Tension textures need to be added to the **Schematic** viewport.

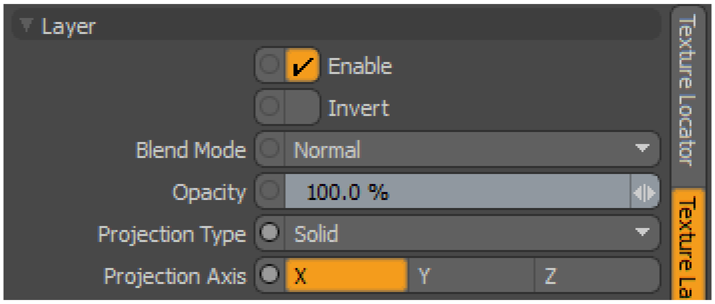
   You also need to add in the **Opacity** channel for the Channel Modifier if it isn't added automatically.
4. Now connect the **Texture Value** output on the Tension texture to the newly added **Opacity** channel input on the modifier.
5. Set the key on the Tension texture layer in the **Shader Tree**. The Channel Modifier input is looking for a **Value** input, so that is the input gradient that needs to be modified. You can middle-click in the middle of its Value input gradient to define a key then drag the resulting key up or down with the left mouse button to set its value.

   Dragging all the way downward sets its value to zero. Now deforming the target surface with a deformer (it can't be a modeling deformation applied in one of the component modes) should show the sculpting layer only in the creased areas of the deformation. You may need to adjust the texture's key positions to produce the desired results.

Tip: A useful troubleshooting tip is to change the Tension layer's **Effect** to something more easily visible in the **Preview** window to adjust its values more easily, such as **Diffuse Amount.**

Layer Properties

>



Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

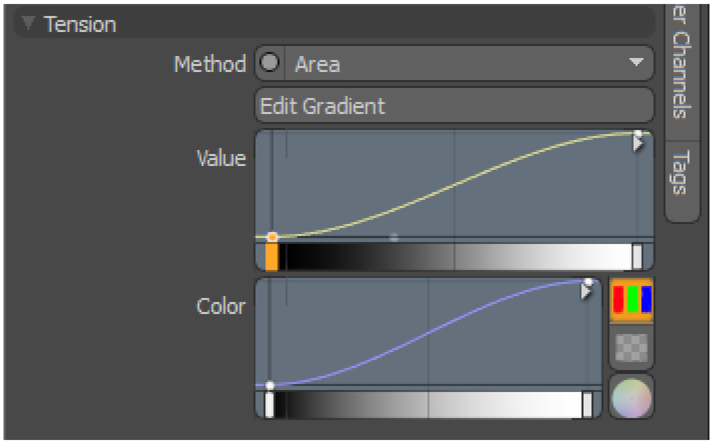
Projection Type: Solid

Projection Axis: X Y Z

Layer	
Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across

	Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Projection Type	Defines how a material us applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

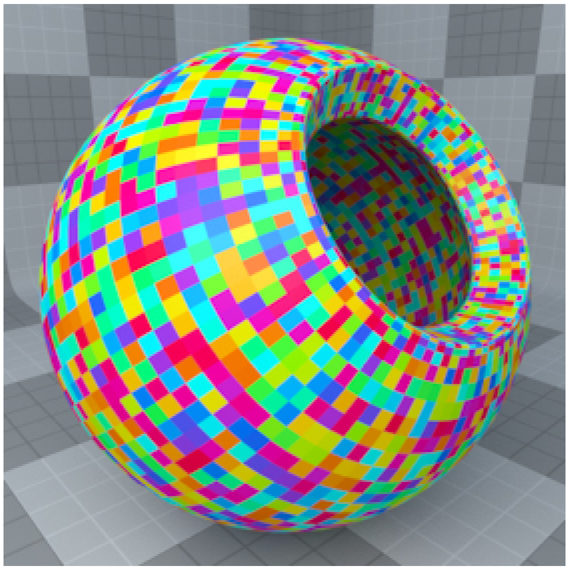
Tension Texture Properties


Method Area

Tension	
Method	Determines the way the polygon deformations are evaluated. The **Angle** option bases the shading by the amount of shearing, the rotational difference between the deformed and undeformed states. The **Area** option is based on the change in size between the deformed and undeformed states, which allows for both negative and positive value inputs with negative evaluations for getting smaller and positive for getting larger.

Value/Color	Inline mini gradient editors allowing you to define gradient value keys easily without having to leave the basic Modo interface. Much in the same way a keyframe stores transform values for items, gradient keys store colors or values and softly attenuate between the defined key positions. You can add keys by middle-clicking along the gradient and then clicking and dragging the key up or down, adjusting the value at that position. The gradient bar directly underneath provides a visual reference to the values generated by the gradient. In the color gradient, pressing the color wheel icon allows you to input precise color values using the standard Modo color picker. The evaluated amount of difference between the deformed and undeformed polygons is correlated to a position along the gradient and that correlated point is applied as shading to the surface.

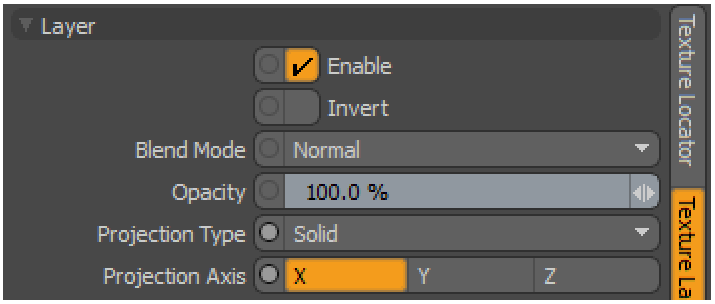
Variation Texture



The **Variation Texture** layer is a means to apply color or value variations to multiple particle objects, surfaces or items. Though similar to the Particle ID Gradient input parameter, the Variations texture is easier to apply and has the additional functionality to add variations to objects that are not particle-based, such as separated polygons and duplicated items. The above sample was created by freezing the subdivided sphere and unwelding all the polygons with the Split command (for more information, see Vertex Tools. From there, the Variations texture was applied using the **Mesh Part** option, and a rainbow gradient was added, providing the colorful randomness for the individual polygons. The Variations texture works by assigning a random numerical float value (a number between 0 and 1). The assigned number then draws the corresponding value or color from the user-defined gradient. For example, if the random number is 0.5, then the value present at the 50% position along the gradient is applied to the chosen variation source.

You can add the Variations texture item to the **Shader Tree** by clicking **Add Layer** > **Processing** > **Variations Texture**. For more information on adding and manipulating **Shader Tree** layers, see Shader Tree. When selected, the Variations texture item reveals its attributes in the **Properties** viewport, where you can adjust its settings and assign the gradient values for the variations.

Layer Properties

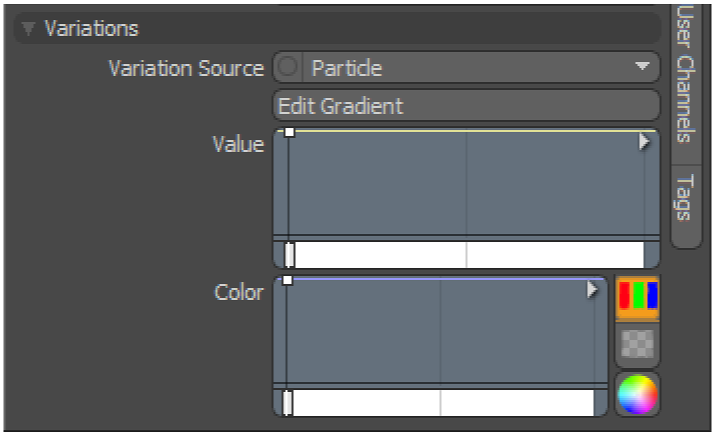


☑ Enable
☐ Invert
Blend Mode: Normal
Opacity: 100.0 %
Projection Type: Solid
Projection Axis: X Y Z

>

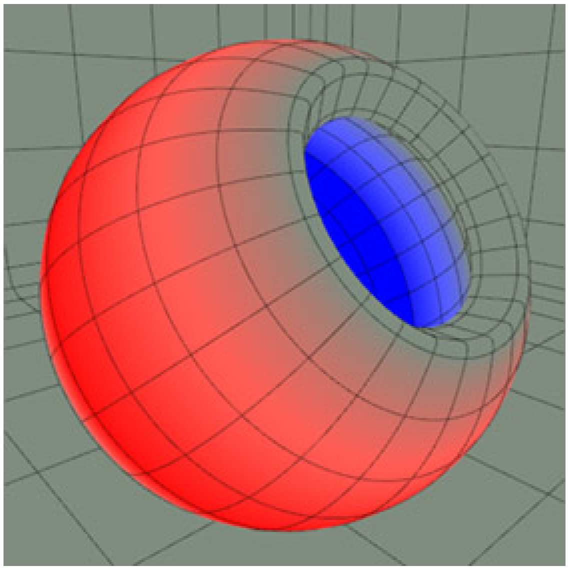
Layer	
Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Projection Type	Defines how a material us applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Variation Texture Properties



Variations	
Variation Source	Specifies what is affected by the gradient. The following options are available. • **Particle**: Applies the random gradient value to items cloned by a Replicator. • **Texture Particle**: Applies the random gradient value to textures cloned by the texture Replicator. • **Mesh Part**: Applies the random gradient value to individual contiguous polygon groups, useful to add variations to multiple objects in a single item layer, for example the leaves of a tree. • **Item**: Applies the random gradient to multiple separate item layers that share the same polygon tag. Especially useful to add variations to instanced elements without the need to create a separate item mask for each one.
Edit Gradient	Opens a full-sized Gradient Editor for creation or refinement of the target gradient at a larger size.
Value/Color	These are inline mini gradient editors allowing you to define gradient value keys easily, without having to leave the basic Modo interface. Much in the same way a keyframe stores transform values for items, gradient keys store colors or values and softly attenuate between the defined key positions. Keys can be added by middle-clicking along the gradient and then clicking and dragging the key up or down, adjusting the value at that position. The gradient bar directly underneath provides a visual reference to the values being generated by the gradient itself. In the color gradient, pressing the color wheel icon allows you to input precise color values, using the standard Modo color picker.

Vertex Map Texture



The Vertex Map Texture allows you to assign certain Vertex Maps for use in the **Shader Tree**. Specifically, vertex RGBA color maps and vertex weight maps. Vertex maps, when used in this way, can provide interesting options for shading a model as they are resolution-independent and can provide a variety of effects with much less system overhead than similar bit-mapped textures. Additionally, the information is embedded into the model and requires no additional external files, making them highly portable.

Usage

Vertex color maps need to be created for the target geometry first, which embeds the information into the vertices. Maps are first generated for a Mesh Item in the Vertex Map List. When selected, the resulting Vertex Map can be painted using the Modo's vertex painting brushes. Vertex color maps can also be generated automatically based on the scene's current surfacing and lighting with the Vertex Shader Baking command.

Once the map is created, you can add the Vertex Map Texture layer to the **Shader Tree** by clicking **Add Layer** > **Processing** > **Vertex Map Texture**.

Note: For information regarding adding and working with **Shader Tree** item layers, see Shader Tree.

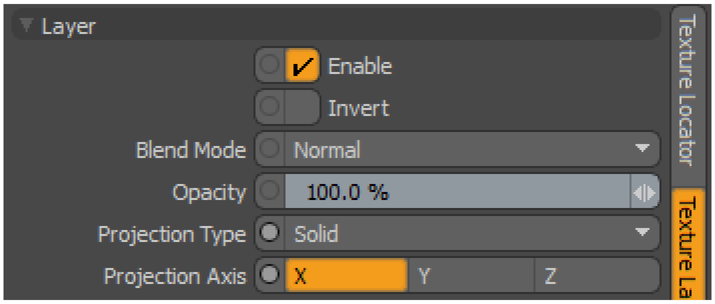
In the resulting Vertex Map Texture layer you can select the appropriate RGBA vertex color map from the **Vertex Map** list menu. You can obtain the best results by using a layer effect that requires color information. When using a RGBA vertex color map, the **Value**, **Color**, and **Alpha** options have no effect on the map.

For vertex weight maps, these need to be created as well. This can be done using the vertex painting brushes, but the brushes assign weight values instead of colors. You can also automatically create vertex weight maps with the Vertex Curvature Baking command, which may be used to produce some interesting effects.

Once the map is created, you can add the Vertex Map Texture layer to the **Shader Tree** using the **Add Layer** control. In the resulting Vertex Map Texture layer, you can select the appropriate vertex weight map from the **Vertex Map** list menu. Depending on the type of effect chosen, you can adjust the **Value**, **Color**, and **Alpha** options to produce the desired look. Additionally, the **Bias** and **Gain** options control the attenuation between the weight map values in the map. When using a layer effect that requires color values (such as **Diffuse Color**), you can assign colors to **Color 1** and **Color 2**, where **Color 1** maps to the 0 value areas of the weight map, and **Color 2** maps to the 1 value areas of the weight map. Where the effect type requires grayscale input (such as with a layer or group mask), you can adjust the **Value 1** and **Value 2** inputs, where **Value 1** maps to the 0 value areas of the weight map, and **Value 2** maps to the 1 value areas of the weight map. When the Vertex Map Texture item is selected, the attributes are visible in the **Properties** panel.

Layer Properties

>



☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

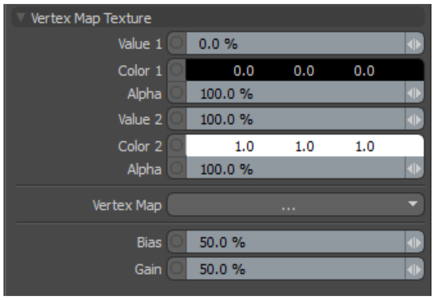
Projection Type: Solid

Projection Axis: X Y Z

Layer	
Enable	Toggles the effect of the layer on and off. This duplicates toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects.

	For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Projection Type	Defines how a material us applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Vertex Map Texture Properties



Vertex Map Texture
Value 1 0.0 %
Color 1 0.0 0.0 0.0
Alpha 100.0 %
Value 2 100.0 %
Color 2 1.0 1.0 1.0
Alpha 100.0 %
Vertex Map ...
Bias 50.0 %
Gain 50.0 %

Option	Description
Layer	
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the weight values for the layer producing a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the **Shader Tree**, if present, or dims the effect of the layer, itself, on the surface.

Option	Description
Vertex Map Texture	
Value 1	Determines the value of the texture for weight values of 0, ramping toward **Value 2** based on the weight map itself. This control is only active when the texture layer's **Effect** is set to a non-color attribute.
Color 1	Determines the color of the texture for weight values of 0, ramping toward **Color 2** based on the weight map itself. This setting only applies to surfaces that require a color value.
Alpha 1	The opacity of the **Color 1** zone. **Note:** You can adjust the **Alpha** value using the **Color Picker**.
Value 2	Determines the value of the texture for weight values of 1. This control is only active when the texture layer's **Effect** is set to a non-color attribute.
Color 2	Determines the color of the texture for weight values of 1. This setting only applies to surfaces that require a color value.
Alpha 2	The opacity of the **Color 2** zone.
Vertex Map	Specifies an existing weight map or vertex color map.
Bias	The bias control causes the texture to favor either the primary or secondary value or color. By increasing the bias, more of the **Color/Value 1** appears, while decreasing the number forces more of the **Color/Value 2** into the gradient blend.
Gain	This setting is similar to a contrast control that effects the falloff of the gradient ramp between the **Color/Value 1** and **Color/Value 2**. Setting the **Gain** to 100% creates a very sharp contrast, whereas setting the value to 0% reduces the contrast between the two to almost imperceptible amounts.

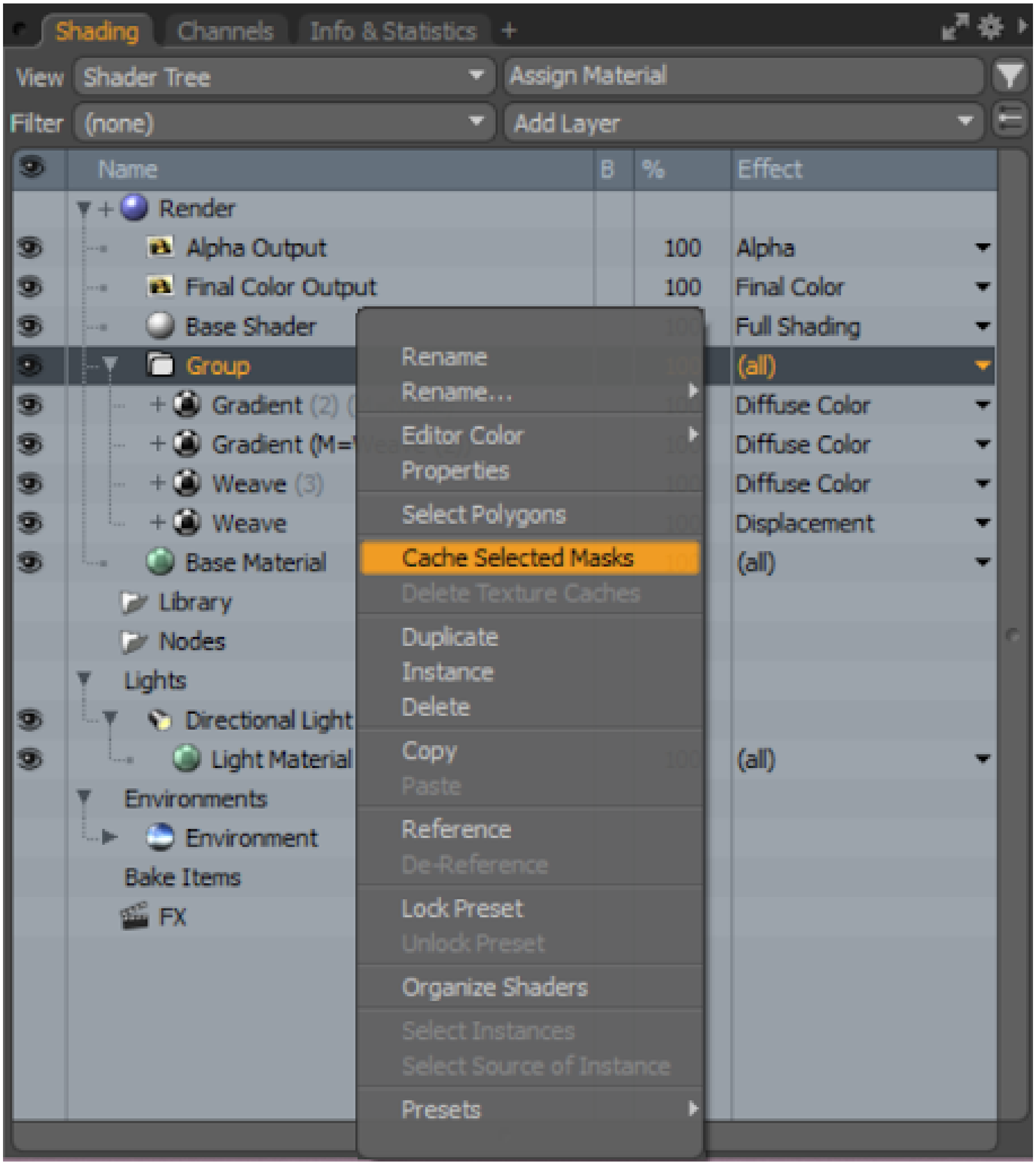
Textures

This topic is an overview of the procedurally generated textures provided with Modo. These procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. Click on the topics below to be directed to the appropriate topic page.

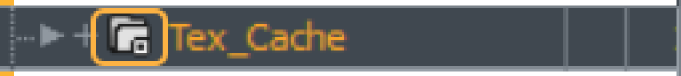
Optimize Your Textures Using the Texture Cache

Modo's texture cache lets you bake high-resolution texture maps to a lower resolution, allowing you to easily view your textures outside the render preview window and in the viewport. Like the **Bake To Texture** command, the texture cache enables you to bake multiple textures and masks within a group. However, unlike **Bake to Texture**, the baking process will occur in the background, and shaders that contribute to the baked image remain editable.

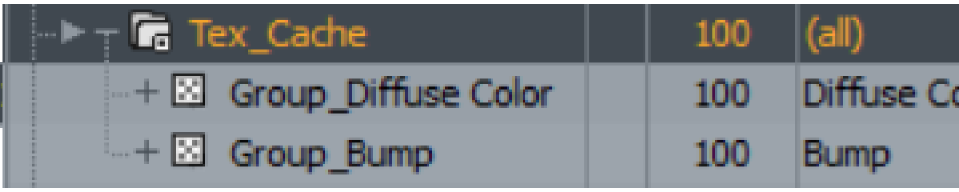
To create cached images from a material group, right-click the group you want to be cached and select **Cache Selected Masks**.


Shading
Channels
Info & Statistics
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
B
%
Effect
Render
Alpha Output
100
Alpha
Final Color Output
100
Final Color
Base Shader
Full Shading
Group
(all)
Gradient (2)
Diffuse Color
Gradient (M=
Diffuse Color
Weave (3)
Diffuse Color
Weave
Displacement
Base Material
(all)
Library
Nodes
Lights
Directional Light
Light Material
(all)
Environments
Environment
Bake Items
FX
Rename
Rename...
Editor Color
Properties
Select Polygons
Cache Selected Masks
Delete Texture Caches
Duplicate
Instance
Delete
Copy
Paste
Reference
De-Reference
Lock Preset
Unlock Preset
Organize Shaders
Select Instances
Select Source of Instance
Presets

Once the textures have been baked down, the group item's icon will change, and all texture layers will be visible as attributes under the + sign.

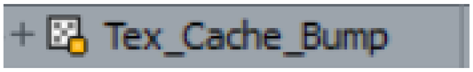


Updated texture group icon

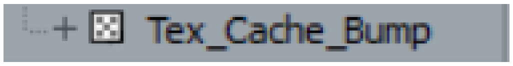

Tex_Cache
100
(all)
Group_Diffuse Color
100
Diffuse C
Group_Bump
100
Bump

Expanded texture group to show Cache Attributes

When baking, cache items will bake one at a time and display a preview resolution. While baking, the item's icon will update to include a yellow icon to show it is baking. Once baking is finished, the yellow icon will disappear.


Tex_Cache_Bump

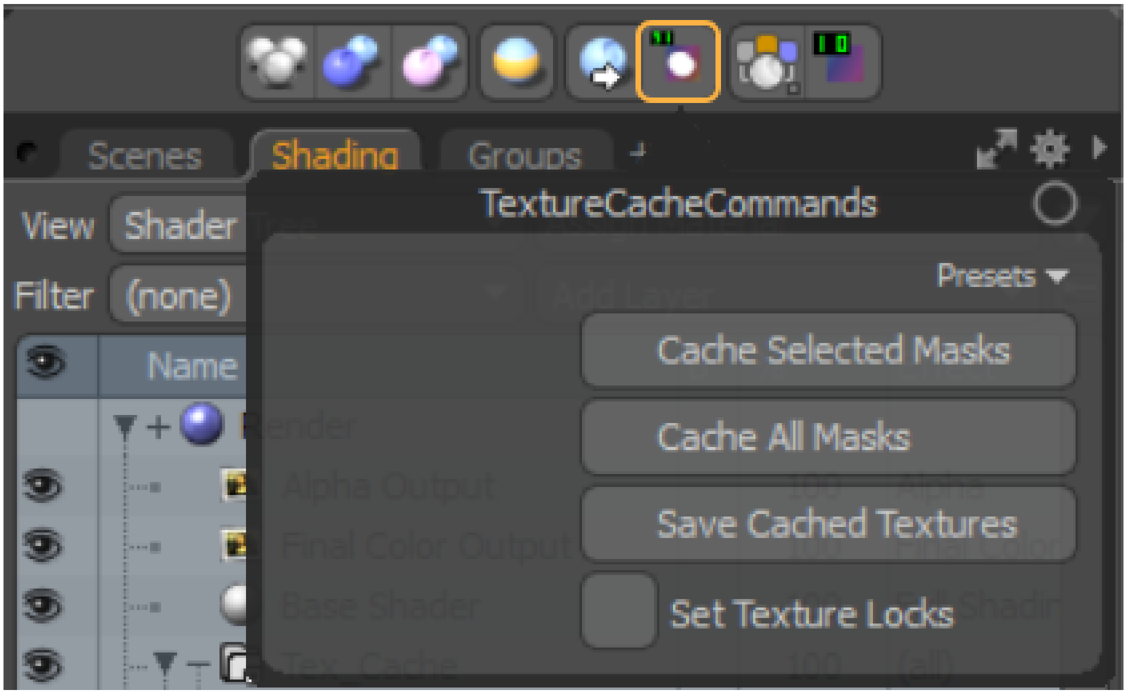
Icon indicating that baking is occurring


Tex_Cache_Bump

Icon updated after baking is complete

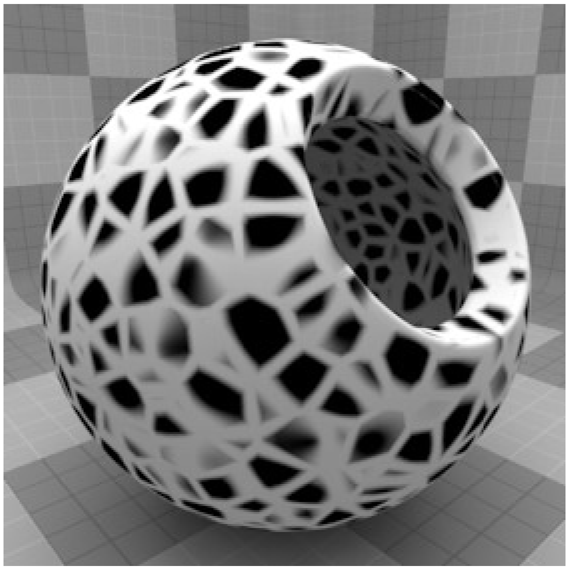
Any changes to the shading stack that the cache was baked from will force the texture cache's images to be re-baked.

Commands related to texture caching can also be accessed by clicking the **Texture Cache Commands** button above the shader tree.


Scenes
Shading
Groups
View
Shader
Filter
(none)
Name
TextureCacheCommands
Presets
Cache Selected Masks
Cache All Masks
Save Cached Textures
Set Texture Locks

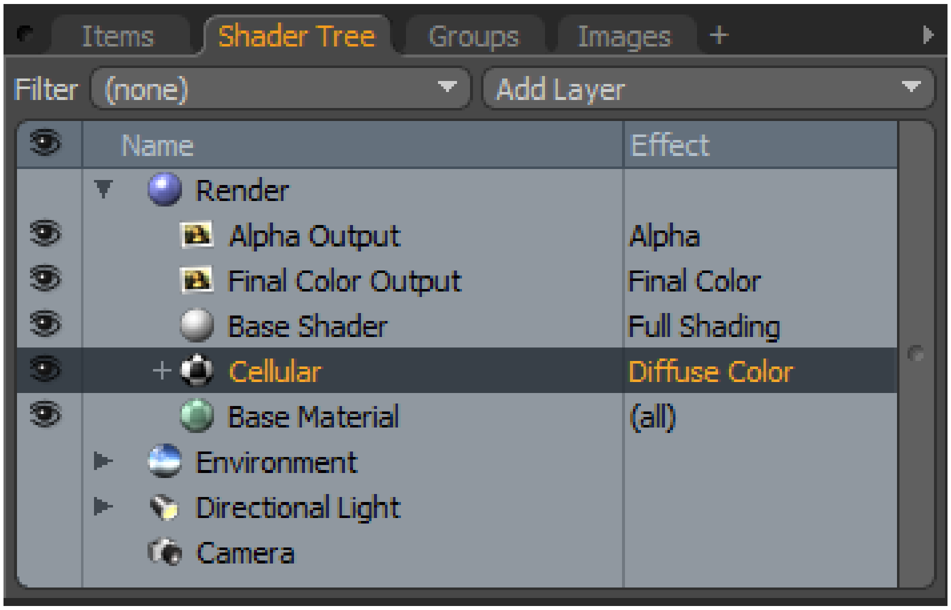
- **Cache Selected Masks** is the same command that is run when you right-click a group
- **Cache All Masks** runs the cache selected masks command across all materials in a scene
- **Save Cache Images** converts all Texture Cache items in a scene into regular images. This is useful for exporting when your shading has been finalized, or when exporting to another format.
- **Set Texture Lock** 'locks' Texture Cache items in a scene, so that any edits to the shader stack don't force the Texture Cache Images to update. When the locks are unset, the changes to the shader stack force the texture cache's images to be re-baked.

Cellular



The **Cellular** texture is one of the many procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Cellular** texture can be addressed by its two zones: the **Cell** and **Filler**. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Cell** and **Filler**. The **Cellular** texture is very versatile and can create effects ranging from veins and cobblestone to crumply water and lizard skins.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.



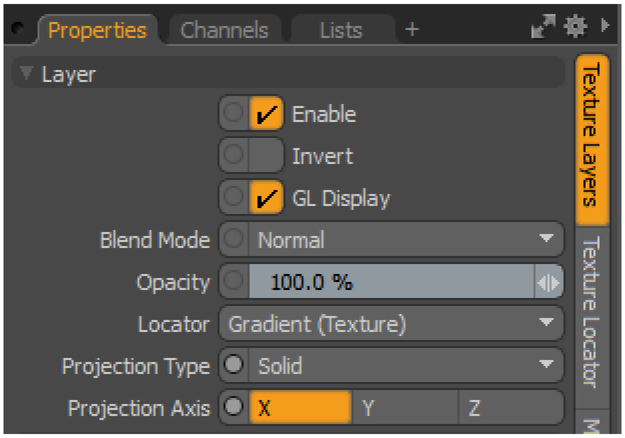
Filter (none)

Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Cellular	Diffuse Color
Base Material	(all)
Environment	
Directional Light	
Camera	

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

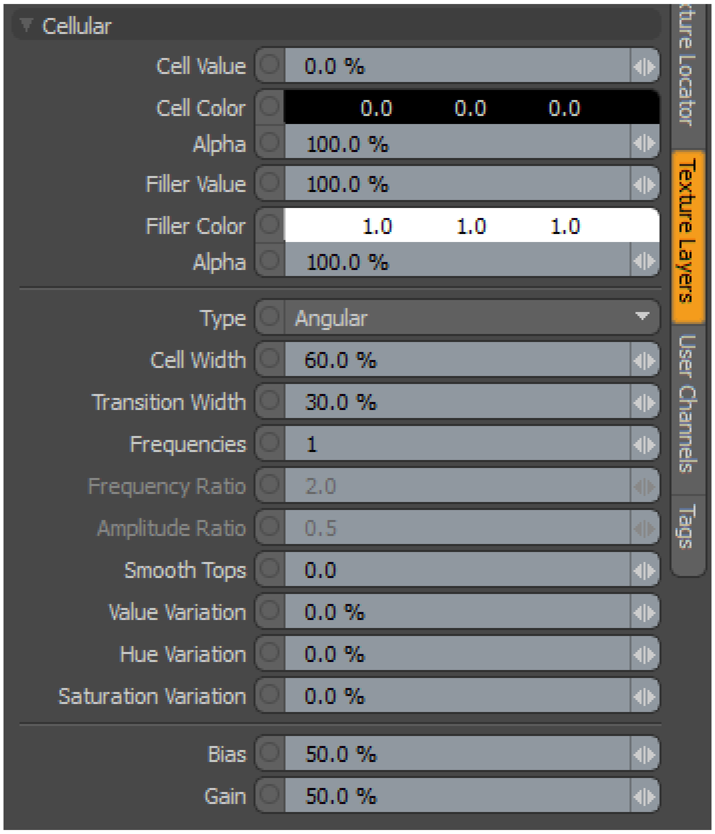
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Cellular Properties

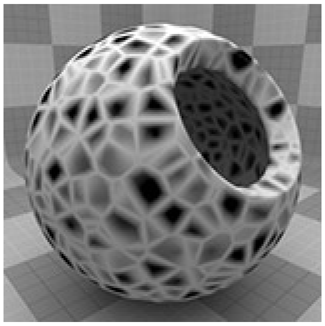


Cell Value 0.0 %
Cell Color 0.0 0.0 0.0
Alpha 100.0 %
Filler Value 100.0 %
Filler Color 1.0 1.0 1.0
Alpha 100.0 %
Type Angular
Cell Width 60.0 %
Transition Width 30.0 %
Frequencies 1
Frequency Ratio 2.0
Amplitude Ratio 0.5
Smooth Tops 0.0
Value Variation 0.0 %
Hue Variation 0.0 %
Saturation Variation 0.0 %
Bias 50.0 %
Gain 50.0 %

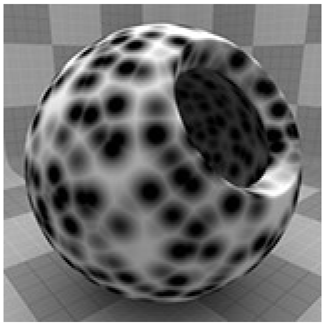
Cellular	
Cell Value	Determines the magnitude of the texture where the cell is most concentrated. This value ramps to the **Filler Value**. (This is only available when the texture layers effect is set to a non-color attribute.) **Note:** When using **Bump** or **Displacement** effects, the **Cell Value** can be used to adjust the height of the cell.
Cell Color	Determines the color of the texture where the cell is most concentrated. This color ramps to the **Filler Color**. You only set the **Cell Color** when the texture layer is set to an effect that requires color. The **Alpha** value determine how transparent the cell zone is.
Alpha	Determines the transparency of the cells. This is used with the **Cell Color** setting. **Note:** You can adjust the **Alpha** value using the **Color Picker**.
Filler Value	Determines the magnitude of the texture between the cells. This value ramps to the **Filler Value**. (This is only available when the texture layers effect is set to a non-color attribute.) **Note:** When using **Bump** or **Displacement** effects, the **Filler Value** can be used to adjust the height of the filler.
Filler Color	Determines the color of the texture between the cells. You only set the **Fill Color** when the texture layer is set to an effect that requires color.
Alpha	Determines the transparency of the texture between the cells. This is used with the **Filler Color** setting.
Type	Specifies the styles of cellular texture and determines how the texture interpolates between its sample points. Click to view examples of the two Type options. **Angular**- Creates a somewhat linear interpolation between the sample points in the texture. With the default settings, this results in an effect similar to cobblestone. You can use this to generate effects ranging from lunar surface to craggy skin. **Round**- Creates circular spots randomly distributed based on the texture. With the default settings, the texture looks like overlapping, soft-edged dots. By adjusting the

	settings, you can create effects such as cauliflower or chipped rock.
Cell Width	Determines the maximum scale of the texture's details. If you set this to **100%** , the maximum scale of any single detail in the texture is equal to the texture size as set by the texture locator size values. Use this to change the scale of the details of the texture. To set the overall scale of the texture, you modify the texture scale on the **Texture Locator** tab. Click to view examples of various Cell Width settings.
Transition Width	Controls the amount of gradient falloff around each of the texture details. Setting this value greater than **100%** can yield interesting effects. Click to view examples of various Transition Width settings.
Frequencies	Determines the number of virtual layers used by the texture. With this value at **1**, Modo disables the **Frequency Ratio** and the **Amplitude Ratio**. Adding more than one frequency increases the detail in the procedural texture, which increases computation time. Click to view examples of various Frequencies settings. You can layer all noise-style procedures to create a fractal effect. Traditionally, to create a fractal you layer a noise pattern multiple times. With each new layer, you increase the frequency (to have smaller details and more of them) and fade the opacity value. When you add a **Cellular** or **Noise** texture in Modo, the default settings are not "fractal" in that they use a default **Frequencies** value of **1** for having only one layer of the procedural effect. Adding multiple frequencies creates a fractal effect and invokes the **Frequency Ratio** and **Amplitude Ratio** options.
Frequency Ratio	Determines how much detail to add into each additional application of the noise texture when the **Frequencies** value is set to a value greater than **1**. With a **Frequency Ratio** value of **2**, the second frequency of the texture is twice as high as the first. Visually, this pattern seems to be half as large but with twice as many occurrences. With a **Frequencies** setting greater than **2**, each additional layer of the noise has twice the frequency of the previous layer.
Amplitude Ratio	Determines the strength of each additional layer of noise when the **Frequencies** value is set to a value greater than **1**. By default, the setting is **.5**, which yields an effect where each additional set of frequencies has half the value of the previous one.
Smooth Tops	Varies the amplitude so as to smooth the outer surface. If you applied the **Cellular** texture as a displacement, this produces a surface that looks like ridges with smooth tops and ragged valleys. If you invert the texture, the valleys become flat-topped mountains, and the ridges become valleys. Click to view examples of various Smooth Tops settings.

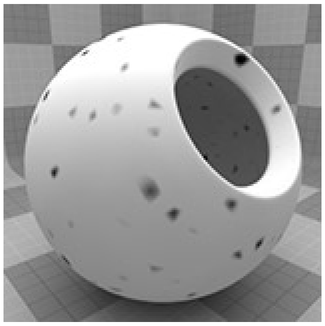
	Note: At the default settings, **Cellular** produces flattened tops when displaced or bumped due to the clear division of color in the pattern. Consequently, **Smooth Tops** has little effect on the surface appearance. However, if you increase the **Frequencies** setting, the surface will roughen and the effect of **Smooth Tops** will increase.
Value Variation	Varies the brightness or luminosity of each individual cell randomly. Higher numbers increase the differences between cells; lower values have decreased brightness variations. Click to view examples of various Value Variation settings.
Hue Variation	Varies the hue (or color) of each individual cell randomly. Higher numbers increase the color differences between cells; lower values have decreased color variations. Click to view examples of various Hue Variation settings.
Saturation Variation	Varies the color saturation of each individual cell. randomly Higher numbers increase the differences between cells; lower values have decreased saturation variations. Click to view examples of various Saturation Variation settings.
Bias	Determines whether Modo favors the **Cell Color** (or **Cell Value**) over the **Filler Color** (or **Filler Value**) or vice versa. Increasing this value favors the **Cell Color** (or **Cell Value**) over the **Filler Color** (or **Filler Value**); decreasing the value favors the **Filler Color** (or **Filler Value**) over the **Cell Color** (or **Cell Value**). Click to view examples of various Bias settings.
Gain	Affects the falloff of the gradient ramp between the **Cell Value** and the **Filler Value**. This is similar to a gamma control. Setting this to **100%** creates a very sharp falloff effect; setting the value to **0** creates a plateau around the value (or the color's mid-point) with a sharp falloff on either extreme of the gradient. Click to view examples of various Gain settings.



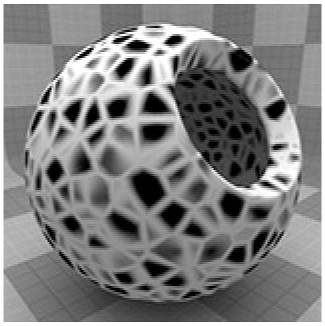
Type Angular



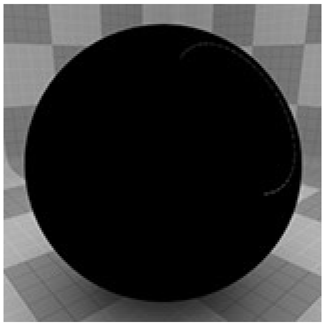
Type Round



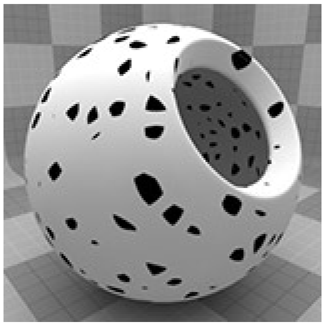
Cell Width 0%



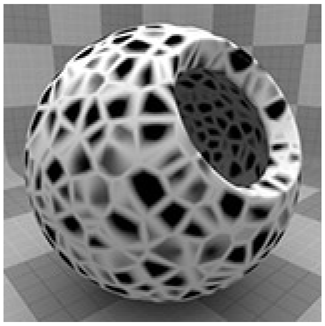
Cell Width 50%



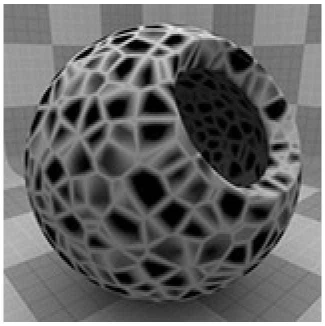
Cell Width 100%



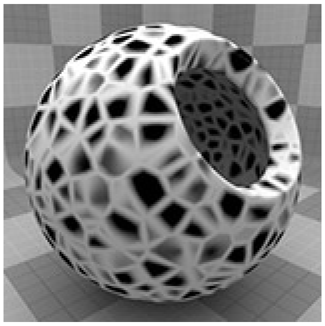
Transition Width 0%



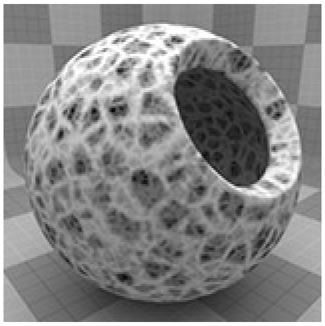
Transition Width 50%



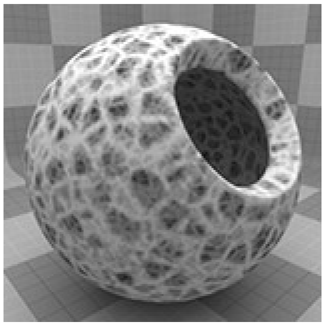
Transition Width 100%



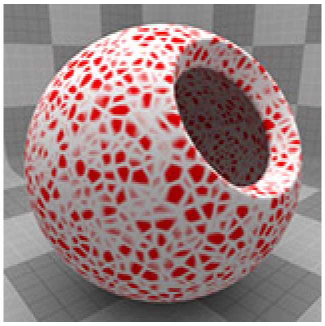
Frequency 1



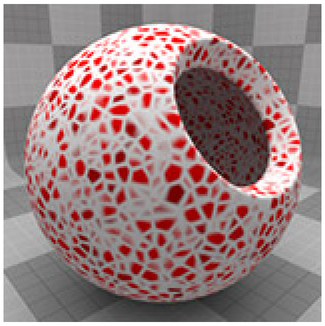
Frequency 3



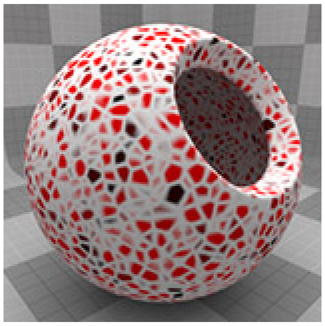
Frequency 5



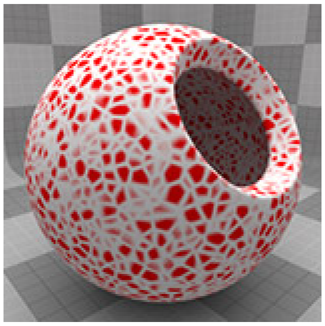
Value Variation 0%



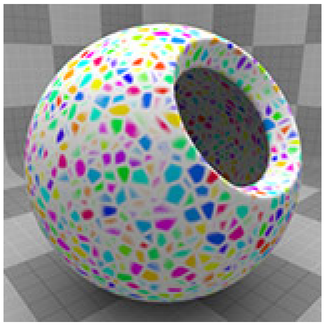
Value Variation 50%



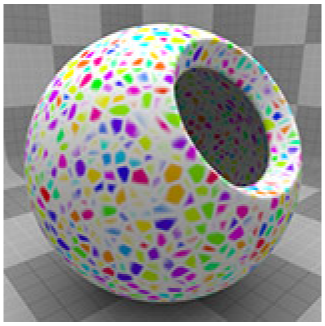
Value Variation 100%



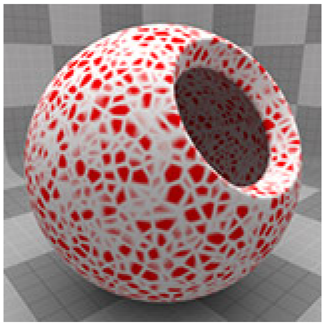
Hue Variation 0%



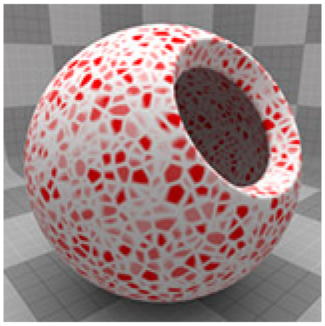
Hue Variation 50%



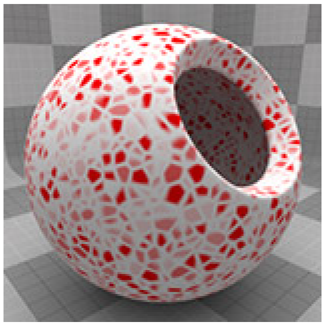
Hue Variation 100%



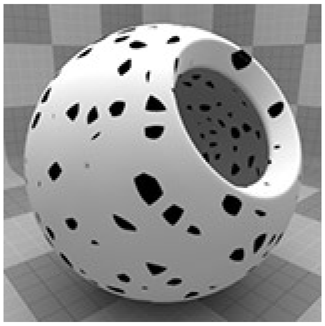
Saturation Variation 0%



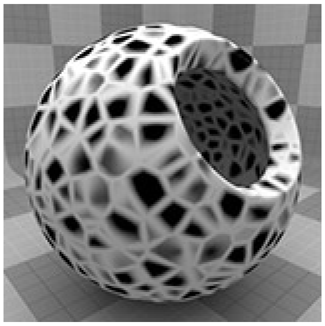
Saturation Variation 50%



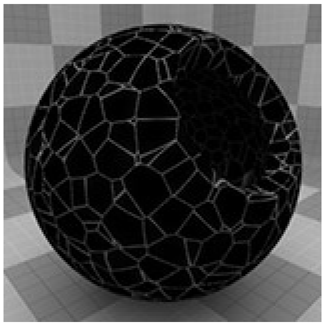
Saturation Variation 100%



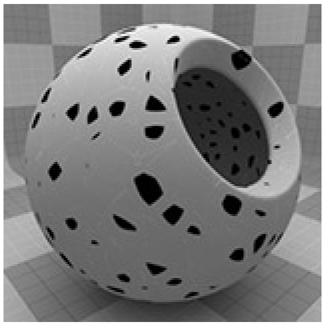
Bias 0%



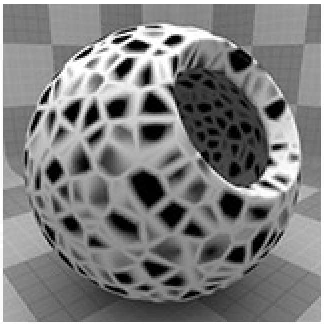
Bias 50%



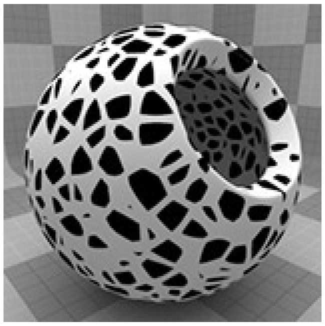
Bias 100%



Gain 0%

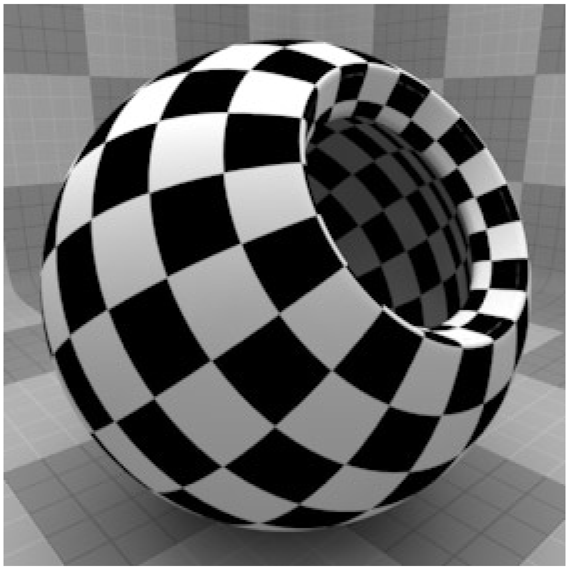


Gain 50%



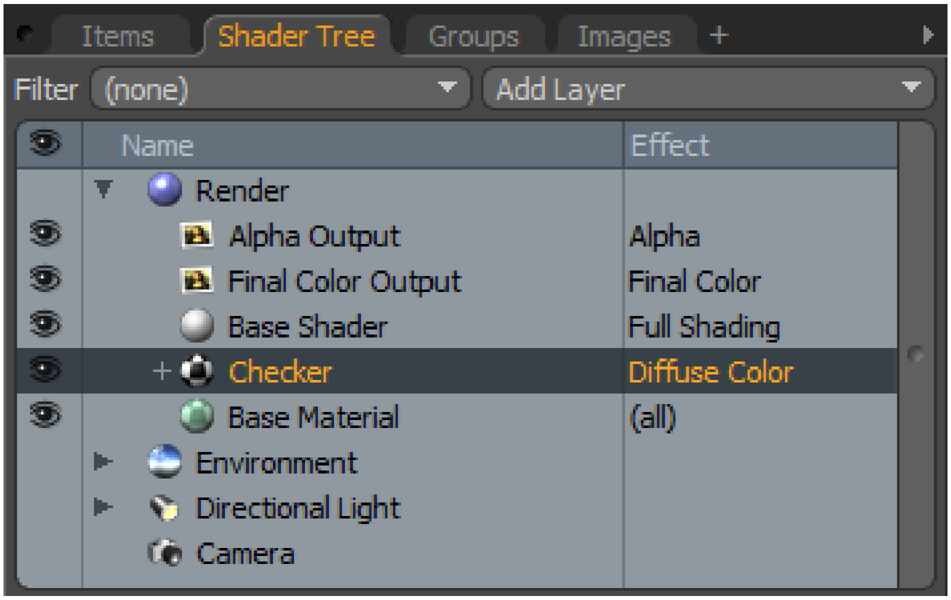
Gain 100%

Checker



The **Checker** texture uses two colors (or values) to create a pattern of alternating squares to resemble a checkerboard. By changing the **Transition Width**, **Bias**, and **Gain** options, you can create a variety of checkered and diamond-like shapes. Additionally, as a procedurally created texture that is mathematically generated at render time, the **Checker** texture has no fixed resolution and can be magnified nearly infinitely with no visual loss in detail.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

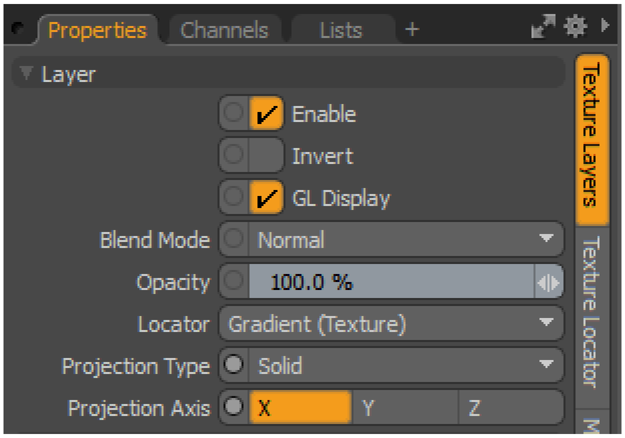

Items
Shader Tree
Groups
Images
+
Add Layer

Filter (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Checker	Diffuse Color
Base Material	(all)
Environment	
Directional Light	
Camera	

Layer Properties

>

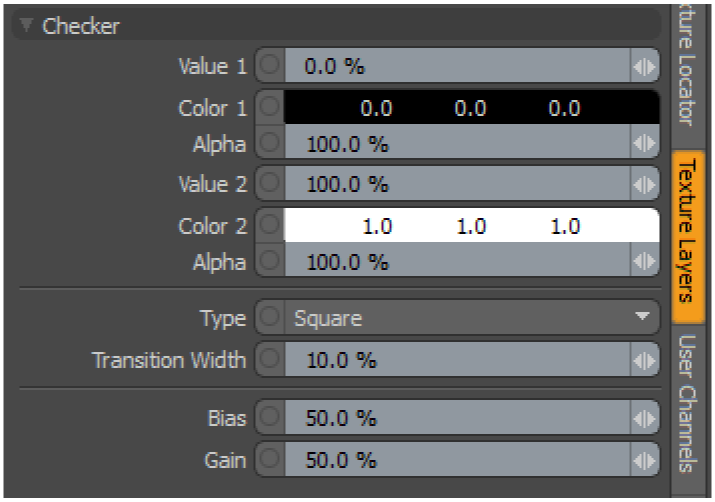

Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator
M

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

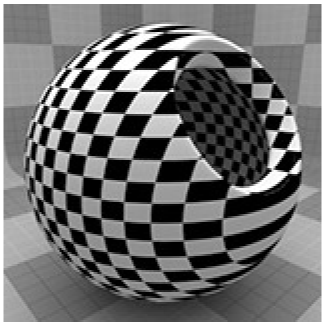
Checker Properties



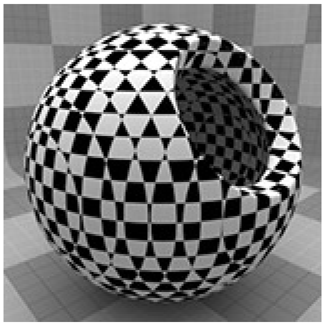
Value 1 0.0 %
Color 1 0.0 0.0 0.0
Alpha 100.0 %
Value 2 100.0 %
Color 2 1.0 1.0 1.0
Alpha 100.0 %
Type Square
Transition Width 10.0 %
Bias 50.0 %
Gain 50.0 %

Checker	
Value 1	The magnitude of the texture on the first set of alternating squares. The color ramps to **Color 2** based on the **Transition Width**. When using **Bump** or **Displacement** effects, this value describes the height of the first set of squares.
Color 1	The color of the texture on the first set of alternating squares. This color ramps to

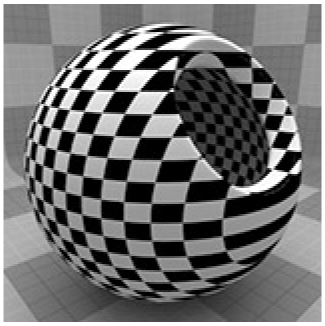
	Color 2 based on the **Transition Width**. The **Alpha** value determines the level of transparency.
Alpha	The level of transparency for **Color 1**. **Note:** You can adjust the **Alpha** value using the **Color Picker**.
Color 2	The color of the texture on the second set of alternating squares.
Value 2	The magnitude of the texture on the first set of alternating squares. The color ramps to **Color 1** based on the **Transition Width**. The **Alpha** value determines the level of transparency. When using **Bump** or **Displacement** effects, this value describes the height of the second set of squares.
Alpha	Determines the level of transparency for **Color 2**.
Type	Specifies the style for the **Checker** texture and determines the method of projection. See Type for examples. • **Square** - Projects the checkered grid from a single axis (which you define in the Texture Locator). • **Cubic** - Projects the checker grid from all three axes (this may look odd on curved surfaces because the edges interact with each other).
Transition Width	Controls the amount of gradient falloff across each of the texture edges. Driving this value over 100% can yield interesting effects. See Transition Width for examples.
Bias	Determines whether the texture favors **Color 1** or **Color 2**. Increasing this value causes the texture to favor **Color 1**; decreasing the value favors **Color 2**. This value is dependent on the **Transition Width** as it works across the gray tones of the image. See Bias for examples.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the two colors. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme. This value is dependent on the **Transition Width** as it works across the gray tones of the image. See Gain for examples.



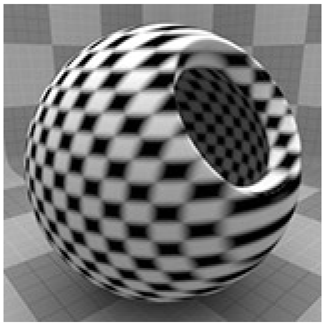
Type Square



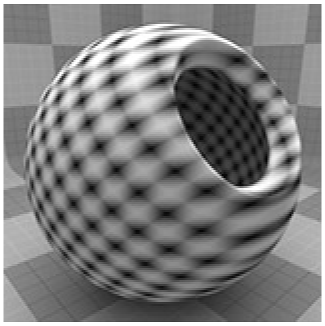
Type Cube



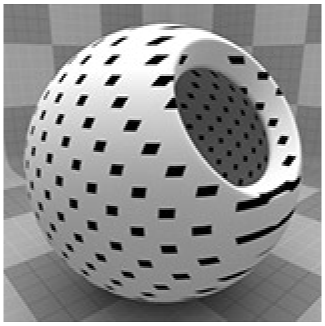
Transition Width 0%



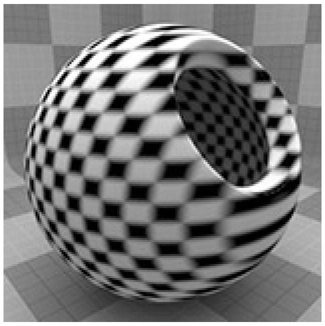
Transition Width 50%



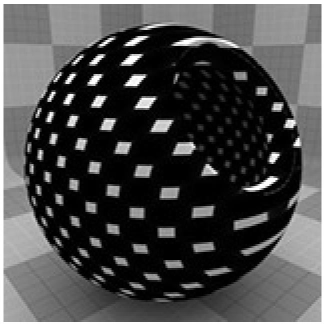
Transition Width 100%



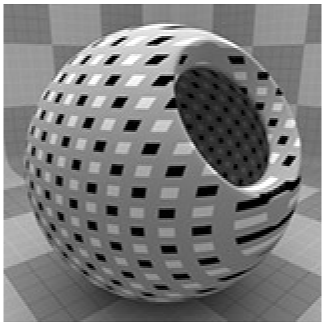
Bias 0%



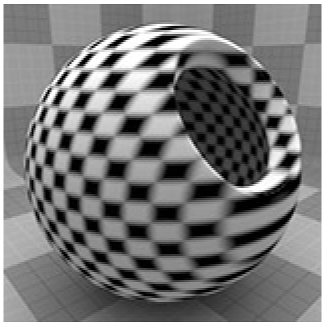
Bias 50%



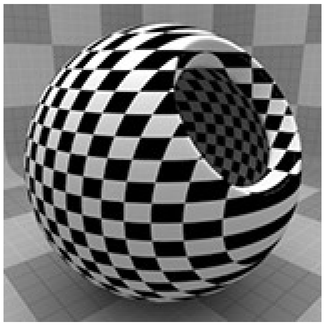
Bias 100%



Gain 0%

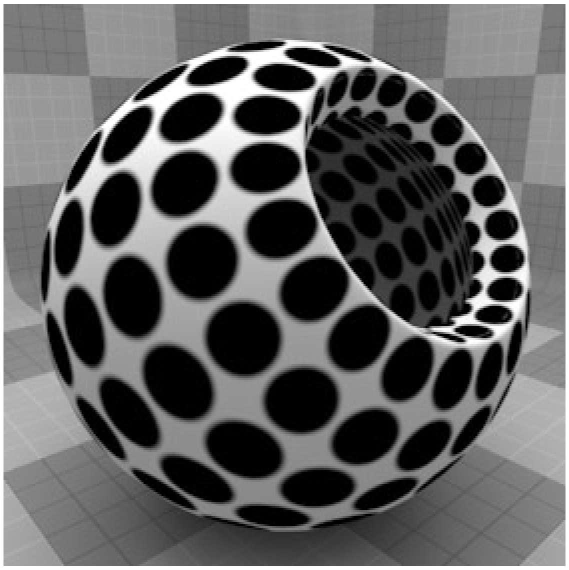


Gain 50%



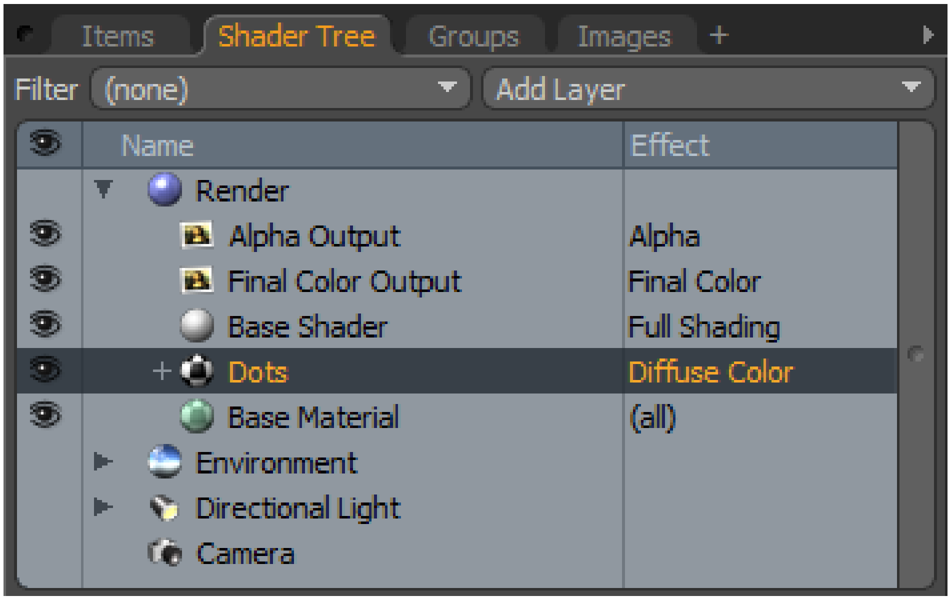
Gain 100%

Dots Procedural



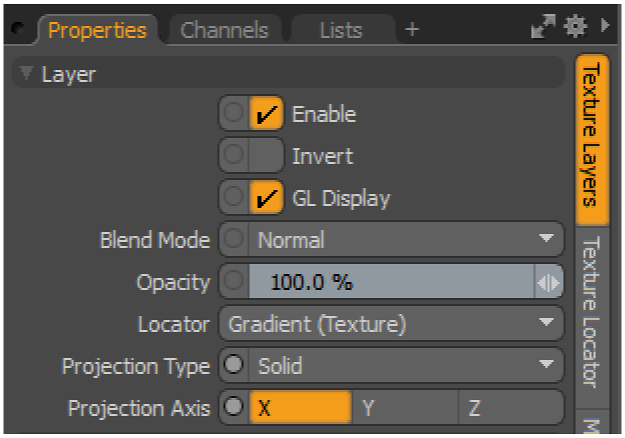
The **Dots** texture creates a grid-like pattern of dots across the surface of the mesh. The dots default to a fairly uniform position with hard edges, but you can modify the options to have different placements or soft-ramped dots. The dots are procedurally created at render time; the texture has no fixed resolution and can be magnified with no visual loss in detail.

Note: For information about adding and working with **Shader Tree** item layers, see the Shader Tree topic.


Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Dots	Diffuse Color
Base Material	(all)
Environment	
Directional Light	
Camera	

Layer Properties


Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

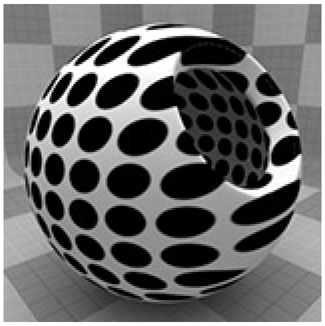
Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

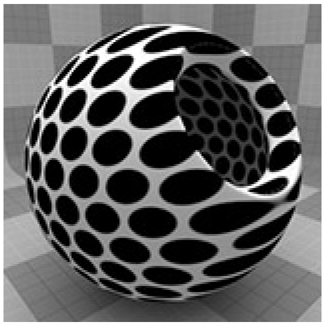
Dots Properties

Dots	
Dot Value	Determines the similarity the color of the dots to the **Filler Color**. You can use this to adjust the dot intensity. At **0%**, the dots are the **Dot Color**; at **100%**, the dots are the **Filler Color**. **Note:** When using **Bump** or **Displacement** effects, the **Dot Value** can be used to adjust the height of the dot.
Dot Color	Determines the color of the dots. This color ramps to the **Filler Color** based on the **Transition Width** value.
Alpha	The opacity of the dot. **Note:** You can adjust the **Alpha** value using the **Color Picker**.
Filler Value	The intensity of the area between the dots. This ramps from the **Dot Color** to the **Filler Color**. At **0%**, the filler is the **Dot Color**; at **100%**, the filler is the **Filler Color**.

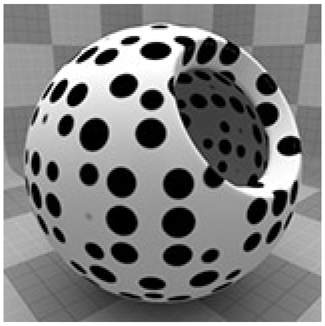
	Note: When using **Bump** or **Displacement** effects, the **Filler Value** can be used to adjust the height of the filler.
Filler Color	The color of the area between the dots.
Alpha	The opacity of the filler area.
Type	Specifies the method for applying the dots to the surface. See Type for examples. • **Square**- Creates a series of perfectly aligned rows and columns of dots that are projected in 2D on a single axis against the mesh. This is the default setting. • **Hexagon**- Offsets the dots in alternating rows to create a hexagonal pattern. As with the **Square** type, Modo projects the dots in 2D on a single axis against the mesh. • **Cube**- Creates a three-dimensional array of spheres that intersects with the surface of the mesh. For any projection type other than UV (which causes the dots to lay perfectly on the surface), the mesh intersects the spheres at different cross-sections resulting in dots of varying size and thickness.
Dot Width	Controls the diameter of the dots within the grid. At **100%** the dots touch edge to edge. Click to view examples of various Dot Width settings.
Transition Width	Creates a smooth, ramped blend from the **Dot Color** (or value) to the **Filler Color** (or value). This is a percentage of the distance between the dot and the filler. You can use extreme values to create interesting effects. Click to view examples of various Transition Width settings.
Bias	Determines whether the texture favors the **Dot Color** or the **Filler Color**. Increasing this value causes the texture to favor the **Dot Color**; decreasing the value favors the **Filler Color**. This value is dependent on the **Transition Width** value as it works across the gray tones of the image. Click to view examples of various Bias settings.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the **Dot Color** and the **Filler Color** values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme. Click to view examples of various Gain settings.



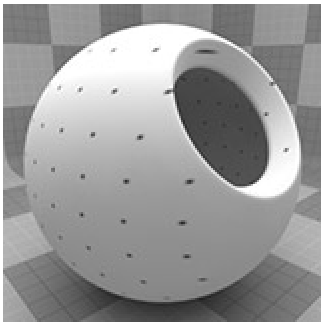
Type Square



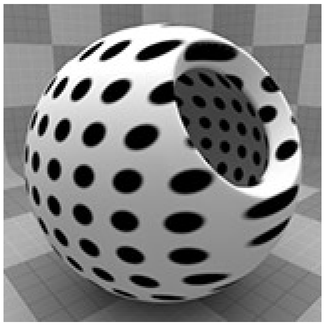
Type Hexagon



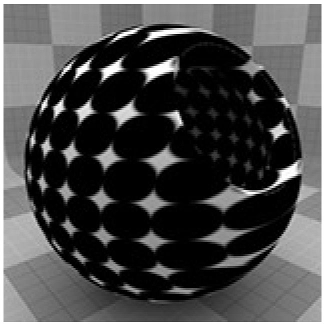
Type Cube



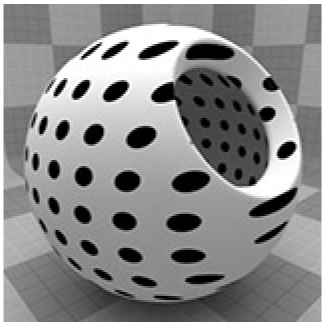
Dot Width 0%



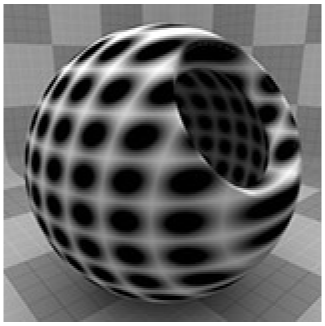
Dot Width 50%



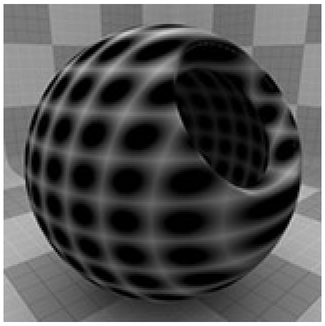
Dot Width 100%



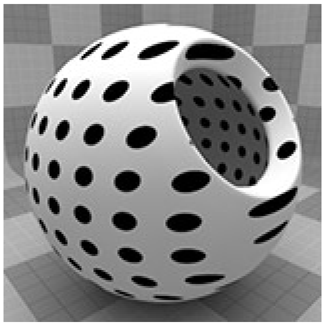
Transition Width 0%



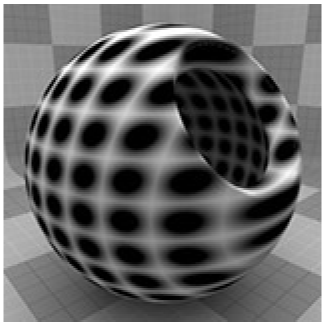
Transition Width 50%



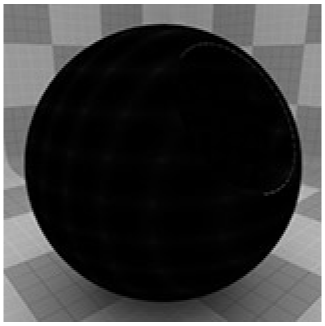
Transition Width 100%



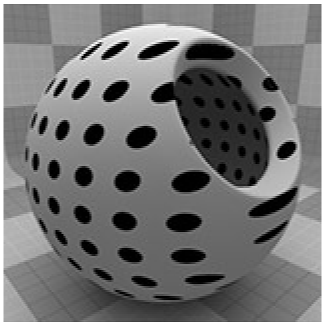
Bias 0%



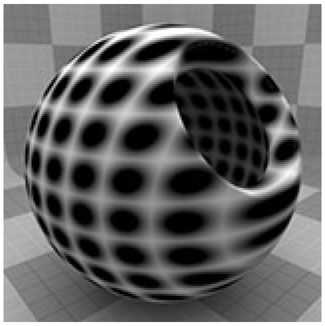
Bias 50%



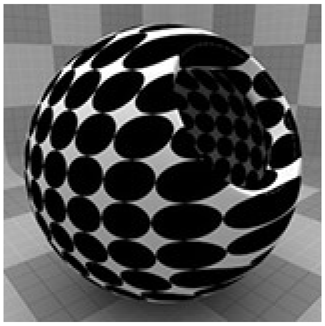
Bias 100%



Gain 0%



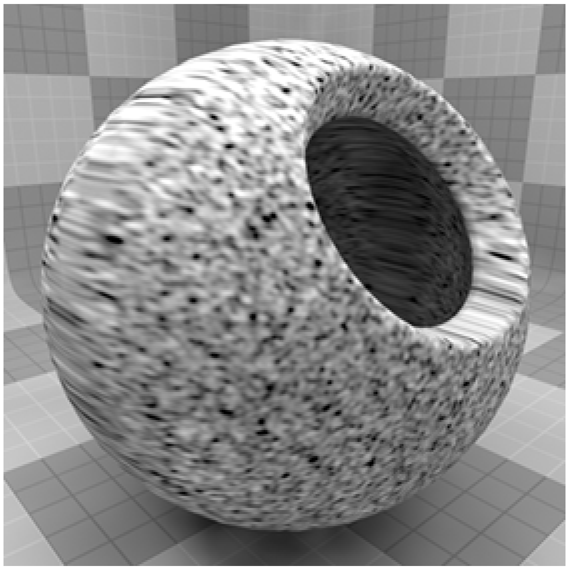
Gain 50%



Gain 100%

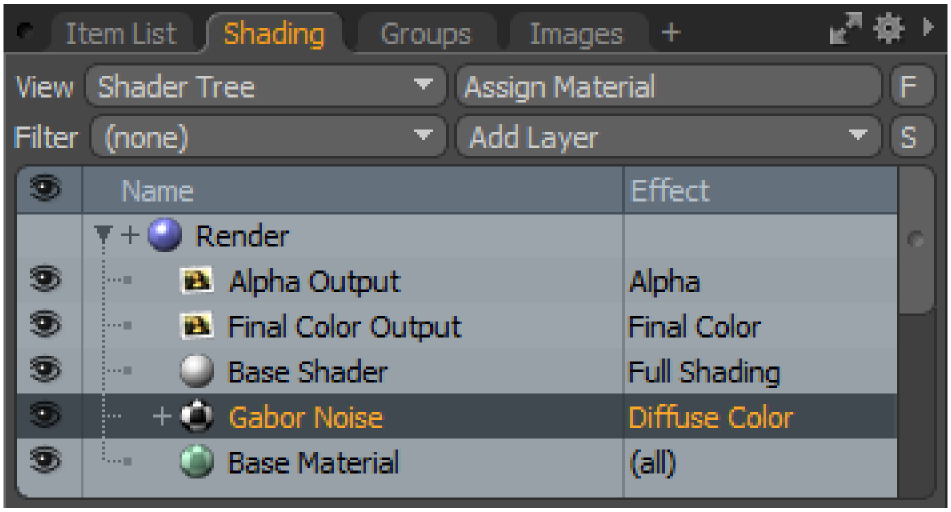
Gabor Noise

The **Gabor Noise** texture is one of Modo's procedurally generated textures. Procedural textures are mathematically created at render-time, and therefore have no fixed resolution; they can be magnified nearly infinitely with no visual loss in detail. Gabor Noise is a simple procedural noise texture that generates patterns which can be stretched and rotated, like brush strokes. It also offers direct control over the frequency characteristics of the pattern, meaning that you can specify the feature density and smoothness very easily.



The texture has two zones, the **Color/Value 1** and **Color/Value 2**. The texture modulates between the two based on the particular algorithm and settings. Essentially there are pseudo-random points created in 3D space and the texture blends from one value to the next, based on the proximity of one sample point to the next. Each zone can have a value, color and alpha value. Which of these are used is dependent on the **Effect** channel to which the texture is applied. For instance, if the texture is to be used as a displacement, the **Value** settings would be utilized whereas setting the texture **Effect** to **Luminous Color** uses the **Color** and **Alpha** settings for **Value/Color 1** and **Value/Color 2**.

For information regarding adding and working with Shader Tree Items Layers, see Shader Tree.

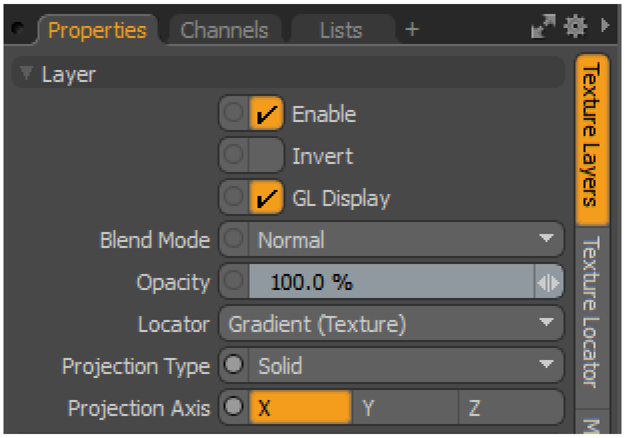

Item List
Shading
Groups
Images
+
View
Assign Material
F
Filter
Add Layer
S

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Gabor Noise	Diffuse Color
Base Material	(all)

Layer Properties

> 
Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator
M

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

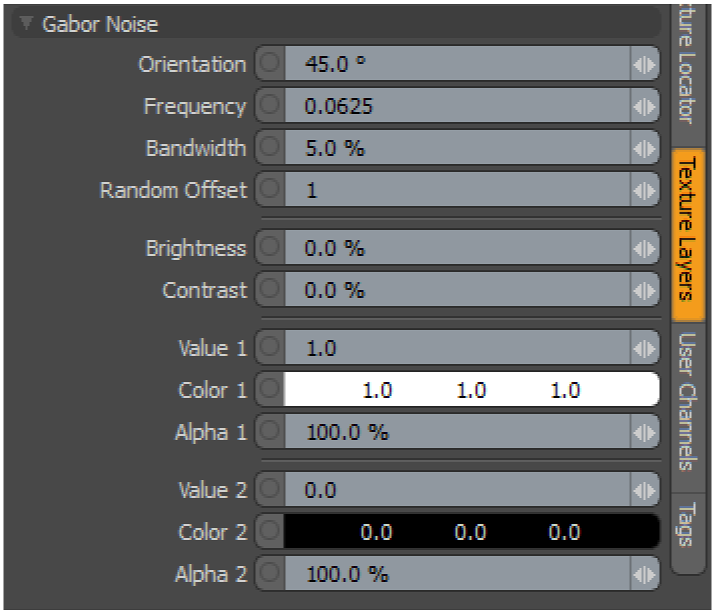
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you

	can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Gabor Noise Properties



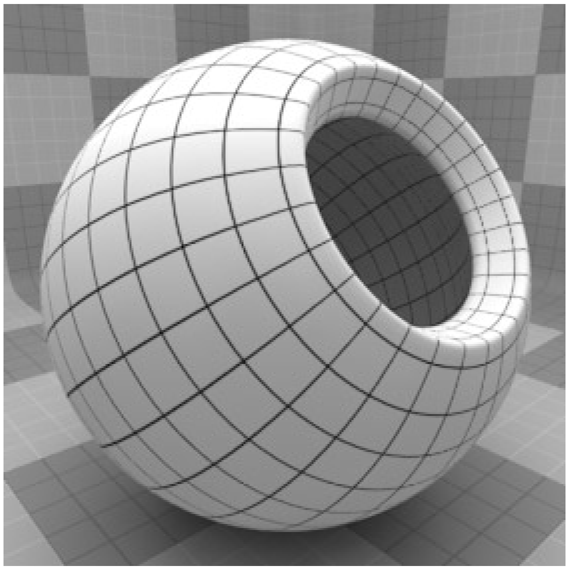
Orientation 45.0 °
Frequency 0.0625
Bandwidth 5.0 %
Random Offset 1
Brightness 0.0 %
Contrast 0.0 %
Value 1 1.0
Color 1 1.0 1.0 1.0
Alpha 1 100.0 %
Value 2 0.0
Color 2 0.0 0.0 0.0
Alpha 2 100.0 %

Gabor Noise

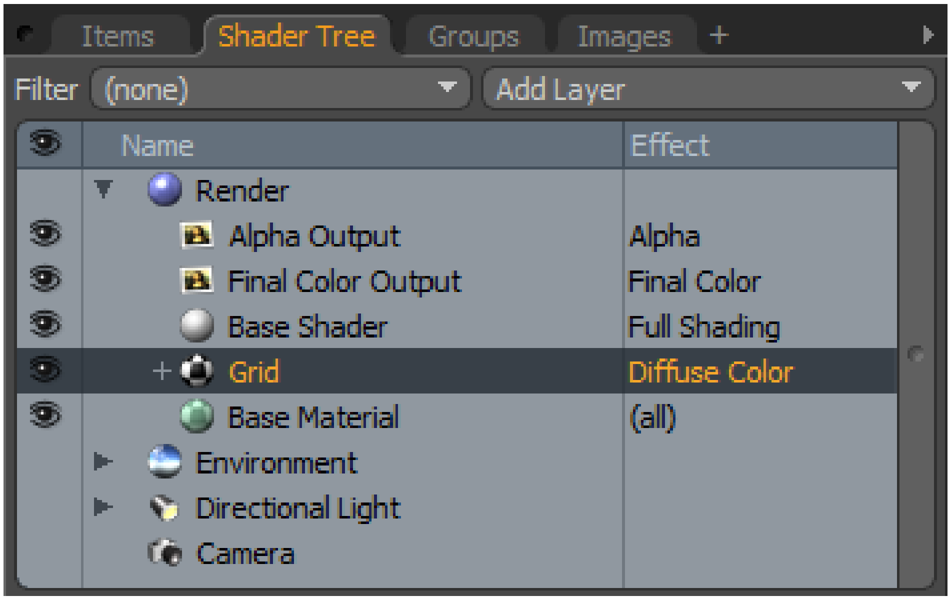
Orientation	Controls the rotation of the striped/stretched sections of the Gabor Noise, essentially rotating the noise around the defined axis (as set in the Texture Locator).
Frequency	Controls the rate of modulation of the noise features.
Bandwidth	Controls the frequency spread between the two different values, low values preserve the striped appearance, while higher values increasingly randomize the look of the pattern.
Random Offset	Shifts the location of the noise across the surface.
Brightness	Adjusts the apparent lightness to darkness intensity, modifying the fade between the **Value**/**Color 1** setting and the **Value**/**Color 2** setting.
Contrast	Adjust the apparent light to dark ratio, modifying the fade abruptness between the **Value**/**Color 1** setting and the **Value**/**Color 2** setting.
Value /Color/**Alpha** (**1** and **2**)	Depending on the defined Effect type, Modo either looks for a **Value** input such as for a **Bump Map** or a **Color** input for a **Diffuse Color** setting. The two separate values attenuate between each other, modified by the **Brightness** and **Contrast** settings. The **Alpha Values** define opacity of the values allowing underlying Shader Tree layers to show through. **Note:** You can adjust the **Alpha** value using the **Color Picker**.

Grid Procedural

The **Grid** texture creates a grid pattern between two value or color settings: **Line Value**/**Color** and **Filler Value**/**Color**. The pattern of the lines is determined by the **Type** setting, and their thickness by **Line Width**. Between the lines and the filler exists a transition zone, which is determined by the **Transition Width**. The grid is procedurally created at render time and therefore has no fixed resolution; it can be magnified with no visual loss in detail.



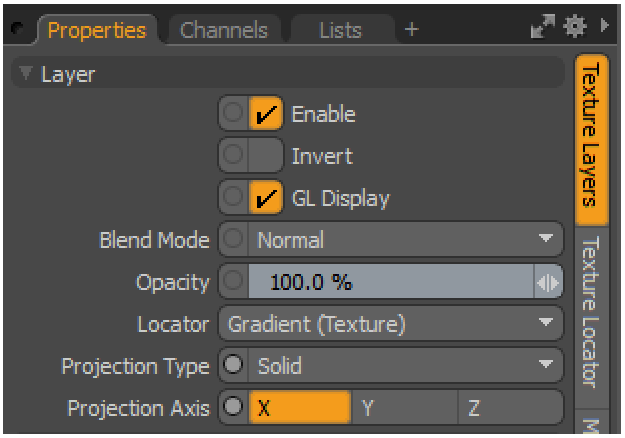
For information regarding adding and working with Shader Tree Items Layers, see Shader Tree.


Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
+ Grid	Diffuse Color
Base Material	(all)
Environment	
Directional Light	
Camera	

Layer Properties

>



☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

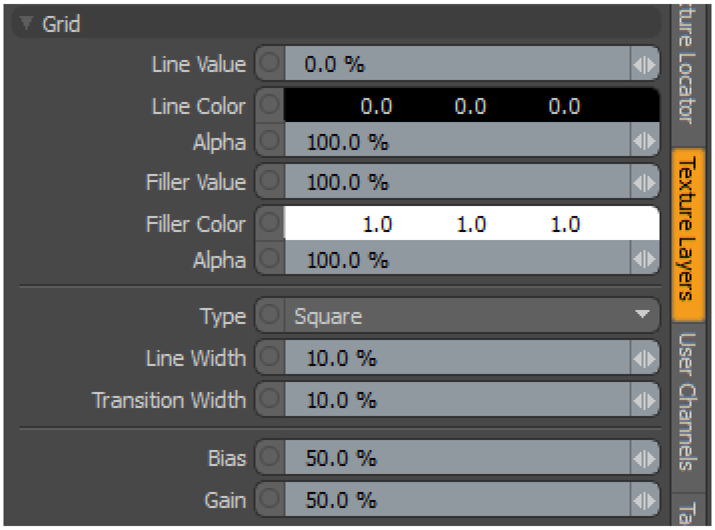
Locator: Gradient (Texture)

Projection Type: Solid

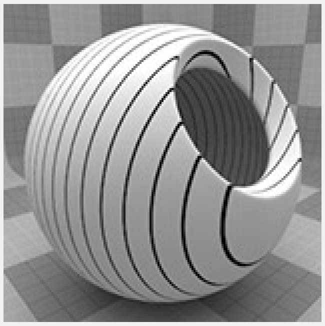
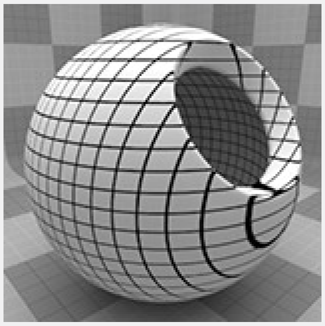
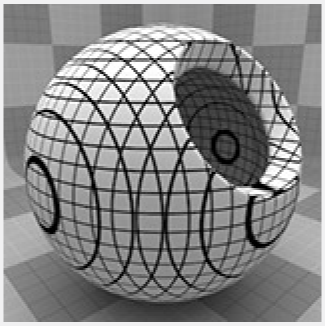
Projection Axis: X Y Z

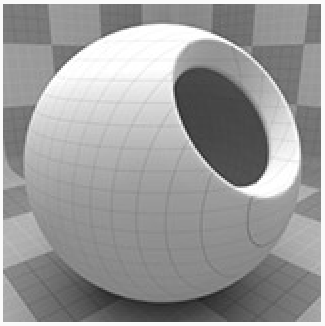
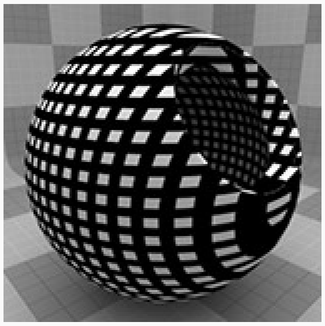
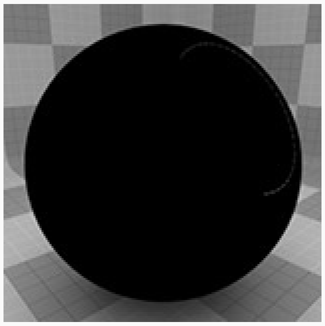
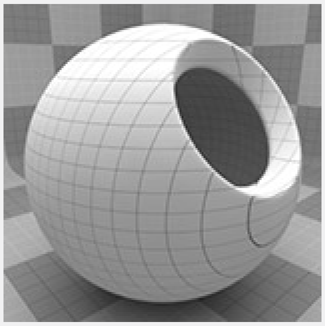
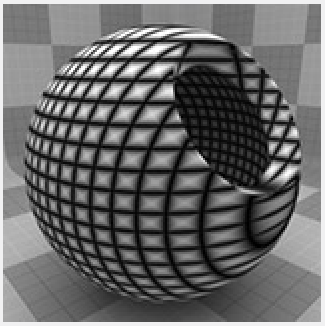
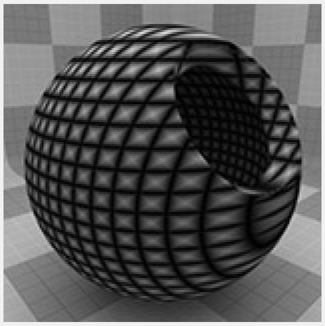
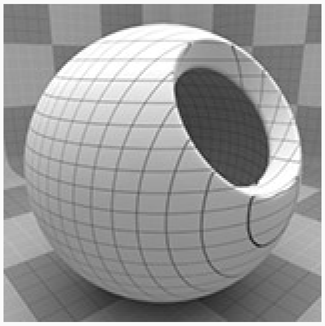
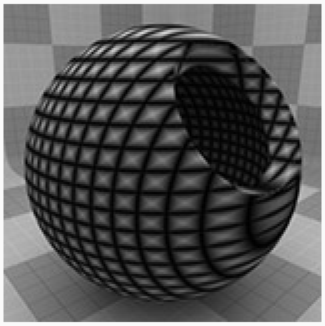
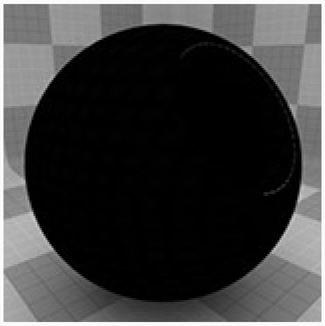
Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

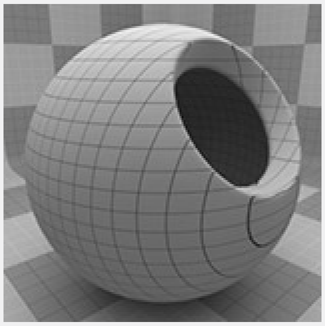
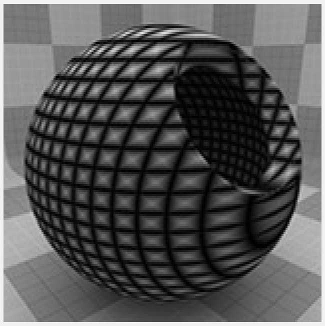
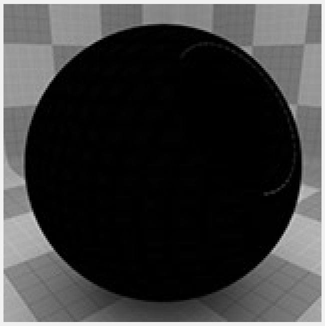
Grid Properties


Grid
Line Value
0.0 %
Line Color
0.0
0.0
0.0
Alpha
100.0 %
Filler Value
100.0 %
Filler Color
1.0
1.0
1.0
Alpha
100.0 %
Type
Square
Line Width
10.0 %
Transition Width
10.0 %
Bias
50.0 %
Gain
50.0 %
ture Locator
Texture Layers
User Channels
Ta

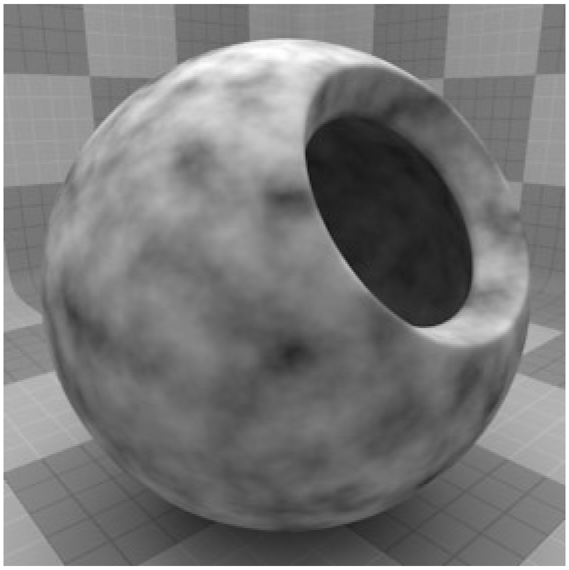
Grid	
Line Value	The similarity of the line color to the color of the surroundings (**Filler Color**). You can use this to adjust the line intensity. At **0%**, the lines are the **Line Color**; at **100%**, the lines are the **Filler Color**.
Line Color	Determines the color of the texture where lines appear, ramping toward the **Filler Color** based on the **Transition Width**. The **Alpha** value determines the level of transparency.
Alpha	The opacity of the line. **Note:** You can adjust the **Alpha** value using the **Color Picker**.
Filler Value	The level of the filler color along the range from the **Line Color** to the **Filler Color**. At **0%**, the filler color will be the same as the **Line Color**; at **100%**, the color matches the **Filler Color**.
Filler Color	Determines the color of the areas between the lines. The **Alpha** value determines the level of transparency.
Alpha	The opacity of areas between the lines.
Type	The **Grid** can be generated in many different styles. These styles are determined by the **Type** setting:

	• **Line** - This pattern consists of parallel straight lines. The orientation of the lines is determined by the **Texture Locator Axis** setting. • **Triangle** - This pattern interconnects a series of parallel lines with additional diagonal lines that intersect to create triangles. • **Square** - Two sets of lines running perpendicular to each other to create a square shaped grid. This is the default setting, as it creates what we traditionally visualize when thinking of a grid. • **Hexagon** - One series of parallel lines is intersected by a series of interconnecting lines that create hexagons at each of the intersection points. • **Cube** - This pattern is similar to Square, generating a grid pattern with a third set of parallel lines projected from the Z axis to produce a grid with depth.		
			
	Line	Square	Cube
Line Width	This setting determines the width of the lines as a percentage of the texture area. Setting this value higher, results in thicker lines until the lines completely cover the surface at 100%. Reducing the value results in thinner lines. **Tip:** When using extremely small values to create very thin lines, make sure to leave a small amount of **Transition Width** to provide an anti-aliasing effect on the fine texture.		

			
	Line Width 0%	Line Width 50%	Line Width 100%
Transition Width	This setting determines the width of the gradient ramp between the line value or color, and the filler value or color.		
			
	Transition Width 0%	Transition Width 50%	Transition Width 100%
Bias	Increasing this value causes the texture to favor the first color or value over the other, whereas decreasing the value causes the secondary color or value to be favored. This value is dependent on the **Transition Width** as it works across the gray tones of the image.		
			
	Bias 0%	Bias 50%	Bias 100%
Gain	This setting is similar to a gamma control that effects the falloff of the gradient ramp between the two color values. Setting the Gain to 100% creates a very sharp falloff effect		

	whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient. This value is dependent on the **Transition Width** as it works across the gray tones of the image.		
			
	Gain 0%	Gain 50%	Gain 100%

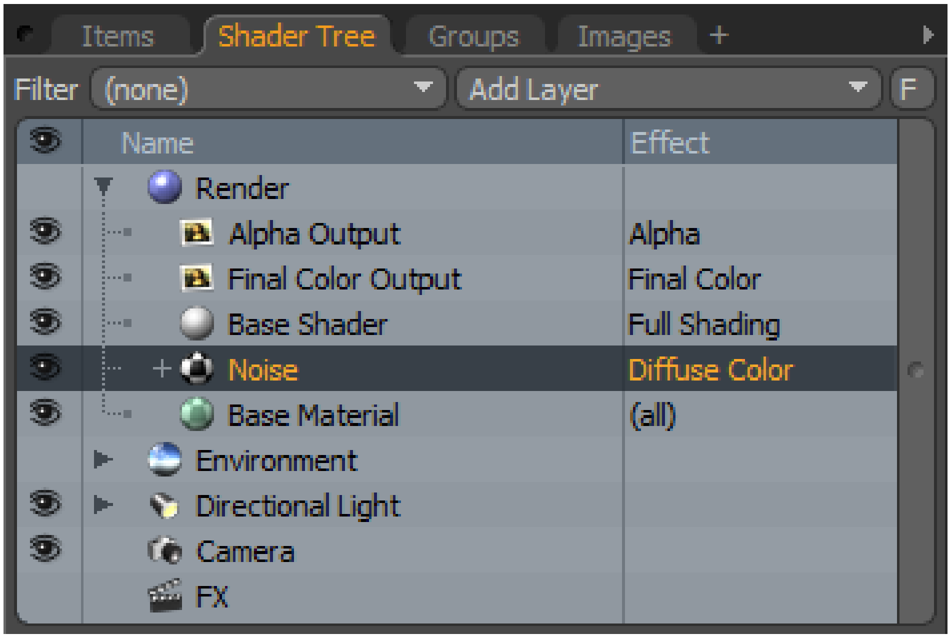
Noise Procedural



The **Noise** texture is one of the procedurally generated textures provided with Modo. Procedural textures are mathematically created at render-time and so have no fixed resolution. They can be magnified almost infinitely with no visual loss in detail. The **Noise** texture has two zones: **Color 1/Value 1** and **Color 2/Value 2**. The texture modulates from one zone to the other based on the algorithm and your settings. Essentially, Modo creates pseudo-random points in 3D space, and the texture blends from one value to the next based on the proximity of one sample point to the next.

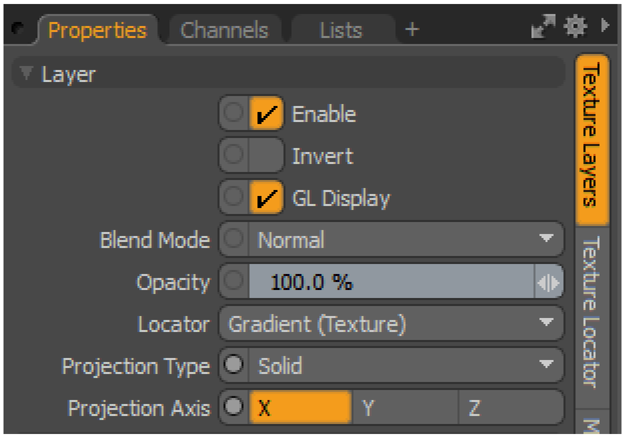
Each zone can have either a **Value** or a **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Luminous Color** , Modo uses the **Color** and **Alpha** settings for the **Color 1** and **Color 2**.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.


Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer
F

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Noise	Diffuse Color
Base Material	(all)
Environment	
Directional Light	
Camera	
FX	

Layer Properties

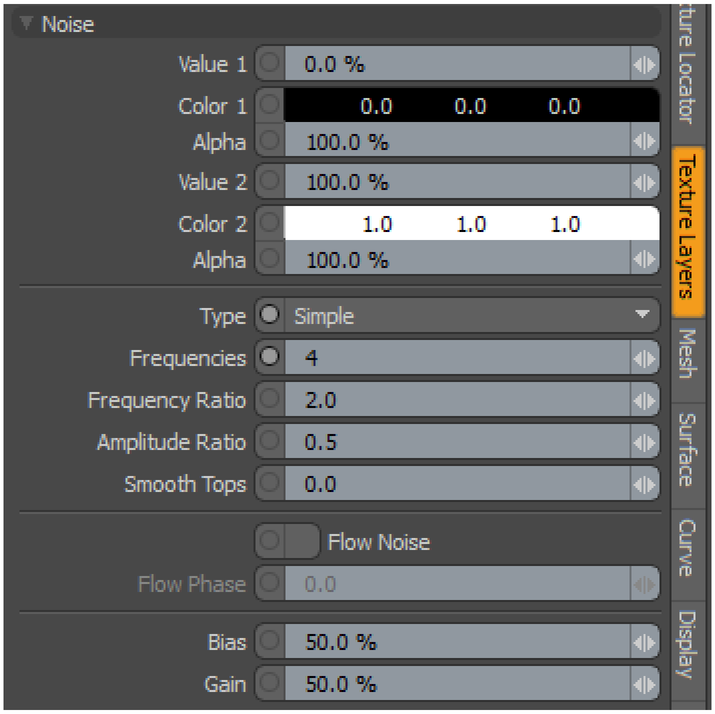

Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Noise Properties


Noise

Value 1 0.0 %

Color 1 0.0 0.0 0.0

Alpha 100.0 %

Value 2 100.0 %

Color 2 1.0 1.0 1.0

Alpha 100.0 %

Type Simple

Frequencies 4

Frequency Ratio 2.0

Amplitude Ratio 0.5

Smooth Tops 0.0

Flow Noise

Flow Phase 0.0

Bias 50.0 %

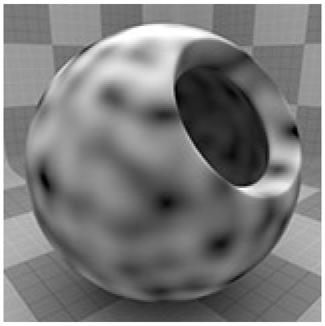
Gain 50.0 %

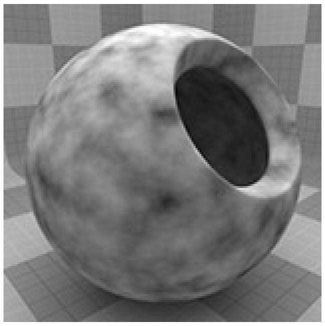
Noise	
Value 1	Determines the magnitude of the texture at the center of each sample location. Modo blends this value into **Value 2** based on proximity to surrounding sample points. **Note:** You would normally use the **Value 1** option when you set the texture layer to an effect that does not require color, such as **Bump** or **Displacement**. In this case, the value would affect the height at the center of each sample location.
Color 1	Sets the color that Modo uses at the center of each sample location. Modo blends this color into **Color 2** based on proximity to the surrounding sample points. You only use the **Color 1** option when you set the texture layer to an effect that requires color.
Alpha	Determines how transparent the **Color 1** zone is. You only use the dedicated **Alpha** option for **Color 1** when you set the texture layer to an effect that requires color.

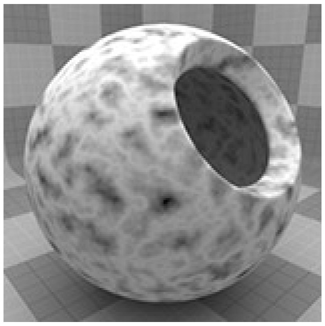
	Note: You can adjust the **Alpha** value using the **Color Picker**.
Value 2	Determines the magnitude of the texture at the center of each secondary sample location. Modo blends this value into **Value 1** based on proximity to surrounding sample points. **Note:** You would normally use the **Value 1** option when you set the texture layer to an effect that does not require color, such as **Bump** or **Displacement**. In this case, the value would affect the height at the center of each secondary sample location.
Color 2	Sets the color that Modo uses at the center of each secondary sample location. Modo blends this color into **Color 1** based on proximity to the surrounding sample points. You only use the **Color 2** option when you set the texture layer to an effect that requires color.
Alpha	Determines how transparent the **Color 2** zone is. You only use the dedicated **Alpha** option for **Color 2** when you set the texture layer to an effect that requires color.
Type	Specifies the style of the **Noise** texture. You can layer all noise-style procedurals to create a fractal effect. The traditional style of creating fractal effects includes layering a noise pattern multiple times with each new layer increasing in frequency (with smaller details and more of them) and fading in opacity. When you add a **Cellular** or **Noise** texture in Modo, the default settings (with the **Simple** type) are not "fractal" in that they only use a **Frequencies** value of **1** and, therefore, only a single application of the procedural effect. Adding multiple frequencies creates a more fractal look to the texture and activates the **Frequency Ratio** and **Amplitude Ratio** controls. Click to view examples of the various Type options. • **Simple**- Modulates between **Color 1** (or **Value 1**) and **Color 2** (or **Value 2**). With this **Noise** style, you cannot add additional frequencies (a **Frequencies** value greater than 1) and, therefore, cannot use for a fractal look. • **Fractal**- Allows for a **Frequencies** setting greater than 1; otherwise, this is the same as the **Simple** type. Once you indicate additional frequencies for a noise (or cellular) pattern, they then become fractal. Modo layers each additional frequency of the texture over the previous layer. The **Frequency Ratio** and **Amplitude Ratio** values modulate the relative detail and amplitude of the additional frequencies. • **Turbulence**- Modulates the gradient ramp between the two colors (or values) up

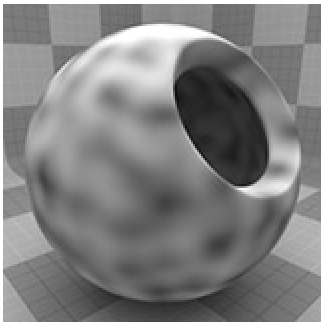
	and down rather than with a straight ramp. Otherwise, the **Turbulence** type is very similar to the **Fractal** type.
Frequencies	Determines the number of layers used by the **Noise** texture. When set to 1, the **Frequency Ratio** and **Amplitude Ratio** are disabled. Adding more than one frequency increases the detail in the procedural texture, which increases computation times. Click to view examples of various Frequencies settings.
Frequency Ratio	Determines how much detail to add into each additional application of the **Noise** texture when the **Frequencies** value is greater than 1. The default value of **2** results in the second frequency of the texture being twice as high as the first. Visually this pattern seems to be half as large but with twice as many occurrences. With more than 2 frequencies each additional layer of the noise has twice the frequency of the last. Click to view examples of various Frequency Ratio settings.
Amplitude Ratio	Determines the strength of each additional layer of noise when the **Frequencies** value is greater than 1. By default, the setting is **.5**, which yields an effect where each additional set of frequencies has half the value of the previous. Click to view examples of various Amplitude Ratio settings.
Smooth Tops	Varies the noise amplitude depending on the texture value. If you applied the **Noise** texture to a surface as a displacement, this produces a surface that looks like mountains with a smooth top and ragged valleys. (The noise amplitude increases as you go down.) If you invert the texture, the mountains look ragged and the valleys are smooth (more as they appear in nature). Click to view examples of various Smooth Tops settings.
Flow Noise	When enabled, animates the noise field without you having to physically translate (or move) the noise across the surface it is shading, which may look artificial. **Flow Noise** is a Perlin noise type that simulates advection. (The movement of something from one place to another — especially when they are different temperatures, such as cream into coffee.) The results could be described as turbulence or boiling depending on how this is used. You may find this setting useful when animating volumetrics to produce the time-lapse-clouds look that is difficult to create otherwise. After you enable **Flow Noise**, then specify a **Flow Phase** value for the animation. Click to view an example animation from enabling **Flow Noise** and setting a **Flow Phase** value.
Flow Phase	Produces a flowing type of animation to the noise that looks far more realistic than simply translating (moving) the noise across the surface or volume after enabling **Flow Noise**. To apply a continuously moving phase, set a keyframe at frame **0** and

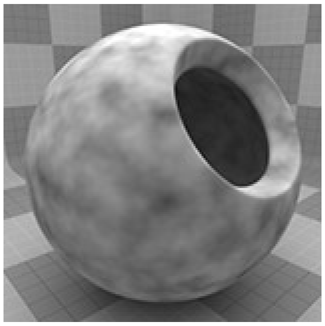
	). If the animation graph is set to **Linear** and the **Post Behavior** is set to **Linear** as well, **Flow Phase** continuously animates the length of the sequence. You can then adjust the second keyframe's value up and down to control the speed of the **Flow Phase**. Click to view an example animation from enabling **Flow Noise** and setting a **Flow Phase** value.
Bias	Determines whether the texture favors **Color 1** (or **Value 1**) over **Color 2** (or **Value 2**). Increasing this value causes the texture to favor **Color 1** (or **Value 1**) over **Color 2** (or **Value 2**). Decreasing the value causes the texture to favor **Color 2** (or **Value 2**). Click to view examples of various Bias settings.
Gain	Affects the falloff of the gradient ramp between **Color 1** (or **Value 1**) and **Color 2** (or **Value 2**). Setting the **Gain** to **100%** creates a very sharp contrast. Setting this value to **0%** reduces the contrast between the two to an almost imperceptible amount. Click to view examples of various Gain settings.

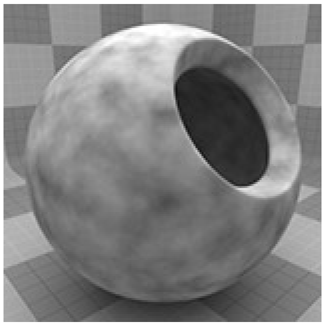

Simple

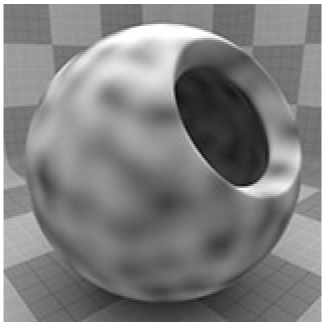

Fractal


Turbulence

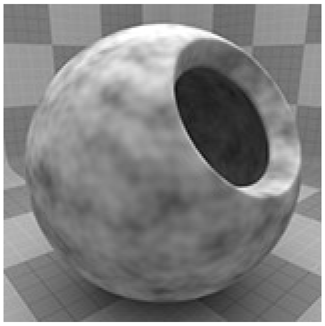

Frequency 1


Frequency 3

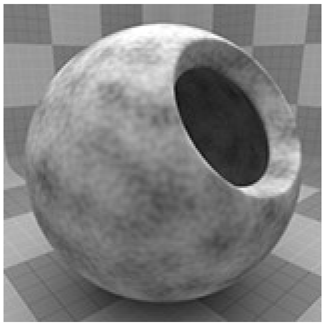

Frequency 5



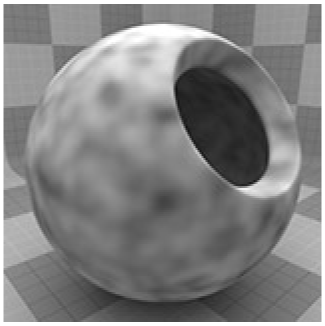
Frequency Ratio 1



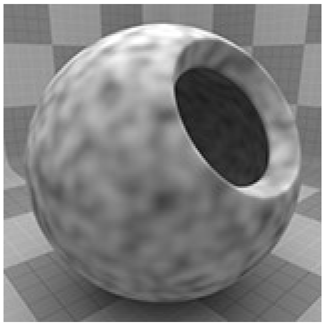
Frequency Ratio 3



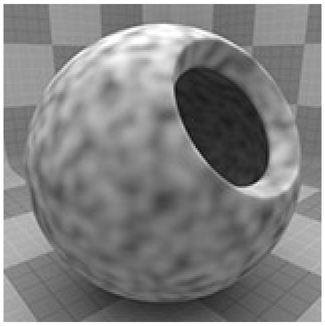
Frequency Ratio 5



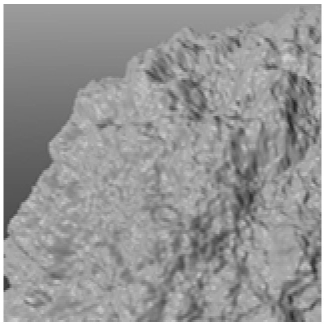
Amplitude Ratio 0.5



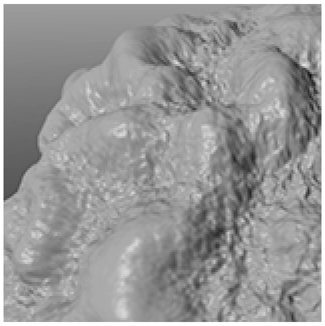
Amplitude Ratio 1.5



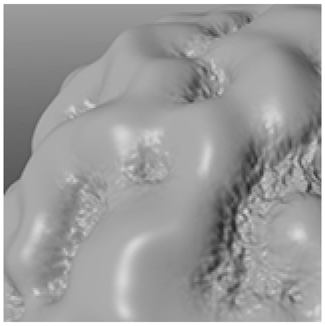
Amplitude Ratio 2.5



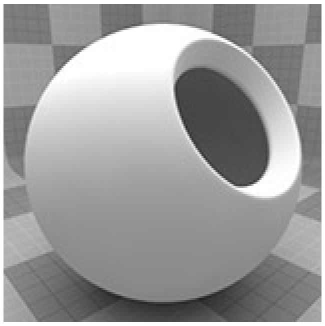
Smooth Tops 0.0



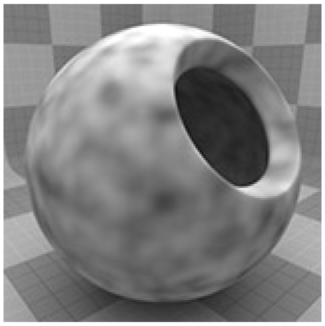
Smooth Tops 0.2



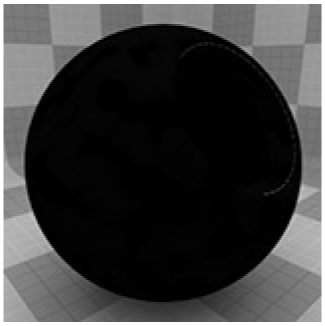
Smooth Tops 0.8



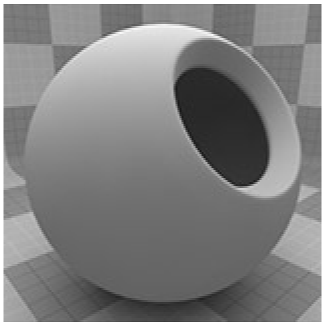
Bias 0%



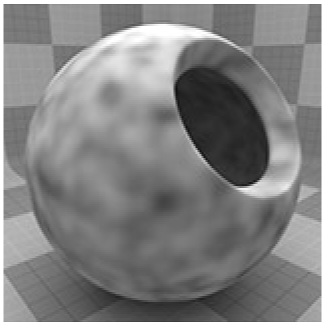
Bias 50%



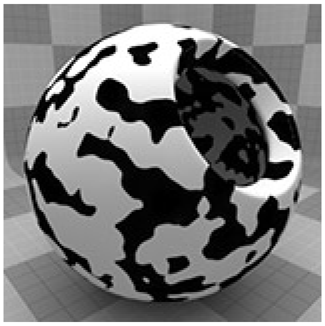
Bias 100%



Gain 0%

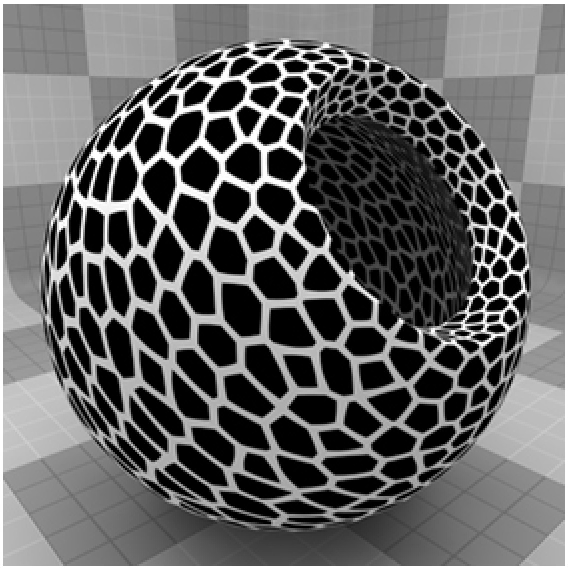


Gain 50%



Gain 100%

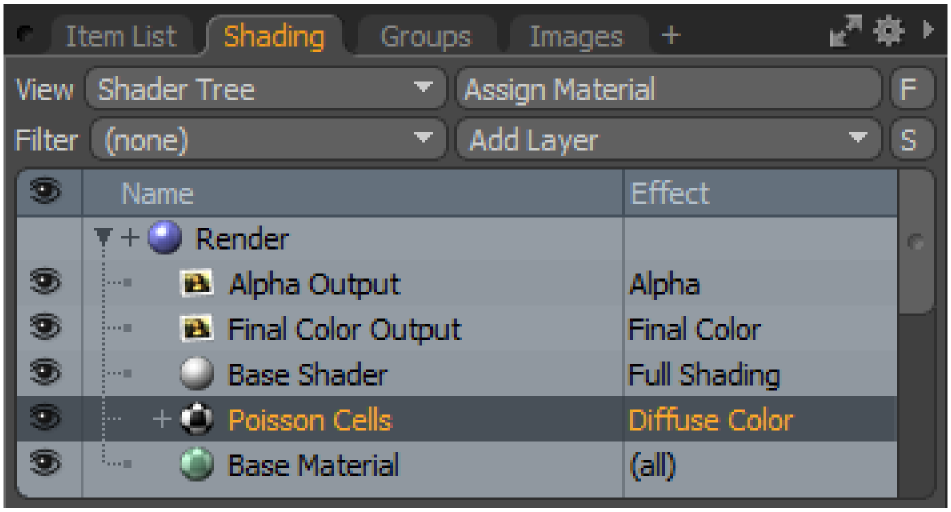
Poisson Cells



The **Poisson Cells** texture is a procedurally generated texture. Procedural textures are mathematically created at render-time and, therefore, have no fixed resolution. They can be magnified almost infinitely with no loss in detail. The **Poisson Cells** texture can be addressed by its two zones: the **Cell** and **Filler** values. The texture modulates from one zone to the other based on your settings. Each zone can have either a **Value** setting or **Color** and **Alpha**. The applied zone is dependent on the Layer Effects to which the texture is applied. For example, if you apply the texture as a **Displacement**, then Modo uses the **Value** settings, but if you set the texture effect to **Diffuse Color**, Modo uses the **Color** and **Alpha** settings for the **Cell** and **Filler** values.

The **Poisson Cells** texture is very similar to the standard Cellular texture in many ways, but it produces a more regular distribution of cells across a surface. You can achieve the best results when you apply the texture to a surface with a UV map or one of the planar projection types. By doing this, the texture can seamlessly tile outside of the defined boundary.

> **Note:** For information about adding and working with **Shader Tree** item layers, see the Shader Tree topic.

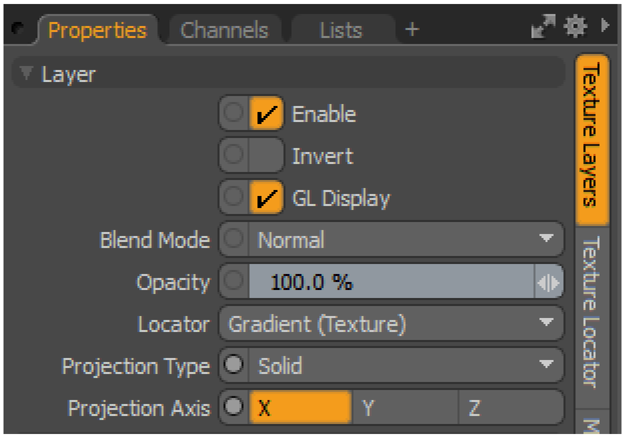

Item List
Shading
Groups
Images
+
Assign Material
F
Add Layer
S

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Poisson Cells	Diffuse Color
Base Material	(all)

Layer Properties


Properties
Channels
Lists
+
Texture Layers
Texture Locator

Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

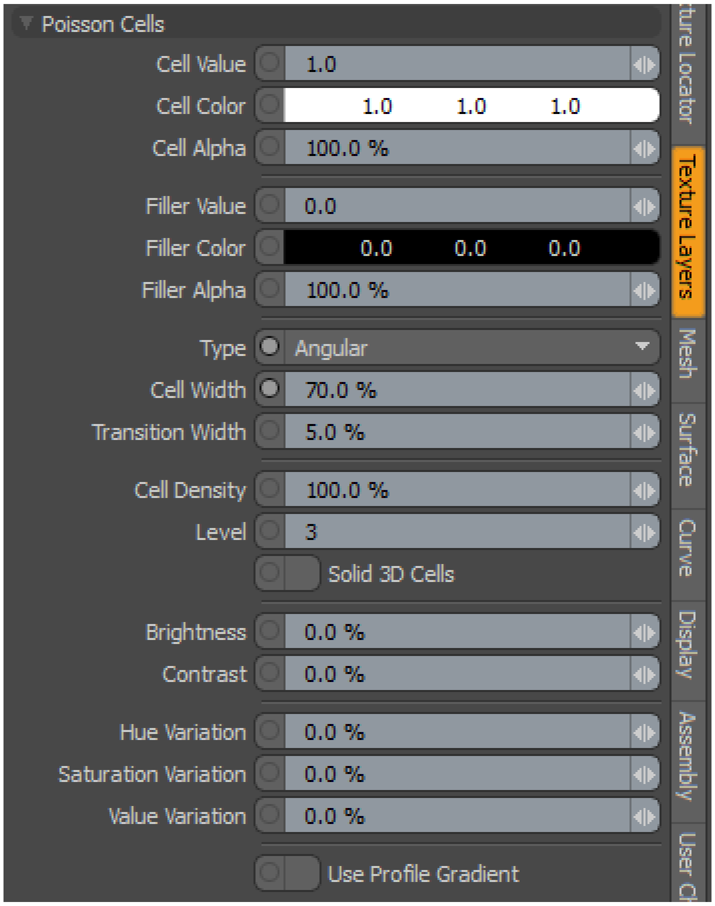
Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they

	are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Poisson Cell Properties



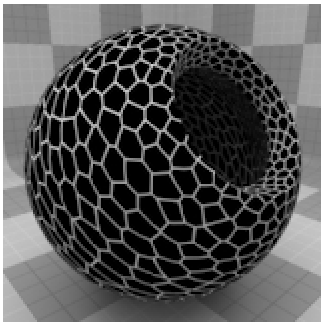
Poisson Cells
Cell Value 1.0
Cell Color 1.0 1.0 1.0
Cell Alpha 100.0 %
Filler Value 0.0
Filler Color 0.0 0.0 0.0
Filler Alpha 100.0 %
Type Angular
Cell Width 70.0 %
Transition Width 5.0 %
Cell Density 100.0 %
Level 3
Solid 3D Cells
Brightness 0.0 %
Contrast 0.0 %
Hue Variation 0.0 %
Saturation Variation 0.0 %
Value Variation 0.0 %
Use Profile Gradient

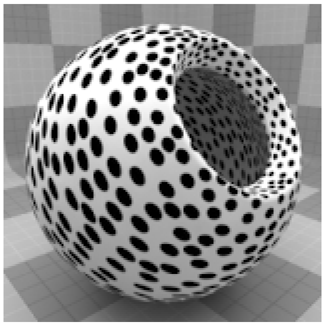
Poisson Cell	
Cell Value	Determines the magnitude of the texture where the cell is most concentrated. This value ramps to the **Filler Value**. You only use the **Cell Value** option when you set the texture layer to an effect that does not require color. **Note:** When using **Bump** or **Displacement** effects, the **Cell Value** can be used to adjust the height of the cell.
Cell Color	Determines the color of the texture where the cell is most concentrated. This color ramps to the **Filler Color**. You only use the **Cell Color** option when you set the texture layer to an effect that requires color.

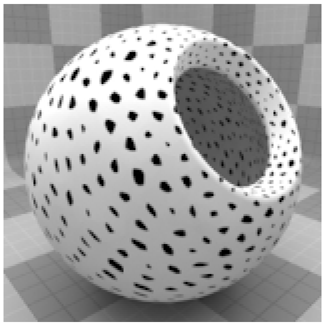
Cell Alpha	Determines the opacity of the **Cell Color** value. When the value is below **100%**, underlying **Shader Tree** layers show through attenuating to fully transparent at **0%**. **Note:** You can adjust the **Alpha** value using the **Color Picker**.
Filler Value	Determines the magnitude between the cells. This value ramps to the **Filler Value**. You only use the **Filler Value** option when you set the texture layer to an effect that does not require color. **Note:** When using **Bump** or **Displacement** effects, the **Filler Value** can be used to adjust the height of the cell.
Filler Color	Determines the color of the texture between the cells. This color ramps to the **Filler Color**. You only use the **Filler Color** option when you set the texture layer to an effect that requires color.
Filler Alpha	Determines the opacity of the **Filler Color** value. When the value is below **100%**, underlying **Shader Tree** layers show through attenuating to fully transparent at **0%**.
Type	Specifies the style of the **Poisson Cell** texture. These options determine how the texture interpolates between its sample points. • **Angular**- Creates a somewhat linear interpolation between the sample points in the texture for an effect similar to cobblestones (with the default setting). You can use this to generate effects ranging from a lunar surface to craggy skin. • **Round**- Creates circular spots randomly distributed based on the texture. In the default settings, the texture looks like overlapping, soft-edged dots. Adjusting the settings can yield effects ranging from cauliflower to chipped rock. (In the example, these are stretched because of the UV map stretching around the sphere.) See Type for examples.
Cell Width	Determines the maximum scale of the texture's features. At **100%** the maximum scale of any single detail in the texture is equal to the texture size as set on the **Texture Locator** tab. To change the scale of the details of the texture, use this value. To set the overall scale of the texture, modify the texture scale on the **Texture Locator** tab. See Cell Width for examples.

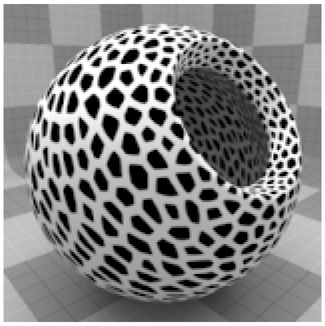
Transition Width	Controls the amount of gradient falloff around each of the texture details. Setting this value over 100% can yield interesting effects. See Transition Width for examples.
Cell Density	Controls the cell size by determining the amount of coverage for each cell within the defined grid area. Outside the grid area, the Poisson cells tile infinitely, so increasing the density makes more cells within a single tile. See Cell Density for examples.
Level	Defines the number of cells in a given solution. Each increase doubles the number of cells from the previous level. See Level for examples.
Solid 3D Cells	When enabled, allows you to apply a non-planar solid projection on the target surface. Doing this, loses the tiling functionality, and the corners of some cells may intersect with the surface causing non-uniform cells to work more like the regular cellular texture. Normally, the **Poisson Cells** texture applies a planar-type projection to a surface, where curved glancing angle areas stretch the texture. By applying the texture as a 3D solid, the target surface carves out the texturing from the 3D volume.
Brightness	Determines whether the texture favors the **Cell** values or the **Filler** values. Increasing this value causes the texture to favor the **Cell** values over the **Filler** values. Decreasing the value causes Modo to favor the **Filler** values. See Brightness for examples.
Contrast	Affects the falloff of the gradient ramp between the **Cell** and **Filler** values, which is similar to a gamma control. Setting the **Contrast** to **100%** creates a very sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with sharp falloff on either extreme of the gradient. See Contrast for examples.
Hue Variation	Varies the hue (or color) of each individual cell randomly. Higher numbers increase the color differences between cells; lower values create fewer color variations. See Hue Variation for examples.
Saturation Variation	Varies the color saturation of each individual cell randomly. Higher numbers increase the color differences between cells; lower values create fewer saturation variations.

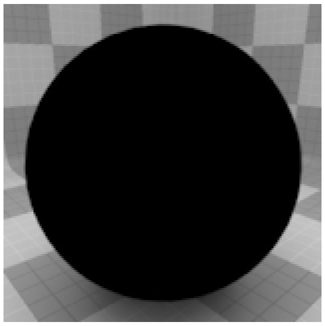
	See Saturation Variation for examples.
Value Variation	Varies the brightness (or luminosity) of each individual cell randomly. Higher numbers increase the differences between cells; lower values create fewer brightness variations. See Value Variation for examples.
Use Profile Gradient	When enabled, uses the embedded gradient control to get extra control over the appearance of the cells. This gradient replaces the smooth linear transition between the **Cell** and **Filler** values specified by the **Transition Width** setting. When you enable this, you make the embedded gradient profile visible.

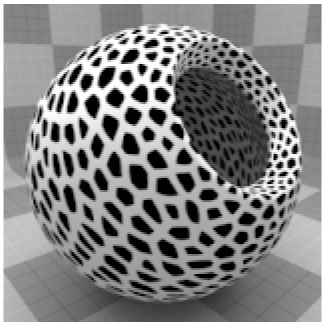

Type Angular

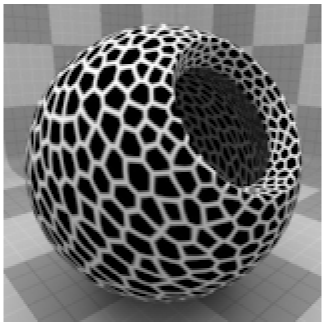

Type Round

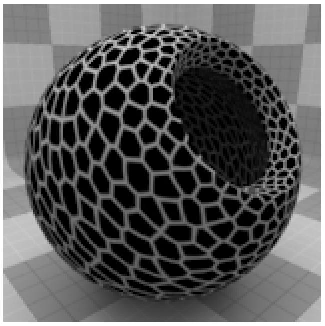

Cell Width 0%

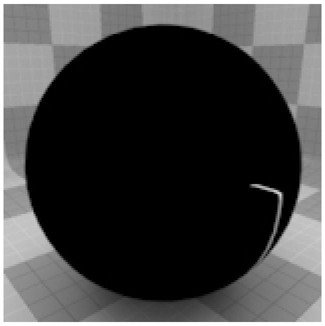

Cell Width 50%

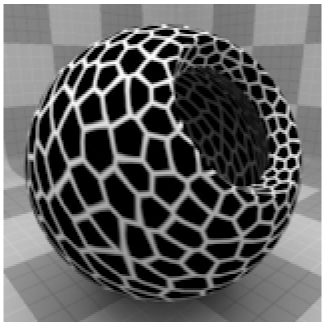

Cell Width 100%

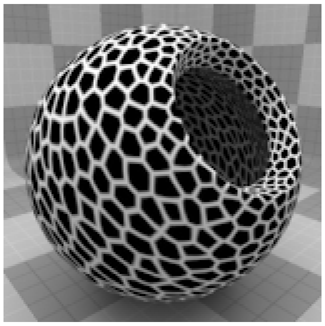

Transition Width 0%

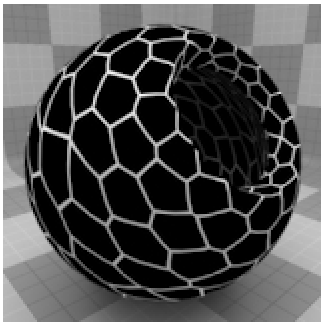

Transition Width 50%

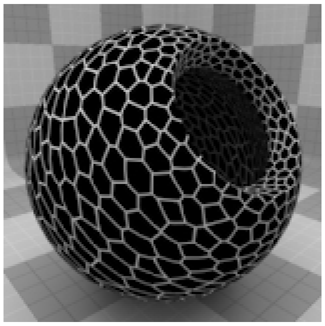

Transition Width 100%

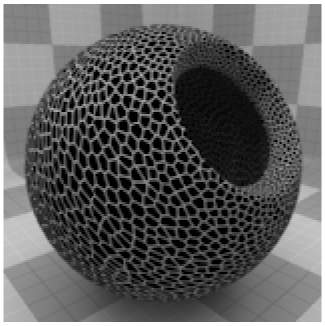

Cell Density 0%

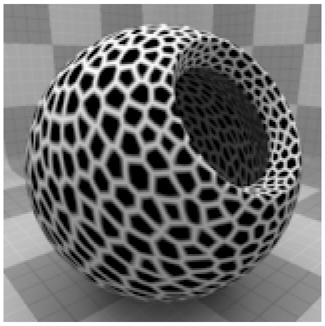

Cell Density 50%

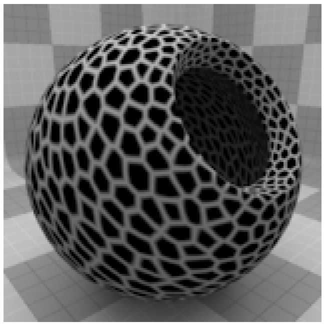

Cell Density 100%

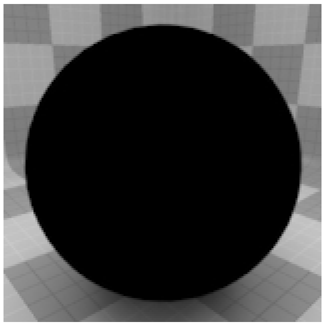

Level 1

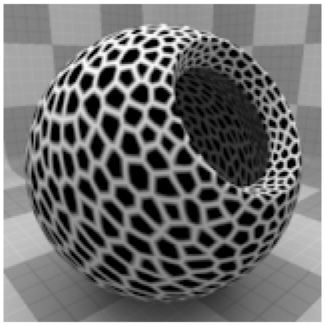

Level 3

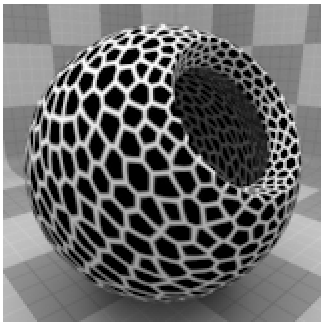

Level 5

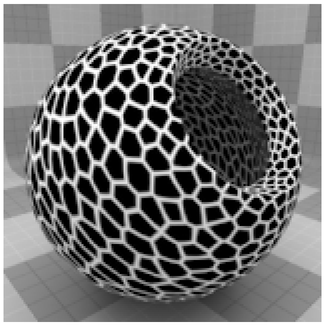

Brightness 0%

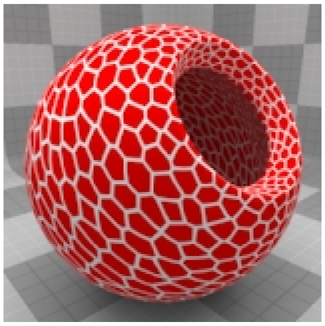

Brightness 50%


Brightness 100%

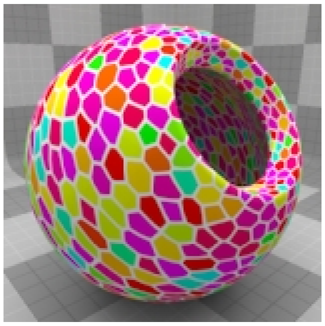

Contrast 0%


Contrast 50%

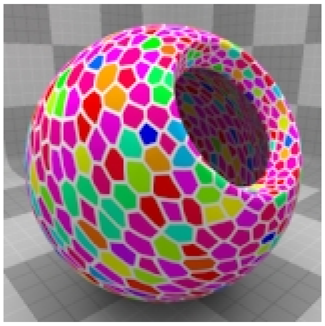

Contrast 100%



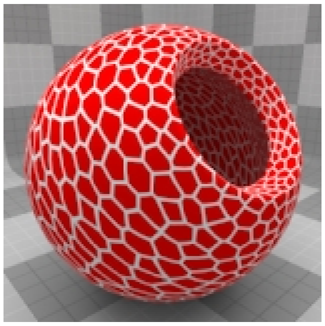
Hue 0%



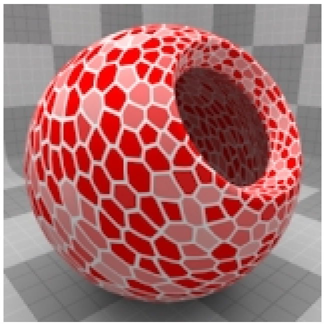
Hue 50%



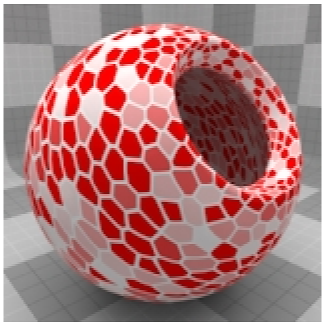
Hue 100%



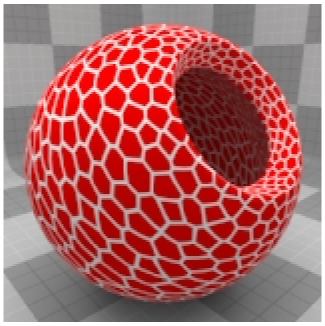
Saturation 0%



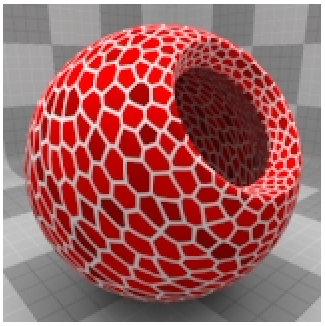
Saturation 50%



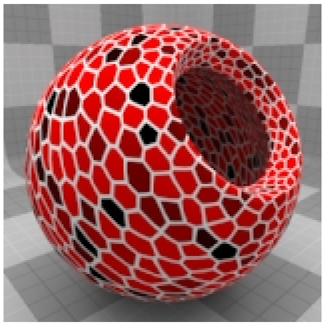
Saturation 100%



Value 0%

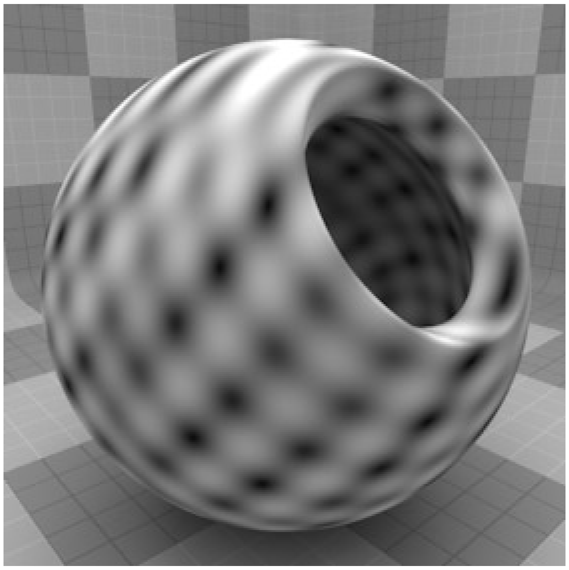


Value 50%



Value 100%

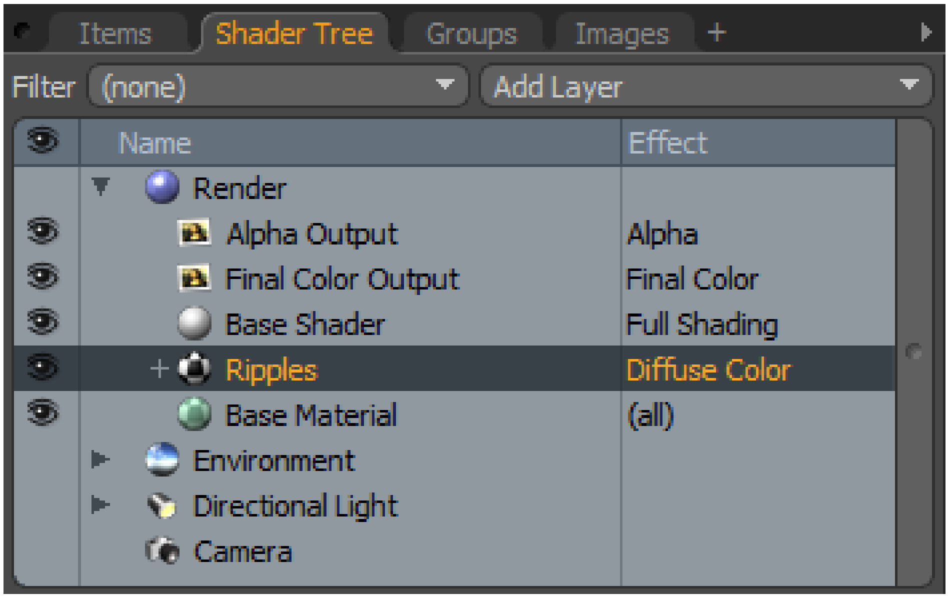
Ripples Procedural



The **Ripples** procedural texture creates a series of concentric circles that originate within a given area, and continue outward toward infinity (such as raindrops in a puddle). Procedural textures are mathematically created at render-time and have no fixed resolution. They can be magnified almost infinitely with no loss in detail.

The texture scale determines the area where the ripples originate. The values for **Sources**, **Wavelength**, and **Phase** determine the specific look of the ripples. The **Ripples** texture is a good choice when you need to simulate undulating waves.

Note: For information about adding and working with **Shader Tree** item layers, see the Shader Tree topic.

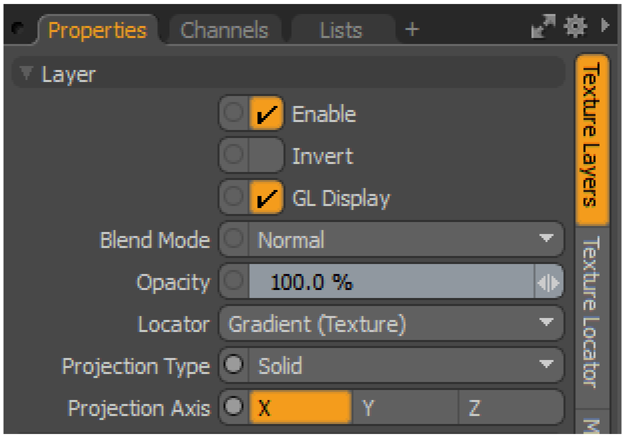


Filter (none) Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Ripples	Diffuse Color
Base Material	(all)
Environment	
Directional Light	
Camera	

Layer Properties

>

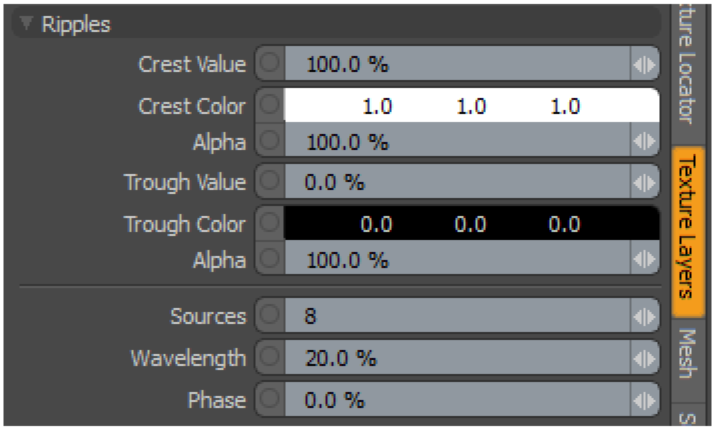


☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.

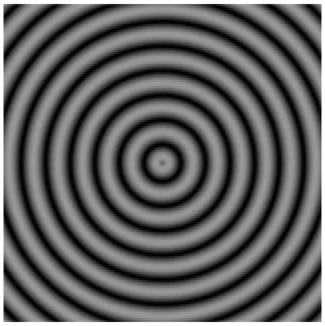
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Ripples Properties

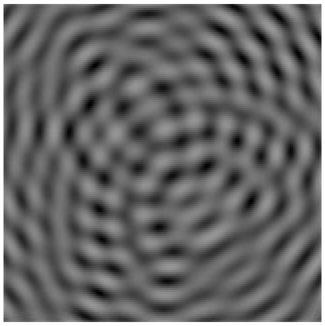

Ripples
Crest Value 100.0 %
Crest Color 1.0 1.0 1.0
Alpha 100.0 %
Trough Value 0.0 %
Trough Color 0.0 0.0 0.0
Alpha 100.0 %
Sources 8
Wavelength 20.0 %
Phase 0.0 %

Ripples	
Crest Value	The intensity or height of the top of the wave. This is only available when you apply **Ripples** to a non-colored channel, such as **Bump** or **Displacement**.
Crest Color	The RGB color value for the top part of the wave that ramps toward the **Trough Color**. This is only available when you set the texture layer to an effect (such as **Diffuse Color**) that requires color.
Alpha	The opacity of the **Crest Color**.

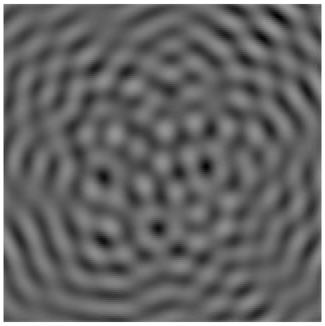
	Note: You can adjust the **Alpha** value using the **Color Picker**.
Trough Color	The RGB color value for the lowest part of the wave that ramps toward the **Crest Color**. This is only available when you set the texture layer to an effect (such as **Diffuse Color**) that requires color.
Alpha	Determines how transparent the **Trough Color** is.
Trough Value	The intensity or depth of the bottom of the wave. This is only available when you apply **Ripples** to a non-colored channel, such as **Bump** or **Displacement**.
Sources	The number of ripple generators the surface has. Larger values produce more complex rippled surfaces; lower values show the concentric circles the ripples are made from more obviously. See Sources for examples.
Wavelength	The distance between each ring. Higher values move the rings further away from each other; lower values increase the frequency of the rings to move them closer together. See Wavelength for examples.
Phase	Determines the position of the wave as it radiates from its center point. If this value is negative, the waves move outward. If the value is positive, the waves move inward.



Sources 1

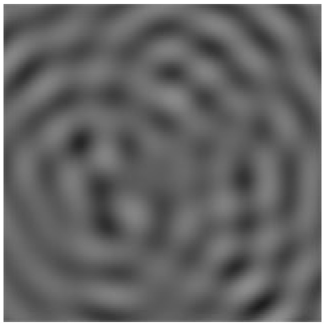


Sources 3



Sources 5

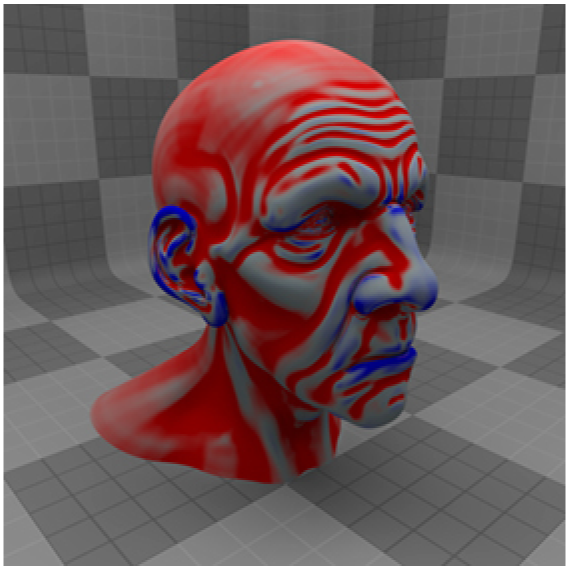
Wavelength 0.1



Wavelength 50

Wavelength 100

RT Curvature



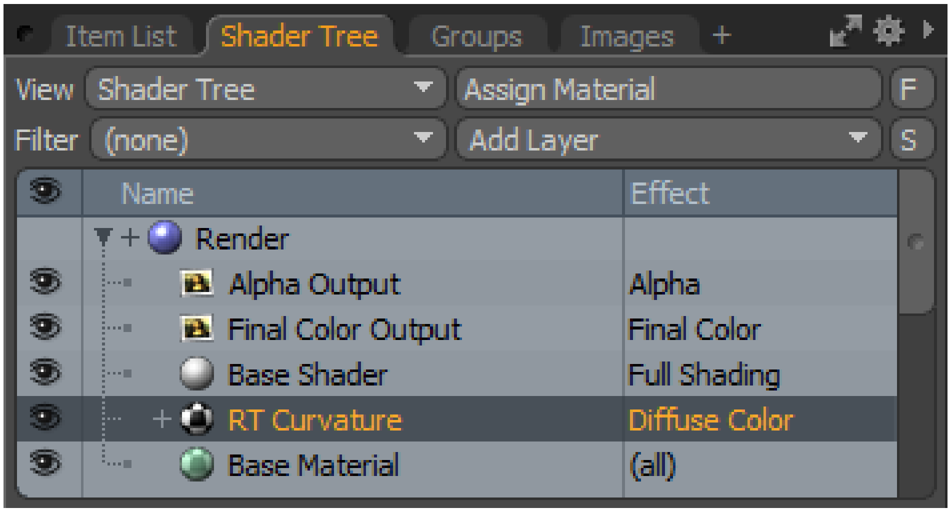
The **RT Curvature** shader is a fast and robust method of shading surface discontinuity. This type of shading is most useful in enhancing surfacing details, both modeled, and that which is added by displacement. The shading can be enhanced further when used to modulate other types of shading effects. The result is similar to a dirt or an accumulation shader that limits shading to recessed (concave) or exposed (convex) areas of a surface, and its really fast.

It works by analyzing the surface in screen-space (whats visible to the viewer) recovering the normal information and then building the shading sample off the defined thresholds. This method of calculation is much faster than fully three dimensional solutions and more robust than strictly 2D solutions.

The **RT Curvature** shader works by applying the gradient values designated in the shader's properties. The outside ends of the gradient terminate in the concave and convex areas of the surfaces topology. The degree of its application can be modified with the **Support Size** and **Magnitude** controls. For best results, the shader should be applied to surfaces with visible details. Plain, smooth surfaces, especially those of the basic primitives offer too little topology changes for interesting shading results. Also, the shader is extremely sensitive to surface normal changes and may reveal the individual polygons, this can be reduced or eliminated by increasing the subdivision level of the geometry, or by increasing the **Coefficient Antialiasing** value of the shader.

Curvature can be used to apply color directly, but it is more often used as a mask for other shading layers.

Note: For information about adding and working with **Shader Tree** item layers, see the Shader Tree topic.

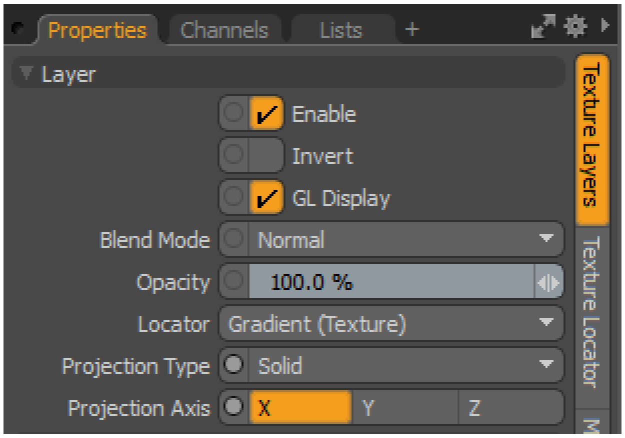

Item List
Shader Tree
Groups
Images
+
Assign Material
F
Add Layer
S

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
RT Curvature	Diffuse Color
Base Material	(all)

Layer Properties


Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator
M

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

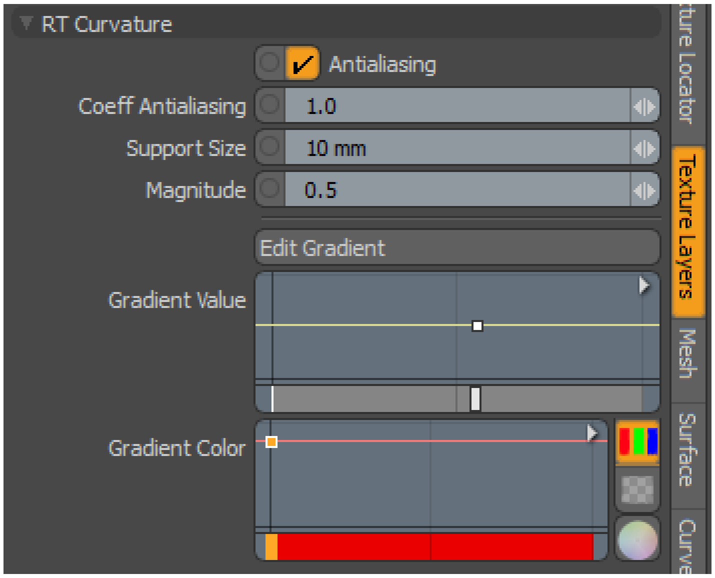
Projection Type: Solid

Projection Axis: X Y Z

>

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

RT Curvature Properties



☑ Antialiasing

Coeff Antialiasing 1.0

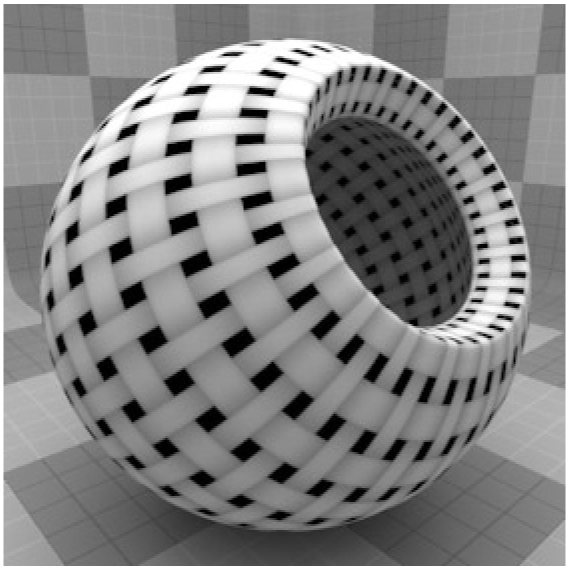
Support Size 10 mm

Magnitude 0.5

RT Curvature	
Antialiasing	Used to multi-sample the curvature shading results, producing a smoother blending between the contours. This is especially useful for surfaces where the shader tends to highlight the polygons of the underlying geometry. When enabled, the antialiasing is sampled based on the **Coefficient Antialiasing** setting. When disabled, no antialiasing is calculated.
Coeff Antialiasing	The scale of the antialiasing (when enabled). Larger values produce smoother shading results.
Support Size	The rough scale of the maximum surface features size used to calculate the curvature shading. You should set this setting to the same approximate scale as the size of the surface details that you need to highlight.
Magnitude	The absolute maximum curvature value used to compute the curvature amount. Higher values produce sharper contrasts across the shading of opposing surface details, while lower values produce smoother, broader shading.
Edit Gradient	Opens a full-sized Gradient Editor for creating or refining the target gradient at a larger size. With a default **Gradient** view, you need to create the target values in between the 0.0 and 1.0, with the 0.0 position terminating on the outermost convex area attenuating toward the 1.0 position terminating at the innermost concave areas.

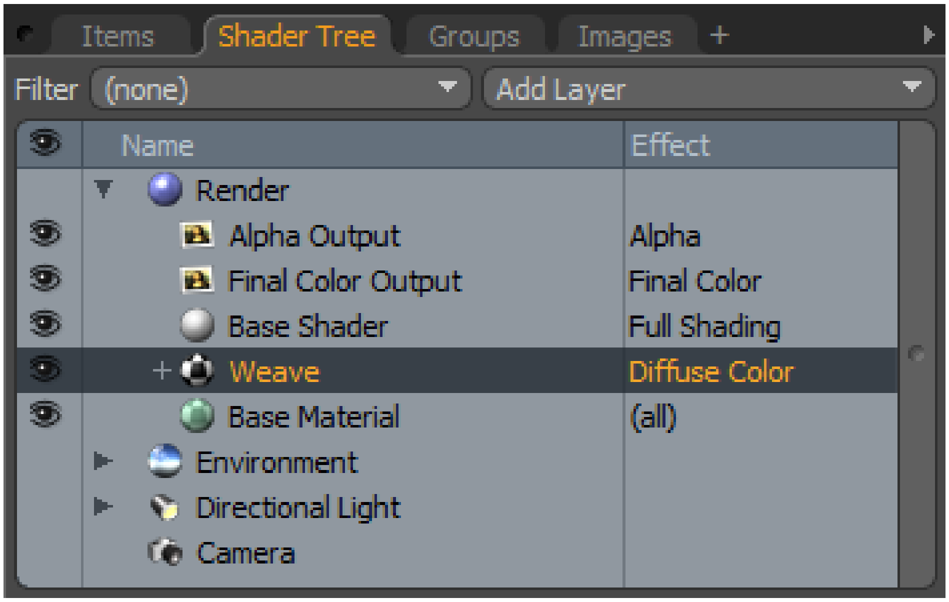
Gradient Value/Color	Inline mini gradient editors allowing you to define gradient value keys easily without having to leave the basic Modo interface. Just as a keyframe stores transform values for items, gradient keys store colors or values and softly attenuate between the defined key positions. You can add keys by middle-clicking along the gradient and then click and drag the key up or down, adjusting the value at that position. The gradient bar directly underneath provides a visual reference to the values being generated by the gradient itself. In the **Color** gradient, clicking the color wheel icon allows precise input of color values using the standard Modo color picker. The left side of the gradient editor terminates on the outermost convex area attenuating toward the right-hand side of the gradient terminating at the innermost concave areas.

Weave Procedural

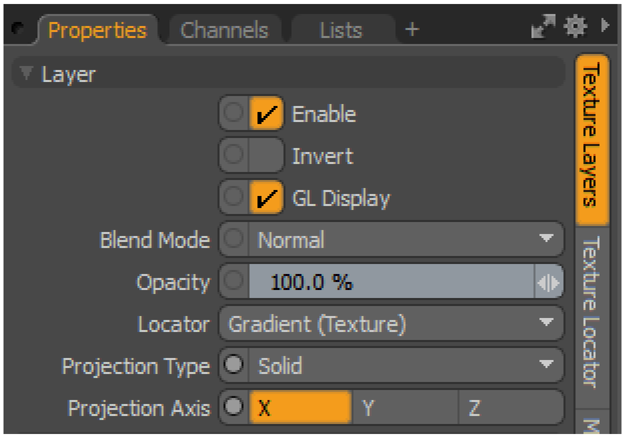


The **Weave** procedural texture simulates woven yarn in a basic basket-weave pattern. This pattern shows a series of parallel rows combined with columns that thread over and under alternating rows. Modo creates procedural textures mathematically at render time and, therefore, they have no fixed resolution. You can magnify procedural textures almost infinitely with no visual loss in detail. The **Yarn Width** and **Roundness** values control the look of the actual woven strands.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.


Items
Shader Tree
Groups
Images
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Weave
Diffuse Color
Base Material
(all)
Environment
Directional Light
Camera

Layer Properties


Properties
Channels
Lists
Layer
Texture Layers
Texture Locator

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

>

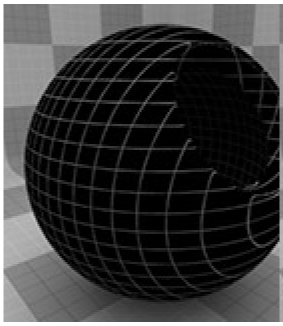
Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.

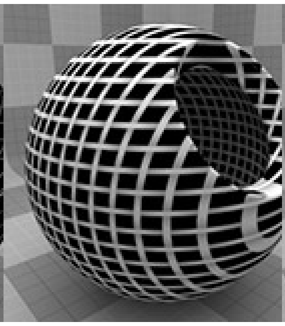
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

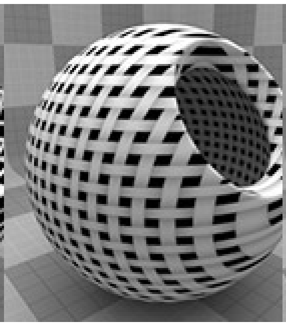
Weave Properties

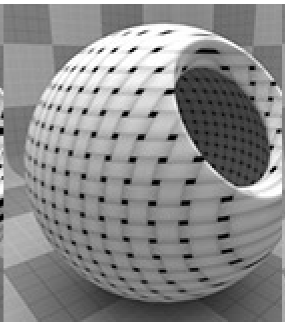
Weave	
Yarn Value	The intensity of the virtual yarn. **Note:** When using **Bump** or **Displacement** effects, the **Yarn Value** can be used to adjust the height of the yarn.
Yarn Color	Specifies the RGB color value for the columns and rows of virtual yarn. This is only available when you apply **Weave** to a color channel (such as **Diffuse Color**).
Alpha	**Yarn Color** transparency. **Note:** You can adjust the **Alpha** value using the **Color Picker**.

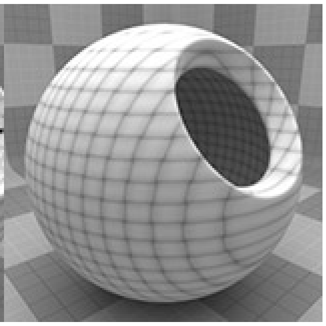
Gap Value	The intensity of the gaps between the virtual yarn. **Note:** When using **Bump** or **Displacement** effects, the **Gap Value** can be used to adjust the height of the gap.
Gap Color	The RGB color value for the spaces between the virtual yarn. This is only available when you apply **Weave** to a color channel (such as **Diffuse Color**).
Alpha	**Gap Color** transparency.
Yarn Width	The thickness of the virtual columns and rows of yarn. A value of **100%** completely eliminates the adjoining gaps to look like a tightly woven basket. Smaller values simulate loosely woven meshes (such as a net or screen). See Yarn Width for examples.
Roundness	The roundness of the strands of virtual yarn. A value of **0%** creates flat strands; a value of **100%** renders strands as woven tubes. See Roundness for examples.

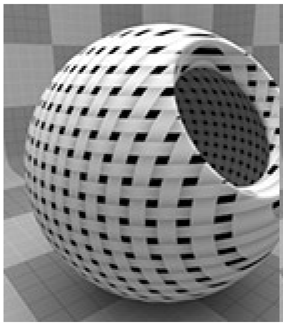

Yarn Width 0.1%

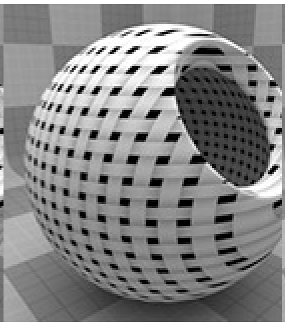

Yarn Width 25%

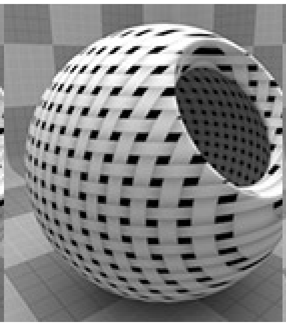

Yarn Width 50%

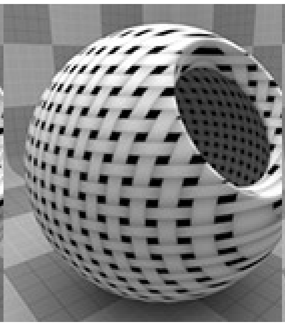

Yarn Width 75%

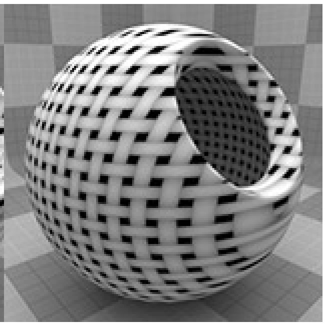

Yarn Width 100%


Roundness 0%

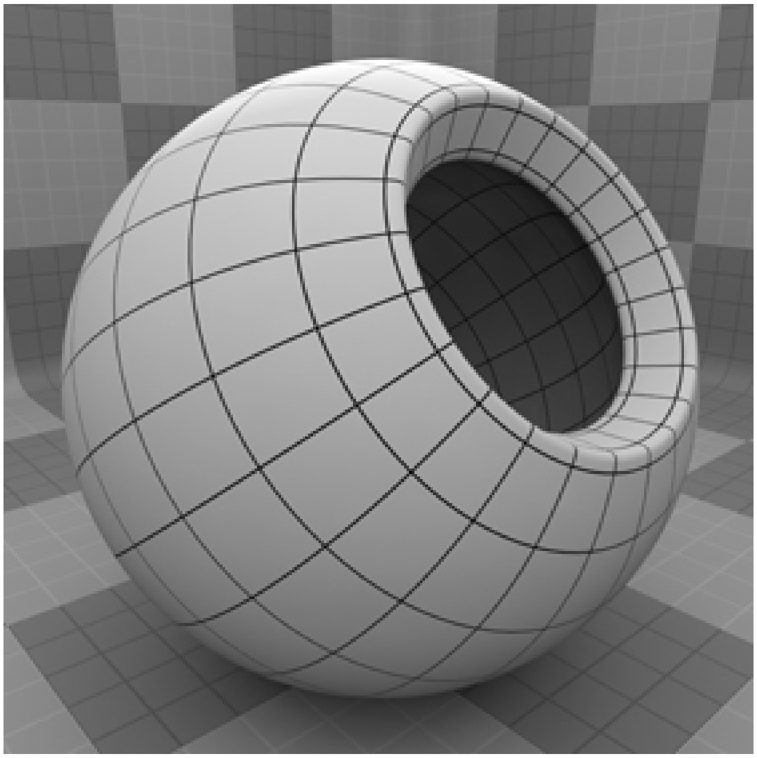

Roundness 25%


Roundness 50%


Roundness 75%


Roundness 100%

Wireframe Texture

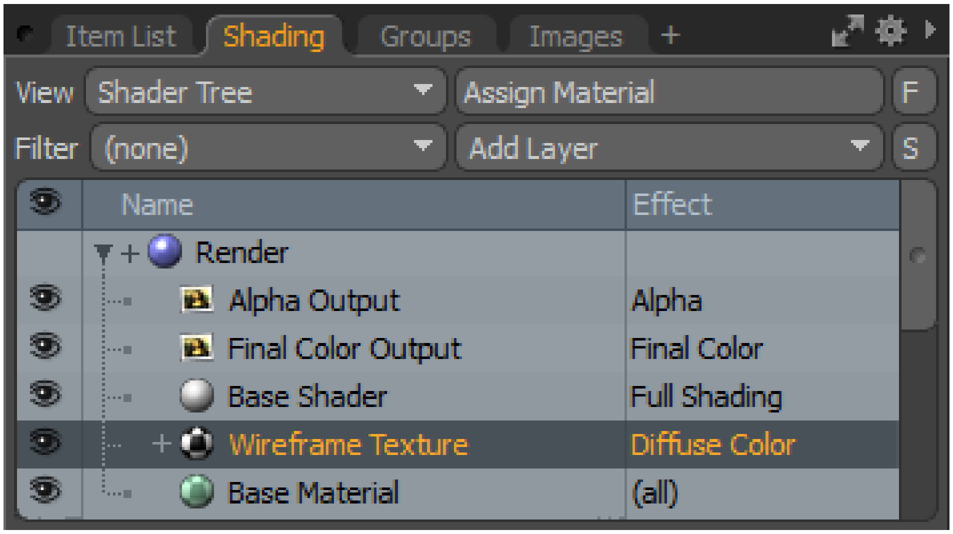


The **Wireframe** texture is a special type of procedural texture that can be applied to draw lines along the polygon edges of the target surface. It is useful to produce the same type of wireframe line rendering used in the 3D Viewports, but in a final rendered image. You can assign the Wireframe texture in the **Shader Tree** under **Add Layer** > **Special** > **Wireframe Texture**.

Note: The Wireframe texture cannot be applied to deforming meshes, but it can be baked into a UV mapped texture and applied to the deformed mesh.

The shader gives you control over the line color and thickness, and a selection set can be assigned to limit the lines to just a specific area of the target surface. The Wireframe texture can be applied in any way that a normal **Shader Tree** texture layer can be used, making it useful for other purposes as well, such as layer masks, unique displacement effects and to create interesting layer blending effects.

Note: For information regarding adding and working with **Shader Tree** item layer, see the Shader Tree topic.


Item List
Shading
Groups
Images
+
View
Assign Material
F
Filter
Add Layer
S

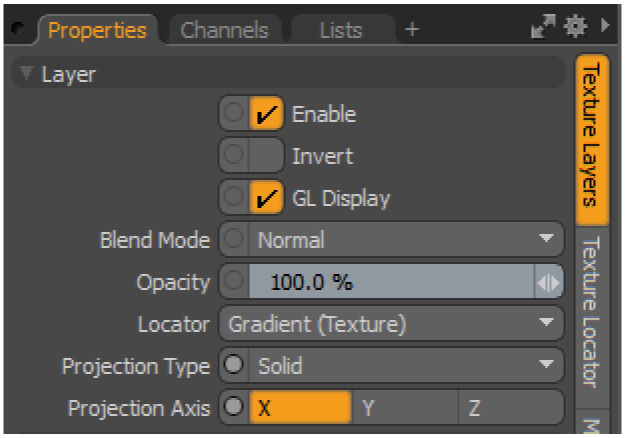
View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Wireframe Texture	Diffuse Color
Base Material	(all)

Layer Properties

>


Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator
M

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

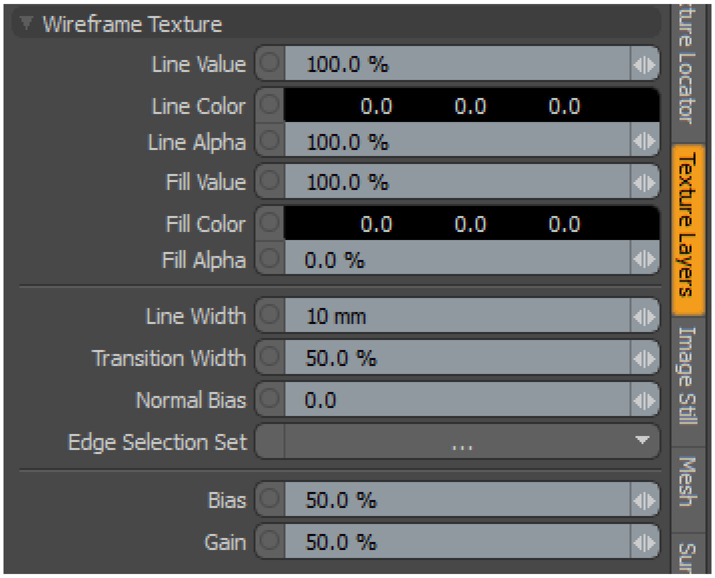
Projection Type: Solid

Projection Axis: X Y Z

Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in

	the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

Wireframe Properties



Wireframe Texture

Line Value 100.0 %

Line Color 0.0 0.0 0.0

Line Alpha 100.0 %

Fill Value 100.0 %

Fill Color 0.0 0.0 0.0

Fill Alpha 0.0 %

Line Width 10 mm

Transition Width 50.0 %

Normal Bias 0.0

Edge Selection Set ...

Bias 50.0 %

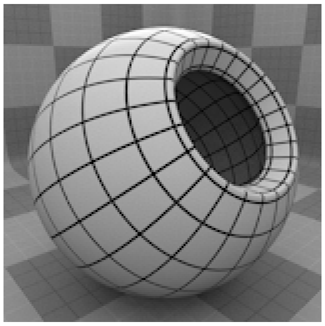
Gain 50.0 %

Wireframe Texture	
Line Value/Color	Depending on the **Effect** definition for the layer, the line itself is represented on the surface by either a **Value** amount, such as with bump maps and specular amount values, or by a **Color** such as with diffuse color or reflection color.
Line Alpha	Determines the opacity of the resulting lines on the surface. At 100%, the line is fully

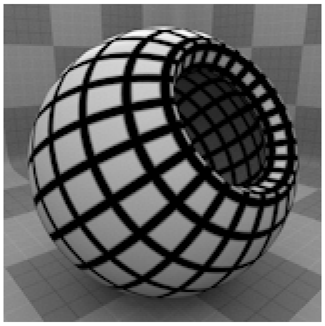
	opaque, attenuating toward 0%, where the line is fully transparent.
Fill Value/Color	Depending on the **Effect** definition for the layer, the fill area is represented on the surface by either a **Value** amount, such as with bump maps and specular amount values, or by a **Color**, such as with diffuse color or reflection color. By default, the **Fill Alpha** value is set as fully transparent so that only the lines show, rendering the fill area completely transparent by default (different from any other texture layer defaults when added) so changes to the fill might not be visible at first because of this behavior.
Fill Alpha	This value determines the opacity of the area between the lines on the surface. At 100%, the fill is fully opaque, attenuating toward 0% where the line is fully transparent. **Note:** By default, the **Fill Alpha** value is set as fully transparent so that only the lines show, rendering the fill area completely clear by default (different from any other texture layer defaults when added).
Line Width	Specifies the thickness of the line area of the texture. For examples, see Wireframe Texture Examples.
Transition Width	Defines an area of transition between the line and fill areas, defined as a percentage of the **Line Width**. Even when a soft edge isn't desired, a small amount can aid the anti-aliasing function, producing a smoother line. For examples, see Wireframe Texture Examples.
Normal Bias	You can use this option to flatten the wireframe shading toward the surface, reducing crosstalk from opposing surfaces in close proximity to each other. The control is activated when the **Normal Bias** value is greater than 0. Crosstalk artifacts can also be minimized by applying two separate Wireframe textures on each opposing surface.
Edge Selection Set	Defines an edge selection set to limit the application of the Wireframe shading to specific areas of the surface. Edge selection sets are regular selection sets made while in **Edges Selection mode**. Once a named set is defined, it can be applied here using the **Edge Selection Set** option by selecting the named set from the pop-up.
Bias	When a **Transition Width** amount is defined, the **Bias** value pushes the shading to favor more either the fill or line area of the shading, depending on the value entered.

	The default 50% value produces a linear transition across the defined width. For examples, see Wireframe Texture Examples.
Gain	This setting is similar to a Gamma control that affects the falloff across the transition width. Setting the **Gain** to 100% creates a very sharp falloff effect, whereas setting the value to 0% creates a plateau around the value or color mid-point with sharp falloff on either extreme of the gradient. This value depens on the **Transition Width**, as it works across the gray tones of the image. For examples, see Wireframe Texture Examples.

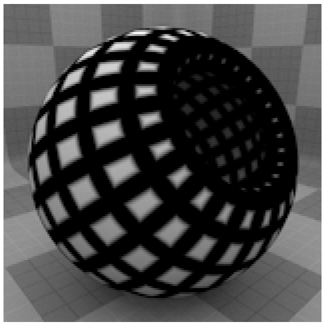
Wireframe Texture Examples



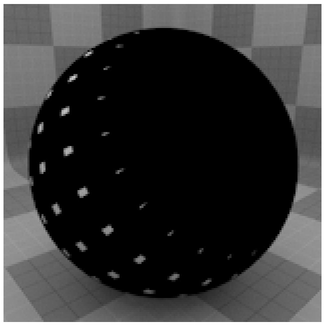
Line Width 1 mm



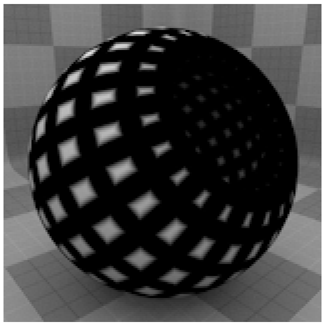
Line Width 3 mm



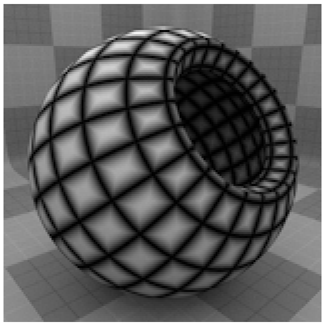
Line Width 5 mm



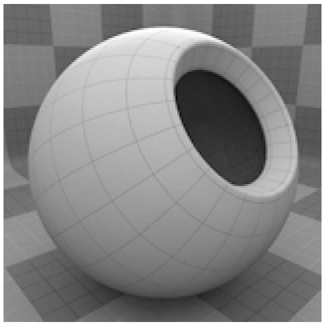
Transition Width 0%

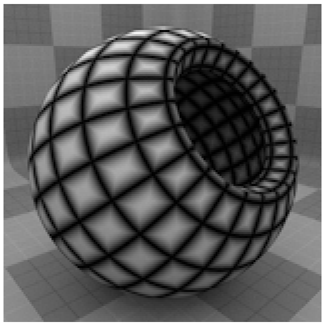


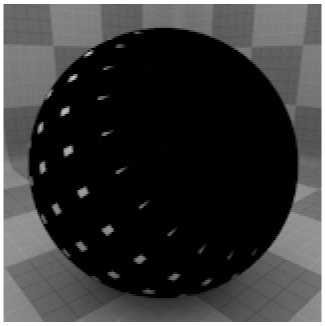
Transition Width 50%



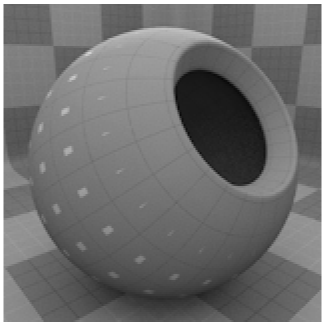
Transition Width 100%





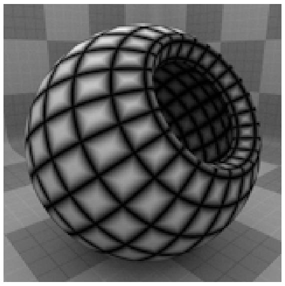


Bias 0%



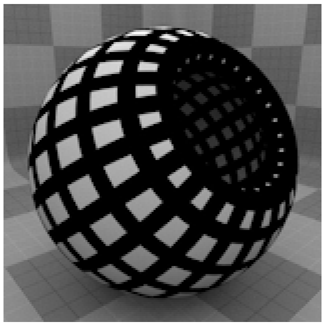
Gain 0%

Bias 50%



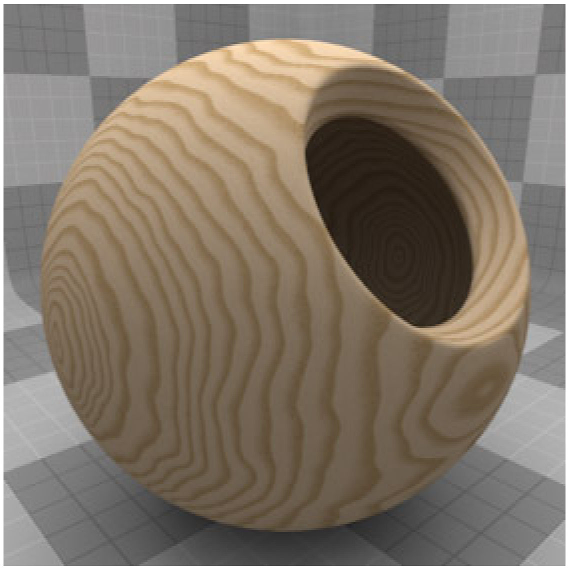
Gain 50%

Bias 100%



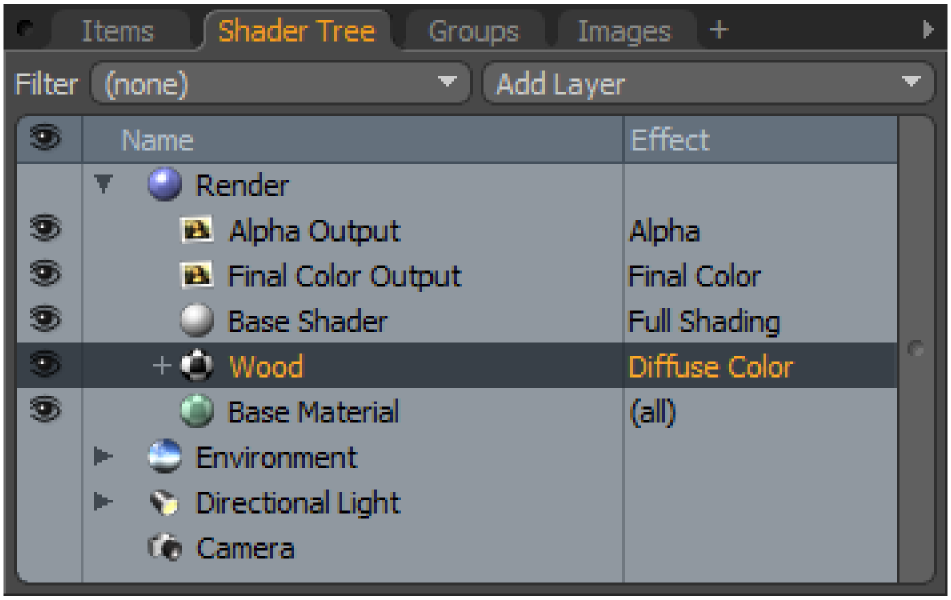
Gain 100%

Wood Procedural



The **Wood** procedural texture is a powerful texture that is designed to simulate wood grain. Of course, you can use the **Wood** texture for effects other than wood. Modo creates procedural textures mathematically at render time and, therefore, they have no fixed resolution. You can magnify procedural textures almost infinitely with no visual loss in detail. This texture has rings, which are distorted by waves and a layer of noise, that simulate wood grain.

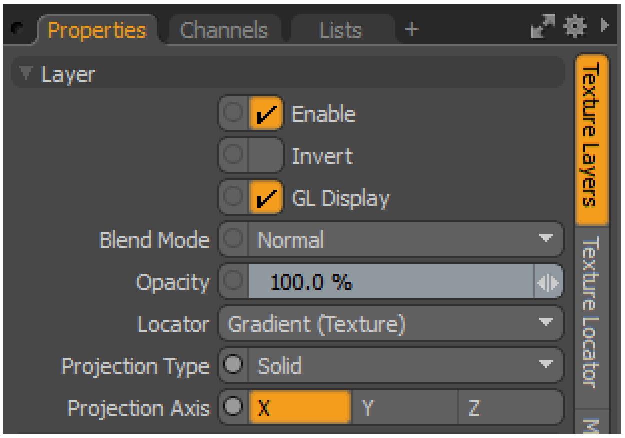
Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.


Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Wood	Diffuse Color
Base Material	(all)
Environment	
Directional Light	
Camera	

Layer Properties

>


Properties
Channels
Lists
+
Layer
Texture Layers
Texture Locator

☑ Enable
☐ Invert
☑ GL Display
Blend Mode: Normal
Opacity: 100.0 %
Locator: Gradient (Texture)
Projection Type: Solid
Projection Axis: X Y Z

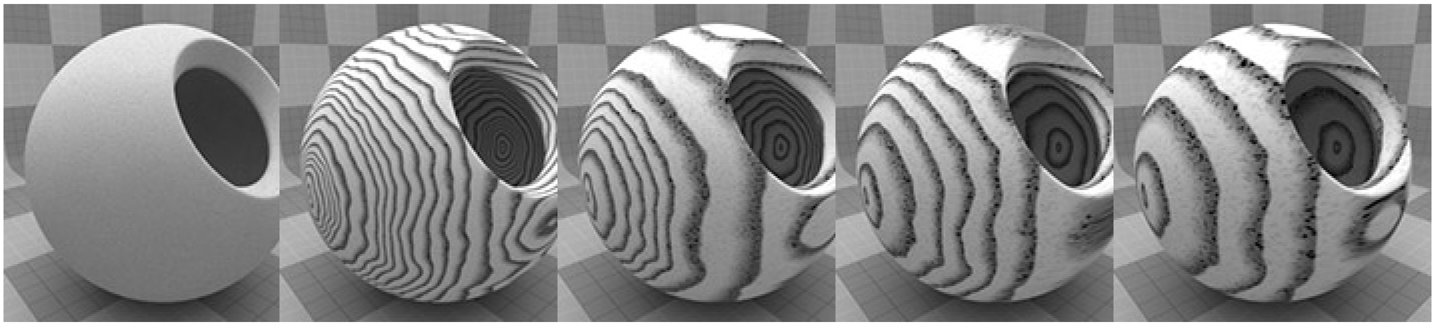
Enable	Toggles the effect of the layer on and off. This duplicates the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a negative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects.

	For more about blending, see Layer Blend Modes.
Opacity	Changes the transparency of the current layer. If there are layers below this layer in the **Shader Tree**, reducing this value increasingly reveals the lower layers. Reducing the value always dims the effect of the layer.
Locator	Sets the association for the Texture Locator. Most texture layers have a **Texture Locator** that Modo automatically creates in the **Item List**. This defines the mapping of the texture (how Modo applies the texture) to the surface. You can specify alternate locators, but this is normally not required. Although you may want multiple texture items to share a single locator.
Projection Type	Defines how a texture/material is applied to a 3D surface. Types vary significantly in their effects. For a guide to each **Projection Type** see Projection Type Samples.
Projection Axis	The texture/material is projected down this axis. This applies to **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and, **Light Probe** projection types.

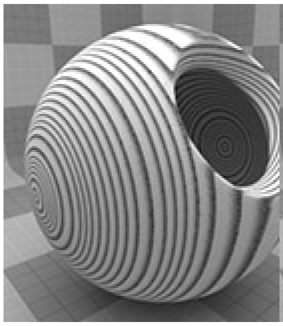
Wood Properties

Wood	
Ring Value	Determines the magnitude of the texture where the ring pattern occurs. This value ramps to the **Filler Value**. This is only available when the texture layer is set to a non-color attribute.
Ring Color	Specifies the RGB color value of the texture where the ring pattern occurs. This color ramps to the **Filler Color**. This is only available when the texture layer is set to an effect that requires color.
Alpha	The opacity of the **Ring Color**. **Note:** You can adjust the **Alpha** value using the **Color Picker**.
Filler Value	Determines the magnitude of the texture where the filler pattern occurs. This is only available when the texture layer is set to a non-color attribute.
Filler Color	Specifies the RGB color value of the texture where the filler pattern occurs. This is only available when the texture layer is set to an effect that requires color.

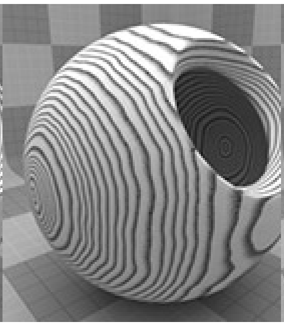
Alpha	Determines how transparent the **Filler Color** is.
Ring Scale	Determines the size of the rings in the texture. Click to view examples of various Ring Scale settings.
Waviness	Controls the amount of distortion that Modo applies to the rings in the texture. Click to view examples of various Waviness settings.
Wave Scale	Sets the size of the wave distortion that deforms the rings in the texture. Click to view examples of various Wave Scale settings.
Graininess	Determines how visible the grain effect is when it is combined with the ring texture. Decreasing this value creates a more subtle grain appearance. Click to view examples of various Graininess settings.
Grain Scale	Determines how large the grain spots are. Click to view examples of various Grain Scale settings.
Bias	Determines whether the texture favors the **Ring Color** (or **Ring Value**) over the **Filler Color** (or **Filler Value**). Increasing this value causes the texture to favor the **Ring Color** (or **Ring Value**); decreasing the value favors the **Filler Color** (or **Filler Value**). Click to view examples of various Bias settings.
Gain	Determines the falloff effect for the texture. This is similar to a gamma control that affects the falloff of the gradient ramp between the ring and filler values. Setting the **Gain** to **100%** creates a sharp falloff effect; setting the value to **0%** creates a plateau around the value (or color mid-point) with a sharp falloff at either gradient extreme. Click to view examples of various Gain settings.



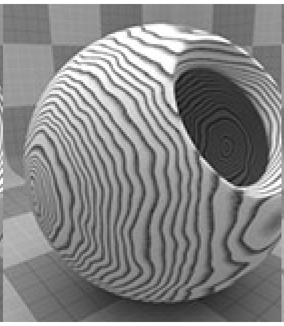
Ring Scale 0.1%   Ring Scale 25%   Ring Scale 50%   Ring Scale 75%   Ring Scale 100%



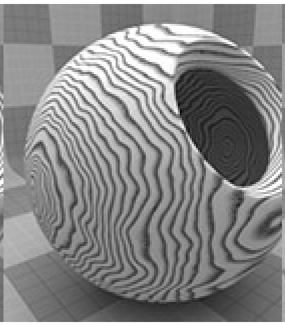
Waviness 0%



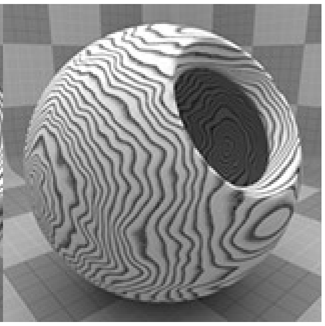
Waviness 25%



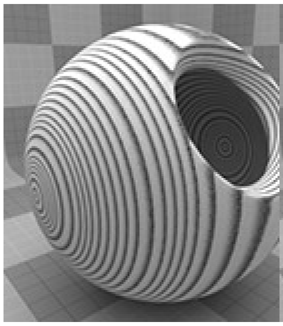
Waviness 50%



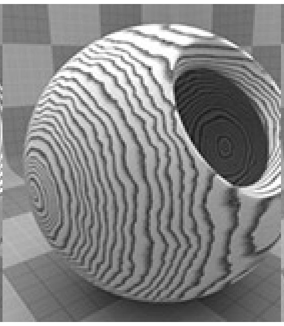
Waviness 75%



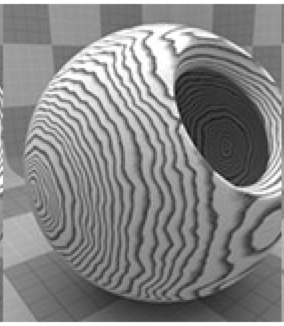
Waviness 100%



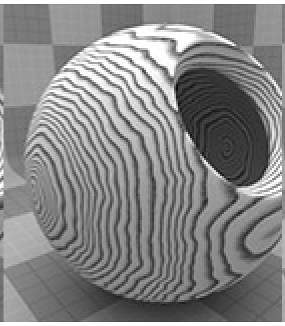
Wave Scale 0.1%



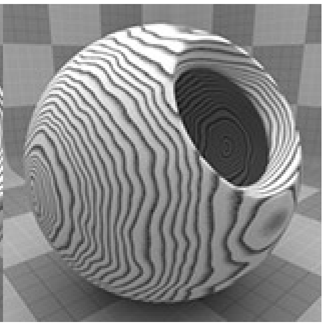
Wave Scale 25%



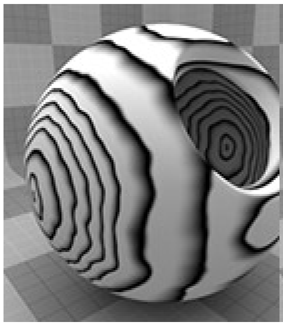
Wave Scale 50%



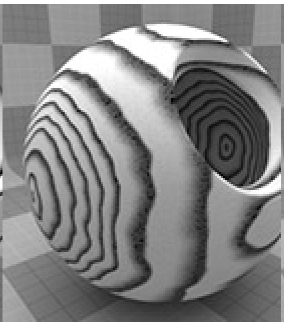
Wave Scale 75%



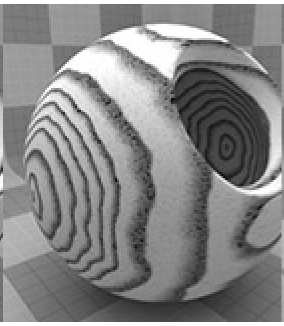
Wave Scale 100%



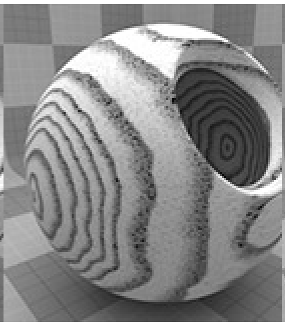
Graininess 0%



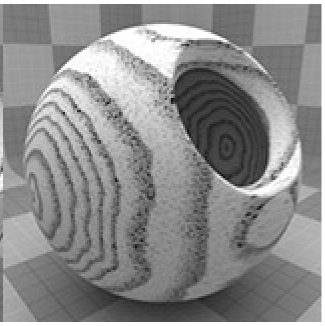
Graininess 25%



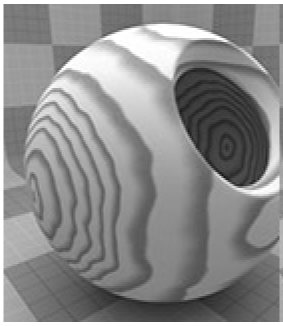
Graininess 50%



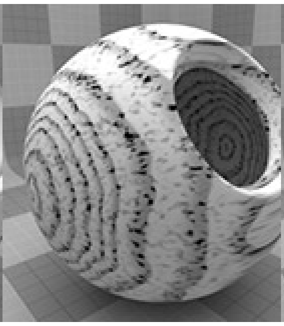
Graininess 75%



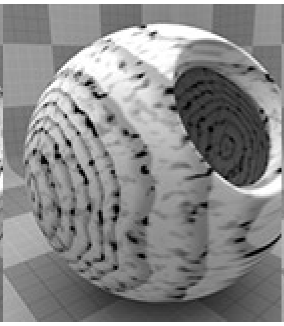
Graininess 100%



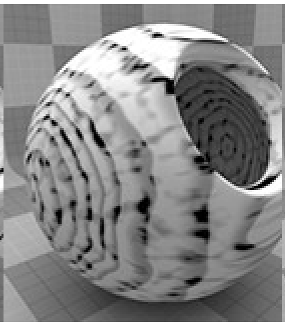
Grain Scale 0.1%



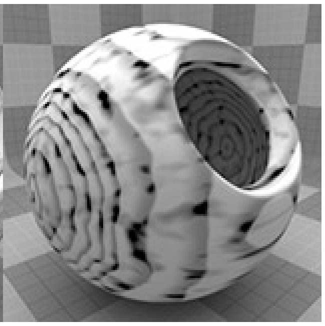
Grain Scale 25%



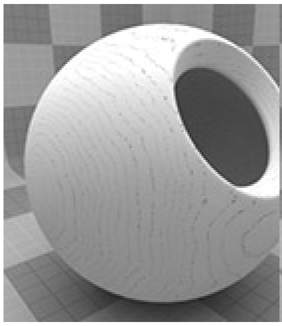
Grain Scale 50%



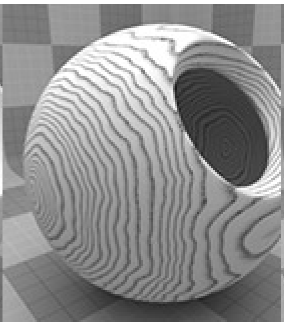
Grain Scale 75%



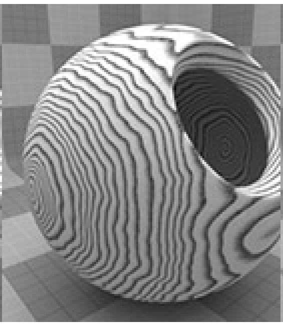
Grain Scale 100%



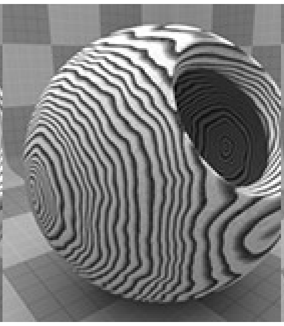
Bias 0%



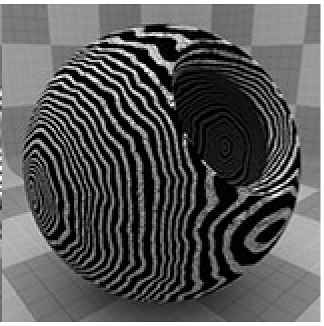
Bias 25%



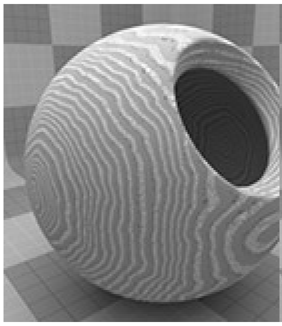
Bias 50%



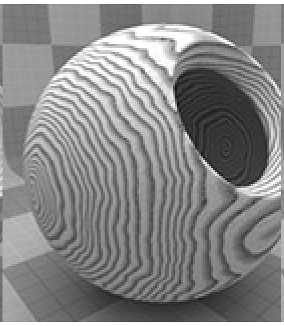
Bias 75%



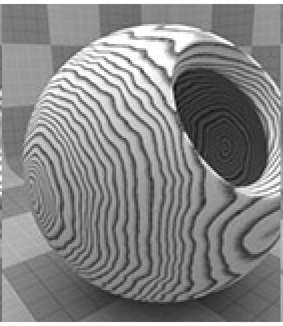
Bias 100%



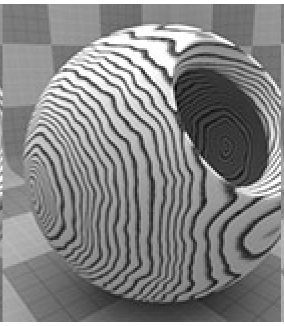
Gain 0%



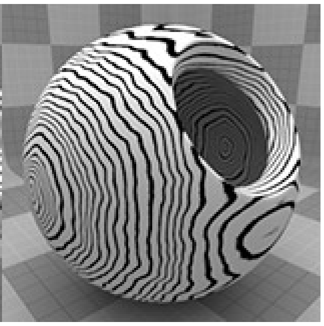
Gain 25%



Gain 50%



Gain 75%



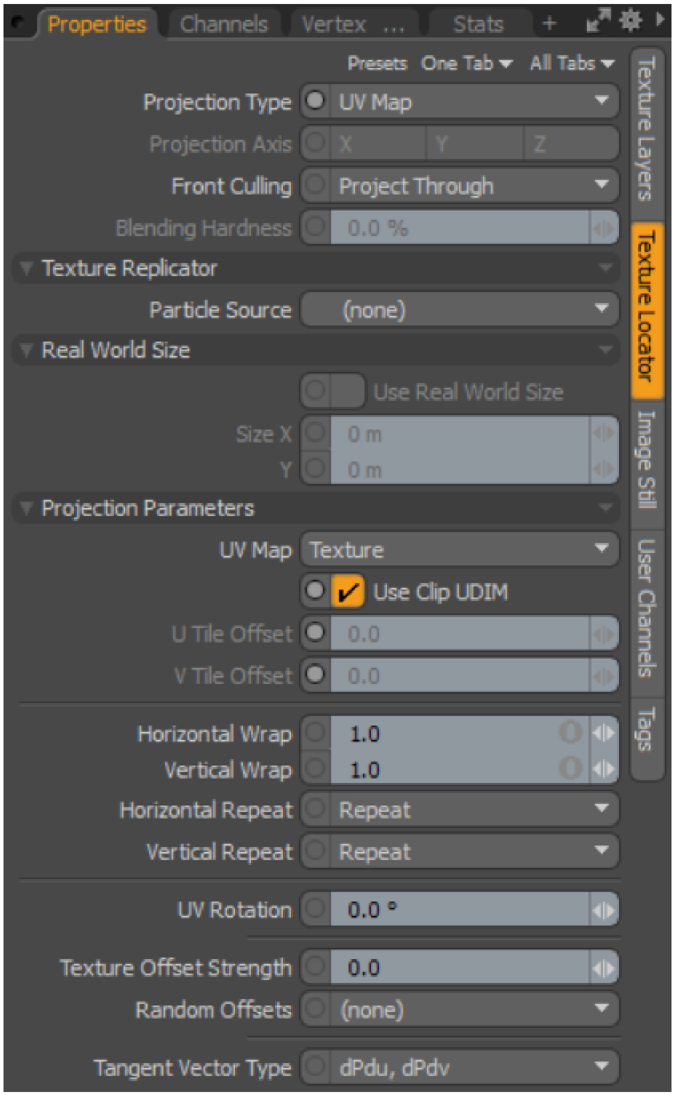
Gain 100%

Texture Locator

When any texture layer is created in the Shader Tree, image-based or procedural, an associated **Texture Locator** item is automatically generated and added to the **Items** list and linked by the texture layer's **Locator** property. By default, texture locators are generally added to a folder (really a group locator item) called a **Texture Group**. This makes it easy to toggle the visibility of the locators in the 3D Viewports, as you might find them distracting. You may select the Texture Locator directly in the **Items** list, or by selecting the locator icon in any 3D viewport. Most often though, you select them in the **Shader Tree**. When any texture layer is selected in the tree, the Locator is available in the named sub-tab on the right side of the item's associated **Properties** panel. Additionally, the locator may be toggled by the small **+** sign preceding the texture layer's name in the **Shader Tree**; clicking the **+** button reveals the layer.

Texture Locators provide a very important function, as they control the application of the texture layer over the rendered surface. A Texture Locator is like a regular Locator Item, in that it has **Position**, **Rotation**, and **Scale** information. Additionally, there are settings specific to controlling the image's projection method, meaning how it is mapped or applied to the surface. For certain projection modes, such as **Planar**, you can interactively position textures in the 3D viewport by transforming the Texture Locator item itself. Other modes, UV in particular, completely ignore the positional and scale attributes, as the mapping is defined by the UV map itself.

Texture Locator Properties


Properties
Channels
Vertex ...
Stats
Presets
One Tab
All Tabs
Texture Layers
Texture Locator
Image Still
User Channels
Tags

Projection Type: UV Map
Projection Axis: X Y Z
Front Culling: Project Through
Blending Hardness: 0.0 %
Texture Replicator
Particle Source: (none)
Real World Size
☐ Use Real World Size
Size X: 0 m
Y: 0 m
Projection Parameters
UV Map: Texture
☑ Use Clip UDIM
U Tile Offset: 0.0
V Tile Offset: 0.0
Horizontal Wrap: 1.0
Vertical Wrap: 1.0
Horizontal Repeat: Repeat
Vertical Repeat: Repeat
UV Rotation: 0.0 °
Texture Offset Strength: 0.0
Random Offsets: (none)
Tangent Vector Type: dPdu, dPdv

Projection Type	A projection defines how a texture, be it image-based or procedural, is applied to a dimensional 3D surface. The various projection types can greatly affect the final rendered outcome. Modo offers a variety of projection types (illustrated in Projection Type Samples).
Projection Axis	Defines an axial direction for any of the primitive projection types: **Planar**, **Cylindrical**, **Spherical**, **Cubic**, **Box**, and **Light Probe**.

Front Culling	Creates a texture shadow that only maps the parts of the geometry directly seen by the camera. The following options are available: • **Project Through** - Projects the texture on the back faces of the mesh. • **Cull Backfacing** - Projects the texture only on the front of the mesh. • **Flip Backfacing** - Projects and flips the texture on the back of the mesh along the U and V directions and appears upside down. • **Flip U Backfacing** - Projects and flips the texture on the back of the mesh along the U direction only and appears upside down. Flip V Backfacing - Projects and flips the texture on the back of the mesh along the V direction only and appears upside down.
Blending Hardness	When the **Projection Type** is set to **Triplanar Projection**, blends the planar projections at the location of seams to create the illusion of a single seamless texture across the entire surface. The default, 0%, is a soft blend off all three projections and 100% is a hard blend or zero blending of the planes.

Texture Replicator Properties

Particle Source	You can specify a Mesh item layer as a point source for the cloned textures, when doing so, the **Texture Replicator** option is effectively enabled and its settings supersede all others of the **Texture Locator**. Once a mesh layer is defined, a single copy of the texture is positioned and aligned to the vertices normal at each vertex point. Typically, you would specify the mesh layer of the actual target surface with this option. However, interesting effects can also be obtained when specifying alternate mesh layers. Keep in mind that the clones are aligned to the surface normal of the source mesh, so surfaces would need to be in somewhat close proximity and need to have similarly facing normals.

Real World Size

Use	Enable to use the real world size for this texture. Some textures, such as AxF material textures,

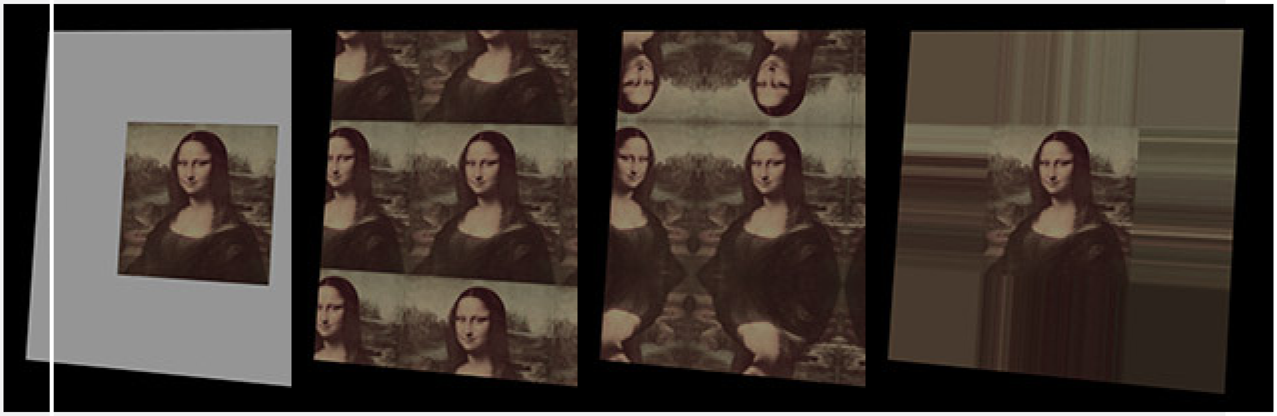
Real World Size	are imported with a real world size, which is set automatically in the **Size X** and **Y** fields. You can also specify the real world size of the texture manually using the **Size X** and **Y** properties. **Note:** You must have real world size assigned to the selected UV map in order to enable this property. See Adding Real World Size to UV Maps for more information. **Tip:** Modo's USD import functions also support UV tiling **Real World Size** information in the Texture Locator tool's properties viewport.
Size X	The real world size of the texture in millimeters or meters along the X axis.
Y	The real world size of the texture in millimeters or meters along the Y axis.

Transform Properties

Position	An **Item** transform that allows you to numerically position the texture locator item in XYZ space, thus defining the position of the projected texture. This position defines the center of the texture's overall volume as defined by the **Size** setting. Useful in the positioning of textures when using standard projections such as **Planar**, **Cylindrical,** and **Cubic**, among others. You can automatically define a texture's **Position** and **Size** using the **Auto Size** function.
Rotation	An **Item** transform that allows you to numerically set the rotation of the texture volume. The **Rotation** transform originates from the locator item's position value.
Order	Sets the order that rotations are applied to the Locator Item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock when animating.
Size	Sets the textures projection size (or boundary, as the case may be), defining the texture's overall volume. You can automatically define a texture's **Position** and **Size** using the **Auto Size** function.
Auto Save	Automatically sets the **Position** and **Size** values, and is determined by the

	overall bounding box volume of the tagged surface(s).
World Transforms	Typically, if you move a texture locator around, the texture moves over a surface that is sitting still. This allows for animated textures. If you parent the texture locator to the item and move the item around, both are moving in sync, so the texture stays still. With **World Transforms** enabled, the parented texture animates over the surface, as if the surface was sitting still and the locator was moving by itself. It's a subtle difference to **World Coordinates**.
World Coordinates	Typically, if you move an item, the texture moves with it, as if painted on. When **World Coordinates** is enabled, textures act as if frozen in place, with surfaces swimming through the texture. This option is extremely useful for texturing cloned or replicated items with procedural textures (that are NOT animated) as it provides the look of randomizing the texturing, without the need to modify any other surface attributes.

Projection Parameters Properties

Horizontal/Vertical Repeat	Sets how bitmap images repeat beyond the defined image size/UV space boundary. Several methods can be chosen: **Reset** disables image tiling, **Repeat** tiles repeatedly, **Mirror** tiles the image, flipping each subsequent image to produce a backward and forwards type tiling, and **Edge** simply repeats the edge row of pixels. You should note that these options are not visible in the GL Viewports.  Reset Repeat Mirror Edge
UV Rotation	Rotates the image plane relative to the UV space, eliminating the need to rotate the UV

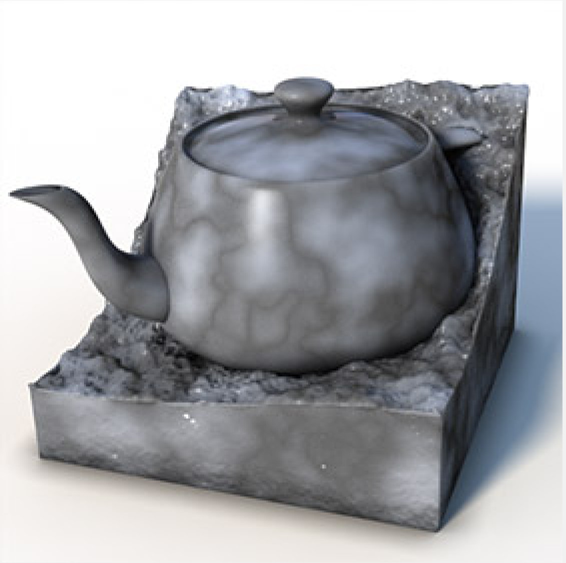
	mapping values manually. Rotation revolves around the 0,0 UV origin.
Texture Offset Strength	Determines the global strength of both the **Random Offsets** option, when one is defined, and for the layer effect **Texture Offset**. A value of '0.0' produces no offset, a value of '10.0' is offset up to 100% of the original location. When using both the **Random Offsets** and a **Texture Offset Strength** effect, the offset effect can be controlled independently by changing the texture layers opacity.
Random Offsets	Controls the relative positions of bitmapped textures on a surface based on the designated criteria. Working best with tiling textures, where there would be no distinct seam across edges, this option helps to hide or eliminate repetition by moving the texture a random direction across the surface. The amount of offset is determined by the **Texture Offset Strength** option. The different offset criteria can be based on: • **(none)** - disables the **Random Offset** feature when selected. • **Particle and Replicas** - offsets are generated per particle and/or per **Replicator** item. • **Mesh Parts** - offsets are generated based on individual mesh parts: multiple, individual geometric objects within a single item layer. • **Surfaces** - offsets are based off discreet sections of a single, named surface across continuous geometry.
Tangent Vector Type	Determines the tangent vector basis for an associated image map set as the surface normal, affecting both the baking and the display of normal maps. Normal maps are generally baked from high resolution geometry onto low resolution geometry to give the impression of higher resolution. There are two options available: **dPdu dPdv** defines a tangent space normal, the **dPdu Cross Product** option defines a bi-tangent space normal that is compatible with many game engines. When baking image-based normal maps, you need to define the proper vector basis prior to actually baking the image out for the proper results.

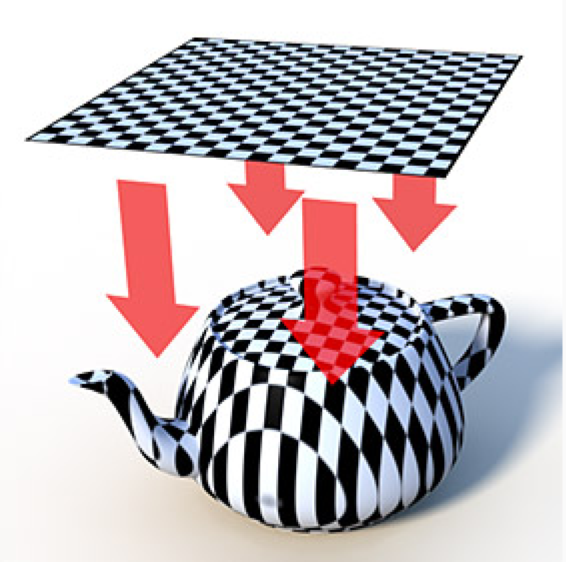
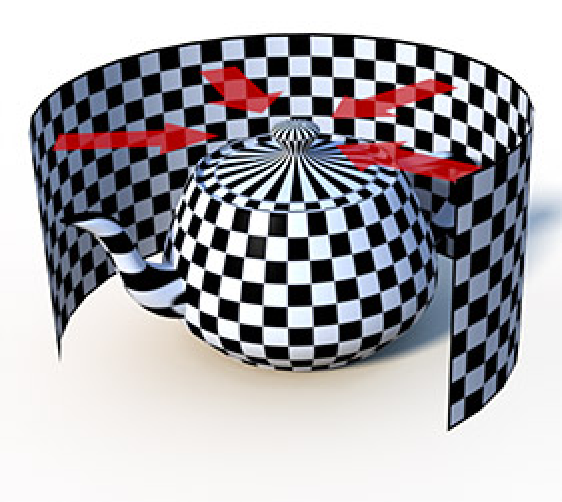
Falloff Properties

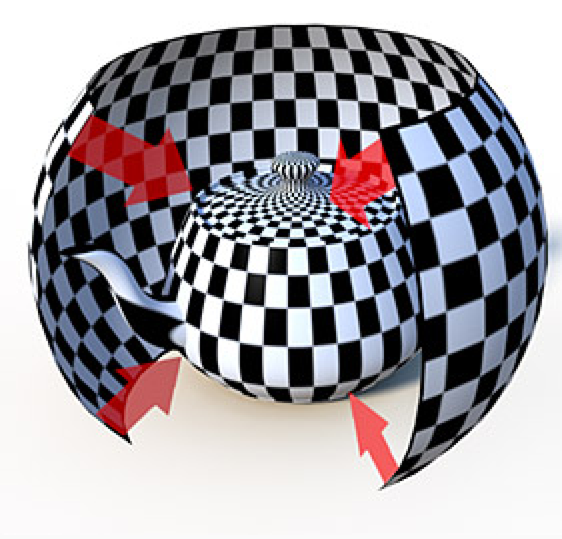
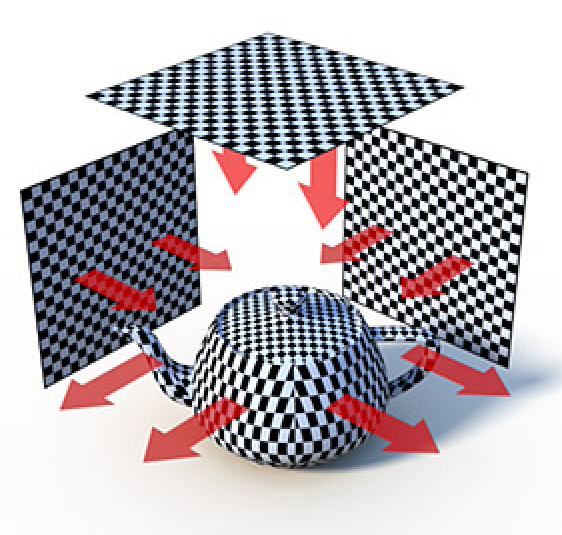
Falloff Type	Fades the texture layers opacity over the specified distance. There is cubic (3D), spherical (3D), and Linear X, Y, or Z. For more information, see Using Falloffs.

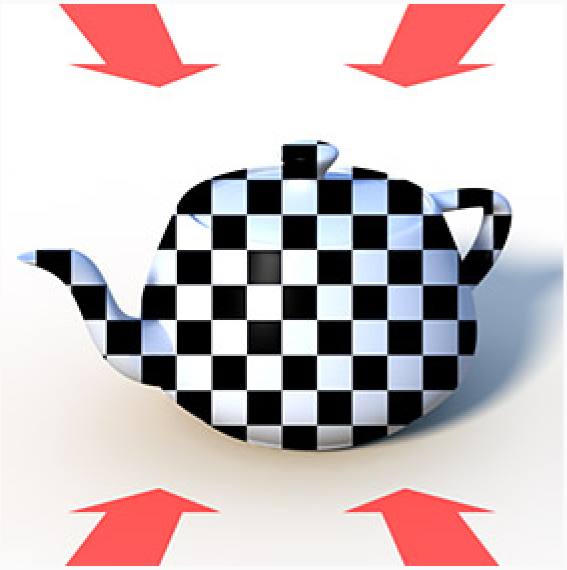
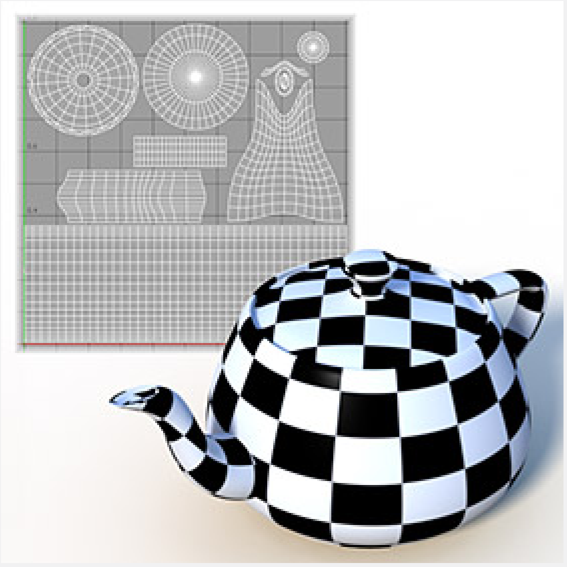
Falloff X/Y/Z	A setting of '0' disables falloff. The falloff amount act like a multiplier of the size. So a setting of 0.5 on a 1 meter texture scale fade the texture to fully transparent over 2 meters from the locators position, a setting of 0.1 would scale it to fully transparent at 10 meters.

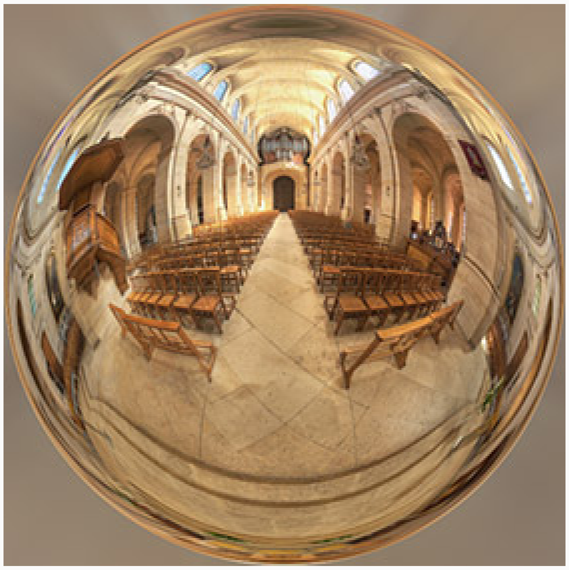
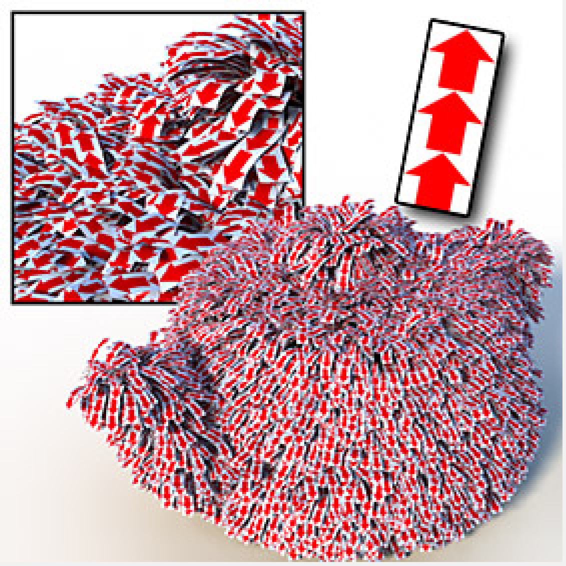
Projection Type Samples

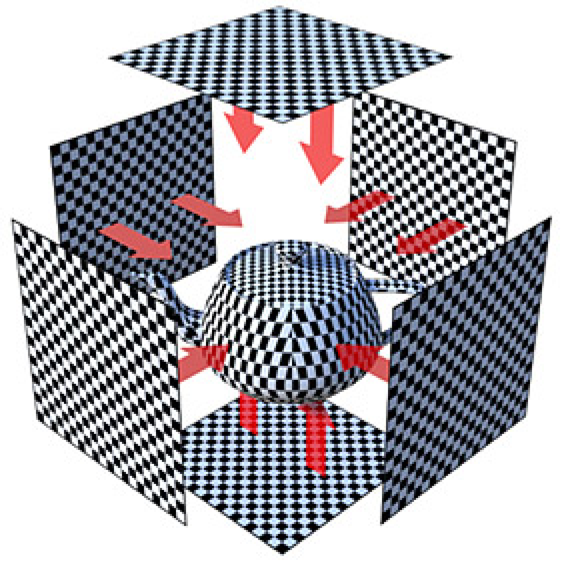
Projection Type Sample	Description
	**Solid** - a **Solid** projection is only applicable for procedural texture layers. If you think of a procedural texture as a solid volume, this projection type renders a particular value at the intersection point of the surface, within the virtual volume of the procedural. A block of marble is a good example: the surface look as if it were carved from the procedural volume revealing the surface.

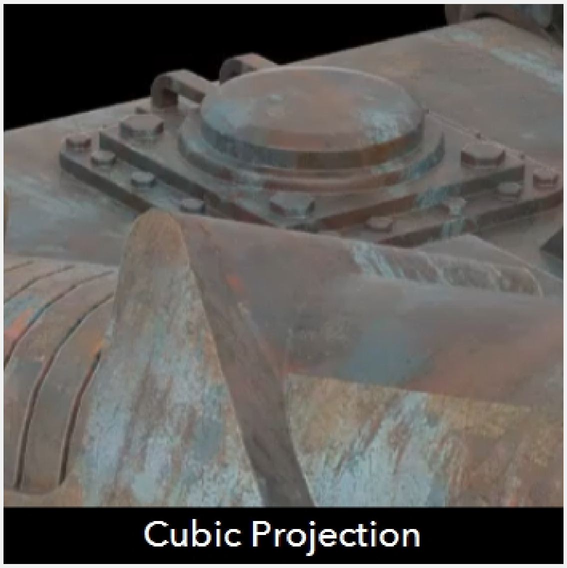
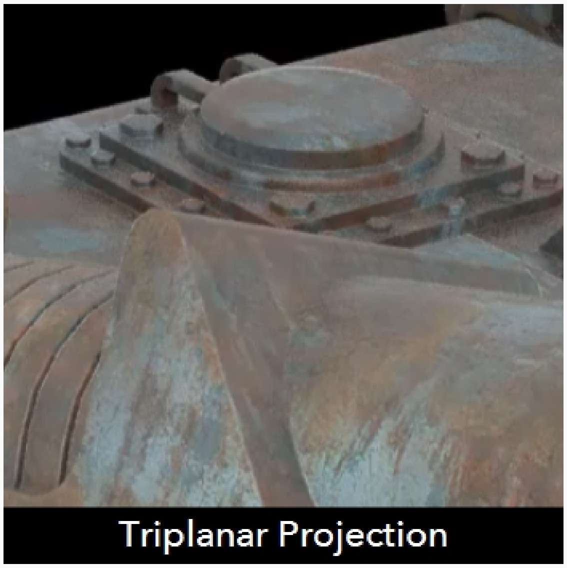
Projection Type Sample	Description
	**Planar** - a **Planar** projection is similar in concept to a movie projector, but the associated image is projected onto the surface orthographically, meaning the projection rays travel perpendicular from the virtual projection plane onto the surface. This projection type is great for flat or nearly flat surfaces. Other instances are likely to cause undesirable stretching of the texture. You can specify the axis direction, either X, Y, or Z, that you want the projection to face. **Note:** As of Modo 16.1, the performance of Planar Projections improved dramatically, allowing you to work with Planar Projections in the viewport without interruptions. For more information on Viewport Updates in Modo 16.1, see What's New in Modo 16.1.
	**Cylindrical** - for the **Cylindrical** projection, the texture image is warped into a cylindrical shape and projected onto the surface. Very useful in texturing cylindrical shapes, for example, when labeling on various cans and bottles. Surfaces perpendicular to the projection produce undesirable stretching of the texture as exhibited by the lid of the teapot. You can specify the axis direction for the cylinder along the X, Y, or Z axes.

Projection Type Sample	Description
	**Spherical** - for the **Spherical** projection, the texture image is warped into a spherical shape and projected onto a surface. This is useful in texturing round objects, for example, planets and sports balls. Spherical projections are also useful in mapping 360° panoramic images for use as environment maps. Surfaces perpendicular to the projection produce undesirable stretching of the texture, as exhibited by the spout of the teapot. You can specify the axis direction for the sphere along the X, Y, or Z axes.
	**Cubic** - for the **Cubic** projection, the texture is planar-projected through a surface from all three axial directions, X, Y, and Z. A polygon receives a certain projection, based on its normal direction. This projection type is best on cube-shaped objects, and occasionally on detailed surfaces where texture seams are not of great concern. The main drawback is for textures that have a proper orientation are flipped on reverse faces. This is resolved by using the **Box** projection method.

Projection Type Sample	Description
	**Front Projection** - a special projection type, useful in composing 3D elements into a photograph. A **Front Projected** image (also called a plate) is projected out from a camera or light's position (as specified by the **Projection Camera** setting) often onto low resolution proxy geometry, or as a background environment. If the camera's position and view settings are set appropriately, rendered elements can be made to match with elements of the projected background.
	**UV Map** - UV mapping is a technique where you can precisely plot how a 2D flat image is applied onto a 3D surface by defining a 'U' horizontal and 'V' vertical position for each polygon vertex. Many primitive objects are created with automatic UV maps. Otherwise, UV maps can be easily created using any of the various UV tools available in the **Texture** > **UV Tools** menu. For surfaces with a UV map defined, this is the default projection method.

Projection Type Sample	Description
	**Light Probe** - Light probe images are images used to light scenes by way of image-based lighting (refer to Image Based Lighting for more information). They are typically high dynamic range imagery (HDRI) that, when applied to a scene as an **Environment** (and when Global Illumination is enabled), brightness information embedded in the image is used to light the scene. This produces very realistic and natural results. A popular method of of generating these HDRI images is based on taking multiple photographs of a reflective sphere. The **Light Probe** projection properly unwraps this chrome-sphere style of HDRI image, projecting it out as a proper spherical background.
	**Implicit UV** - Implicit UVs are procedurally-generated implied UVs, used to apply image maps onto fur material fibers individually. Every fiber, when set as **Strips**, fill the 0-1 UV space, allowing appropriate images to be mapped onto the fibers. This provides a means to make incredibly realistic grass and feathers, among other things.

Projection Type Sample	Description
	**Box** - similar to **Cubic** projection, but the **Box** projection type projects a texture in a planar fashion from all six directions (like an inward facing cube), eliminating reverse projections on rear facing polygons inherent to the **Cubic** method. This projection type is best on cube-shaped objects, and occasionally on detailed surfaces where texture seams are not of great concern.

Projection Type Sample	Description
	**Triplanar Projection** - combines planar projections together in a similar way to **Cubic** and **Box**, but blends the projections according to the **Blending Hardness** control to cover the whole object. Triplanar projection can be used to project repeating noise and patterns, such as rust or tiles, without creating visible seams. The first image on the left shows **Cubic** projection creating a seam between rust textures. The second image shows **Triplanar** projection blending the seam between the textures.
	

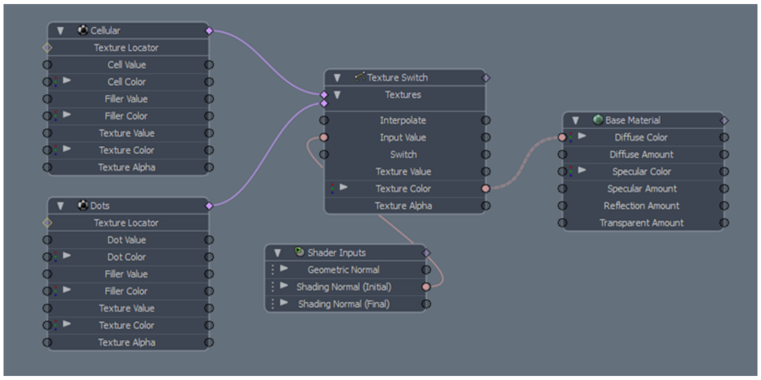
Texture Switch Node

The Texture Switch node allows you to dynamically choose between multiple textures based on an **Input Value**. The input textures' connections are ordered from bottom to top, so that when **Interpolate** is on, only adjacent neighbors are interpolated. The **Input Value** is also wrapped, meaning that the last texture is interpolated to the first, and negative values are interpolated in the reverse direction.

Usage

You can add the Texture Switch node to the **Schematic** viewport using the **Add** function, found in the pop-up menu under **Shader Inputs** > **Texture Switch**. The textures are connected to the node by way of their relationship connection (rather than a texture output channel connected to an input). An input must also be connected to the **Input Parameter** to determine the switch criteria. This can be a Particle ID feature when used with Replicators (and obviously a particle source of some type) or switched on ray types or surface features, such as normal. For information on working with node graphs, see the Schematic Viewport topic.

Schematic Node: Texture Switch

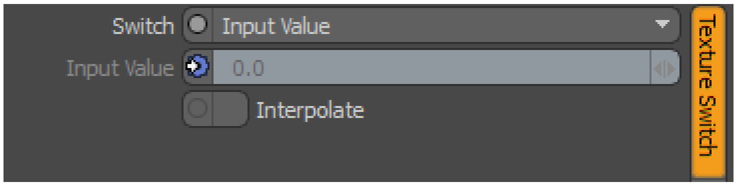

Cellular
Texture Locator
Cell Value
Cell Color
Filler Value
Filler Color
Texture Value
Texture Color
Texture Alpha
Dots
Texture Locator
Dot Value
Dot Color
Filler Value
Filler Color
Texture Value
Texture Color
Texture Alpha
Texture Switch
Textures
Interpolate
Input Value
Switch
Texture Value
Texture Color
Texture Alpha
Shader Inputs
Geometric Normal
Shading Normal (Initial)
Shading Normal (Final)
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount

- **Textures**: This parameter receives the various relationship inputs from the source nodes. The input texture's connections are ordered from bottom to top, so that when **Interpolate** is on, only directly adjacent neighbors are interpolated. The **Input Value** is also wrapped, meaning that the last texture is interpolated to the first, and negative values are interpolated in the reverse direction.

  In addition to linking textures as inputs, you can also link groups, and any textures contained within the group is used.
- **Interpolate**: When disabled, only a single input is expressed in the result, when enabled it acts as a blend between the various inputs.
- **Input Value**: This is used for the switching criteria. Input Value is only used if **Switch** is set to **Input Value**. You can drive the **Input Value** any way you like, for example, you can drive it with a gradient to control the weighting of the texture choice.
- **Switch**: This specifies the parameter that is to be used by the switching criteria. By default, it uses the **Input Value** channel but it can also be set to **Particle ID** or **Surface ID**. If the **Switch** is set to an ID then it uses the ID value from the surface directly and **Input Value** has no effect. This provides a quick way of using the ID. The range of the input is evenly split according to the number of texture inputs.

Particle ID is expressed as a random value between 0 and 1. This can be used when applying surfacing to a group of replicators. For two inputs this would show the first texture for any IDs between 0.0 - 0.5, and the second texture for any ID between 0.5 - 1.0. For a surface normal input the value switches based on the angle of incidence relative to the camera.

The image below shows the detail from the **Properties** panel for the **Texture Switch** node in the diagram above.

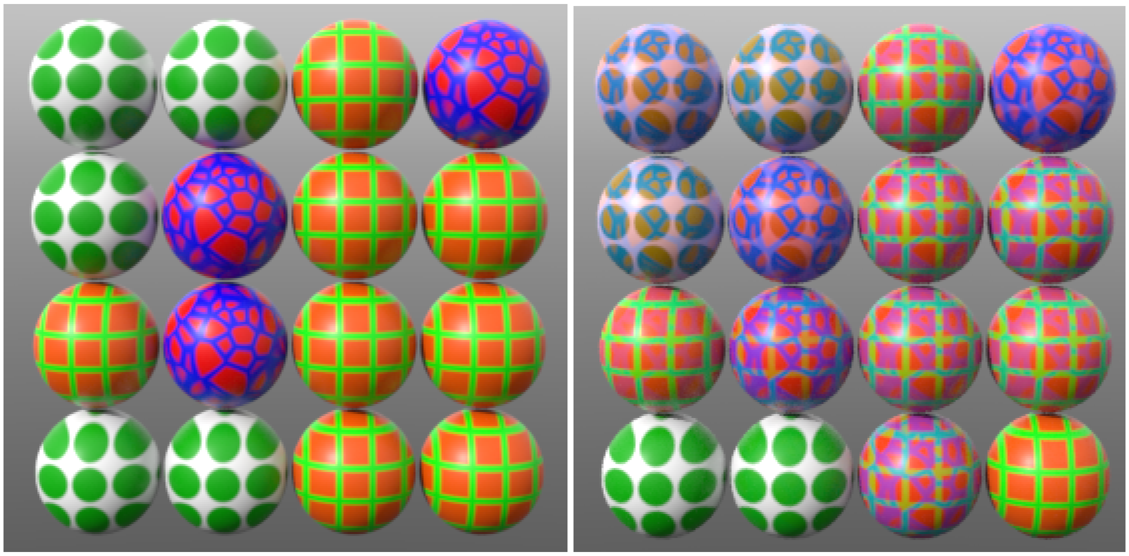


Switch ○ Input Value

Input Value 0.0

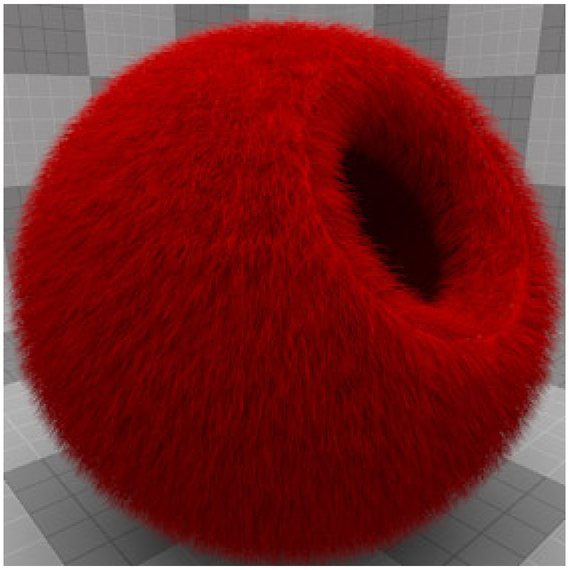
○ Interpolate

Example



In this example, a sphere is cloned using a Replicator and a particle source. The Texture Switch node is output to the surface's **Diffuse Color** channel and chooses between three possible texture inputs: Grid, Cellular, or Dots. **Switch** is set to **Particle ID**, which is a random value between **0** and **1**. In the image on the left, with **Interpolate** off, we see that replicas with ID's between **0** and **1/3** use the **Dots** texture, between **1/3** and **2/3** use the **Cellular** texture, and between **2/3** and **1** get the Grid texture. In the image on the right, with **Interpolate** on, we get a blend of the adjacent textures.

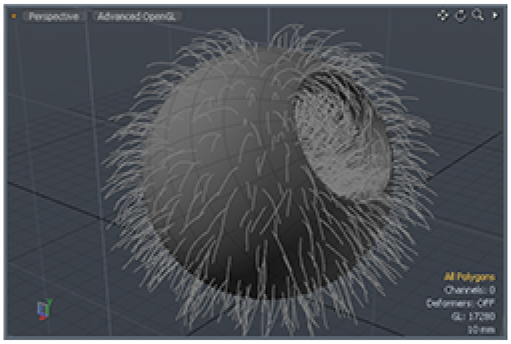
Fur Material



Endless fields of well-manicured lawns, small woodland creatures, shag carpeting, and werewolves - what do these all have in common? They are all possible with Modo's fur material. This flexible texture layer covers surfaces with various fuzzy, frazzled, fluffy, or fibrous fur. The provided functions have a range of creative options. (For example, you can create tame or wild lawns, all manner of fibrous carpeting, little fuzzy animals, or a conifer with pine needles.) Modo renders all of these fibers with amazing speed — even when using global illumination.

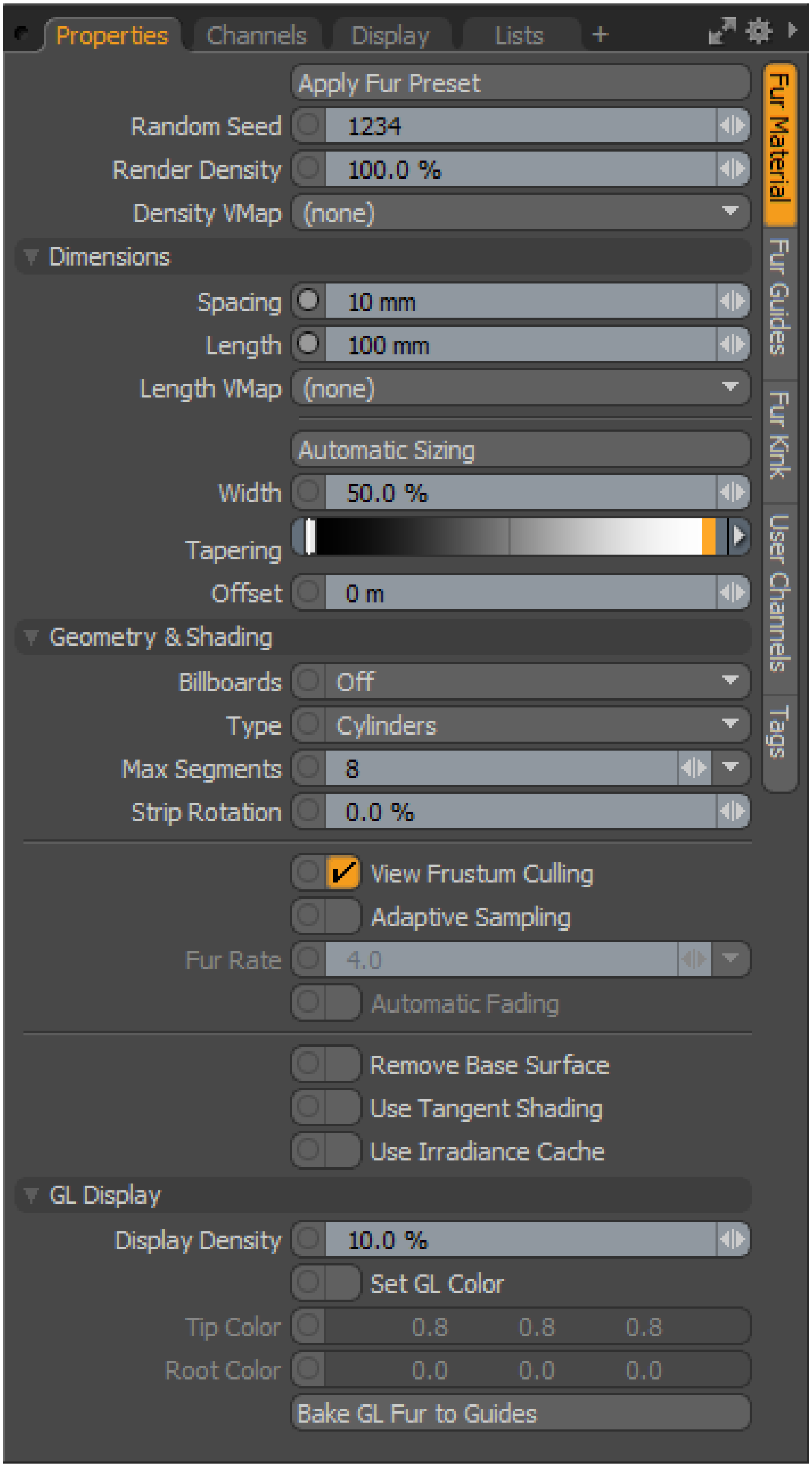
To apply fur to a surface, start by adding a **Fur Material** item to the target surface in the **Surface** viewport.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic. Also, view the Working with Fur topic that provides additional information about using fur in Modo.



Once you add a fur layer, lines now extend from the target object (or surface). These lines may be long or short depending on the object's relative scale. You can't select these lines directly because they're only a rough preview to represent the fibers now growing from the surface. As you make changes in the material properties, these lines update in the viewport to provide feedback about the various parameters. There are many options available in the **Properties** panel, which are in multiple sub-tabs. In the **Fur Material** sub-tab, you specify the settings for the fibers' density, size, and shape as well as how they render. In the Fur Guides sub-tab, you adjust the fibers themselves — whether they are curly or straight, or wavy or limp. In the Fur Kink sub-tab, you specify the **Frizz** and **Kink** values. You can also control fur with guides that you can style with Modo's painting tools. (See the Working with Fur topic for details.)

Fur Material


Properties
Channels
Display
Lists
+
Fur Material
Fur Guides
Fur Kink
User Channels
Tags
Apply Fur Preset
Automatic Sizing
Bake GL Fur to Guides

Random Seed 1234
Render Density 100.0 %
Density VMap (none)

Dimensions

Spacing 10 mm
Length 100 mm
Length VMap (none)
Width 50.0 %
Tapering
Offset 0 m

Geometry & Shading

Billboards Off
Type Cylinders
Max Segments 8
Strip Rotation 0.0 %

☑ View Frustum Culling
☐ Adaptive Sampling
Fur Rate 4.0
☐ Automatic Fading

☐ Remove Base Surface
☐ Use Tangent Shading
☐ Use Irradiance Cache

GL Display

Display Density 10.0 %
☐ Set GL Color
Tip Color 0.8 0.8 0.8
Root Color 0.0 0.0 0.0

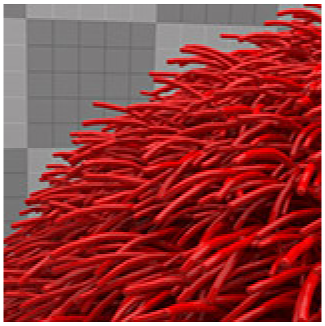
The **Fur Material** sub-tab has general-purpose options that can produce a wide variety of results. Knowing where to start to achieve a particular effect can daunting. Therefore, Modo provides a preset to get you started.

Apply Fur Preset	Opens a dialog for you to specify options — each of which is a variation on a way to use fur items. The selected preset includes general settings for the fur item, itself, that are specific to the named function. Depending on your use of the fur, you may need to make adjustments to the preset values, which you should consider a starting point to getting the best results for your use.
Random Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.
Render Density	Adjusts the density or overall number of fibers on the surface. Increasing this value increases the number of fibers to create a thicker mat. Lower values thins the number of overall fibers. This option only affects the number of fibers that Modo renders. Realize that increased numbers of fibers require more memory to calculate and subsequently take longer to render.
Density VMap	Specifies an embedded Weight Map on the target fur surface and uses its values to control the density (thickness) of the fur fibers across the surface. Doing this eliminates the need to create custom image maps. You can select any available Weight Map from this list to control the final density of the resulting fur. Weight Map values work as a multiplier over the normal rendered density. A value of **1** produces the unmodified 100% density and fades toward **0** to produce 0% density.
Dimensions	
Spacing	Sets the average space between individual fibers. Smaller values result in thicker, denser fur; larger values result in thinner fur. This setting works with the **Width** percentage to determine the thickness of the fiber, itself.
Length	Specifies the overall length of the fibers from the root to the tip.
Length VMap	Indicates the embedded Weight Map for the target fur surface and uses its values to control the length of the fur fibers across the surface. Using an existing Weight Map eliminates the need to create custom image maps. You can select any available Weight Map from this list. Weight Map values work as a multiplier over the normal rendered length. A value of **1** produces the unmodified **100%** length and fades toward **0** to produce a **0%** length.

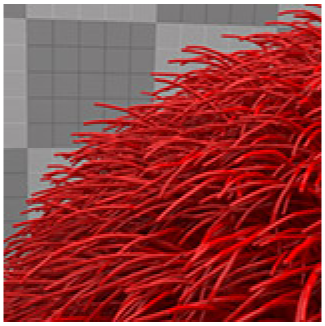
Automatic Sizing	Sets an initial value for the fur's spacing and length based on the size of the fur surface's bounding box. When you click this button, Modo calculates the size based on **0.5%** percent of the maximum extent of surface bounding box (the maximum length of the bounding box's diagonals). The fur length is then 10 times that value. This should only be considered a starting point.
Width	Specifies the width or thickness of the individual fibers at their root. This is a percentage of the **Spacing** value. For example, if you have **Spacing** set to **10 mm** and **Width** set to **10%**, then the fibers would be **1 mm** wide at the root. With **Width** set to **50%**, the fibers would be **5 mm** wide.
Tapering	Modifies the width of the individual fibers along their length. By using a mini-gradient input, you can modify the width with gradient keys that represent a percentage of the original fiber's width. Click to view examples of various Tapering settings. Much like how a keyframe defines a value in time, gradient keys represent values along a fiber's length for a fiber to thicken and narrow in width any number of times along its length. The gradient input's width represents the individual fiber's overall length with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black produces a fiber at its full, defined width; white tapers the width to **0%** (no width); and the shades of gray attenuate the fiber's width in between. Complex tapering input also requires additional geometry for Modo to render it correctly. Therefore, increasing the number of segments and increasing the amount of RAM required to render the scene may be necessary with complex tapering.
Offset	Pushes the root of each fiber away from the surface by the specified distance. Fibers retain their overall length.
Geometry & Shading	
Billboards	Specifies whether to set up billboard polygons by distributing flat polygon planes across a surface. When you map images to the planes, the planes look more complex than their shape suggests. By default, the **Billboard** setting is **Off** and has no effect on the fibers. When you select another option, that option provide a one-click setup of popular functions and overrides other settings such as **Type** and **Tapering**. • **Off** - Disables the **Billboard** option. • **Trees** - Replaces each fiber with a flat, upright, and simple polygon plane that

	always faces the current render camera. • **Leaves** - Replaces each fiber with a flat, simple polygon plane. • **Feathers** - Creates a multi-segmented plane that can be deformed by using the **Bend** options but, otherwise, is similar to the **Leaves** option. The root orientation remains perpidicular to the bending direction as well. To simulate feathers, apply a Vector Vmap to control the orientation of the feather bending with the Bend Direction set to **Normal**. You can adjust the **Bend Amount** and **Root Bend** options as needed.
Type	Specifies the method Modo uses to generate the fur fibers when **Billboards** is set to **(none)**. Click to view examples of the two Type options. • **Cylinders** - Creates tiny tubes for fur and hair. • **Strips** - Creates flat planes for making grass and similar flat fibers.
Max Segments	Smooths fibers. Just as a curved surface requires more polygons to produce a smooth surface, increasing **Max Segments** produces smoother curves for generated fibers. If you set this value too high, Modo requires more memory and processing time to calculate the fur layer. Generally, you can use fewer segments on fibers that bend only slightly; however, long or curly hair likely requires more segments to produce a smooth result. Click to view examples of various Max Segments settings.
Strip Rotation	Rotates individual fibers randomly at their root. This option is only effective when you select **Type > Strip**. Setting a higher value increases the amount of rotation applied (up to 180 degrees at **100%**). (You can also change the direction using the **Jitter Direction** option in the **Fur Shape** tab. **Strip Rotation** retains the same bend direction, but **Direction Jitter** does not.) Click to view examples of various Strip Rotation settings.
View Frustrum Culling	When enabled, removes the fur outside the view of the camera completely. Overall, this results in faster fur generation time and reduced memory usage. It is especially useful with large, grass-covered landscapes where much grass (or many fibers) are outside of the camera's view.
Adaptive Sampling	When enabled, decreases the density of a fur layer as it recedes from the camera (based on the **Fur Rate**). This reduces the overall memory for calculating the layer. To avoid 'popping' as fibers disappear, also enable **Automatic Fading**.
Fur Rate	Specifies the threshold that **Adaptive Sampling** uses to reduce fibers. Modo calculates this as an average distance, in pixels, between two fibers. Use **Adaptive Sampling** and **Fur Rate** with scenes such as grassy fields or any situation where

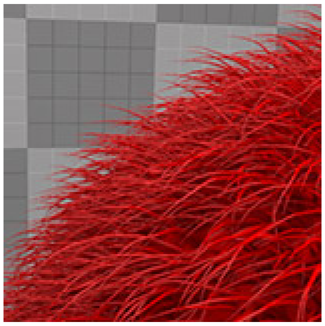
	large amounts of fibers recede into the distance.
Automatic Fading	Fades fibers in and out as Modo creates and destroys them based on **Adaptive Sampling**. This is ideal for animated sequences, but render times increase because semi-transparent geometry takes longer to render.
Remove Base Surface	When enabled, makes the base polygon invisible (similar to setting the **Dissolve Amount** to **100%**). Normally, fur takes its color and shading information from the surface material from which it grows. For example, you could create a green polygon with green fur growing from it. In this case, you likely want the grass shading to be different from the surface it's attached to. For more information, see the Working with Fur topic.
Use Tangent Shading	When enabled, rotates the specular shading normal toward the camera to create highlights that run parallel to the fur's direction. This provides a more realistic hair-type highlight.
Use Irradiance Cache	When enabled, uses irradiance caching instead of the default the Monte Carlo method. For rendering, the irradiance caching options are quite fast, but when rendering fur, they can be quite slow because of the number of curved surfaces in close proximity. Therefore, by default, for fur, Modo uses the Monte Carlo method for calculating global illumination to override the GI settings of the render item. In some special cases, you may want to use irradiance caching instead.
GL Display	
Display Density	Affects the number of preview fibers displayed in the 3D viewport. To speed up the on-screen display, you can view fewer fibers than Modo renders normally.
Set GL Color	When enabled, uses the **GL Color** option to set the color for the guide hairs visible in the 3D GL viewport. This is useful in helping to differentiate fibers from other scene elements by visually differentiating them from the rest of the scene.
Tip Color	Defines a color for each fiber's tip that blends with the **Root Color**. You can then see the difference between the roots and tips of the fur fibers.
Root Color	Defines a color for each fiber's root that blends with the **Tip Color**. You can then see the difference between the roots and tips of the fur fibers.
Bake GL Fur to Guides	Converts the visible guides within the 3D viewport into actual polyline guides that you can style and modify by using the brush-based styling tools. Click to covert the guides.



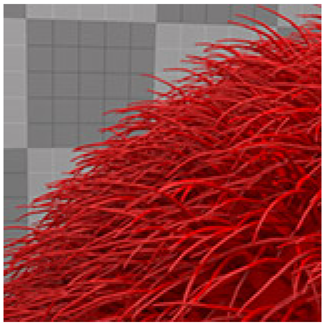
Linear Taper No Key



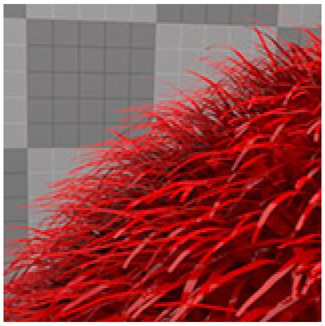
Linear Taper 50% key



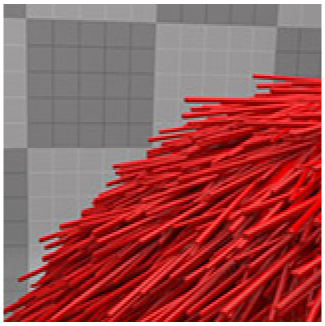
Linear Taper 100% key



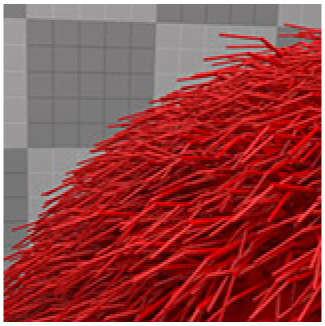
Type Cylinder



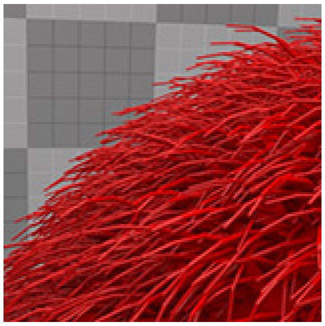
Type Strips



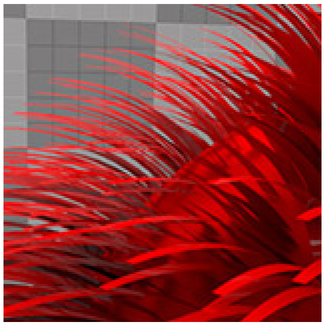
Segments 1



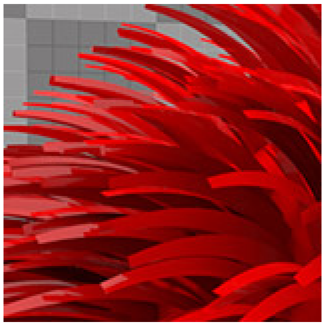
Segments 3



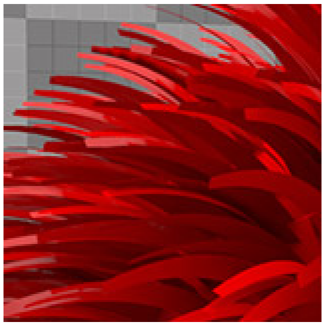
Segments 5



Strip Rotation 0%

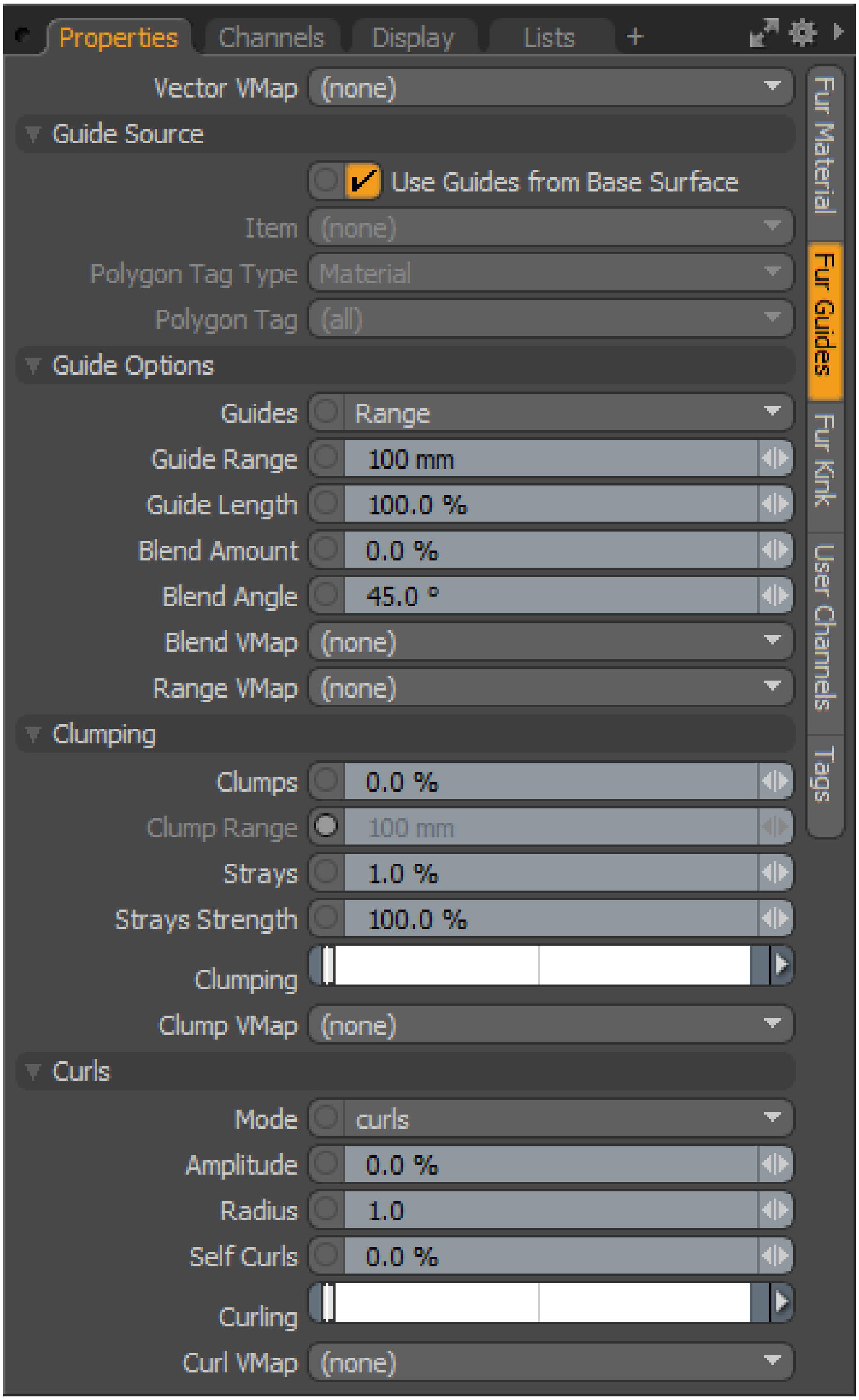


Strip Rotation 50%



Strip Rotation 100%

Fur Guides


Properties
Channels
Display
Lists
+
Fur Material
Fur Guides
Fur Kink
User Channels
Tags

Vector VMap (none)

Guide Source

☑ Use Guides from Base Surface

Item (none)

Polygon Tag Type Material

Polygon Tag (all)

Guide Options

Guides Range

Guide Range 100 mm

Guide Length 100.0 %

Blend Amount 0.0 %

Blend Angle 45.0 °

Blend VMap (none)

Range VMap (none)

Clumping

Clumps 0.0 %

Clump Range 100 mm

Strays 1.0 %

Strays Strength 100.0 %

Clumping

Clump VMap (none)

Curls

Mode curls

Amplitude 0.0 %

Radius 1.0

Self Curls 0.0 %

Curling

Curl VMap (none)

The **Fur Guides** sub-tab has options for adjusting the fibers themselves. In general, you use guides to adjust the shape of the fur fibers.

Also, on this sub-tab, you can specify guides to control styling (both combing and length) of fur by polygon tag and by object. Therefore, you can separate the guides from the surface on which hairs grow and can have a mesh with just the guides to control the hairs growing on another mesh. You can also tag the guides for each fur layer for more sophisticated hair styling. You can apply tags to guides by selecting the curves in Polygons mode and applying the tags similar to applying a standard polygon tag.

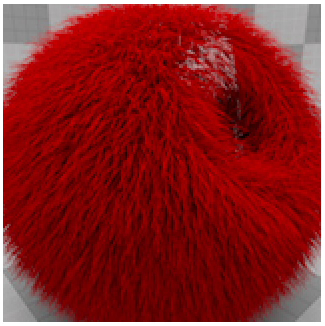
Option	Description
Vector VMap	Creates a vector-style Vertex Map to style the resulting fur and control both the length and orientation of the fur (as if combed). Vertex maps have benefits over using bitmap-based vector maps in that they become part of the object and, therefore, part of the scene. There is no external file to keep track of, and you can edit them using Modo's tools. Although a vector Vertex Map cannot match the resolution offered by image maps, for most fur-styling tasks, a vector Vertex Map is adequate. To create a vector Vertex Map, click **Vector VMap** and choose **New Map**. In the dialog, type a name (or accept the default name) and click **OK**. With a vector Vertex Map specified, you can use the styling tools to modify the map and control the resulting fur fibers.
Guide Source	
Use Guides From Base Surface	When enabled, uses the guides associated with the surface from which the fur is growing (where guides and polygons are sharing a material tag). When disabled, you can specify tagged guides to control the styling for the fur layer. When disabled, the **Item**, **Polygon Tag Type**, and **Polygon Tag** options become available.
Item	Specifies a guide layer from an available Item (layer) that has guides. This is only available if you disable **Use Guides from Base Surface**.
Polygon Tag Type	Specifies one of Modo's polygon tags. (You can define a surface mask based on Modo's polygon tags as given in the Material Group topic. See the Defining Shaders topic for information about creating tags.) This is only available if you disable **Use Guides from Base Surface**.
Polygon Tag	Specifies an existing tag name for Modo to use when creating the curves for the fur material. This is only available if you disable **Use Guides from Base Surface**, and you specify a **Polygon Tag Type**.

Option	Description
Guide Options	
Guides	Determines how guides affect the generated fur material. Guides are the drawn curves (geometry) that Modo uses to style the fur (controlling the length and direction among other attributes). You can create curves for fur in several ways. See the Working with Fur and the Styling Fur topics. The **Guides** options are: • **None** - Renders fur texture based on the **Fur Materials** settings. Guides have no affect on the fur texture. • **Clump** - Uses guides for clumping. With this option, hairs grow as usual but clump around the closest guides. Fiber tips clump (or gather) toward the guides based on the **Clumps** percentage. • **Direction** - Guides are only used to decide the root direction, based on the guide root's direction. Hair direction is then interpolated between the neighbour guides. • **Direction+Length** - Similar to **Direction**, but the hair length is computed from the guide length, and multiplied by the main **Length** setting. The length parameter still controls the maximum length, but the guides can control overall variations in length. Shortened guides define shorter fur. • **Shape** - Grows hairs so that they match the neighbor guides' shape. It works by finding the three closest guides and interpolating the shape of the guides. Follows the overall shape of the guides for the fur. This mode is ideal for modeling short fur, but not recommended for long hairs. **Note:** Hair length is still set by the main **Length** setting. • **Range** - Hairs are grown around each guide. The range is the influence radius around each guide. Outside of the range no hairs are grown, and the size of hairs fades as you get close to the edge. If one hair falls into the range of influence of several guides, it is interpolated between them. This mode is really well suited for long hairs as it follows the shape and length of the guides exactly either coming from simulation or from sculpting. **Note:** Hair length is dictated by the guide length, so the main **Length** setting is ignored.

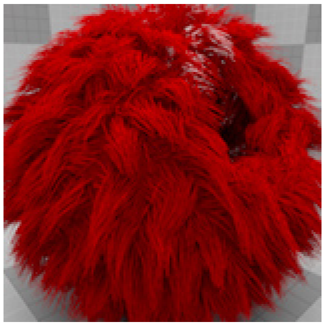
Option	Description
Guide Range	Sets the effective area around the root of the guide where fibers are generated when the **Guides** type is set to **Range**. When there is a **Guide Range** overlap, fibers blend between the two guides based on the blend amount. There is also a small amount of falloff around the **Guide Range** border.
Guide Length	Adjusts the length of fur that is generated by guides without needing to edit the guides, themselves. A value of **100%** equals the overall guide length; lower values reduce the length by the appropriate percentage.
Blend Amount	Controls to what degree fibers generated between guides blend between the two positions when the **Guide Range** setting creates areas of overlapping fibers.
Blend Angle	Sets the maximum angle between two adjoining guides that create a blend of fibers.
Blend VMap	Specifies an embedded Weight Map on the target fur surface and uses its values to control the blending of the fur fibers across the surface. This eliminates needing to create custom image maps. You can select any available Weight Maps from this list. Weight Map values work as multipliers over the defined **Blend Amount** value. Weight values of **1** produce the unmodified, 100% **Blend Amount** and fade toward **0** to produce no blending of the fibers. Values above **1** (100%) increase the effect.
Range VMap	Specifies an embedded Weight Map on the target fur surface and uses its values to control the **Guide Range** amount across the surface. This eliminates needing to create custom image maps. You can select any available Weight Maps from this list. Weight Map values work as multipliers over the defined **Guide Range** value. Weight values of **1** produce the unmodified, 100% **Guide Range** and fade toward **0** to produce no blending of the fibers. Values above **1** (100%) increase the effect.
Clumping	
Clumps	Determines how the fiber tips gather in small groups (clumping). With the guide **Type** set to **Clump**, the higher the **Clumps** value is, the more the tips of fibers tend toward the guides. When there are no guides, the **Clump Range** setting determines the area of fibers that clump together based on the **Clumps** setting. Higher values bring the tips into tighter groups; lower values have less of an effect. Click to view examples of various Clumps settings.
Clump Range	Sets the average range (or area) of fibers that gather together based on the **Clumps** setting.

Option	Description
Strays	Randomizes (with a user-specified percentage) the individual fibers to produce a more realistic, random look to fur generated with guides. This option applies to fur with **Clumping**; otherwise, this setting produces no results. You can specify the number of fibers affected as a percentage of all fibers to define the number of fur fibers that are affected by the **Strays Strength** value.
Strays Strength	Determines the amount of randomization applied to the fibers chosen as strays defined as a Perlin noise function that deviates the fiber along its length away from the guide. With **Clumps**, this value determines the strength a fiber strays away from the clump group.
Clumping	Influences the clumping amount along the clump's length. Using a mini-gradient input, you can modify the clumping amount with gradient keys that represent a percentage of clumping. Much like how a keyframe defines a value in time, gradient keys represent a clumping amount along the clump's overall length for finer control. The gradient input's width represents the clump's overall length with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black produces no clumps; white produces maximum clumping; and the shades of gray attenuate the clumping in between. Complex clumping input also requires additional geometry for Modo to render it correctly. Therefore, increasing the number of segments and increasing the amount of RAM required to render the scene may be necessary with complex tapering.
Clump VMap	Specifies an embedded Weight Map on the target fur surface and uses its values to control the **Clumps** amount across the surface. Doing this eliminates the need to create custom image maps. You can select any available Weight Map from this list to control the clumping of the resulting fur. Weight Map values work as a multiplier over the defined **Clumps** strength setting. A value of **1** produces the unmodified 100% clumps and fades toward **0** to produce no clumping of the fibers. Values above **1** (100%) increase the effect.
Curls	
Mode	Specifies the type of curling for the fur. • **Curls** - Produces a corkscrew-like curving to the fibers.

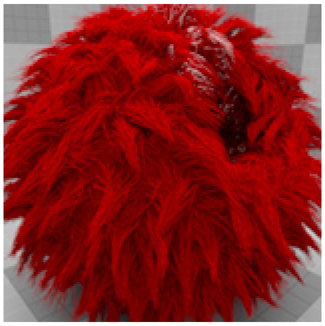
Option	Description
	• **Waves** - Produce an undulating wave-like curving to the fibers.
Amplitude	Determines the effect of the curls. With smaller values, fibers are only slightly wavy or curly; larger values increase the effect to create stronger curls or waves. To create tight, corkscrew-type curls, set the value greater than **100%**. (**500%** is a good starting point.) Then, make sure there are enough segments on the fiber to support the desired details.
Radius	Determines the scale of the **Curls** or **Waves**.
Self Curls	Adds an additional amount of curling to the tube-like form created by the initial deformation (for example, curling of the curls or waves).
Curling	Modulates the curl's strength along the individual fiber's length (acting as a multiplier of the curl's value). Using a mini-gradient input, you can modify the curl's strength with gradient keys that represent the amount of curl. Much like how a keyframe defines a value in time, gradient keys represent the curl strength along the fiber's length for finer control. The gradient input's width represents the length of the fiber with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black produces no curl; white produces maximum curling (based on the curl's value); and the shades of gray attenuate the curling in between. **Curling** is a multiplier of the curl's amount that requires a value above **0** before **Curling** produces results. Curling creates complex fiber twisting that requires additional geometry for Modo to render it correctly. Therefore, increasing the number of segments and increasing the amount of RAM required to render the scene may be necessary with complex tapering.
Curl VMap	Specifies an embedded Weight Map on the target fur surface and uses its values to control the curls' **Amplitude** amount across the surface. Doing this eliminates the need to create custom image maps. You can select any available Weight Map from this list to control the clumping of the resulting fur. Weight Map values work as a multiplier over the defined **Amplitude** strength setting. A value of **1** produces the unmodified 100% amplitude and fades toward **0** to produce no curling of the fibers. Values above **1** (100%) increase the effect.



Fur Clumps 0%

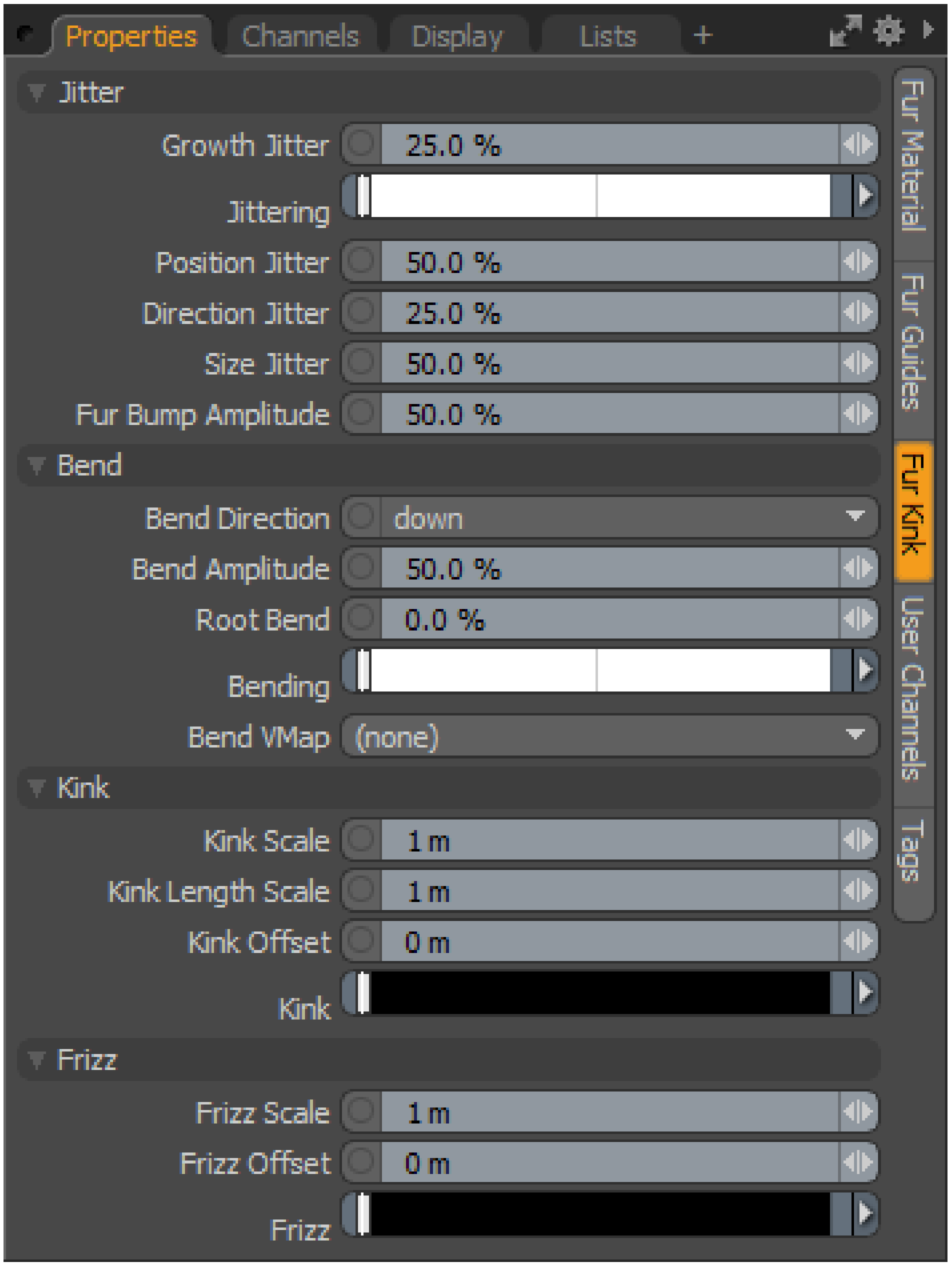


Fur Clumps 50%



Fur Clumps 100%

Fur Kink



Jitter

Growth Jitter	25.0 %
Jittering	
Position Jitter	50.0 %
Direction Jitter	25.0 %
Size Jitter	50.0 %
Fur Bump Amplitude	50.0 %

Bend

Bend Direction	down
Bend Amplitude	50.0 %
Root Bend	0.0 %
Bending	
Bend VMap	(none)

Kink

Kink Scale	1 m
Kink Length Scale	1 m
Kink Offset	0 m
Kink	

Frizz

Frizz Scale	1 m
Frizz Offset	0 m
Frizz	

Based on a noise modifier, the **Kink** effect applies a chaotic twisting to the individual fibers to make them look tangled. The detail introduced is dependent on the number of segments in the fiber. A greater number of segments produce more detailed results, but more segments also require additional memory and time to render. To produce a result, there must be some value of **Kink** along the gradient.

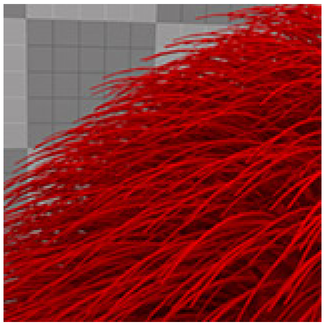
The **Frizz** effect applies a somewhat random waviness to the individual fibers to making them look messy. The detail introduced is dependent on the number of segments in the fiber. A greater number of segments produce more detailed results, but more segments also require additional memory and time to render. To produce a result, there must be some value of **Frizz** along the gradient.

Option	Description
Jitter	
Growth Jitter	Increases the randomness of a fiber along its length. A setting of **0%** produces straight fibers; increasing values produce more jagged and random fibers. Click to view examples of various Growth Jitter settings.
Jittering	Modulates the jitter strength along the individual fibers length (acting as a multiplier of the **Growth Jitter** value). Using a mini-gradient input, you can modify the jittering strength with gradient keys that represent the amount of jittering. Much like how a keyframe defines a value in time, gradient keys represent the jittering strength along the fiber's length for finer control. The gradient input's width represents the length of the fiber with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black produces no jittering; white produces maximum jittering (based on the **Growth Jitter** value); and the shades of gray attenuate the jittering in between. Jittering creates complex fiber twisting that requires additional geometry for Modo to render it correctly. Therefore, increasing the number of segments and increasing the amount of RAM required to render the scene may be necessary with complex tapering. Additionally, because **Jittering** is a multiplier of the **Growth Jitter** amount, a value above **0** is necessary before Jittering produces results.
Position Jitter	Increases the randomness of the root positions of fibers. When creating multi-layered fur, subsequent fiber layers grow from identical positions. If this is not the intended result, you can increase the **Position Jitter** value to eliminate the overlap.
Direction Jitter	Randomizes the angle of rotation for the root of a fiber. This is similar to **Strip Rotation** on the **Fur Material** sub-tab; however, **Direction Jitter** also randomizes a fiber's bend direction. A setting of **0%** produces fibers in a uniform direction. Values greater than **0%** increase the randomness of rotations for each fiber. Click to view examples of various Direction Jitter settings.
Size Jitter	Randomizes the overall scale of each fiber. A setting of **0%** produce fibers of equal size; values greater than **0%** increase the random scaling of fibers. Click to view examples of various Size Jitter settings.

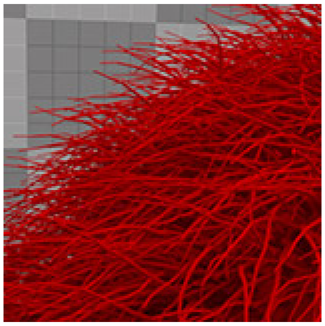
Option	Description
Fur Bump Amplitude	Controls the strength of influence a texture layer assigned to the **Fur Bump** layer effect has over the **Root Bend** and **Bend Direction** values of the fur effectively to control the growth direction. This is similar in function to the **Fur Direction** controller used for combing fur; however, a **Fur Bump** layer can be an arbitrary image map or a procedural texture to control the direction much like a bump map perturbs the surface rays and simulates bumpy surfaces.
Bend	
Bend Direction	Controls the bending orientation of individual fibers. • **Down** - Flexes the fibers downward as if gravity were affecting them. • **Normals** - Bends the fibers based on the normal direction of the surface directly under the fiber. This is useful when the Billboards setting is **Feathers**.
Bend Amplitude	Adds bending or drooping to individual fibers as if they were affected by gravity. For fibers growing straight up (such as grass), a value of **0%** produces perfectly straight fibers, and setting of **100%** produces a fiber tip that is bent 180° from its root. Fibers growing downward are less affected by this setting. To produce smooth bend results, you may need to increase the number of **Segments**. Click to view examples of various Bend Amplitude settings.
Root Bend	Determines the direction of a fiber. With a value of **0%**, a fiber's root direction is based on the direction of the corresponding smoothed polygon's normal. (Whatever direction the polygon is facing, the fiber grows from that surface in the same direction.) Increasing this value rotates the root to alter the fiber's growth direction. A setting of **100%** produces a fiber that grows perpendicular to the normal (a rotation of 90 degrees from the surface). Click to view examples of various Root Bend settings.
Bending	Modulates the **Bend Amplitude** along the individual fiber's length (acting as a multiplier of the **Bend Amplitude** value). Using a mini-gradient input, you can modify the bending strength with gradient keys that represent the amount of Bending. Much like how a keyframe defines a value in time, gradient keys represent the bend strength along the fiber's length for finer control. The gradient input's width represents the length of the fiber with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to

Option	Description
	how the values fade from one key to the next. Black produces no bending; white produces maximum bending (based on the **Bend Amplitude** value); and the shades of gray attenuate the bending in between. Additionally, because **Bending** is a multiplier of the **Bend Amplitude** amount, a value above **0** is necessary before **Bending** produces results.
Bend VMap	Specifies an embedded Weight Map on the target fur surface and uses its values to control the bend strength across the surface. Doing this eliminates the need to create custom image maps. You can select any available Weight Map from this list to control the clumping of the resulting fur. Weight Map values work as a multiplier over the defined **Bend Amplitude** setting. A value of **1** produces the unmodified 100% bending amount and fades toward **0** to produce no bending of the fibers. Values above **1** (100%) increase the effect.
Kink	
Kink Scale	Determines the size of the noise that produces the kinks. Smaller values have a more dramatic tangled look, and larger values look less tangled.
Kink Length Scale	Determines the scale of the kink along the length of the individual fiber. Large values create fibers that look more like waves.
Kink Offset	Re-positions the noise generator for the kinks. When animated, you can use this option, for example, to simulate a grass field blowing in the wind.
Kink	Modulates the kink amount along the individual fiber's length. Using a mini-gradient input, you can modify the amount with gradient keys. Much like how a keyframe defines a value in time, gradient keys represent the kink amount along the fiber's length for finer control. The gradient input's width represents the length of the fiber with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black produces no kink; white produces a maximum kink (based on the **Kink Scale**, **Kink Length Scale**, and **Kink Offset** values); and the shades of gray attenuate the kinking in between. The **Kink** effect creates complex fiber twisting that requires additional geometry for Modo to render it correctly. Therefore, increasing the number of segments and increasing the amount of RAM required to render the scene may be necessary with complex tapering.

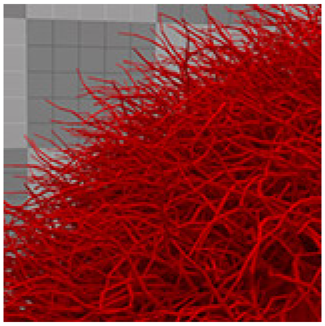
Option	Description
Frizz	
Frizz Scale	Determines the size of the waviness applied to the fibers. Smaller values produce finer waves; larger values increase the size of the waves.
Frizz Offset	Re-positions the noise generator for the frizz. When animated, you can use this option, for example, to simulate a grass field blowing in the wind.
Frizz	Modulates the frizz amount along the individual fiber's length. Using a mini-gradient input, you can modify the amount with gradient keys. Much like how a keyframe defines a value in time, gradient keys represent the frizz amount along the fiber's length for finer control. The gradient input's width represents the length of the fiber with the base on the left side. You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. Black produces no frizz; white produces a maximum frizz (based on the **Frizz Scale** and **Frizz Offset** values); and the shades of gray attenuate the frizziness in between. The **Frizz** effect creates complex fiber twisting that requires additional geometry for Modo to render it correctly. Therefore, increasing the number of segments and increasing the amount of RAM required to render the scene may be necessary with complex tapering.



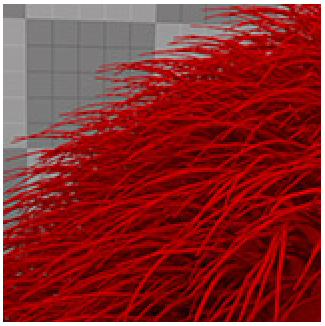
Growth Jitter 0%

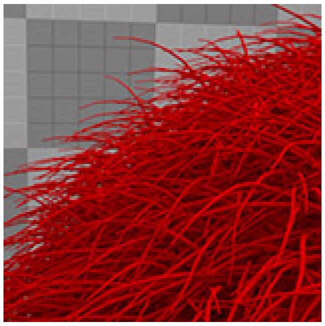


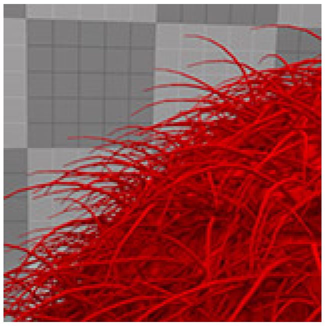
Growth Jitter 50%



Growth Jitter 100%



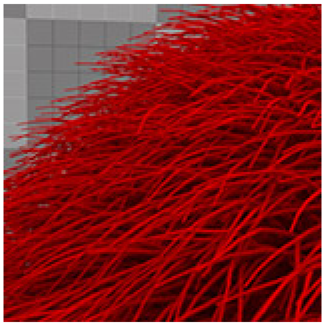




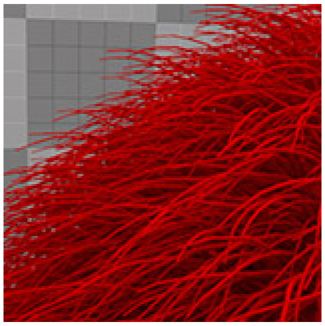
Direction Jitter 0%

Direction Jitter 50%

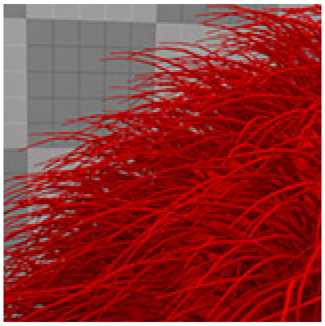
Direction Jitter 100%



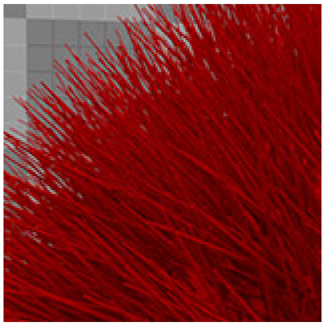
Size Jitter 0%



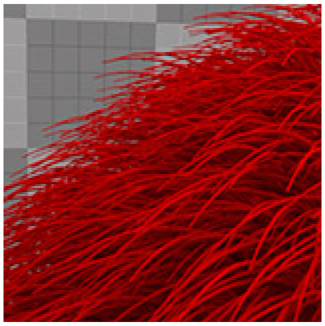
Size Jitter 50%



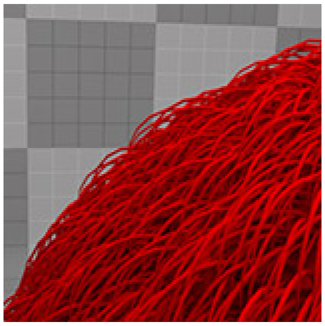
Size Jitter 100%



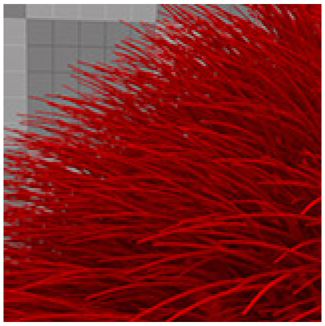
Bend Amplitude 0%



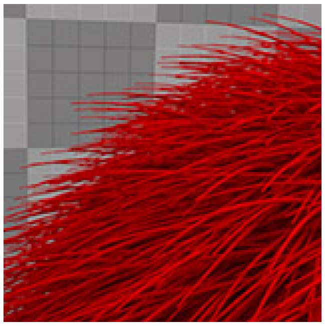
Bend Amplitude 50%



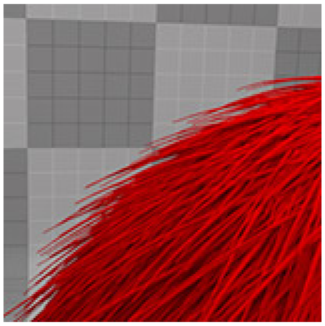
Bend Amplitude 100%



Root Bend 0%



Root Bend 50%



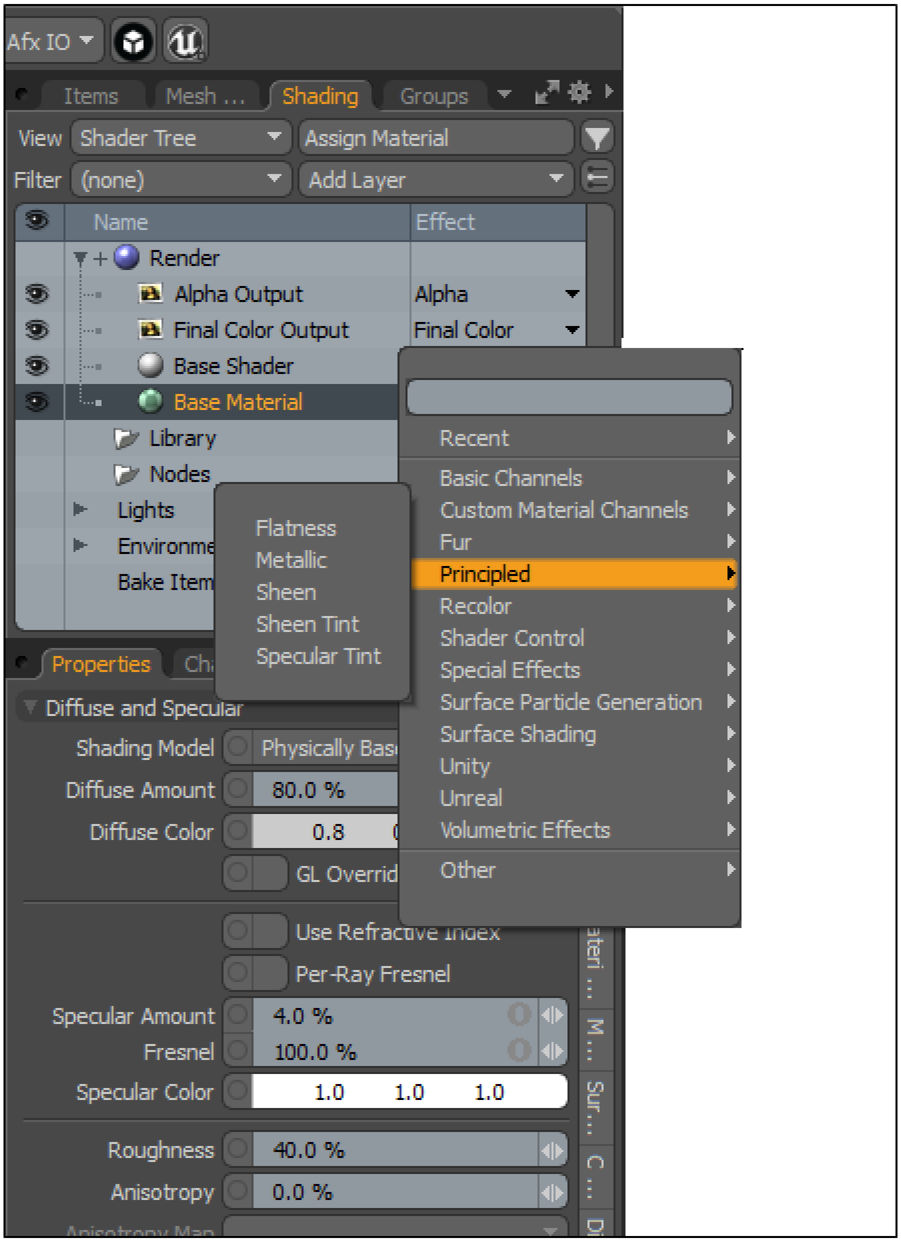
Root Bend 100%

Principled Shader

The **Principled** shading model is based on the Principled BRDF model, which attempts to give simplified, artist-friendly, parameters to control your shading. Use this shader model on a Modo Material.

Note: For information about adding and working with **Shader Tree** item layers, see the Shader Tree topic.

Tip: Import textures for your Principled shader using the PBR loader. See PBR Textures for more information.


Afx IO
Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Name
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Base Material
Library
Nodes
Lights
Environme
Bake Item
Flatness
Metallic
Sheen
Sheen Tint
Specular Tint
Recent
Basic Channels
Custom Material Channels
Fur
Principled
Recolor
Shader Control
Special Effects
Surface Particle Generation
Surface Shading
Unity
Unreal
Volumetric Effects
Other
Properties
Ch
Diffuse and Specular
GL Overrid
Use Refractive Index
Per-Ray Fresnel
Anisotropy Map

Shading Model: Physically Bas
Diffuse Amount: 80.0 %
Diffuse Color: 0.8
Specular Amount: 4.0 %
Fresnel: 100.0 %
Specular Color: 1.0 1.0 1.0
Roughness: 40.0 %
Anisotropy: 0.0 %

Principled Shader Types

The following Principled Shader types are available:

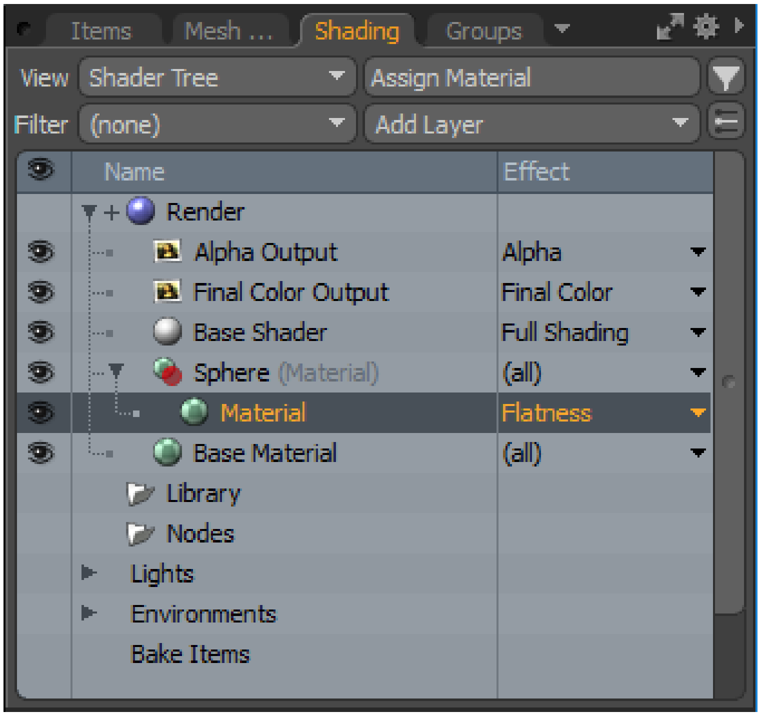
Shader Type	Description
Flatness	**Flatness** is a faked sub-surface effect, which flattens out the diffuse highlight in order to fake some amount of subsurface scattering and translucency. **Note:** No subsurface rays are fired when **Flatness** is applied. Only a modification of the diffuse lighting profile is applied.
Metallic	Controls the diffuse and specular response of the material. At 0%, it has no effect. As it approaches 100%, the **Specular Color** blends towards the given **Diffuse Color**, while the **Diffuse Color** used for shading is blended towards black.
Sheen	**Sheen** is an additional grazing value. For example, use this shader type to create "peach fuzz" effect. The default **Sheen** color is white. As **Sheen** increases, an additional grazing component is added to the surface. **Note:** As **Metallic** reaches 100%, the effect of the **Sheen** property is decreased to 0%.
Sheen Tint	Increasing the **Sheen Tint** value blends the **Sheen** color towards the **Diffuse Color**.
Specular Tint	As you increase the **Specular Tint** property, the **Specular Color** is blended towards the **Diffuse Color**. **Note:** The **Specular Color** is assumed to be white (For example, a dielectric surface).

Setting the Selected Principled Shader Percentage Value

You can quickly change the percentage for the selected Principled Shader value.

To set a percentage value:

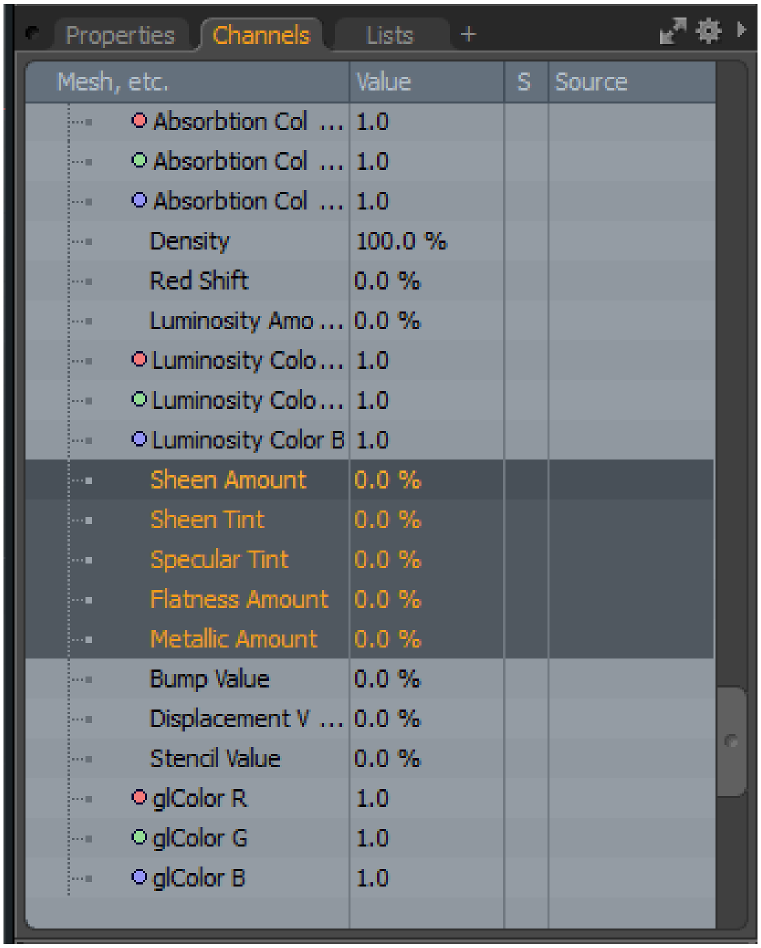
1. Open the **Shading** tab on the right panel.
2. Select a **Material** item in the list.
3. From the **Effect** dropdown menu, select **Principled** and select a shader type. In this example, we have selected **Flattness**.



Items
Mesh ...
Shading
Groups
View
Shader Tree
Assign Material
Filter
(none)
Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Sphere (Material)	(all)
Material	Flatness
Base Material	(all)
Library	
Nodes	
Lights	
Environments	
Bake Items	

4. On the right panel, open the **Channels** tab, and scroll down to the one of the **Principled** shader types.



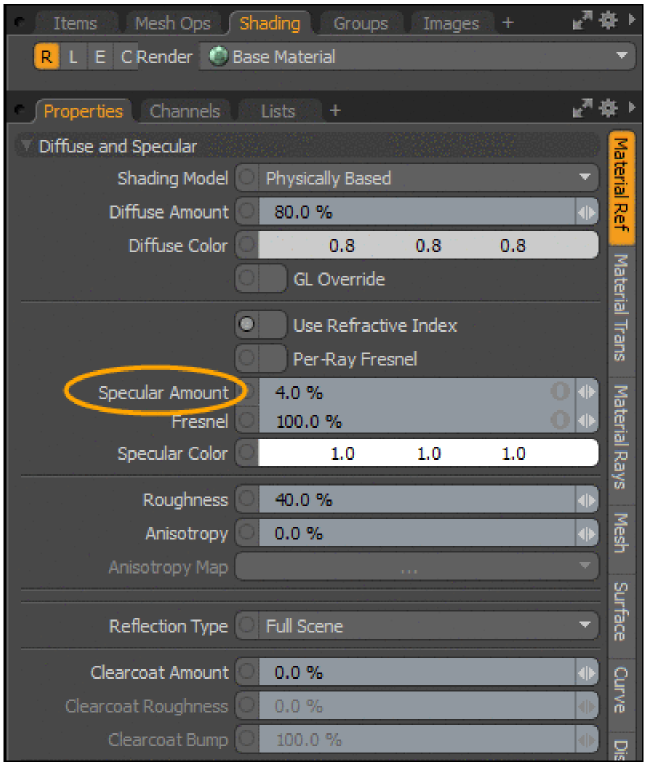
Mesh, etc.	Value	S	Source
○ Absorbtion Col ...	1.0		
○ Absorbtion Col ...	1.0		
○ Absorbtion Col ...	1.0		
Density	100.0 %		
Red Shift	0.0 %		
Luminosity Amo ...	0.0 %		
○ Luminosity Colo ...	1.0		
○ Luminosity Colo ...	1.0		
○ Luminosity Color B	1.0		
Sheen Amount	0.0 %		
Sheen Tint	0.0 %		
Specular Tint	0.0 %		
Flatness Amount	0.0 %		
Metallic Amount	0.0 %		
Bump Value	0.0 %		
Displacement V ...	0.0 %		
Stencil Value	0.0 %		
○ glColor R	1.0		
○ glColor G	1.0		
○ glColor B	1.0		

5. Click in the **Value** field of the selected Principled shader and type in your desired percentage value.

Setting the Refractive Index from the Specular Amount

Modo provides a **Refractive Amount** option, which is exposed on a **Material** when using the **Principled** shading model with the **Use Refractive Index** is enabled.

Click on the image below to see an animation.


Items
Mesh Ops
Shading
Groups
Images
R L E C Render
Base Material
Properties
Channels
Lists
Material Ref
Material Trans
Material Rays
Mesh
Surface
Curve
Dis

Diffuse and Specular

Shading Model: Physically Based
Diffuse Amount: 80.0 %
Diffuse Color: 0.8 0.8 0.8
○ GL Override

● Use Refractive Index
○ Per-Ray Fresnel
Specular Amount: 4.0 %
Fresnel: 100.0 %
Specular Color: 1.0 1.0 1.0

Roughness: 40.0 %
Anisotropy: 0.0 %
Anisotropy Map: ...

Reflection Type: Full Scene

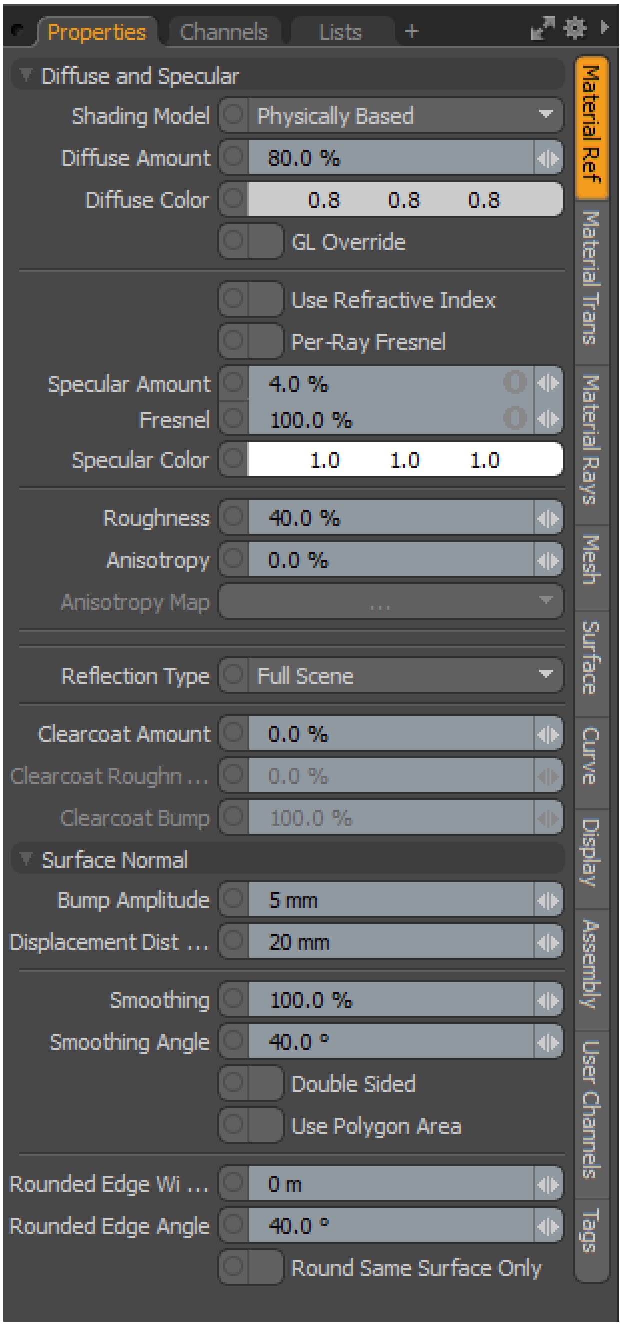
Clearcoat Amount: 0.0 %
Clearcoat Roughness: 0.0 %
Clearcoat Bump: 100.0 %

The **Use Refractive Index** option attempts to derive an index of the refraction from the **Specular Amount** effect; enabling a control over the Refractive Index (**IOR**) with a more familiar **Specular Amount** property.

Effectively, the **Use Refractive Index** option remaps the standard **Specular Amount** range (0% - 100%) to an **Index of Refraction** range (1.0 - 1.8), which attempts to cover majority of materials. For example, with **Specular Amount** set to 50% , the **Refractive Index** (**IOR**) is 1.5, which is a good approximate value for most materials (most of which fall within the Refractive Index (**IOR**) range of 1.4 - 1.6).

Principled Properties

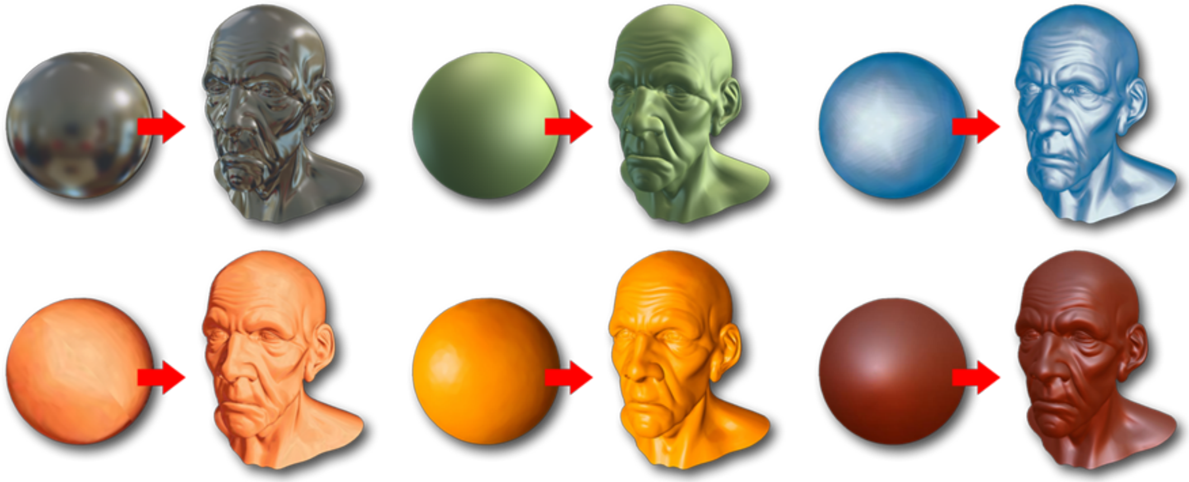
The following options are available. For more details on these options, see Material.


Properties
Channels
Lists
+
Material Ref
Material Trans
Material Rays
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

Diffuse and Specular
Shading Model: Physically Based
Diffuse Amount: 80.0 %
Diffuse Color: 0.8 0.8 0.8
GL Override
Use Refractive Index
Per-Ray Fresnel
Specular Amount: 4.0 %
Fresnel: 100.0 %
Specular Color: 1.0 1.0 1.0
Roughness: 40.0 %
Anisotropy: 0.0 %
Anisotropy Map: ...
Reflection Type: Full Scene
Clearcoat Amount: 0.0 %
Clearcoat Roughn ...: 0.0 %
Clearcoat Bump: 100.0 %
Surface Normal
Bump Amplitude: 5 mm
Displacement Dist ...: 20 mm
Smoothing: 100.0 %
Smoothing Angle: 40.0 °
Double Sided
Use Polygon Area
Rounded Edge Wi ...: 0 m
Rounded Edge Angle: 40.0 °
Round Same Surface Only

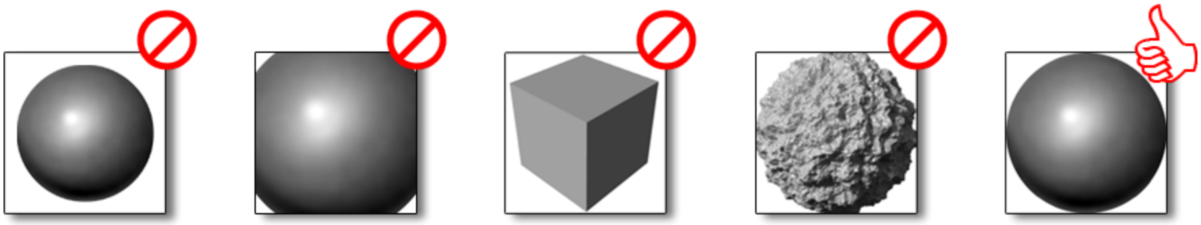
MatCap Shader

MatCaps, or "material captures" allows you to create a surface material and lighting environment simply by painting an object so that it looks like how you want your surface to appear. What happens is the shading of a spherical image is applied to the target surface by mapping corresponding normals from the captured surface onto that of the rendered surface. This opens up all sorts of interesting possibilities for nonphotoreal image rendering demonstrated by these examples below; the original MatCap is on the left, with the resulting shaded surface on the right.



Creating Custom MatCaps

The MatCap shader requires a spherical image as a source. You can create or photograph any spherical shape for application to the surface. However, make sure the final image is cropped in exactly to the outer edge of the sphere and that the image has the same width and height (square aspect ratio). When creating MatCaps, avoid surfaces with high-frequency details, these don't typically map well to complex surfaces, and they end up looking like the image is a reflection onto the surface instead; soft or subtle transitions work best. Resolution-wise, make sure the image itself captures the necessary details for the target output resolution, but for soft, blurry MatCaps, sizes in the 512x512 to 1024x1024 pixel resolution works just fine for most cases.

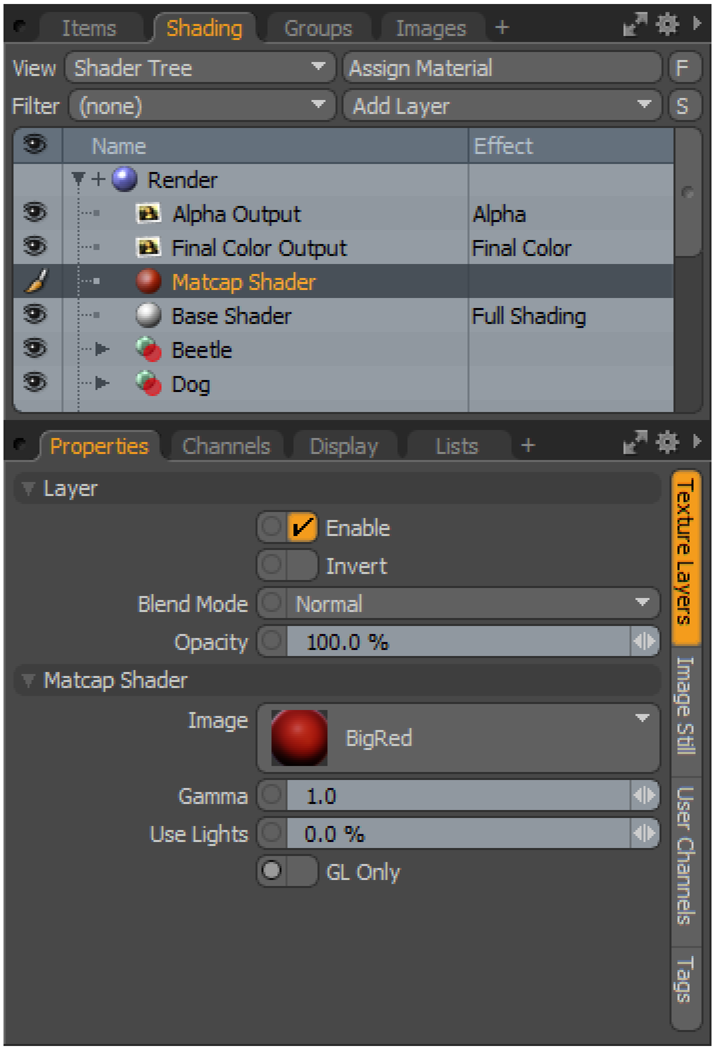


Applying MatCaps

MatCaps are applied with the MatCap shader, and can be added to the **Shader Tree** by clicking **Add Layer** > **Special** > **MatCap**. Initially, Modo places the MatCap at the top of the **Shader Tree**, affecting all elements in the scene. You can leave the item in this position and enable the **GL Only** option so the surfacing only shows in the 3D Viewports, not in rendered images, a great option for sculpting.

Tip: To experiment with MatCap shaders, you can select them from the 3D Viewport Viewport Textures button without having to add any items in the Shader Tree. Using this feature enables you to experiment with different textures before actually creating **Shader Tree** layers. If a MatCap image is set it completely replaces the surface shader in the **Shader Tree** with just that MatCap image.

The MatCap layer works like a replacement shader, placing itself above the base shader, it overwrites the settings of the base shader. In order to localize the effects of the MatCap, you need to create an appropriate mask for the target item or surface and place that group with the MatCap layer above the base shader. For more information on working with **Shader Tree**, see the Shader Tree topic.


Items
Shading
Groups
Images
+
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Properties
Channels
Display
Lists
+
Texture Layers
Image Still
User Channels
Tags

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Matcap Shader	
Base Shader	Full Shading
Beetle	
Dog	

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Matcap Shader

Image: BigRed

Gamma: 1.0

Use Lights: 0.0 %

○ GL Only

Option	Description
Layer	
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a photonegative effect.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the **Shader Tree**, if present, or dims the effect of the layer, itself, on the surface.
MatCap Shader	
Image	Specifies the spherical MatCap image that is to be applied to the target surface. If no images are loaded in the scene, a clip browser window opens, allowing you to load an image for use. If images are already present in the scene, you can specify any existing image for use. You may need to load the image separately, using the **Images** viewport before applying a MatCap shader.
Gamma	This setting can be used as an image adjustment control to lighten or darken the MatCap image without the need to externally adjust the image in an external bitmap editor. The **Gamma** adjustment is non-linear and affects the mid-tones of the image greater than the shadow and highlight areas of the image.
Use Lights	The shading of MatCaps surfaces are controlled directly by the MatCap image itself, and are not affected by scene lighting. This option provides a means to incorporate the lights in the scene, blending them with the texture of the MatCap shader. A value of 0% disables all lights on the scene, so they don't have any effect on the surface, moving up toward 100% which means the surface is completely controlled by the lights of the scene, in effect changing the MatCap shader into a diffuse color channel for the surface.
GL Only	When this toggle is enabled, the MatCap shader only affects the OpenGL viewport representation of the surface, and the rendered surface displays as normal (as if the

Option	Description
	MatCap shader was disabled). This is useful for when you wish to use MatCaps as a sculpting or modeling aid, giving you a clearer view of the geometry but still have the surface render normally otherwise.

Material Presets

The Material Preset is a layer in the Shader Tree that references a material preset from the Preset Browser, and allows you to easily change the materials on an item.

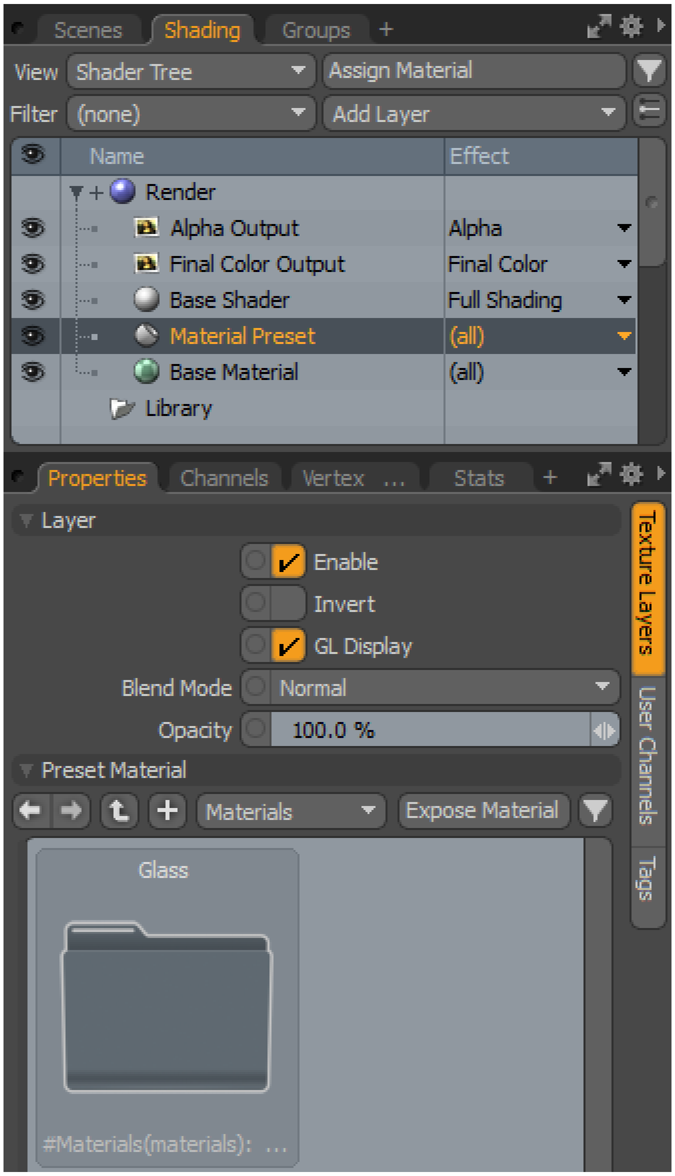
Tip: For more information on using the Preset Browser, see Preset Browser.

To add a Material Preset:

1. In the Shader Tree, click **Add Layer**, and under **Special**, click **Material Preset**.

   The Material Preset layer is added to the Shader Tree, and its properties open in the lower-right **Properties** panel. You can see the available materials in the **Preset Material** section. Only the materials on your hard drive can be found here.

   You can use the standard Preset Browser buttons to navigate among your materials and adjust the layout of the section.


Scenes
Shading
Groups
+
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Properties
Channels
Vertex ...
Stats
+
Texture Layers
User Channels
Tags
Materials
Expose Material
Glass
#Materials(materials): ...

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Material Preset	(all)
Base Material	(all)
Library	

Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

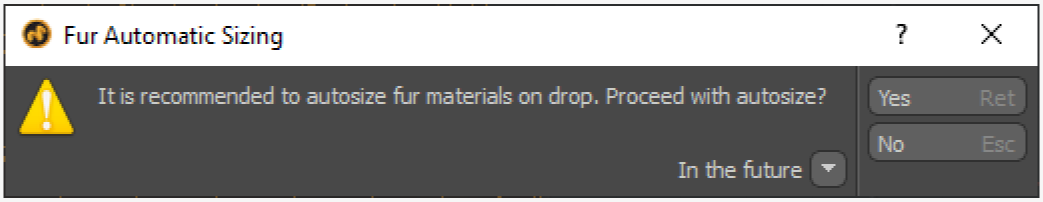
Opacity: 100.0 %

Preset Material

2. In the **Properties** panel, under **Preset Material**, click the material you want to load.

   The selected material is applied to your item.

Note: When selecting a fur material, Modo displays a warning, prompting you to autosize the material:


Fur Automatic Sizing
?
X
Yes Ret
No Esc
In the future

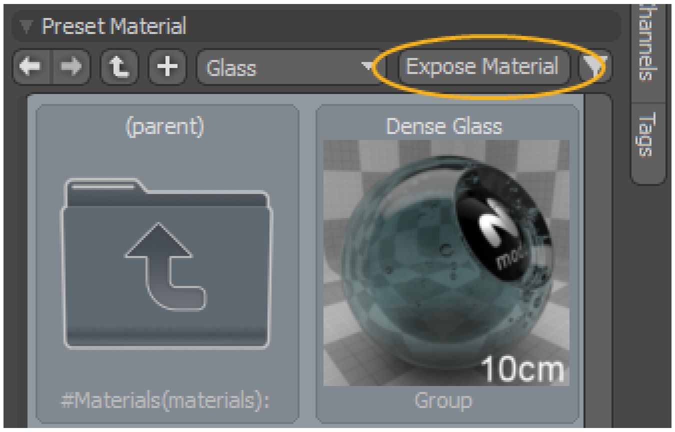
It is recommended to autosize fur materials on drop. Proceed with autosize?

3. If you decide to use a different material, click on another one in the Preset Browser.

Note: Material Presets may not always show the correct preset as selected in the item properties. You can look at the **filepath** channel in the **Channels** list if you need to know which preset file a material preset item is referencing.

You can convert the Material Preset to a standard texture group by clicking the **Expose Material** button in the **Properties** panel. This allows you to fully edit all layers in the group.

Tip: For more information on material groups, see Material Group.


Preset Material
Glass
Expose Material
Channels
Tags
(parent)
Dense Glass
10cm
#Materials(materials):
Group

Refractive Index Table

You may specify these settings as the **Refractive Index** setting in the **Materials** panel.

Gemstones

Agate 1.544 - 1.553
Alexandrite 1.746 - 1.755
Almandine 1.75 - 1.83
Amber 1.539 - 1.545
Amethyst 1.532 - 1.554
Ammolite 1.52 -1.68
Andalusite 1.629 - 1.650
Apatite 1.632 - 1.42
Aquamarine 1.567-1.590
Axenite 1.674 - 1.704
Beryl 1.57 - 1.60
Beryl, Red 1.570 - 1.598

Chalcedony 1.544 - 1.553
Chrome Tourmaline, 1.61 - 1.64
Citrine 1.532 - 1.554
Clinohumite 1.625 - 1.675
Coral 1.486 - 1.658
Crystal 2.000
Crysoberyl, Catseye 1.746 - 1.755
Danburite 1.627 - 1.641
Diamond 2.417
Emerald 1.560 - 1.605
Emerald Catseye 1.560 - 1.605
Flourite 1.434
Garnet, Grossular 1.72 - 1.80
Garnet, Andradite 1.88 - 1.94
Garnet, Demantiod 1.880 - 1.9
Garnet, Mandarin 1.790 - 1.8
Garnet, Pyrope 1.73 - 1.76
Garnet, Rhodolite 1.740 - 1.770
Garnet, Tsavorite 1.739 - 1.744
Garnet, Uvarovite 1.74 - 1.87
Hauyn 1.490 - 1.505
Iolite 1.522 - 1.578
Jade, Jadeite 1.64 - 1.667
Jade, Nephrite 1.600 - 1.641
Jet 1.660
Kunzite 1.660 - 1.676
Labradorite 1.560 - 1.572
Lapis Lazuli 1.50 - 1.55
Moonstone 1.518 - 1.526
Morganite 1.585 - 1.594
Obsidian 1.50
Opal, Black 1.440 - 1.460
Opal, Fire 1.430 - 1.460
Opal, White 1.440 - 1.460
Oregon Sunstone 1.560 - 1.572
Padparadja 1.760 - 1.773
Pearl 1.53 - 1.69
Peridot 1.635 - 1.690
Quartz 1.544 - 1.553
Ruby 1.757 - 1.779
Sapphire 1.757 - 1.779
Sapphire, Star 1.760 - 1.773

Spessarite 1.79 - 1.81
Spinel 1.712 - 1.717
Spinel, Blue 1.712 - 1.747
Spinel, Red 1.708 - 1.735
Star Ruby 1.76 - 1.773
Tanzanite 1.690-1.7
Tanzanite 1.692 - 1.700
Topaz 1.607 - 1.627
Topaz, Imperial 1.605-1.640
Tourmaline 1.603 - 1.655
Tourmaline, Blue 1.61 - 1.64
Tourmaline, Catseye 1.61 - 1.64
Tourmaline, Green 1.61 - 1.64
Tourmaline, Paraiba 1.61 - 1.65
Tourmaline, Red 1.61 - 1.64
Zircon 1.777 - 1.987
Zirconia, Cubic 2.173 - 2.21

Common Liquids
Acetone 1.36
Alcohol, Ethyl (grain) 1.36
Alcohol, Methyl (wood) 1.329
Beer 1.345
Carbonated Beverages 1.34 - 1.356
Fruit Juice
Chlorine (liq) 1.385
Cranberry Juice (25%) 1.351
Glycerin 1.473
Honey, 13% water content 1.504
Honey, 17% water content 1.494
Honey, 21% water content 1.484
Ice 1.309
Milk 1.35
Oil, Clove 1.535
Oil, Lemon 1.481
Oil, Neroli 1.482
Oil, Orange 1.473
Oil, Safflower 1.466
Oil, vegetable (50 C) 1.47
Oil of Wintergreen 1.536
Rum, White 1.361
Shampoo 1.362
Sugar Solution 30% 1.38

Sugar Solution 80% 1.49
Turpentine 1.472
Vodka 1.363
Water (0 C) 1.33346
Water (100 C) 1.31766
Water (20 C) 1.33283
Whisky 1.356<

Common Transparent Materials

Eye, Aqueous humor 1.33
Eye, Cornea 1.38
Eye, Lens 1.41
Eye, Vitreous humor 1.34
Glass, Arsenic Trisulfide 2.04
Glass, Crown (common) 1.52
Glass, Flint, 29% lead 1.569
Glass, Flint, 55% lead 1.669
Glass, Flint, 71% lead 1.805
Glass, Fused Silica 1.459
Glass, Pyrex 1.474
Lucite 1.495
Nylon 1.53
Obsidian 1.50
Plastic 1.460 - 1.55
Plexiglas 1.488
Salt 1.516

Shader

By default, each new scene has a single base **Shader Item**. This item reads the information from the entire **Shader Tree** and computes the effects of the illumination on all the surfaces in the scene. This makes the base Shader Item the most important layer in the tree. Shaders also control the **Shading Rate**, which is the rate that Modo uses to evaluate pixels before sending them to the render item. This breaks up a single pixel for texture computing and anti-aliasing.

Additionally, shaders control visibility of surfaces in a scene and what rays are seen by specific surfaces. For example, you may want to make only the shadow of an object visible. In most cases, you won't change the settings in the shader; however, where specific surfaces need additional control (such as limiting Monte Carlo shading to a single surface, hiding a group of lights from illuminating a single surface, or even hiding an entire object from a camera), you may need to add an additional shader with appropriate settings applied.

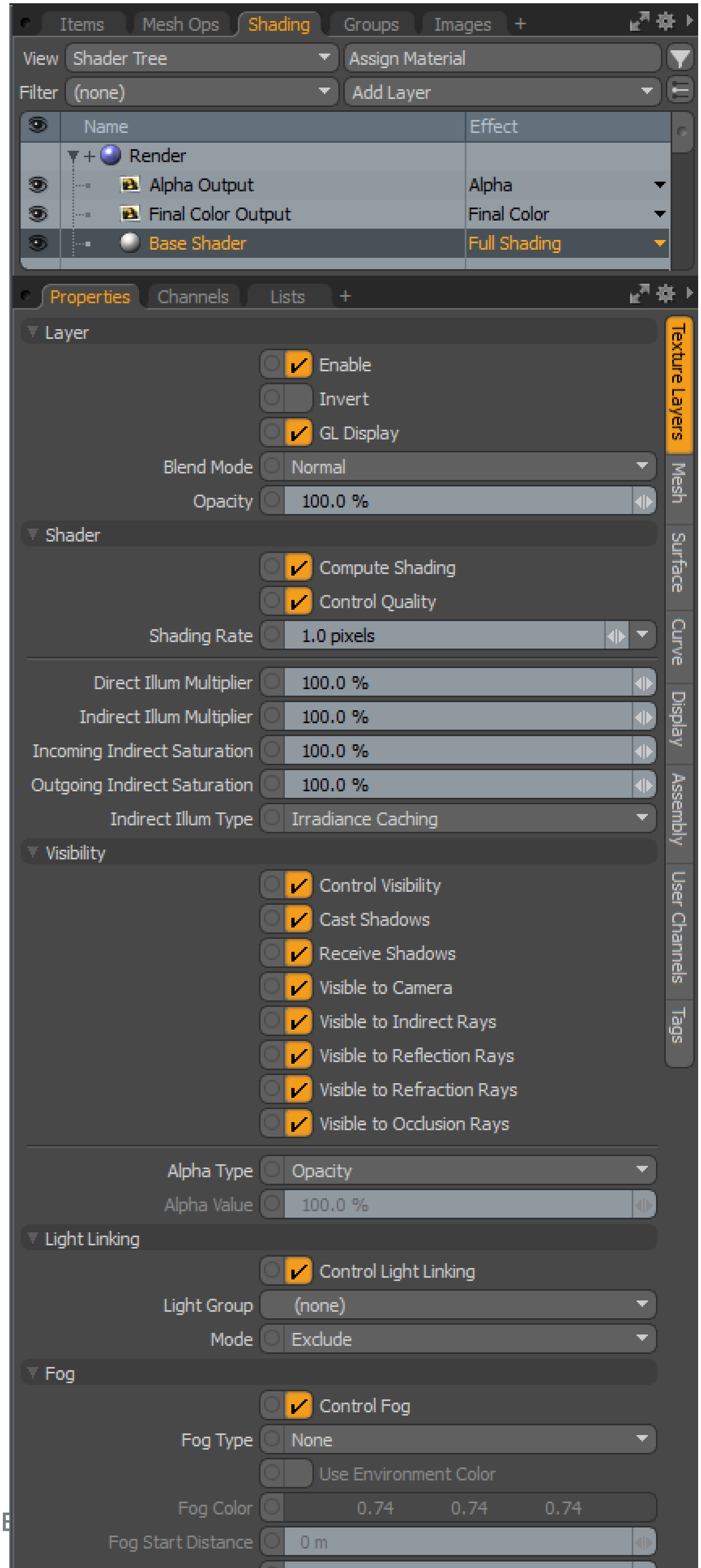
Assigning Item-Specific Shaders

In situations where you need additional shading control for a specific surface or item (such as making a single item invisible to a reflection), you can add another shader to the scene. Because of the layered aspect of the **Shader Tree**, any additional shaders need to be placed above the base shader within its own material group mask. (Otherwise, the base shader settings override it.) A simplified approach is to apply an item-specific shader. These shaders work outside of the **Shader Tree**, itself, to act as individual item-based overrides of the **Shader Tree** settings. You add and select item shaders within the **Items** list. After right-clicking the target layer, choose **Create Item Shader** and assign the shader to the item. Clicking the **+** sign preceding the layer's name displays the shader below the item. You can then select the shader and modify its attributes in the **Properties** panel. Watch the video to learn the process.

There are several options to control visibility of the attributes of the elements feeding the shader. For instance, with the **Visibility** controls, you could create a bright luminous polygon that is visible to indirect illumination but invisible to reflections, ray-traced shadows, reflection, and the camera. This creates a virtual light element that you can place anywhere in the scene for illumination purposes without worrying about it affecting other visual aspects of the project.

When applying the item shader for a specific task, you can disable the **Control** (or **Compute**) options except for the targeted adjustment. For example, to control the visibility of an item, you could disable **Compute Shading**, **Control Quality**, **Control Lighting**, and **Control Fog**. Then disable **Visible to Camera**. Now the target item is invisible to the camera, but it still casts a shadow and shows up in reflections. Shaders applied to items simplify certain tasks like these and keep the **Shader Tree** free of additional layers.

Note: For information about adding and working with **Shader Tree** item layers, see the Shader Tree topic.


Items
Mesh Ops
Shading
Groups
Images
+
Assign Material
Add Layer
Properties
Channels
Lists
+
Texture Layers
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

View: Shader Tree

Filter: (none)

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading

Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Shader

☑ Compute Shading

☑ Control Quality

Shading Rate: 1.0 pixels

Direct Illum Multiplier: 100.0 %

Indirect Illum Multiplier: 100.0 %

Incoming Indirect Saturation: 100.0 %

Outgoing Indirect Saturation: 100.0 %

Indirect Illum Type: Irradiance Caching

Visibility

☑ Control Visibility

☑ Cast Shadows

☑ Receive Shadows

☑ Visible to Camera

☑ Visible to Indirect Rays

☑ Visible to Reflection Rays

☑ Visible to Refraction Rays

☑ Visible to Occlusion Rays

Alpha Type: Opacity

Alpha Value: 100.0 %

Light Linking

☑ Control Light Linking

Light Group: (none)

Mode: Exclude

Fog

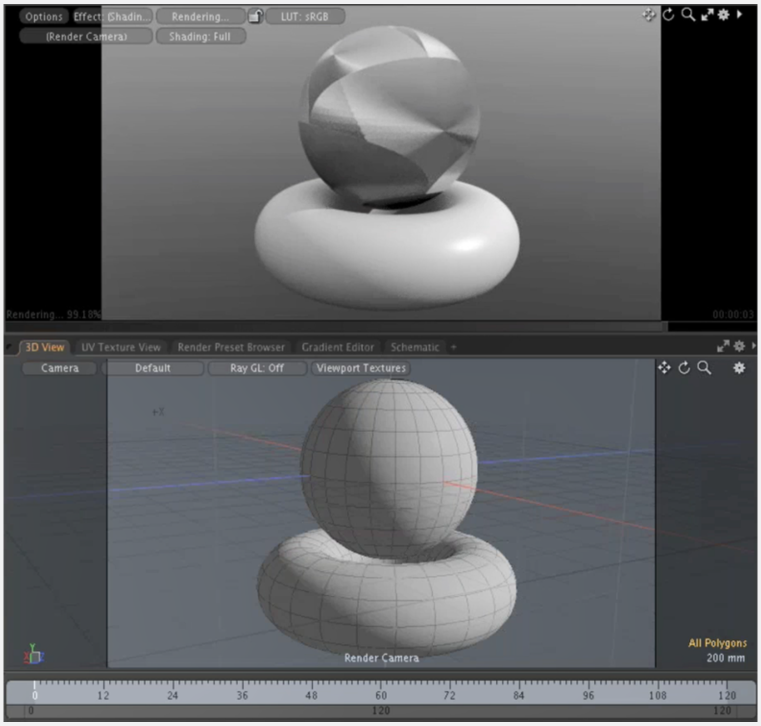
☑ Control Fog

Fog Type: None

☐ Use Environment Color

Fog Color: 0.74 0.74 0.74

Fog Start Distance: 0 m

Option	Description
Layer	
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling visibility in the **Shader Tree**. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a photonegative effect.
GL Display	Toggles on and off the display of textures in the Camera View to improve readability and performance when working on complex scenes. When this option is disabled, textures are displayed in the Preview Viewport but not in the Camera View.  Options Effect: Shadin... Rendering... LUT: sRGB (Render Camera) Shading: Full Rendering... 99.18% 00:00:03 3D View UV Texture View Render Preset Browser Gradient Editor Schematic Camera Default Ray GL: Off Viewport Textures +X Render Camera All Polygons 200 mm 0 12 24 36 48 60 72 84 96 108 120 0 120 120
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the **Shader Tree**, if present, or dims the effect of the layer

Option	Description
	on the surface.
Shader	
Compute Shading	When enabled, computes the shading attributes for the scene (such as **Diffuse** or **Specular**). When applied as an item shader (where the shader, itself, is applied in the **Items** list rather than the **Shader Tree**), Modo computes the shader outside the **Shader Tree** and overrides the **Shader Tree** settings. When disabled, Modo reverts to the **Shader Tree** evaluations.
Control Quality	When enabled, activates the **Shading Rate**, **Multiplier**, **Saturation**, and **Indirect Illum Type** settings. When applied as an item shader (where the shader is applied in the **Items** list rather than the **Shader Tree**), Modo computes the shader outside the **Shader Tree** and overrides the **Shader Tree** settings. When disabled, Modo reverts to the quality setting evaluations of the **Shader Tree**.
Shading Rate	Determines how finely to shade each pixel when the material feeds the selected shader item during initial shading. (The **Shading Rate** is also known as the coarse shading rate.) The default value of **1** indicates that Modo evaluates the pixel once before being evaluated for further refinement. (See the **Refinement Threshold** option in the Render Item: Settings topic.) Decreasing the **Shading Rate** results in a more finely evaluated pixel, which generally yields longer render times. For example, if you change the value to **.5**, Modo cuts the pixel in half horizontally and vertically and evaluates each subpixel. In this case, you reduced the rate by half but increased the number of evaluation 4 times. Small changes to the **Shading Rate** can produce significant changes in performance. Generally, you can leave the **Shading Rate** at **1** and achieve refinement with the **Antialiasing** level or the **Refinement Shading** settings. However, when you see aliasing in reflections (or refractions) or on procedural textures, you could rectify the aliasing with the **Shading Rate** setting. Because doing this affects all the evaluated pixels, it is a brute force solution; and, therefore, you should manually restrict the pixels that are affected by the **Shading Rate**. Create shaders for specific problem areas with masks and only decrease the **Shading Rates** where absolutely necessary.
Direct Illum Multiplier	Affects the direct light intensity at the shader item level. Direct illumination is light coming from traditional 3D light sources (such as directional, spot, area, distant, cylinder, dome, and point lights). By default the value is set to **100%**. For example,

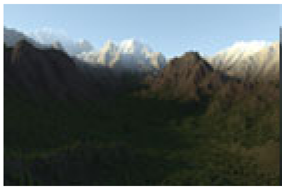
Option	Description
	if your project has three lights of varying intensity (such as 100%, 70%, and 30%), setting the **Direct Illum Multiplier** to **50%** decreases these values to 50%, 35%, and 15% respectively. You can use this value to adjust overall light effects as well as balancing direct and indirect light contributions.
Indirect Illum Multiplier	Provides a global method for controlling the indirect light contribution for the final render. Indirect light comes from global illumination sources (such as luminous geometry, image-based lighting through environments, and simple bounced light from diffuse surfaces). Adjust this option when indirect light sources are too subtle (or too pronounced) or to balance direct and indirect light contributions.
Incoming Indirect Saturation/Outgoing Indirect Saturation	Controls the strength of the colors (or saturation amount) generated by indirect illumination independent of the illumination itself. One of the many benefits of indirect illumination is the color bleeding between surfaces. The rays take into account not only the color of the lights and the environment, but the surface's color as well. You can adjust both incoming and outgoing rays. Increasing the value above **100%** makes the colors stronger and more saturated, and values below **100%** decrease the effect.
Indirect illum Type	Controls the type of global illumination per surface by using a shader with an associated mask. Generally, you control indirect illumination globally through the Render **Properties** panel on the **Global Illumination** tab. With the **Indirect Illum Type** setting, you have the option of using Irradiance Caching for an exterior scene, but using the Monte Carlo method on complex surfaces (such as the trees and bushes). You can create a selection set for just the trees and bushes (by choosing **Select > Assign Selection Set**) and in the **Shader Tree**, add an additional Material Group item above the base shader. In the group **Properties** panel, set **Tag Type** to **Selection Set** and select the named set under **Polygon Tag**. In this group, add its own shader and set the **Indirect Illum Type** to **Monte Carlo**. Now, most of the scene uses Irradiance Caching and only the trees and bushes use the Monte Carlo method. The **Indirect Illum Type** draws its values from the **Render Properties** panel on the **Global Illumination** tab. For the Monte Carlo method, Modo uses the number of rays indicated by the **Indirect Rays** value of the **Properties** tab of the Render Item.

Option	Description
Visibility	
Control Visibility	When enabled, computes the visibility attributes for the item or scene. When applied as an item shader (applied in the **Items** list rather than the **Shader Tree**), Modo computes the shader outside of the **Shader Tree** and overrides the **Shader Tree** settings. When disabled, Modo reverts to the **Shader Tree** evaluations for visibility.
Cast Shadows	When enabled, creates shadows for all elements controlled by the shader. When disabled, causes all elements controlled by the shader to be ignored by shadow rays. Light simply passes through the elements without creating shadows. Disabling this control does not disable shadow effects from indirect illumination because this control only affects direct light sources.
Receive Shadows	When enabled, receives ray-traced shadows from other geometry. When disabled, causes the elements to stop receiving ray-traced shadows from other geometry in the project.
Visible to Camera	When enabled, makes elements controlled by the shader visible to the camera. When disabled, hides any item or element controlled by the shader from the camera. These elements may still appear in shadows, reflections, and refractions even when hidden.
Visible to Indirect Rays	When enabled, uses elements controlled by the shader in any indirect light calculations. Indirect light is any light calculated during global illumination (such as luminous geometry, images used as light sources and other light bounced from geometry). When disabled, removes the elements controlled by the shader from indirect light calculations.
Visible to Reflection Rays	When enabled, reflects elements controlled by the shader in mirror surfaces. When disabled, removes the elements controlled by the shader from reflecting in mirror surfaces.
Visible to Refraction Rays	When enabled, makes elements controlled by the shader visible when looking at them through refractive surfaces. When disabled, causes the elements controlled by the shader to be invisible when looking at them through refractive surfaces.
Visible to Occlusion Rays	When enabled, uses elements controlled by the shader in any occlusion ray calculations. When disabled, causes the elements controlled by the shader to be invisible to occlusion ray calculations.

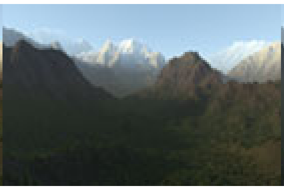
Option	Description
Alpha Type	Specifies how surfaces contribute to a rendered **Alpha** channel. • **Opacity** - Specifies the **Alpha** value based on the transparency of each individual surface. • **Constant** - Uses a specified, fixed **Alpha Value** for all associated surfaces. • **Shadow Catcher** - Makes all associated surface attributes invisible and exclusively renders shadows cast by direct and indirect illumination. This is a slightly different function from the other options. With this you can compose rendered images with background photos right in Modo. You can set a ground plane to catch the shadows cast by another object without contributing any shadows of its own. When you specify a photo as a background, Modo renders all the objects and shadows together in one pass.
Alpha Value	Overrides surface opacity settings and determines to what degree surfaces contribute to a rendered **Alpha** channel when **Constant** is the **Alpha Type**.
Light Linking	
Control Light Linking	When enabled, computes the visibility of light groups for a given item or surface. When applied as an item shader (applied in the **Items** list rather than the **Shader Tree**), Modo computes the shader outside of the **Shader Tree** and overrides the **Shader Tree** settings. When disabled, Modo reverts to the **Shader Tree** evaluations for light linking.
Light Group	Controls, on a shader level, what lights in a scene affect which surfaces. You can specify which group of lights to control. You make light groups in the Groups Viewport. To create a base group, click **new group**, rename it if you wish, and add items by selecting the lights in the 3D viewport. (Press **Shift** to select multiple items.) Now, right-click the group's title and choose **Add Items**. You can also drag items directly from the 3D viewport onto any Group Item to add them.
Mode	Specifies how the lights affect surfaces. If set to **Include**, the lights affect the shaded surfaces. If set to **Exclude**, the lights do not affect shaded surfaces.
Fog	
Control Fog	When enabled, produces a simulated (non-volumetric) fog effect by applying a user-defined color to relevant materials with increased intensity as the geometry

Option	Description
	is further from the camera. You can use this to fade objects into a background by increasing the apparent depth (and scale) of a scene. You can also specify fog in the Environment Material item with similar options. The **Environment Fog** option differs in that it always uses the environment color, but with the shader fog, you can specify any color.
Fog Type	Specifies the fog type. Click to view examples of various Fog Type settings. (Click an example to see a larger image.) • **None** - Disables fog from rendering. • **Linear** - Renders fog in the scene based on the **Fog Start Distance** and the **Fog End Distance**. (There is no fog up to the starting distance, but then surfaces attenuate toward a solid color and maximum fog density at the end distance.) • **Exponential** - Renders fog starting at the camera's position and increases in strength the further the surface is from the camera. Based on exponential values, the strength increases non-linearly and is directly influenced by the **Fog Density** setting. (For very large scenes, use a very small **Fog Density** value.) • **Underwater** - Renders fog starting at the camera's position and increases in strength the further the surface is from the camera — except with this type the fog absorbs the environment's color to produce an environment-colored cast. This type also reduces color saturation with distance. For this type, the example shows a late afternoon environment, so the reds of the setting sun influence the fog's color. For a true underwater look, define a blue or teal color (or set the environment to a 2- or 4-color gradient with blues or greens) and enable **Use Environment Color**. • **Layered** - Produces a ground fog-like effect where the fog density increases with both depth and height.
Use Environment Color	Gives the illusion that the mesh is fading into the background because it colors the fog based on the environment material or textures. This option is only available when you specify **Underwater** as the **Fog Type**.
Fog Color	Specifies the RGB color to apply to materials receiving fog.
Fog Start Distance/Fog End Distance	Specifies the starting and ending distances for the fog effect when the **Fog Type** is set to **Linear**.
Fog Density	Determines how thick the fog is just in front of the camera. The default value of

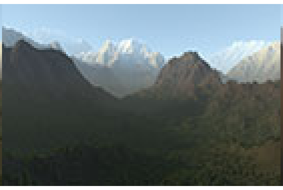
Option	Description
	10% indicates that geometry just in front of the camera has a 10% blend of the fog color. The fog's density increases as it recedes from the camera.



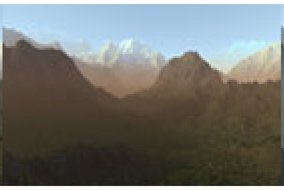
No Fog



Linear Fog



Exponential Fog



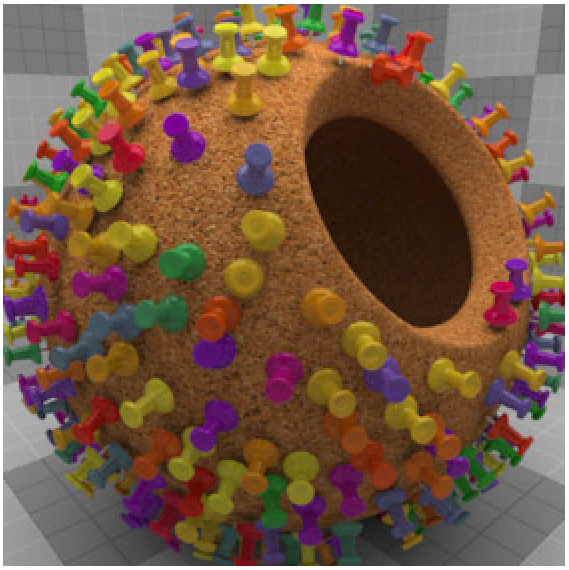
Underwater Fog

Layered Fog

Related Videos

Video: This video shows you how to make a Shadow Catcher in Modo.

Surface Generator



A Surface Generator works in tandem with a Replicator item to offer a greater degree of control over the random placement and density of replicated objects. Once you add a Surface Generator, you can use additional texture layers to control density, size, direction, and even coloration by setting the texture layer to the proper effect.

Note: For information about adding and working with Shader Tree item layers and setting their effect, see the Shader Tree topic.

Replicators, themselves, use vertex data to propagate replicas. Although there is an amount of randomization available for the replicas, there is still a certain degree of regularity to them. By using a Surface Generator, you can have Modo generate random points across the surface of a Mesh Item. The added Surface Generator then passes along this information to the Replicator for prototype propagation.

For example, if you model a tree and want to replicate leaves over its surface, you would want better control of the overall density than you could get by simply replicating them to only vertex positions. Additionally, you could use a Weight Map to limit the leaves' distribution on the branches and not the trunk. You could also make some additional color variations by using a gradient layer to produce far more realistic results.

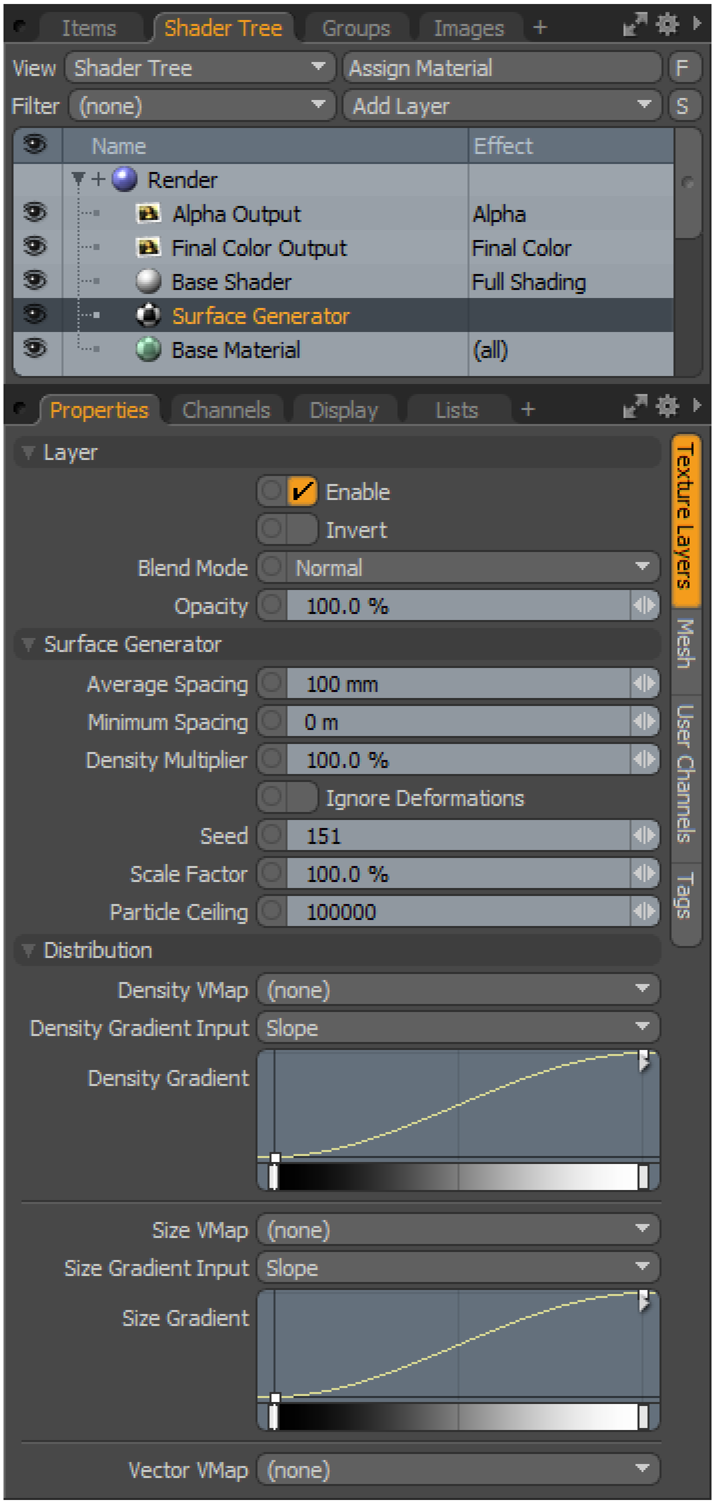
Tip: Similar to using a Surface Generator, using a Surface Particle Generator item is another way to add random placement to Replicator elements. However, because surface particle items work outside of the Shader Tree, the surface particle item cannot see surface deformations (such as **Displacement**). If that is required, then you must use a Surface Generator as a point source.

Adding a Surface Generator

To use a Surface Generator, you must first assign it to the target surface by creating an item mask. (Right-click the target item in the 3D viewport and choose **Create Item Mask**.) Then add the Surface Generator item to the group by adding it as a layer. (Choose **Add Layer > Special > Surface Generator**.) You can also add the layer to an existing surface by placing a Surface Generator into an existing material group. Once you define the Surface Generator for the target surface, you then only need to assign it as the point source in the Replicator Item. You can adjust Surface Generator settings as needed.

Tip: To add some variety to replicated objects, you could use groups. By creating a group from multiple Mesh Items and specifying that group as the prototype to be cloned, the Replicator pulls random items from within the group to add further randomness and variation to the cloned objects. Once you create a group, it shows up in the **Prototype** list of the Replicator item. Additionally, you can make the surface for replicas more random by using gradients set to the **Particle ID** input parameter.

When you select a Surface Generator item, its attributes appear in the **Properties** panel.



Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Surface Generator	
Base Material	(all)

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Surface Generator

Average Spacing: 100 mm

Minimum Spacing: 0 m

Density Multiplier: 100.0 %

☐ Ignore Deformations

Seed: 151

Scale Factor: 100.0 %

Particle Ceiling: 100000

Distribution

Density VMap: (none)

Density Gradient Input: Slope

Size VMap: (none)

Size Gradient Input: Slope

Vector VMap: (none)

Items
Shader Tree
Groups
Images
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S
Properties
Channels
Display
Lists
Density Gradient
Size Gradient
Texture Layers
Mesh
User Channels
Tags

Option	Description
Layer	

Option	Description
Enable	Toggles the effect of the layer on and off to duplicate the functionality of toggling visibility in the Shader Tree. When disabled, the layer has no effect on the shading of the scene. However, Modo saves disabled layers with the scene, and they are persistent across Modo sessions.
Invert	Inverts the colors (RGB values) for the layer to produce a photonegative effect. This is a standard texture layer control and provides no functionality specific to the Surface Generator.
Blend Mode	Affects the blending between different layers of the same effect type. With this, you can stack several layers for different effects. This is a standard texture layer control and provides no functionality specific to the Surface Generator. For more about blending, see the Layer Blend Modes topic.
Opacity	Changes the transparency of the current layer. Reducing this value increasingly reveals lower layers in the Shader Tree, if present, or dims the effect of the layer, itself, on the surface. This is a standard texture layer control and provides no functionality specific to the Surface Generator.
Surface Generator	
Average Spacing	Specifies the average distance between replicated items.
Minimum Spacing	Specifies the minimum amount of space between replicated items. This is especially useful in eliminating overlaps for neighboring replicas.
Density Multiplier	Controls the overall density of replicated items. Lower values reduce the overall number of replicas; higher values increase the overall number of replicas up to the **Particle Ceiling** value.
Ignore Deformations	When enabled, maintains particle positions on a surface through the deformation as if the particle positions are fixed if you apply an animated deformer (such as a morph or weight deformer). When disabled, the changes to the underlying surface from the deformer triggers Modo to re-distribute surface particles in a way that the particles retain their **Average Spacing** and **Minimum Spacing** values.
Seed	Specifies the initial number Modo uses when generating the procedural values. Different **Seed** values produce different random variations and can be useful in changing the texture result; however, you need to use the same **Seed** value when you want items to retain the same variations.

Option	Description
Scale Factor	Controls the scale of replicated objects to eliminating the need for manually scaling the base, prototype item. With a value of **100%**, Modo generates replicated items at the same size of the prototype. Smaller values decrease each replica's size; larger values would increase each replica's size.
Particle Ceiling	Specifies a maximum for the total number of generated replicas.
Distribution	
Density VMap	Specifies an embedded Weight Map on the target surface and uses its values to control the density of the particle distribution. Doing this eliminates the need to create custom image maps. You can select any available Weight Map from this list to control the density of the resulting particles. Weight Map values work as a multiplier over the normal rendered density. A value of **1** produces the unmodified 100% density and fades toward **0** to produce 0% density. Weight values can also be greater than **1** for added density control.
Density Gradient Input	Specifies the input criteria that Modo uses to control the density of the resulting particles based on the **Density Gradient**.
Density Gradient	Modifies the density. By using a mini-gradient input, you can modify the density with gradient keys that represent a percentage of the original density. Much like how a keyframe defines a value in time, gradient keys represent values across the selected input parameter (such as over the angle of **Slope**). You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. White produces a **100%** density attenuating toward black at **0%** density. If you open the Gradient Editor, you can specify values above 100%.
Size VMap	Specifies an embedded Weight Map on the target surface and uses its values to control the scale of the elements connected to the generated particles. Doing this eliminates the need to create custom image maps. You can select any available Weight Map from this list to control the scale of the resulting particles. Weight Map values work as a multiplier over the regular sizing. A value of **1** produces the unmodified 100% size and fades toward **0** to produce 0% scale. Weight values can also be greater than **1** for added scale control.
Size Gradient	Specifies the input criteria that Modo uses to control the scale of the elements

Option	Description
Input	connected to the generated particles based on the **Size Gradient**.
Size Gradient	Modifies the scale. By using a mini-gradient input, you can modify the density with gradient keys that represent a percentage of the original scale. Much like how a keyframe defines a value in time, gradient keys represent values across the selected input parameter (such as over the angle of **Slope**). You can add keys by middle-clicking along the length of the gradient input and then dragging the key icon up or down to adjust the value. The grayscale bar gives visual feedback as to how the values fade from one key to the next. White produces a **100%** scale attenuating toward black at **0%** density.
Vector VMap	Specifies the vector map to use its values to control the orientation and scale of the elements connected to the generated particles. If you create an embedded vertex vector map on the target surface, you don't need to create custom image maps to do so. With a vector map, you can use the sculpting tools (like with hair guides) to control the replicated elements. Elements controlled by vector Vertex Maps automatically adapt to surface deformations because the vector maps are defined in tangent space.

Working with Fur

Modo makes working with fur easy. First, select some polygons, and then create a material tag for the surface. Once you've tagged the surface, Modo automatically creates a Material Group Item with its own material layer in the Shader Tree. Select this new material item and choose **Add Layer > Special > Fur Material**. You now have fur on the surface. Of course, there are many controls to customize the fur to look different.

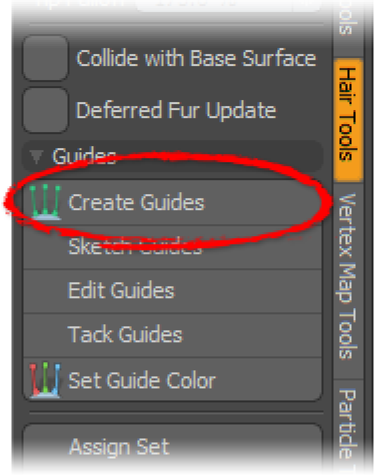
The Fur Material layer, itself, with its various settings offers a degree of control over the fur. In that layer, you set the size, length, kink, waviness, curls and other attributes of the fibers. This topic includes brief overviews of some additional techniques related to working fur. By using these techniques (such as working with guides, implicit UVs, and gradients) you can customize and control the look of your fluffy (or frazzled) fur.

Guides

Guides are useful to control fur in various ways (such as controlling its bending direction or length). In the 3D viewport representations, guides look like sparse fur fibers over the target surface. These guides, shown specifically for fur, are curves from polylines, which are a special type of curve that is made up from multiple, tiny line segments linked together. These curves are easier to manipulate with the styling tools and offer greater performance over regular spline curves. In all other respects they work and act just like the regular curves do.

Modo sets curves to control the fur fibers with the settings of the **Fur Material** sub-tab. As long as these polylines share the same material tag as the fur material layer, they work. To tag the polylines, select them in Polygons mode and assign a tag (as you do for regular polygons). You can draw hair guides manually with the Pen tool or create regular spline curves with the Curve, Bezier, or Sketch tools and then convert them to polyline curves with the Freeze function. (In the **Freeze Polygons** dialog, select **Polylines** from the **Freeze Curves as** list.) When drawing curves as fur guides, the root of the curve needs to originate close to the surface from which the fur grows. (Modo draws a small circle around the root vertex for all curves.)

Hair Guide Tool



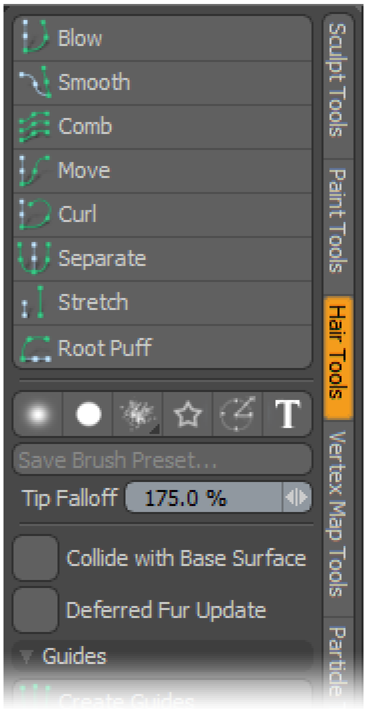
Creating guides manually can be tedious. An easier way to create guides is by using the Hair Guides function in the Painting/Sculpting toolbox under the **Hair Tools** sub-tab. This tool creates guides that inherit the material tag from the selected surface, so name the fur surface with a proper material before activating the tool. Select the polygons you want your fur to grow from and keep them selected when you activate the tool. The selection controls the growth area to limit the guides to the selected geometry.

To apply the tool to an area based on existing materials, you can recall the polygons selection that defined the area by using the Statistics panel and selecting the material by name under the Material section of the panel. Once you have the polygon selection and have activated the tool, with this **Properties** panel, click in the 3D viewport to actually generate the fur guides. With the tool, dragging to the left or right increases or decreases the length of the guides. You can also manually set the length directly in the tool properties. For detailed information about using this tool, see the Hair Guides topic.

Styling

Much like the sculpt and painting tools, Modo offers a variety of tools meant specifically for styling fur. Located in the **Paint** tab of the default workspace, the **Hair Tools** tab has everything you need for combing and sculpting fur. The double-duty tools style both guides and fur (by painting on image maps that control the styling of regular fur when you select **Edit Fur Map**). Image maps can control different aspects of fur material (such as length, density, direction, and curling). All aspects are available if you right-click in the **Effect** column of any procedural or image map texture layer. (Even procedurals can control a fur material layer.)

Hair Styling Tools


Blow
Smooth
Comb
Move
Curl
Separate
Stretch
Root Puff
Save Brush Preset...
Tip Falloff
175.0 %
Collide with Base Surface
Deferred Fur Update
Guides
Create Guides
Sculpt Tools
Paint Tools
Hair Tools
Vertex Map Tools
Particle

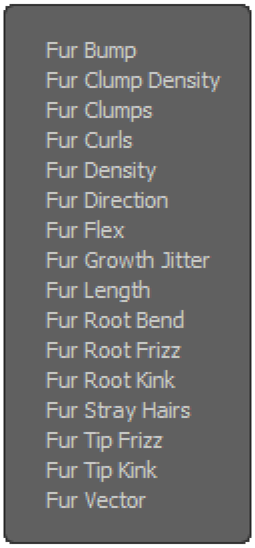
The **Hair Tools** tab has several brushes for styling fur in a variety of ways. See the Hair Brush Reference topic. These brushes work just like the painting and sculpting brushes, and you can enhance their performance by using a pressure-sensitive tablet to adjust the strength values dynamically. These tools are multi-purpose — in that you can use them to affect the guide curves or control the fur indirectly by using a fur vector map. (The application depends on the **Edit Fur Map** setting for each tool.)

Editing fur guides is intuitive. To modify fur guides, select a Hair Tools brush and drag it over some guides. (The guides must be visible on a foreground layer.) If you set the **Offset** amount of the brush high, that brush has a stronger influence over the adjustment; therefore, by adjusting the offset, you can make big changes or subtle adjustments with the same tools.

Controlling Fibers with Vector Maps

If you do not use guides (such as when working on short fur styling for a cat or other small animal), then you can use image-based vector maps to simultaneously control the length and bend direction for the fibers. Modo provides some preset scripts (at the bottom of the Hair Tools toolbox) for the most common fur styling tasks: **Add Fur Vector Texture**, **Add Fur Direction Texture**, and **Add Fur Length Texture**. When you activate these scripts, Modo automatically creates a blank image and applies it with the appropriate effect setting in the Shader Tree for the fur material. (A fur material item must be present first.) Once Modo creates the correct image, you can use any of the available brushes to control the fur layer. When adjusting fur this way, you see the GL preview fibers created by the fur material, itself, move and change as you apply the brush strokes.

Texturing Fur with Shader Tree Layers

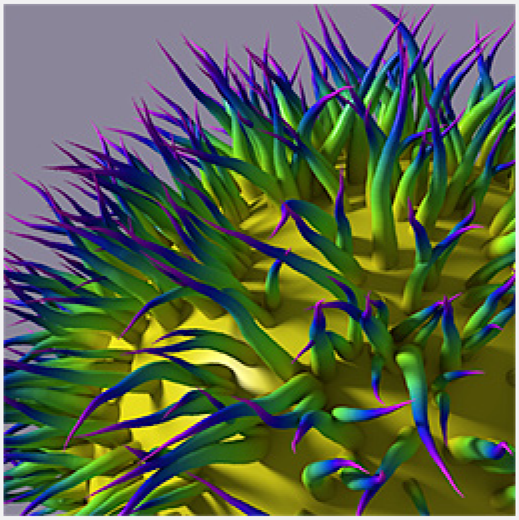


Fur layers take all their color and shading information from the material item in the same layer group. For example, set the material to red and you get red fur; set it to 50% transparent and the fur becomes transparent. Most fur and hair has a great deal of variation to it in length, color, and its growth pattern. The **Fur Material** sub-tab has jitter setting to control the growth part. Texture layers are a great way to control how the fur looks and add any necessary variation. Texture layers can also add specific, highly accurate control to fur styling and coloration. Modo has several effect channels that you can apply to both procedural and image map texture layers.

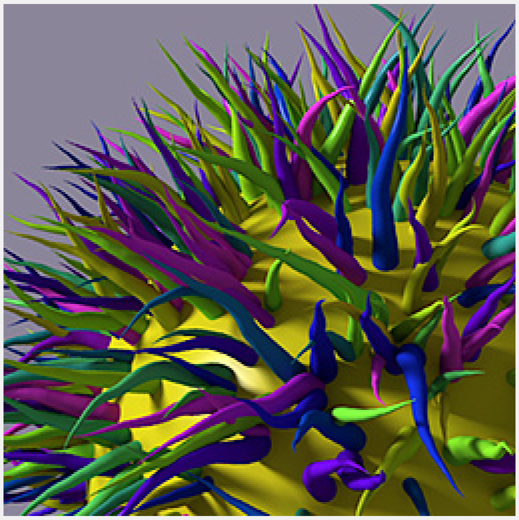
By adding a blank image to your material group (**Images tab > New Image**) and setting the layer to your intended effect, you can use any of the painting tools in Modo to adjust, color, and style fur. (A little paintbrush icon appears in the **eye** column of the Shader Tree to let you know that's the current layer.) The fur material gets its shading from the material layer, so all the ways you adjust material with texture layers are also available for the fur layer. For example, if you apply an image map as diffuse color, each fiber gets its color from the underlying texture layer's color.

Gradients

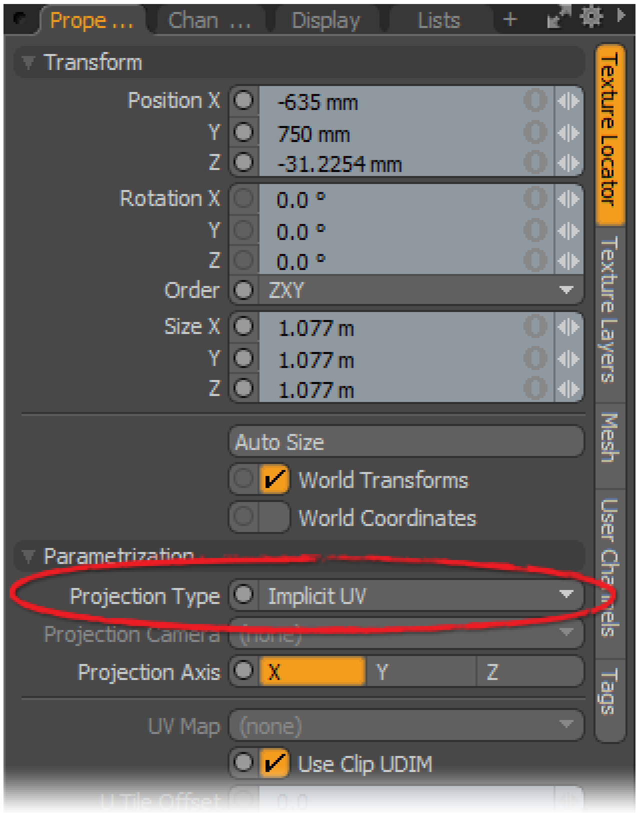
Gradients are also useful for fur layers. Besides all the ways a gradient normally controls a layer's surfacing parameters, you can use gradients to control each fiber along its length by applying the **Fur Parametric Length** setting as the gradient's **Input Parameter**. Now, whatever you set as the effect for the gradient to control (such as **Diffuse Color** or **Transparent Amount**) applies from a fiber's root at **0%** to the tip at **100%** regardless of the length. Perhaps you want to mix up the colors of the fibers or simulate a salt-and-pepper effect. Those are possible with gradients as well. As Modo creates each fiber, it assigns a random value between 0 and 1. By setting the gradient's input parameter to **Particle ID**, you can add some randomization to the individual fibers with ease as it takes that random number and pulls the color from the gradient based on its value.



A fur **Parametric Length** gradient applied along the length of the fur.



A **Particle ID** gradient applied varying the individual colors.



Transform

Position X -635 mm
Y 750 mm
Z -31.2254 mm
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °
Order ZXY
Size X 1.077 m
Y 1.077 m
Z 1.077 m

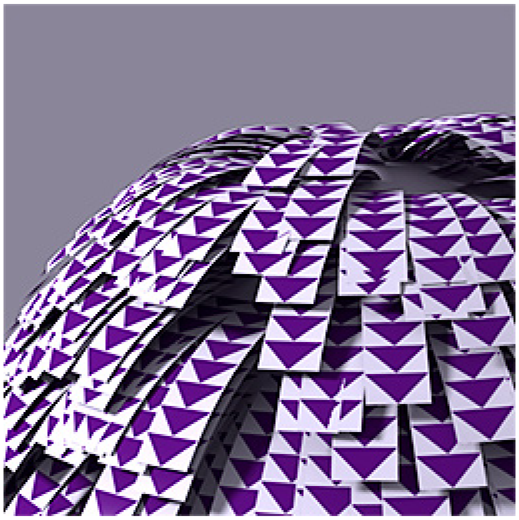
Auto Size
☑ World Transforms
☐ World Coordinates

Parametrization

Projection Type Implicit UV
Projection Camera (none)
Projection Axis X Y Z
UV Map (none)
☑ Use Clip UDIM
U Tile Offset 0.0

Implicit UVs

UV maps can control how Modo applies images to surfaces. It is possible to apply image maps to individual fibers. When you have your fur type set to **Strips**, images map to each strip individually. When the fur type is set to **Cylinders**, images wrap around each fiber cylindrically. You can also map images this way for the Billboard features. Applying images for fur has many uses (such as simulating feathers and making photo-realistic grass). You add a texture to individual fibers by adding an image map layer to the material group. In the Texture Locator item's properties, select **Implicit UV** as the projection type within the texture locator. Once you do that, Modo applies the image to each and every fiber as it renders the fur.



Material Layers and Multi-layered Fur

Each fur layer pulls its shading from the material layer, but you may want the fur material to be different from the underlying surface. That's possible as well.

Video: Watch this video to see how to do this.

Layer Effects

By default, any new texture layer added to the **Shader Tree** modulate the **Diffuse Color** channel, meaning the layer affects the apparent visual color of the surface. If you want any layer to affect the surface differently, such as applying a bump or displacement map, you can specify this in the **Effect** column of the **Shader Tree**. A layer can be set to modulate a specific effect, by right-clicking over the name in the **Effect** column and then selecting a new type from the contextual menu.

Layers can also be added and affect other items, such as changing the function of a render output, or using procedural textures to modulate a volumetric light's intensity.

Depending on the type of layer selected, different contextual menus are displayed, revealing different options. Each is fully covered on their respective pages - click the titles below to view the associated content.

- Basic Render Outputs
- Effect - Texture Item
- Effect - Shader Item

- Effect - Environment Item
- Effect - Light Item
- FX Item

Effect - Render Item

Rendering is the act of converting the scene information into a bitmap image by evaluating all the geometry and lighting information in a given file. Generally, you'd expect the resulting rendered image to be a fully ray traced and shaded image with full shadows, reflection and the like, much like a photograph depicts an entire scene when taking a picture with a camera. In Modo these images are the result of what are called "Render Outputs", the visible representations of the full scene evaluations. Each Render Output is a separate layer in the Shader Tree and produces its own image layer as part of the render calculations. Every time a Render command is invoked, all of the render output layers are simultaneously calculated and represented in the Render Display viewport. The different layers can be viewed by choosing one of the specific 'Output' options within the display window.

Occasionally, only a subset of the full evaluation is required, such as the diffuse color layers without any shadows. Modo offers many different Render Output types useful in a variety of situations ranging from purely diagnostic, to specific compositing utility in an external application. A default scene automatically includes a **Final Color** and **Alpha** output which should be fine for most situations. Additional layers can be added from within the **Shader Tree** itself, by clicking the **Add Layer** option of the full viewport window and selecting **Special** > **Render Output** from the pop-up menu. This adds an additional layer to the top of the tree. Users may then right-click in the **Effect** column of the Render Output for a pop-up context menu that allows one to select from the alternate output types. See the Render Outputs topic for for information regarding the editable attributes of the individual outputs. For more information on the render output types, see Render Output Items.

Effect - Texture Item

By default, any new material tag that you apply to a selection of polygons gains a Material item layer in the **Shader Tree**. Materials control the most common surface settings directly (such as **Diffuse Color**, **Specular Amount**, **Reflection Amount**, and **Transparency**). These settings are quite powerful on their own, but rendered surfaces that solely use the basic material settings can look flat or uninteresting. By adding texture layers (procedural or image-map based ones), you can achieve richer and more realistic surfaces.

Note: For information about adding and working with Shader Tree item layers, see the Shader Tree topic.

Once you add a layer, you can right-click the effect name for the layer (in the rightmost column) and direct how to have the texture affect the surface. Select a **Category** button to see additional samples.

Basic Channels

Clearcoat Amount

Adds an additional unblurred and untinted reflective layer to a surface to simplify the simulation of multiple layers of paint (especially automotive). When you set a texture layer to this effect, it controls the amount of **Clearcoat** across the surface. In the example, an additional **Noise** procedural texture with a high **Gain** value makes the result more apparent. With color image maps, Modo only uses the brightness (luminosity) of the image.

Diffuse Amount

Acts as a multiplier with the **Diffuse Color** setting to represent how much light is reflected from the surface. In the real world, all objects absorb certain wavelengths of light while reflecting others. The light that is reflected is what you see as the color of the object. When you set a texture layer to control the **Diffuse**

Amount, it darkens the color of the surface. The more light that is absorbed, the darker the surface appears. Controlling the **Diffuse Amount** is especially important for reflective surfaces. Reflections are an additive effect and otherwise would appear too bright. You can adjust this automatically by enabling **Conserve Energy** in the material item layer.

Diffuse Color

Determines the color of the surface when lit by sources of pure white light (whether directly or indirectly). This is represents the light that bounces from the surface and, effectively, is the color of the object.

Diffuse Roughness

Simulates a microscopic roughness effect by modifying the standard shading model to produce a more realistic matte surface. Extremely matte surfaces (such as stone, concrete, or plaster) reflect light very differently than smooth or glossy surfaces (such as glass or plastic). This is because on a microscopic level the matte surface is actually quite rough and spreads the light outwards to make the surface appear flatter. Any texture layer set to **Diffuse Roughness** modulates the **Diffuse Roughness** amount. A black value in the texture equals a **0%** value and attenuates toward white, which represents a **100%** value. When using color image maps, Modo uses only the brightness (luminosity) of the image.

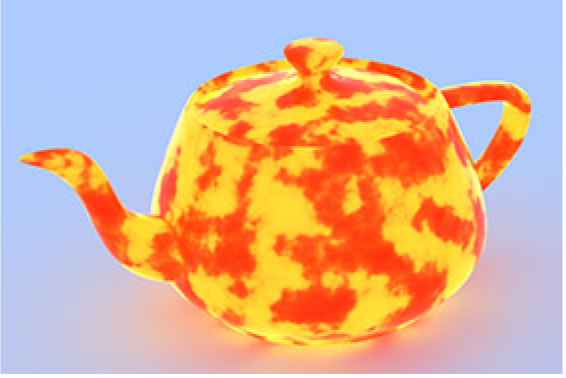
Dissolve

Fades out all surface settings together. The **Dissolve** effect is different from a **Transparency** effect. **Transparency** does not affect additive surface effects (such as **Specular** or **Reflections**), but **Dissolve** does. When you set any texture layer as **Dissolve**, the black areas are completely opaque and they attenuate toward white, which is completely transparent. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Luminous Amount

Modulates the **Luminosity** setting of the Material item layer to control how much illumination radiates from the surface. (Illumination only contributes to scene shading when you enable **Global Illumination**.) A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Luminous Color



Controls the **Luminous Color** setting of the Material item layer to control the color of any illumination radiating from the surface. (Illumination only contributes to scene shading when you enable **Global Illumination**.)

Reflection Amount

Modulates the **Reflection Amount** setting of the Material item layer to control how reflective (mirrored) a surface is. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image. In the rendering process, reflections are an additive process. To avoid reflections from becoming too bright, reduce the **Diffuse Amount**, which is the inverse of the **Reflective Amount**. Modo does this automatically if you enable **Conserve Energy** for the material item.

Reflection Color

Controls the **Reflection Color** setting of the Material item layer to control the color of any reflective areas of a surface.

Reflection Fresnel

Controls the **Reflection Fresnel** setting of the Material item layer to control the amount of any glancing angle (Fresnel) reflectivity. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Refraction Roughness

Modulates the **Refraction Roughness** setting of the Material item layer to control how sharp or blurry refractions are through transparent surfaces. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Roughness

Modulates the **Roughness** setting of the Material item layer to control the apparent roughness of a surface's specularity or reflectivity when you enable **Blurry Reflection**. The rougher the surface is, the more spread out the specular highlights are to simulate the look of microscopic bumps on a surface. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Specular Amount

Modulates the **Specular Amount** setting of the Material item layer to controlling the specular reflection over a surface. Specularity is a simulation of a direct light source's reflection on a surface. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Specular Color

Controls the **Specular Color** setting of the Material item layer to control the color of any specular reflections on a surface.

Specular Fresnel

Controls the **Specular Fresnel** setting of the Material item layer to control the amount of any glancing angle (Fresnel) specularity. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Subsurface Amount

Modulates the **Subsurface Amount** setting of the Material item layer to controlling subsurface scattering on a surface. This is the simulation of light bouncing around inside a surface before exiting. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Subsurface Color

Controls the **Subsurface Color** setting of the Material item layer to control the color of any subsurface scattering on a surface.

Transparent Amount

Modulates the **Transparency** setting of the Material item layer to control how clear (or see through) a surface is. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image. **Transparency** does not remove additive effects (such as **Specularity** and **Reflectivity**). In cases where this is necessary, use the **Dissolve** or **Stencil** effects.

Transparent Color

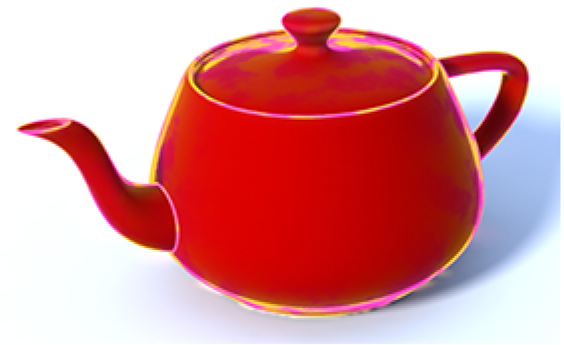
Modulates the **Transparent Color** setting of the surface. Some transparency needs to be present to see its effect on the surface. (Using an image map texture produces a result similar to a projector transparency slide where the image is clearly visible but completely transparent.)

Cel Edge Material Channels

Edge Width

Modulates the **Edge Width** value of the Cel Edges texture layer.

Edge Color



Determines the **Edge Color** of the Cel Edges texture layer.

Hair Material Channels

Glints Color: Determines the **Glints Color** of the Hair Material texture layer.

Primary Highlight Color: Determines the **Primary Highlights Color** of the Hair Material texture layer.

Rimlight Color: Determines the **Rimlight Color** of the Hair Material texture layer.

Secondary Highlight Color: Determines the **Secondary Highlight Color** of the Hair Material texture layer.
Fur Bump

Controls the **Flex** direction for fur growth. Generally, as fur grows, it bends toward the ground, which by default is negative **Y** (assuming this is the direction that gravity pulls the fibers). This is similar in function to **Fur Direction**, but **Fur Direction** requires a 32-bit image map for combing. **Fur Bump** angles the fiber direction similar to how a bump map perturbs the surface rays. Then you can use arbitrary texture layers (such as image maps and procedural textures) to control the fur's direction. The fur item's **Fur Bump Amplitude** setting controls the strength of the **Fur Bump** effect.

Fur Clump Density

Modulates the fur **Clumps** setting of the fur item layer to control the strength of how tightly clumps of fur gather together. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Fur Clumps

Determines whether or not fur clumps together. Clumped fur uses the strength value of the **Clump** input field for clumping areas attenuating toward no clumping. The results of **Clumps** and **Clump Density** are similar. The difference is that **Clumps Density** controls the strength of the clumping, and **Clumps** controls whether or not a fiber clumps. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Fur Curls

Modulates the fur **Curls** setting of the fur item layer that controls the strength of the corkscrew curls added to each individual fiber. This layer may require additional segments to properly render the curly fur. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Fur Density

Modulates the **Render Density** setting of the fur item layer that controls the thickness of the fur (number of fibers). A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Fur Direction

Controls the **Flex** direction of the fur item layer. Generally, as fur grows, it bends toward the ground, which by default is negative **Y** (assuming this is the direction gravity pulls the fibers). The **Fur Direction** effect requires a 32-bit HDR image format(such as EXR) and is similar to **Vector Displacement**. You can create this type of image in Modo by using **Add Fur Direction Texture** (for more information, see Adding Images). Once you create and apply the basic image, you can use the styling tools of the **Paint** tab to control the direction of the fiber's growth independent of other settings. Make sure to enable **Edit Fur Map** of the styling tools when sculpting fur this way. To control both **Growth Direction** and **Length**, use the **Fur Vector** setting.

Fur Flex

Modulates the **Flex** setting of the fur item layer to control the amount of bend applied to individual fibers. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Fur Growth Jitter

Modulates the **Growth Jitter** setting of the fur item layer to control how uniform or random the fiber growth is. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**.

Fur Length

Modulates the **Length** setting of the fur item layer as a percentage of the total length. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Fur Root Bend

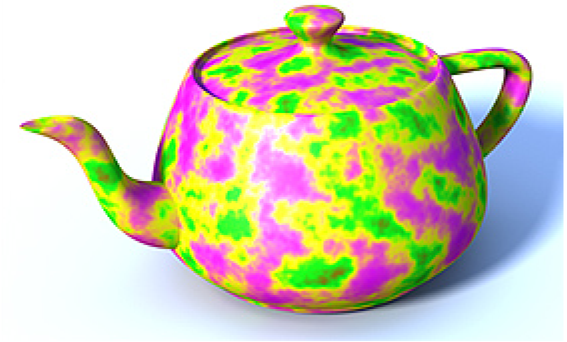
Modulates the **Fur Root Bend** setting of the fur item layer, which controls the angle at which the fiber grows from the polygon surface. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**.

Fur Vector

There are two ways to style fur surfaces in Modo. Guides are the most common as they are easy to visualize and control. You can use 32-bit HDR images as such (much like those meant for **Vector Displacement**). You can apply an HDR image to the Shader Tree and assign the **Fur Vector** effect, which is effective for shorter fur type surfaces than longer, styled hair. You can create this type of image in Modo by using the **Add Fur Direction Texture** option (for more information, see Adding Images). Then, use the styling tools of the **Sculpting** tab to interactively control various properties of the fur. Make sure to enable **Edit Fur Map** of the styling tools when sculpting fur this way.

Shader Control

Driver A, B, C, and D

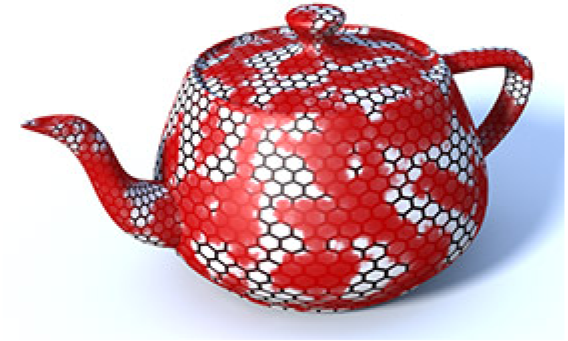


Controls a Gradient layer by using one or more texture layers set to the same drive letter (A, B, C, or D). Modo feeds the values of the layer(s) into any gradient item layer with a matching input parameter. Within the gradient, you may apply colors or values based on the values of the input textures. This is useful to remap the shades of an image map texture or to achieve finer control beyond the **Bias** and **Gain** settings in any procedural texture.

Group Mask

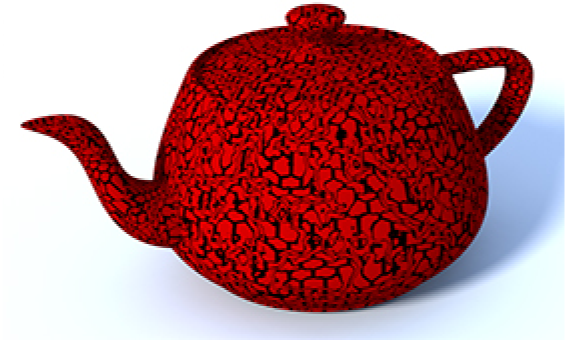
Masks the effect of all layers within a given Group Item. Due to the layered nature of the **Shader Tree**, this reveals any lower Shader Tree layers. Using multiple tags applied to multiple group items produces some interesting effects. The example masks a group item that has a displacement layer and a gradient layer that simulates a muddy look.

Layer Mask



Masks the effect of any layer directly above it in the **Shader Tree**. You can drop any texture or procedural layer onto another layer to automatically set it as a layer mask. With this, you can individually mask the contribution of a single layer in the **Shader Tree**. The example adds a **Grid** texture layer that is masked by the **Noise** procedural layer, which is set to **Layer Mask**, directly below it in the **Shader Tree**.

Texture Offset



Displaces the values of one layer with another (defined as the **Texture Offset**). (This is like a **Vector Displacement**, but for textures instead of surfaces.) The **Texture Offset** layer modulates any and all layers above it in the **Shader Tree** within the same material group mask. (Make sure you don't place any texture layers above it that you don't want distorted.) Textures can be either grayscale or RGB layers. RGB layers produce a more complex offset effect with the additional channel information for multi-directional offset. (This is similar to how RGB values work for **Vector Displacement**.) The example adds a standard **Noise** texture layer as an offset to a hexagon grid pattern. The **Texture Offset Amplitude** setting of the Texture Locator controls the strength of the offset. You can adjust this further by modifying the texture layer's **Opacity** setting.

Special Effects

Anisotropy Direction

Even on very smooth surfaces, microscopic scratches affect the way lights reflect across them. Their effect is most apparent through the reflection and specular attributes drawing them out in a parallel direction to the scratches. Surfaces that are direction dependent are considered anisotropic. It is especially apparent on machined surfaces (such as brushed metals), but this is also visible on many fabrics (such as silks and carbon fiber). Anisotropic maps require a special RGB setup for proper results. To make your own anisotropic maps, see the Anisotropic Direction topic.

RGBA

Makes only the opaque areas of the image visible on the surface when you set a texture layer to **RGBA** and use an image with embedded transparency (such as a TGA or PNG file). All other areas of the surface are invisible — including the associated attributes (such as **Specular** and **Reflections**). You can define both opacity for a surface and the diffuse color with a single texture layer. This is similar to having an image for **Diffuse Color** and then another Alpha image set to **Stencil**. This effect is not appropriate if you need variable transparency.

Stencil

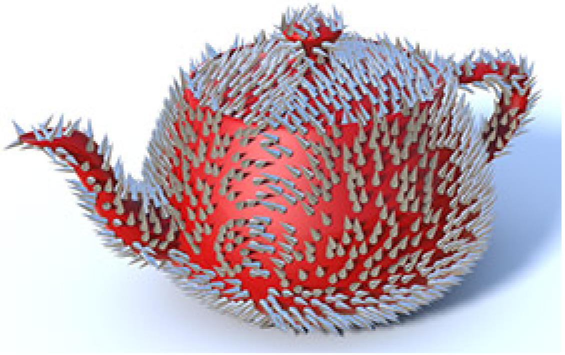
Truncates all surface settings together. **Stencil** is different from '**Transparency**. (**Transparency** does not affect additive surface effects, such as **Specular** or **Reflections**, but **Stencil** does.) **Stencil**'s concept is identical to **Dissolve**'s, but **Stencil** does not allow for gray shades between black and white, so it always produces a hard edge. Because of this, it is also much faster to compute to provide faster results when many transparent surfaces overlap (such as leaves in a tree).

Surface Particle

Surface Particle Density

Controls the distribution of replicas procedurally by using a texture layer when using **Replicators** and **Surface Generators** to duplicate geometry over the surface of an object (for example, to make grass or water droplets). The texture layer attenuates the total density as defined by the **Minimum** and **Average** spacing options of the **Surface Generator**. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Surface Particle Normal



Controls the normal direction of replicas procedurally as an offset from the surface normal when using **Replicators** and **Surface Generators** to duplicate geometry over the surface of an object (for example, to make grass or water droplets).

Surface Particle Size

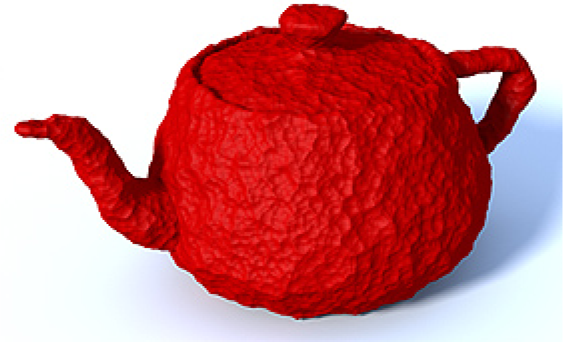
Controls the scaling of replicas procedurally by using a texture layer when using **Replicators** and **Surface Generators** to duplicate geometry over the surface of an object (for example, to make grass or water droplets). The texture layer sizes the replica as a percentage of the default object size. A black value in the texture equals **0%** and attenuates toward white, which represents **100%**. When using color image maps, Modo uses only the brightness (luminosity) of the image.

Surface Shading

Bump

Simulates small surface details that would be difficult or impossible to model directly. Bump maps are similar to displacement maps; however, they do not displace the geometric surface itself, but only the rays that hit it. In extreme cases this weakness shows on silhouette edges. Because of this, keep bump mapping to smaller details that wouldn't be noticeable at glancing angles. because bump maps do not require additional geometry, they render quickly. Bump maps require a grayscale image or a procedural layer. When you apply color images and set the layer as **Bump**, Modo uses only the brightness (luminosity) of the image.

Displacement



Adds detail to a surface (like a Bump map). When using Modo's sculpting tools (or an external application such as ZBrush™ or Mudbox™), displacement maps can provide an organic method of modeling. Displacement maps use the micropolygon tessellation feature of Modo to create additional geometry on subdivision surface models. The displacement map moves the surface a given distance along its normal defined by the **Displacement Distance** setting of the material item and the gray shades of the texture layer, itself. This results in a model with greater surface detail and a more organic form that is easier to obtain. Often, a displacement map has a gray middle ground with the dark values pushing the surface in and white values pushing the surface out. You achieve this by setting a inverse (negative) value for the **Low** setting of the image map layer's property or by setting a negative value in a procedural texture.

Normal

Affects the shading normal of the surface to provide a way to simulate high-resolution detail on a low-resolution polygon model. Normal maps are generally generated through by baking, in which Modo projects the surface detailing of a high-resolution model onto a low-resolution model and maps the resulting image onto it. At render time the low-resolution model look much like the high-resolution version at a much lower processor use. This is a popular method for generating geometry for realtime environments (such as games and simulations). For a normal map to render properly, you must use a UV map.

When you set the texture **Effect** to **Normal** type you may see a dialog asking, "**Warning: Normal clip is using a color correction, do you want to set it to none?**". By default, Modo uses sRGB color correction for all clips. In most cases this works well, however, it is usually not correct if the clip is being used as a normal map. And so the dialog allows you to choose whether to use color correction for the normal map. The dialog has the following options:

- **Yes** - Convert the current clip to (none) color space.
- **No** - Don't convert, accept the image's color space.
- **Yes to All** - Always convert to (none) color space, until Modo quits.
- **No to All** - Always don't convert to (none) color space, until Modo quits.
- **Cancel** - Don't use this feature. Equivalent to **No to All**.

Note: The settings are reset when you relaunch Modo.

Note: You can change the default colorspaces in the Color Management.

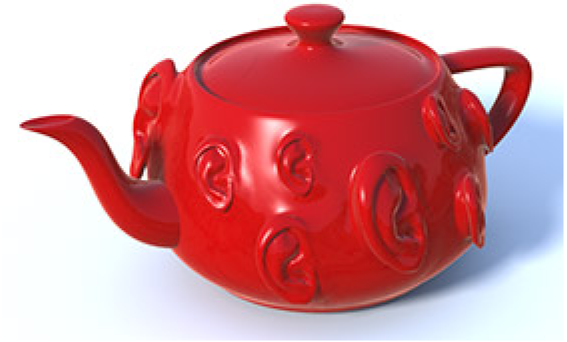
Object Normal

Object Normal maps are typically used in Games workflows for static, non-deforming objects (environment, props, and so on) to render quickly in game engines.

Standard normal maps encode the direction of the surface relative to the tangents of the surface. For example, they describe the surface as it would appear to an ant standing on the surface. Modo encodes the normals relative to the surface tangents. This means, Object Normal maps encode the direction in absolute terms in object space. For example, pure green means the surface is pointing straight up along the object's Y axis.

To set the texture **Effect** to **Object Normal**, in the **Shader Tree**, right-click on the **Effect** column for a material and select **Surface Shading** > **Object Normal**. The baking procedure is the same as Normal texture effect. For more information, see Bake from Object to Selected Texture.

Vector Displacement



Sculpts very highly detailed forms simply. **Vector Displacement** is similar to displacement mapping that pushes micro polygons outward along their normal a given distance. Displacement maps are only capable of moving a polygon in a single forward direction, but a vector displacement value can record all three directional values to the image map. With this, a **Vector Displacement** can produce overhangs and undercuts to a sculpted surface. To store the necessary information, a **Vector Displacement** requires an HDR image format (such as EXR).

Volumetrics

- **Volume Sample Density**: Determines the density value of the volume itself. This value is evaluated when sampling along the ray of a given 3D volume. The ray is used to calculate the overall density of the volume. This option should be used as an input parameter for a Gradient.
- **Volumetric Absorption Amount**: A scale value to define how much light energy is absorbed by the volume.
- **Volumetric Absorption Color**: Determines the color of the light that is absorbed by the volume.
- **Volumetric Ambient Color**: Determines a constant color value for the display of a volume.
- **Volumetric Density**: The density value of the hypertexture. Like a 2D texture used on a surface, a hypertexture is a 3D texture used on a volume.

  Volumetric Density is the base material density value. When the volume is rendered, samples are collected along the ray when the ray intersects a volume. This is a process called Ray Marching. The rays are broken into small steps and for each step along the way, it collects a volume sample. The **Volume Sample Density** is then multiplied by the **Volumetric Density** effect, which is controlled by the Volumetric part of the material. When all samples have been computed, the ray is flattened to produce a single scattering and absorption pair for the ray.
- **Volumetric Level**: Determines the offset of the **Volume Sample Density** value. The default value is 1.0.
- **Volumetric Luminosity Amount**: Determines the scale value to control how much light energy is emitted from the volume. This option is used for self-luminous objects.
- **Volumetric Luminosity Color**: Determines the color of the light that is emitted from the volume. This option is used for self-luminous objects.
- **Volumetrics Scattering Amount**: Specifies the scale value to control how much light energy is scattered from the volume. When set to a smaller value, the scattered energy is less.
- **Volumetrics Scattering Color**: Determines the color of the light that is scattered from the volume.

Volumetric Examples

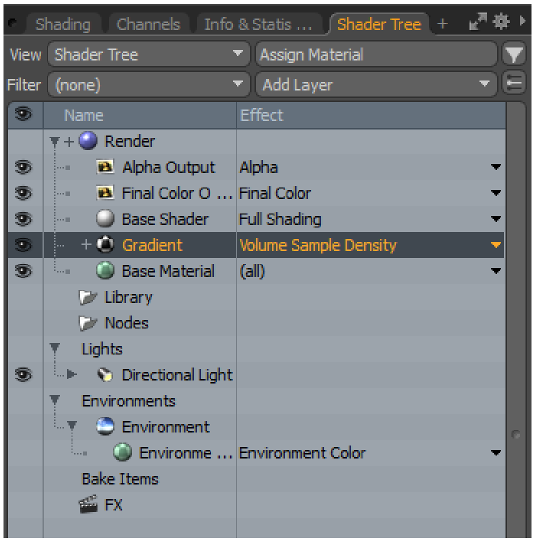
The following procedures outline how to create Volumetric samples using the above parameters. Each sample is built on top of the previous sample.

Creating a Volume Sample Density Sample

1. Click **File** > **New** and then open the **Render** layout.
2. Add a volume to your scene. On the right-hand side pane, open the **Item List**, click **Add Item**, select **Volumes** and then double-click on **Volume**.

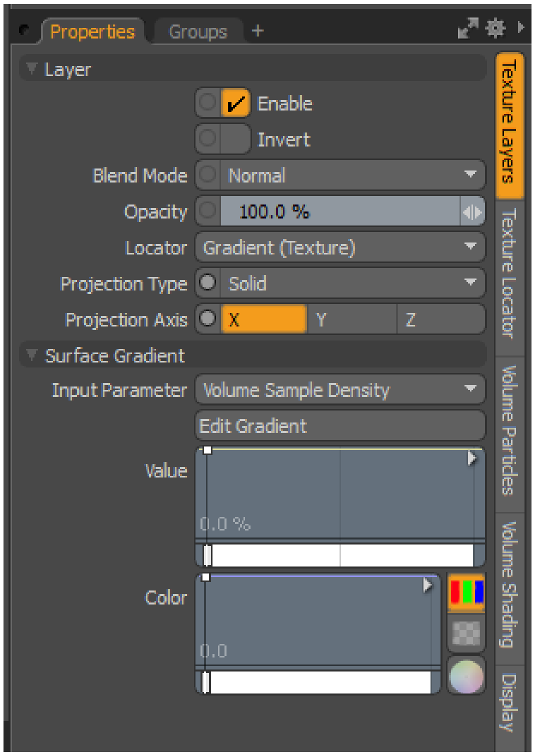
A volume is added to your scene. In addition, by default, a Directional Light and an Environment Material are also added to your scene.

3. In the Shader Tree, expand **Environments** > **Environment**, and toggle the eye button (👁) to off, so that the Volumetric effect is easier to see. Click **Play** in the Render view to see a darker background.
4. In the Shader Tree, click **Add Layer**, and select **Processing** > **Gradient**.
5. In the Gradient **Effect** column, click the down arrow, and select **Volumetric Effects** > **Volume Sample Density**.



Name	Effect
Render	
Alpha Output	Alpha
Final Color O ...	Final Color
Base Shader	Full Shading
Gradient	Volume Sample Density
Base Material	(all)
Library	
Nodes	
Lights	
Directional Light	
Environments	
Environment	
Environme ...	Environment Color
Bake Items	
FX	

6. In the **Properties** panel, open the **Texture Layers** tab, expand **Surface Gradient**, and then set the **Input Parameter** to **Volume Sample Density**.


Properties
Groups
+
Texture Layers
Texture Locator
Volume Particles
Volume Shading
Display

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

Projection Type: Solid

Projection Axis: X Y Z

Surface Gradient

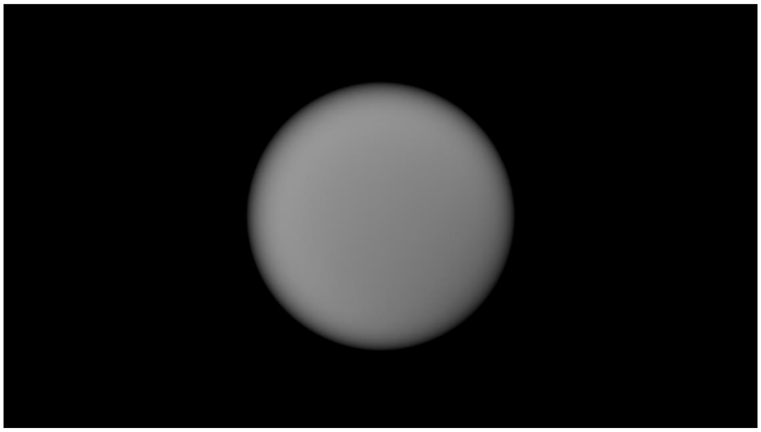
Input Parameter: Volume Sample Density

Edit Gradient

Value: 0.0 %

Color: 0.0

The Render viewport displays something similar to the following.

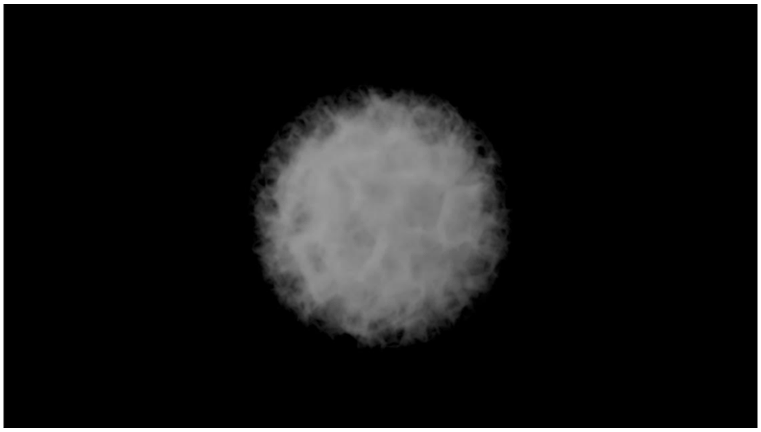


Creating a Volumetric Density Sample

Continue using the same scene.

1. In the Shader Tree, click **Add Layer**, and select **Textures** > **Cellular**.
2. In the Shader Tree, under the **Effect** column, click the down-arrow, and set **Cellular** to **Volumetrics Effects > Volumetric Density**.

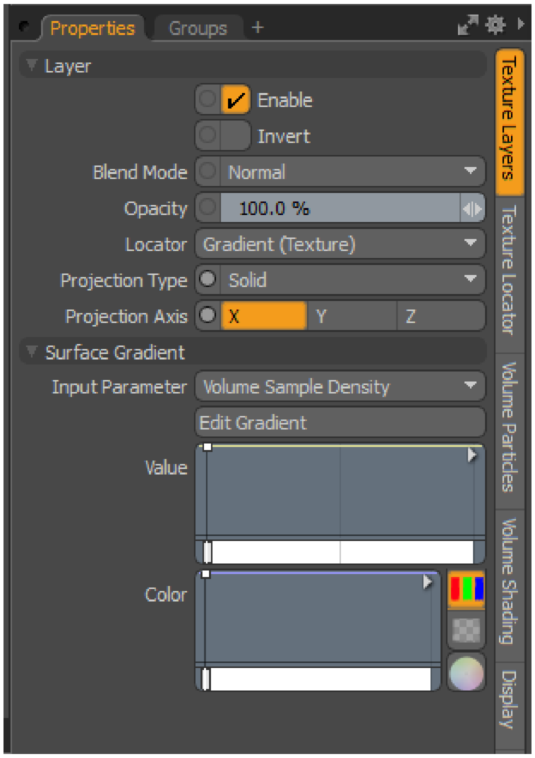
   The Render viewport displays something similar to the following.



Creating a Volumetric Scattering Color Sample

Continue using the same scene.

1. With the **Gradient** texture selected, click the down arrow in the **Effect** column, and select **Volumetrics Effects** > **Volumetrics Scattering Color**.
2. Next, change the color of the Gradient texture. In the **Properties** panel, open the **Texture Layers** tab, and then expand **Surface Gradient** properties.



Properties
Groups
+
Texture Layers
Texture Locator
Volume Particles
Volume Shading
Display
Edit Gradient
Value
Color

Layer

☑ Enable

☐ Invert

Blend Mode: Normal

Opacity: 100.0 %

Locator: Gradient (Texture)

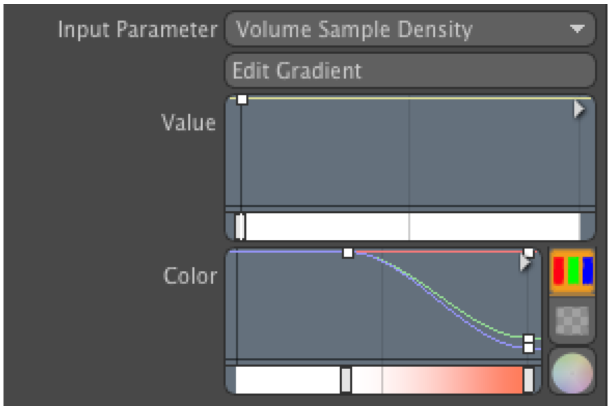
Projection Type: Solid

Projection Axis: X Y Z

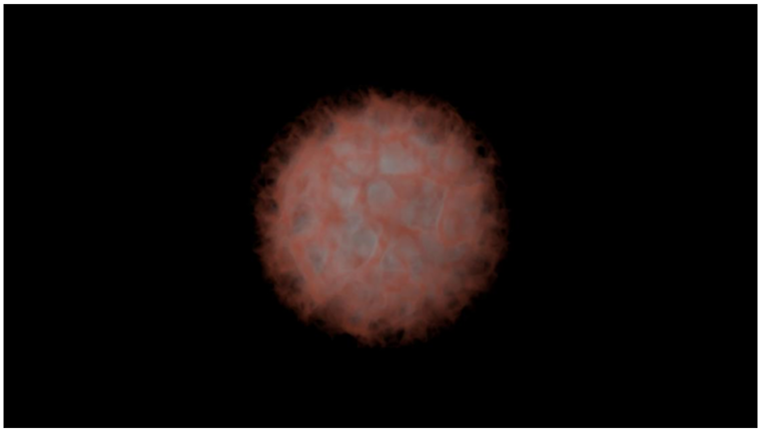
Surface Gradient

Input Parameter: Volume Sample Density

3. In the **Color** graph, select a key frame, and then click the **Color Picker** icon. For more information about how to use the Color Picker, see Color Picker Viewport.
4. Set the Volumetric Scattering Color to red.


Input Parameter Volume Sample Density

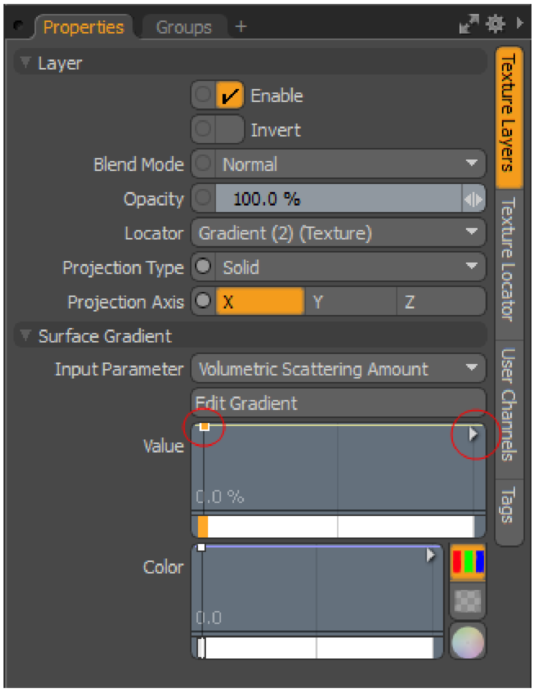
The Render viewport displays something similar to the following.



Creating a Volumetric Scattering Amount Sample

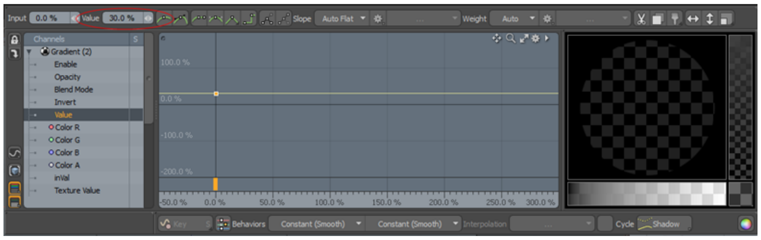
Continue using the same scene.

1. In the Shader Tree, click **Add Layer**, and select **Processing** > **Gradient**.
2. In the Gradient **Effect** column, click the down arrow and select **Volumetric Effects** > **Volume Scattering Amount**.
3. In the **Properties** panel, open the **Texture Layers** tab and set the **Surface Gradient** > **Input Parameter** to **Volume Scattering Amount**.
4. In the **Value** graph, select a frame, and then click the arrow on the right, and click **Open in Editor**.


Properties
Groups
Layer
Surface Gradient
Edit Gradient
Texture Layers
Texture Locator
User Channels
Tags

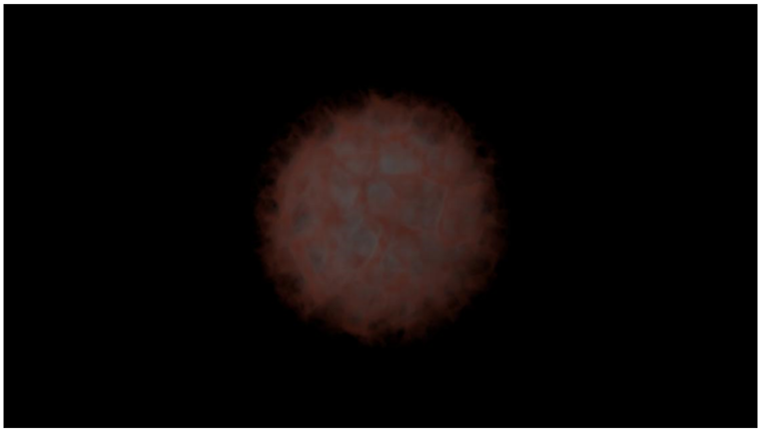
☑ Enable
☐ Invert
Blend Mode Normal
Opacity 100.0 %
Locator Gradient (2) (Texture)
Projection Type Solid
Projection Axis X Y Z
Input Parameter Volumetric Scattering Amount
Value 0.0 %
Color 0.0

5. In the top left side of the **Gradient Editor** dialog, in the **Value** field, type **30%**.


Channels
Gradient (2)
Enable
Opacity
Blend Mode
Invert
Value
Color R
Color G
Color B
Color A
inVal
Texture Value
100.0 %
0.0 %
-100.0 %
-200.0 %
-50.0 %
0.0 %
50.0 %
100.0 %
150.0 %
200.0 %
250.0 %
300.0 %
Key
Interpolation
Cycle
Shadow

Input 0.0 % Value 30.0 % Slope Auto Flat Weight Auto
Behaviors Constant (Smooth) Constant (Smooth)

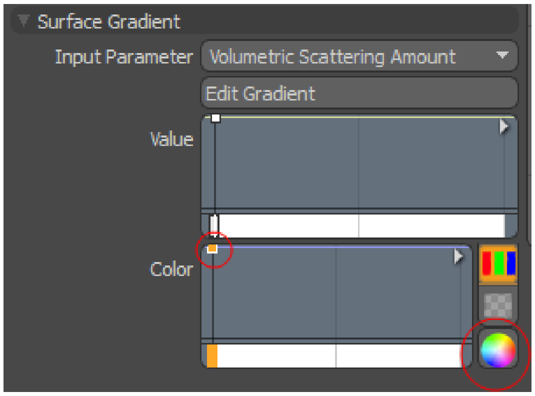
The Render viewport displays something similar to the following.



Creating a Volumetric Luminosity Color Sample

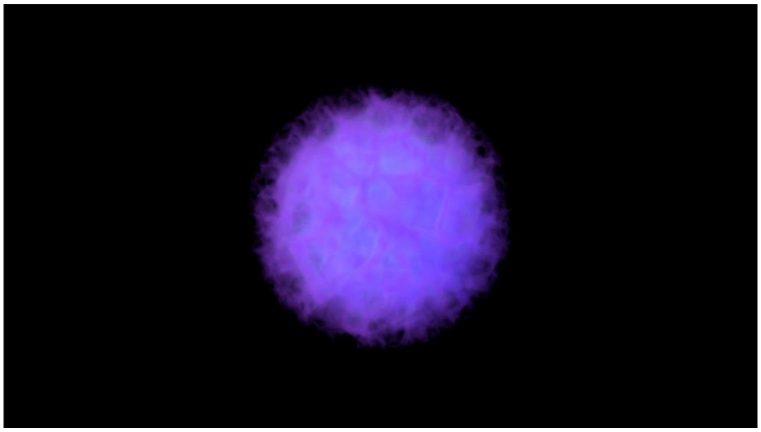
Continue using the same scene.

1. In the Shader Tree, click **Add Layer**, and select **Processing** > **Gradient**.
2. In the Gradient **Effect** column, click the down arrow and select **Volumetric Effects** > **Volumetric Luminosity Color**.
3. In the **Properties** panel, open the **Texture Layers** tab, under **Surface Gradient**, set **Input Parameter** to **Volumetric Scattering Amount**.
4. In the **Color** graph, select a frame and then click the **Color Picker**.


Surface Gradient
Input Parameter Volumetric Scattering Amount

5. Set the color to **Blue**.
6. In the Shader Tree, select the **Base Material**.
7. In the **Properties** panel, open the **Material Rays** tab.
8. Under **Volumetrics**, set **Luminosity Amount** to **100%**.

   The Render viewport displays something similar to the following.



Notice, the volume is displayed in purple because the final energy is combined with **Volumetric Scattering Color** value.

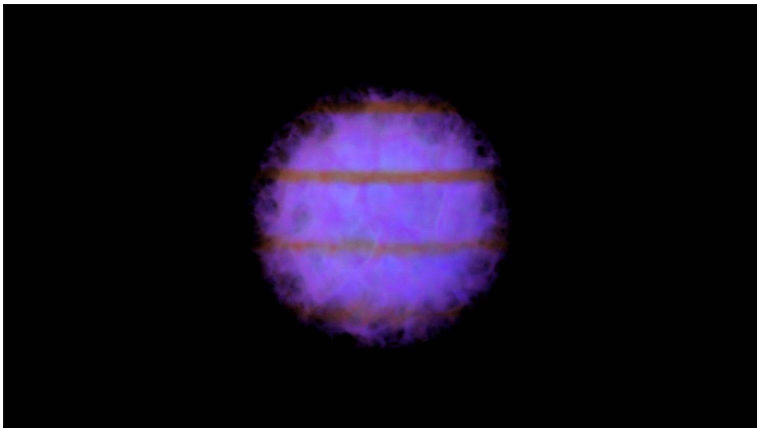
Creating a Volumetric Luminosity Amount Sample

Continue using the same scene.

1. In the Shader Tree, click **Add Layer**, and select **Textures** > **Grid**.

2. With the **Grid** selected, under the **Effect** column, click the down arrow, and select **Volumetric Effects** > **Volumetric Luminosity Amount**.

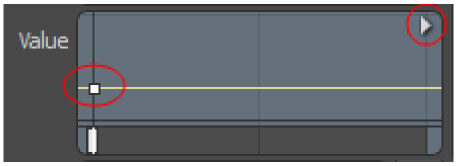
   The Render viewport displays something similar to the following.



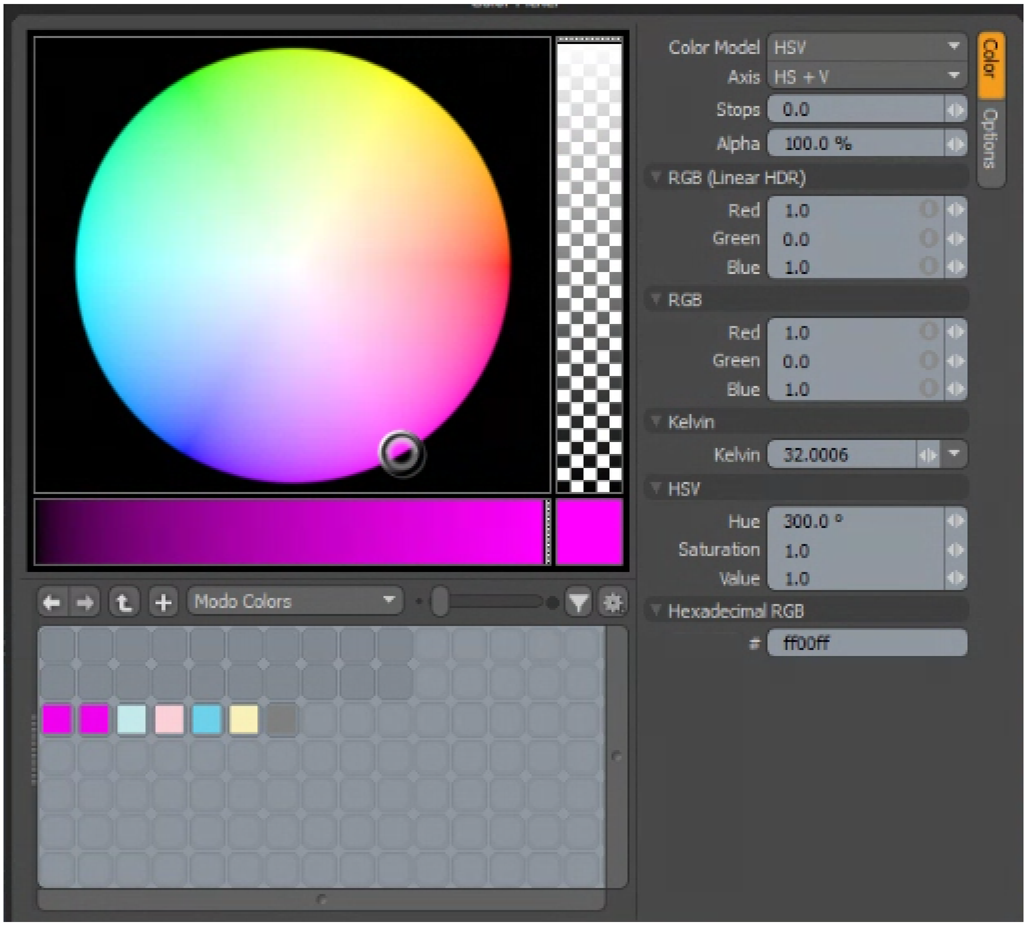
Creating a Volumetric Absorption Color Sample

Continue using the same scene.

1. In the Shader Tree, toggle the eye button (👁) to turn off the **Grid** and the **Gradient** set to **Volumetric Luminosity Color**.
2. In the Shader Tree, click **Add Layer**, and select **Processing** > **Gradient**.
3. In the Gradient **Effect** column, click the down arrow, and select **Volumetric Effects** > **Volumetric Absorption Color**.
4. In the **Properties** panel, open the **Texture Layers** tab, under **Surface Gradient**, set **Input Parameter** to **Volumetric Sample Density**.
5. On the **Value** graph, select a frame, and then click the arrow on the right, and select **Open in Editor**.



6. Set the **Value** field to **30%** and the **Input** to **0%**.
7. On the **Color** graph, select a frame, and then click the **Color Picker**.
8. Set the **RGB** values to **1.0, 0.0, 1.0**.



Color Model HSV
Axis HS + V
Stops 0.0
Alpha 100.0 %

RGB (Linear HDR)
Red 1.0
Green 0.0
Blue 1.0

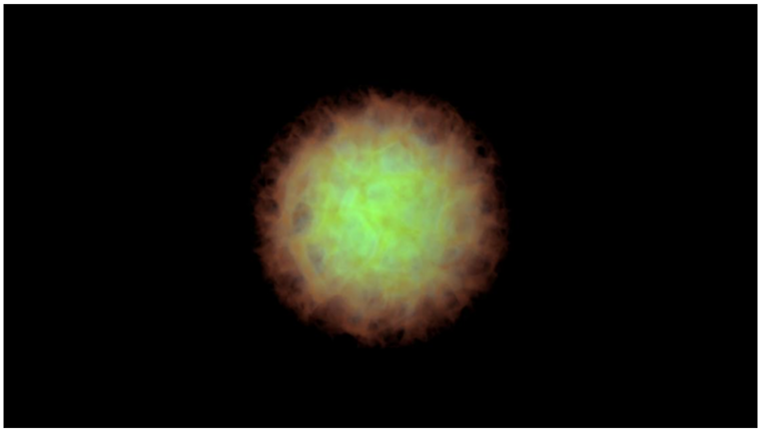
RGB
Red 1.0
Green 0.0
Blue 1.0

Kelvin
Kelvin 32.0006

HSV
Hue 300.0 °
Saturation 1.0
Value 1.0

Hexadecimal RGB
# ff00ff

The Render viewport displays something similar to the following.



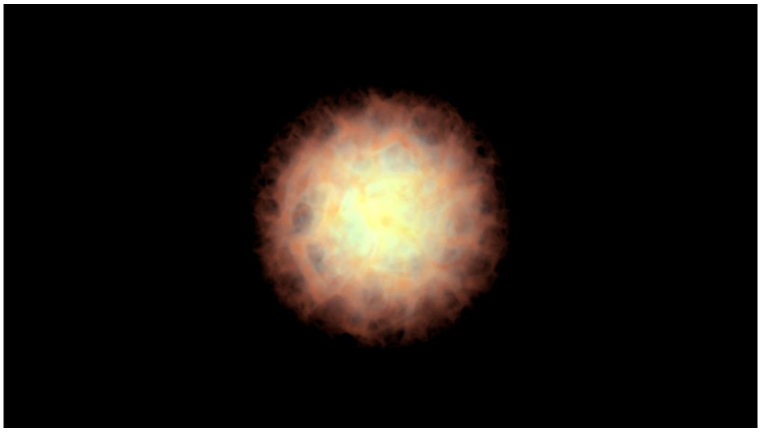
Creating a Volumetric Absorption Amount Sample

Continue using the same scene.

1. In the Shader Tree, click **Add Layer**, and select **Processing** > **Gradient**.

2. In the Gradient **Effect** column, click the down arrow and select **Volumetric Effects** > **Volumetric Absorption Amount**.
3. In the **Properties** panel, open the **Texture Layers** tab, under **Surface Gradient**, set the **Input Parameter** to **Volume Sample Density**.
4. On the **Value** graph, select a frame, and click the arrow on the right, and select **Open in Editor**.
5. Set **Value** to **30%** and **Input** to **0%**.

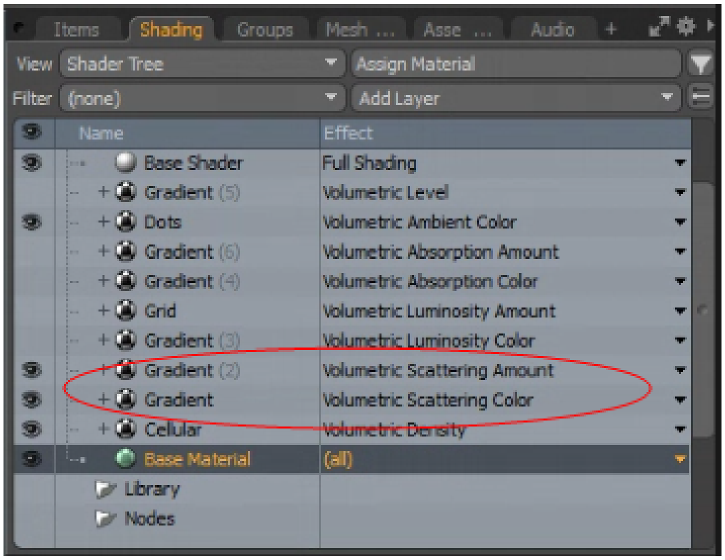
   The Render viewport displays something similar to the following.



Creating a Volumetric Ambient Color Sample

Continue using the same scene.

1. In the Shader Tree, turn the visibility off for all **Gradients** except the first two you created.

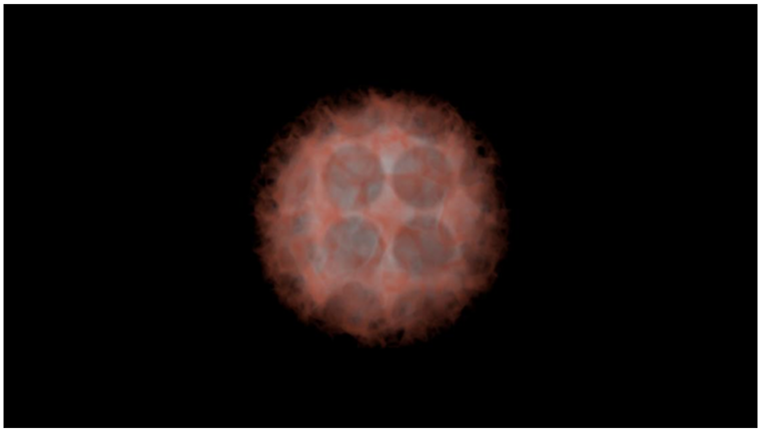


Name	Effect
Base Shader	Full Shading
Gradient (5)	Volumetric Level
Dots	Volumetric Ambient Color
Gradient (6)	Volumetric Absorption Amount
Gradient (4)	Volumetric Absorption Color
Grid	Volumetric Luminosity Amount
Gradient (3)	Volumetric Luminosity Color
Gradient (2)	Volumetric Scattering Amount
Gradient	Volumetric Scattering Color
Cellular	Volumetric Density
Base Material	(all)
Library	
Nodes	

2. In the **Item List**, select **Volume**.

3. In the **Properties** panel, open the **Volume Shading** tab, expand **Shading**, and set the **Ambient** value to **1.0**.
4. In the Shader Tree, click **Add Layer** and select **Textures** > **Dots**.
5. Set the **Effect** to **Volumetric Ambient Color**.
6. With **Dots** selected, in the **Properties** panel, open the **Texture Layers** tab.
7. Under the **Layer** properties, set the **Projection Axis** to **Z**.

   The Render viewport displays something similar to the following.

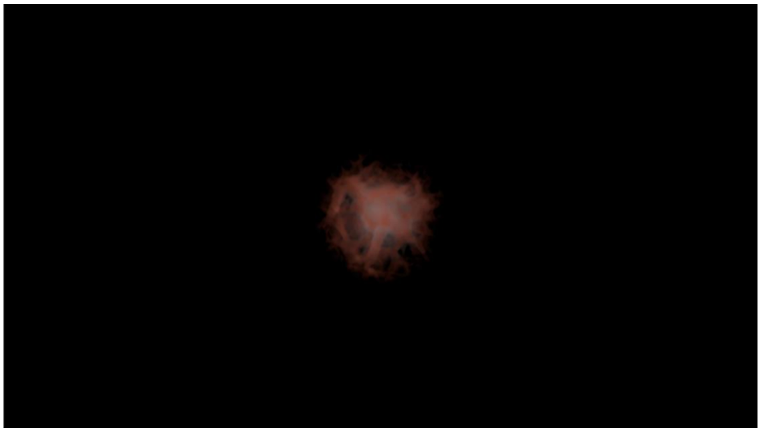


Creating a Volumetric Level Sample

Continue using the same scene.

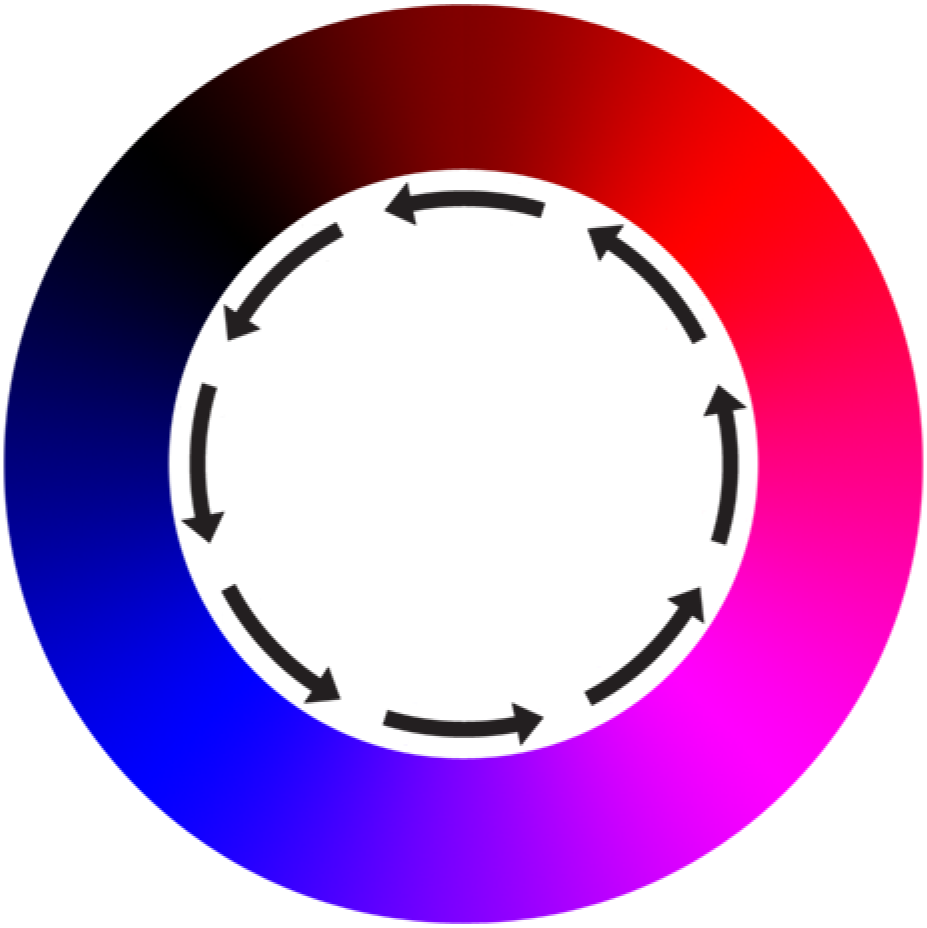
1. In the Shader Tree, hide the visibility of the **Dots**.
2. In the Shader Tree, click **Add Layer**, and select **Processing** > **Gradient**.
3. In the Gradient **Effect** column, click the down arrow, and select **Volumetric Effects** > **Volumetric Level**.
4. In the **Properties** panel, open the **Texture Layers** tab.
5. Under **Surface Gradient**, set the **Input Parameter** to **Volumetric Level**.
6. On the **Value** graph, select the frame and click the arrow on the right, and then select **Open in Editor**.
7. Set the **Value** to **50%**.

   The Render viewport displays something similar to the following.



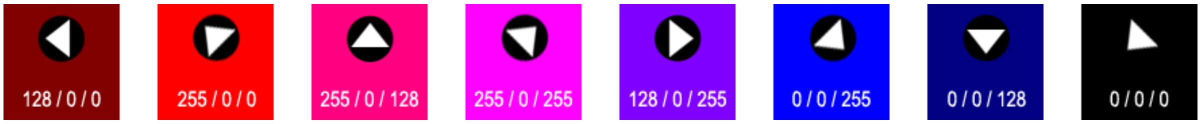
Anisotropic Direction

Anisotropic maps require specific RGB values to control the direction that the specular and reflection shading stretch across. You can see the direction easily in the provided color wheel where the arrows represent the direction for each color.



To recreate the provided anisotropic wheel, you can load a gradient file into the Gradient tool in Photoshop. Then apply it with an angle gradient by dragging up when you create it. If you need assistance, see the Photoshop documentation for information about using the Gradient tool. You can do this also in other image-editing applications by using the provided RGB values in the gradient file.

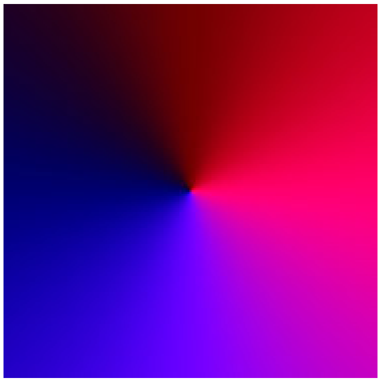
Tip: You can download our example scenes and use **anisotropic_direction.grd**.

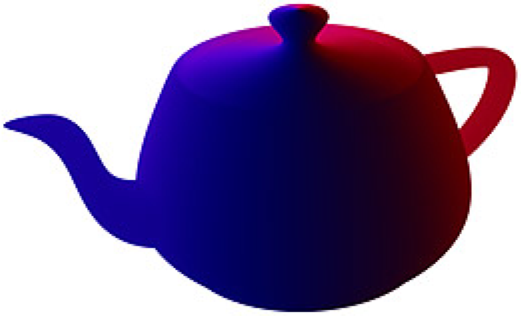


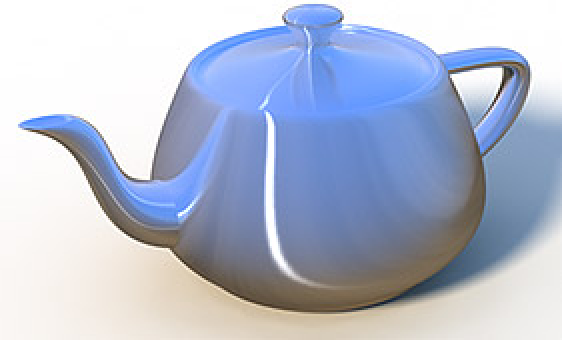
Note: Opposing directions produce identical results (moving left or right to the color makes no difference); therefore, half the wheel is actually unnecessary. It is included here for completeness.

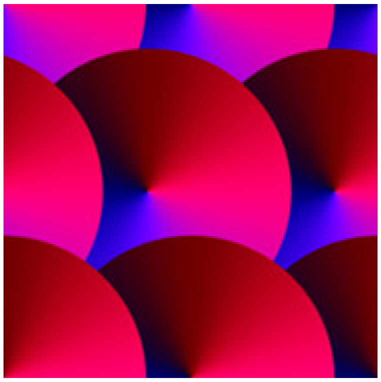
To apply an anisotropic map to a surface, enable **Blurry Reflections** with some value for the **Specular Amount** or **Reflection Amount** in the Material item for the surface. Set the **Anisotropy** value also. (With a UV map defined, the map defines the direction in reference to the UV mapping of the surface.)

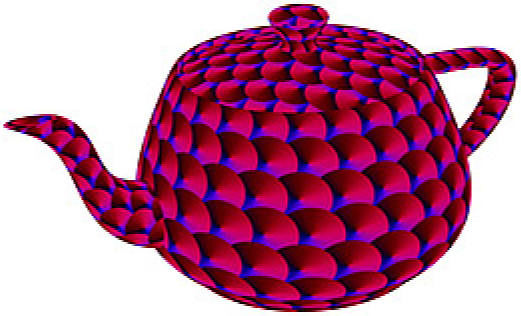
The provided samples use **100%** as the **Reflection Amount** (with **Conserve Energy** enabled) and the **Anisotropy** value set to **100%** as well. Depending on the way you have set up the UV map, Modo can display different effects. The samples show (from left to right): the Anisotropic Map, the Map Colors, and the Resulting Effect.

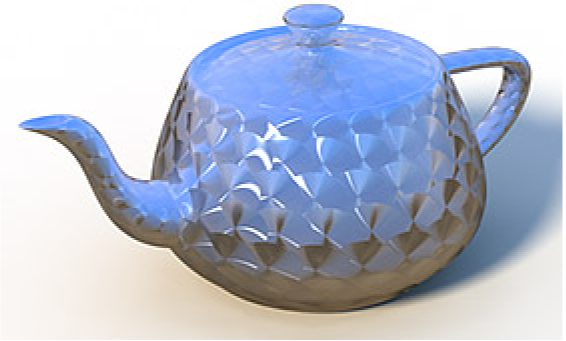


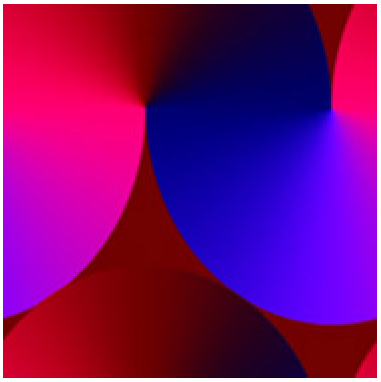


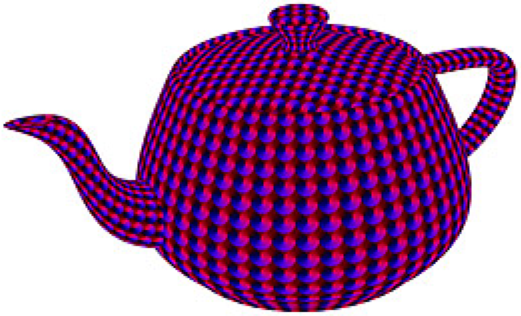


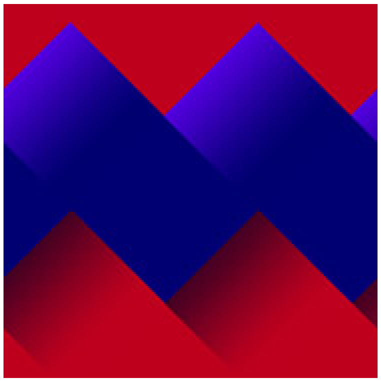


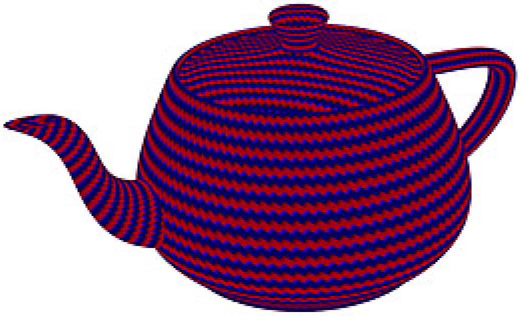


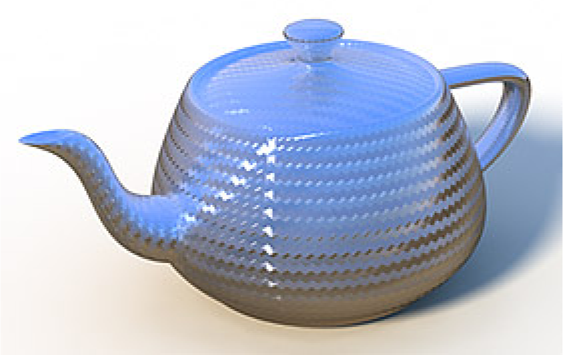












Effect - Shader Item

Shader item layers shade surfaces from direct and indirect light sources, as well as determining the surface visibility. Each new scene, by default, has a single Base Shader item that takes care of these actions, but you may want to add other shader items to the **Shader Tree**. Modifying a shader layer's effect is useful when composing multiple passes out of Modo for a external application (such as Photoshop, After Effects, Nuke, or Fusion). Changing the effect of any shader item limits the output to that type information when passed on to render the output. However, shader items are layers, and you can mask and blend them to create unique possibilities. Unless necessary, use Render Outputs instead of shader items.

The following shader effect types are available:

- **Diffuse Shading**: Limits the output of the shader exclusively to the diffuse shading of the scene.
- **Fog Shading**: Limits the output of the shader exclusively to the fog shading of the scene.
- **Full Shading**: Passes along all the results of the **Shader Tree** to the render outputs. This is the default settings for any shader item.
- **Luminous Shading**: Limits the output of the shader exclusively to the luminous shading of the scene.
- **Reflection Shading**: Limits the output of the shader exclusively to the reflection shading of the scene.
- **Specular Shading**: Limits the output of the shader exclusively to the specular shading of the scene.
- **Subsurface Shading**: Limits the output of the shader exclusively to the subsurface shading of the scene.
- **Transparent Shading**: Limits the output of the shader exclusively to the transparent shading of the scene.

Effect - Environment Item

Every environment in Modo can have a variety of texture layers added to it. These are useful to add depth and complexity to the background of any scene. The layering, opacity, and blend modes all work the same between layers as they do under the render item. You can modify a layer's effect from the default settings by right-clicking the layer in the **Effect** column of the **Shader Tree** and choosing an available command.

The following options are available:

- **Driver A, B, C, and D** - Allows texture layers (either a bitmap, a procedural, or a combination) to drive a gradient layer with a matching input parameter. Modo passes the mapping of the image's brightness (luminosity) onto the gradient. The gradient's keyed values replace those of the driving layers and are output as specified by the gradient layer's **Effect** setting.

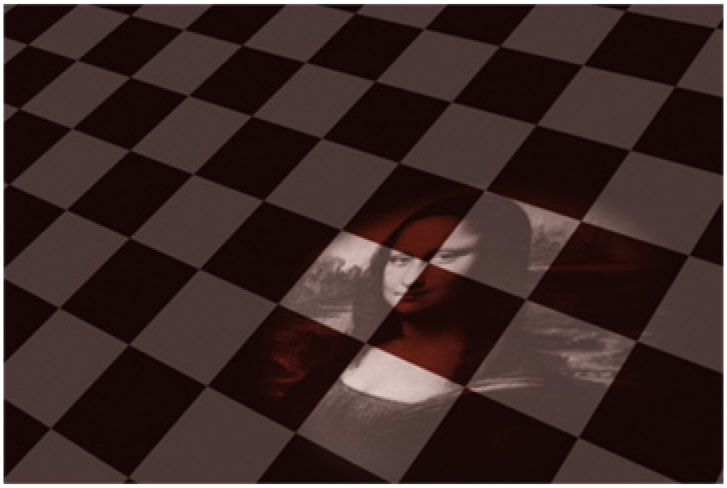
- **Environment Color** - Passes all the color (and illumination information, if applicable, such as in HDR images) to Modo's render engine. This is the default setting for any texture layer added to an environment item.
- **Group Mask** - Masks all layers within the group to reveal any lower texture layers in the Shader Tree. The texture layer must reside within a Material Group item and be specified as a **Group Mask** for the brightness values of the layer to act as a mask. Black areas are completely transparent and attenuate toward white areas that are completely opaque.
- **Layer Mask** - Works identically as the **Group Mask**, but where a **Group Mask** affects an entire group, a layer mask only affects a single layer. When you set any texture layer as a **Layer Mask**, it modulates the texture layer directly below it in the **Shader Tree**. Black areas are completely transparent, and they attenuate toward white areas that are completely opaque.

Effect - Light Item

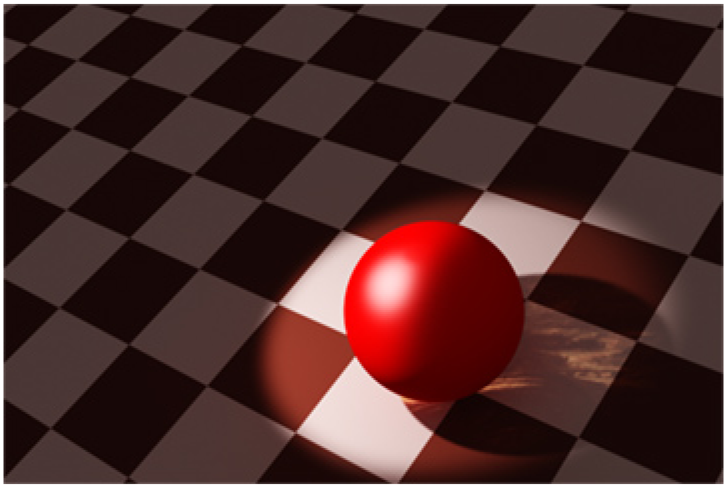
Every light in Modo can have a variety of texture layers added to it. This is useful to add depth and complexity to the lighting of any scene. The layering, opacity, and blend modes all work the same between layers as they do under the render item. You can modify a layer's effect from the default settings by right-clicking the layer in the **Effect** column of the Shader Tree and choosing an available command.

- **Driver A, B, C, and D** - Allows texture layers (either a bitmap, a procedural, or a combination) to drive a gradient layer with a matching input parameter. Modo passes the mapping of the image's brightness (luminosity) onto the gradient. The gradient's keyed values replace those of the driving layers and are output as specified by the gradient layer's **Effect** setting.
- **Group Mask** - Masks all layers within the group to reveal any lower texture layers in the Shader Tree. The texture layer must reside within a Material Group item and be specified as a **Group Mask** for the brightness values of the layer to act as a mask. Black areas are completely transparent and attenuate toward white areas that are completely opaque.
- **Layer Mask** - Works identically as the **Group Mask**, but where a **Group Mask** affects an entire group, a layer mask only affects a single layer. When you set any texture layer as a **Layer Mask**, it modulates the texture layer directly below it in the Shader Tree. Black areas are completely transparent, and they attenuate toward white areas that are completely opaque.
- **Light Color** - Projects an image from the light similar to a slide projector. When enabled as such, this setting also affects the volumetric color. When you set a texture layer to **Light Color**, the color of the image replaces the color specified as the **Light Color** in the light's material panel. For more information on the light material, see Light Material.

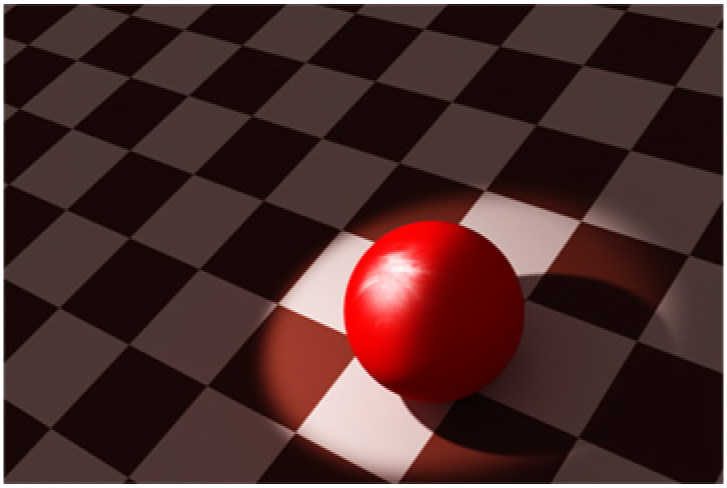
- **Light Diffuse Amount** - Modulates the light's effect on the surface's diffuse amount with the brightness (luminosity) of the texture.



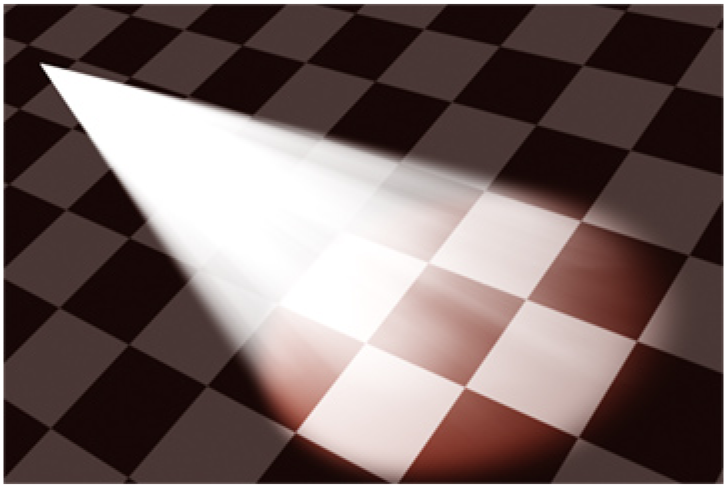
- **Light Shadow Color** - Replaces the color specified as the **Shadow Color** in the light's material panel with the color of the image. For more information on the light material, see Light Material.



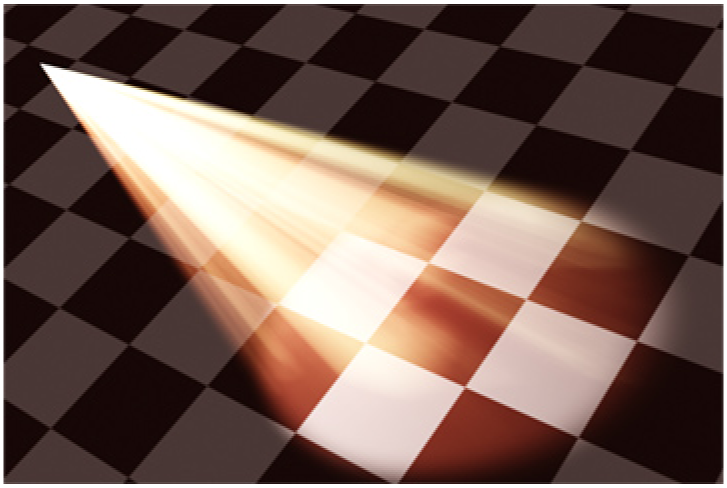
- **Light Specular Amount** - Modulates the light's effect on the surface's specular amount with the brightness (luminosity) of the texture.



- **Volumetric Density** - Modulates the density (thickness) of the volumetric effect. When combined with procedural noise, this is a way to simulate lights shining through smoke.



- **Volumetric Scattering Color** - Modulates the color of the light through the volumetric effect with the colors of the texture layer.



FX Item

The **FX Item** provides a way to control scene related functions that aren't directly surface or shading related, using Shader Tree textures. This could be modulating the strength of a Deformer Falloff using a **Noise** procedural, or controlling the density of particles in a Particle Generator item with a **Gradient**. The same procedures for the rest of the Shader Tree apply to FX Items, but since the target items you would apply these textures to are procedural in nature (generated numerically), they can't be tagged in the traditional sense. For FX Items, textures are applied by way of **Item Masks**.

Creating Item Masks

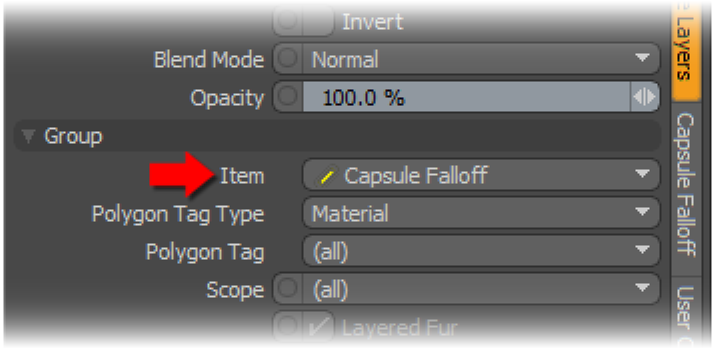
To apply a texture to an item, you first need to create an **Item Mask**:

1. Ensure you're in Item Selection mode.
2. Right-click directly over the target item in the **Items** list or 3D viewport to open the context menu.
3. Select **Create Item Mask** toward the bottom of the menu to create a **Material Group** item.

   Now any textures added to the group only apply to the target item.

Note: In some cases, the items are created properly in the **FX Item** (such as **Falloff**), but in other cases Modo may place the **Item Mask** under the **Render** item, in which case, drag and drop the layer to the proper location.

See Shader Tree for more information on layers and layer hierarchy.


Invert
Blend Mode Normal
Opacity 100.0 %
Group
Item Capsule Falloff
Polygon Tag Type Material
Polygon Tag (all)
Scope (all)
Layered Fur

Layers
Capsule Falloff
User

An Item Mask is nothing more than a Material
Group with the target layer specified under the **Item** option.

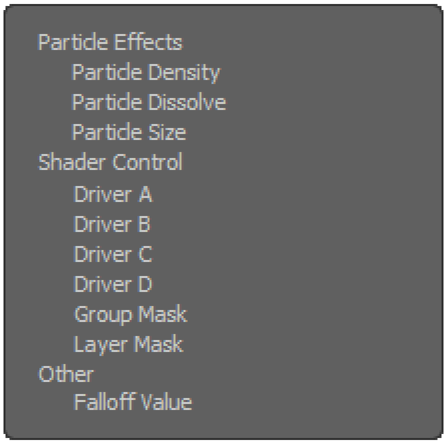
Adding Textures

Once the mask is generated, you can use the **Add Layers** function of the **Shader Tree** to insert the actual texture(s) for use. If the **Group Item** mask is selected when the menu is opened, note that the list of available layers is reduced, as only applicable layers are present. Once the specified layer is added, you can select the layer by left-clicking on the layer name and its associated attributes can be edited in the **Properties** panel. See Shader Tree Item Layers for information on specific texture layers.

In order to direct how the texture layer modulates the target item, specify a layer **Effect** type by right-clicking on the **Effect Title** of the target layer in the **Effect** column and choosing an alternate **Effect** type.

Effect Types

The effect types available to **FX Item** layers are:


Particle Effects
Particle Density
Particle Dissolve
Particle Size
Shader Control
Driver A
Driver B
Driver C
Driver D
Group Mask
Layer Mask
Other
Falloff Value

Particle Density

The **Particle Density** option allows you to modulate the frequency of particles generated by a Surface Particle Generator or a Particle Cloud item using procedural textures with a **Solid** projection.

Particle Dissolve

The **Particle Dissolve** option allows you to modulate the opacity of particles in a **Volume** or a **Sprite** item using procedural textures with a **Solid** projection.

Particle Size

The **Particle Size** option allows you to modulate the size of particles in a **Blob**, **Volume**, or **Sprite** item using procedural textures with a **Solid** projection.

Note: For more information on projections, see Projection Type Samples.

Driver A,B,C, and D

The **Driver** options allow texture layers to modulate a Gradient layer with a matching Input parameter. The mapping of the images brightness (luminosity) is passed onto the **Gradient**. The keyed values in the **Gradient** replace those of the driving layers and are output as specified by the **Gradient** layers **Effect** setting.

Group Mask

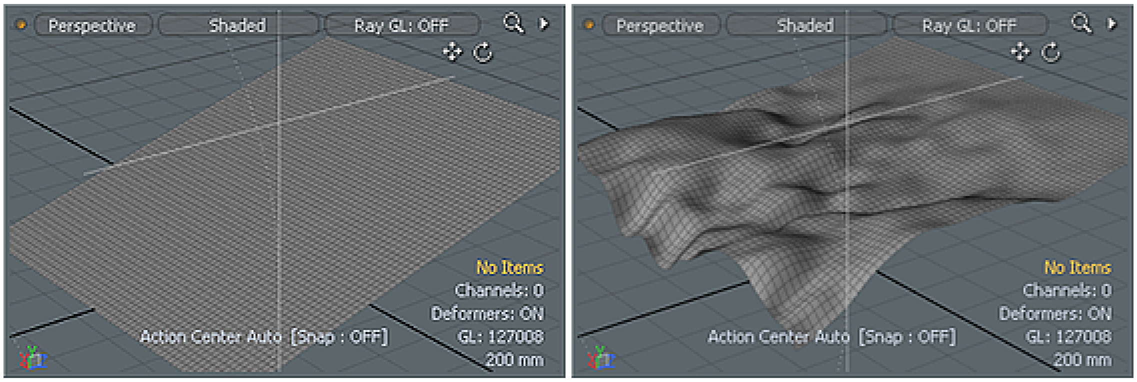
When any texture layer resides within a Material Group item and is specified as a **Group Mask**, the brightness values of the layer masks all layers contained within the group revealing any lower texture layers in the Shader Tree. Black areas are completely transparent, attenuating toward completely opaque (represented by white).

Layer Mask

The **Layer Mask** works identically to the **Group Mask**, but where a **Group Mask** affects an entire group, a layer mask only affects a single layer. When any texture layer is set as such, it modulates the texture layer directly above it in the Shader Tree. Black areas are completely transparent attenuating toward completely opaque represented by white.

Falloff Value

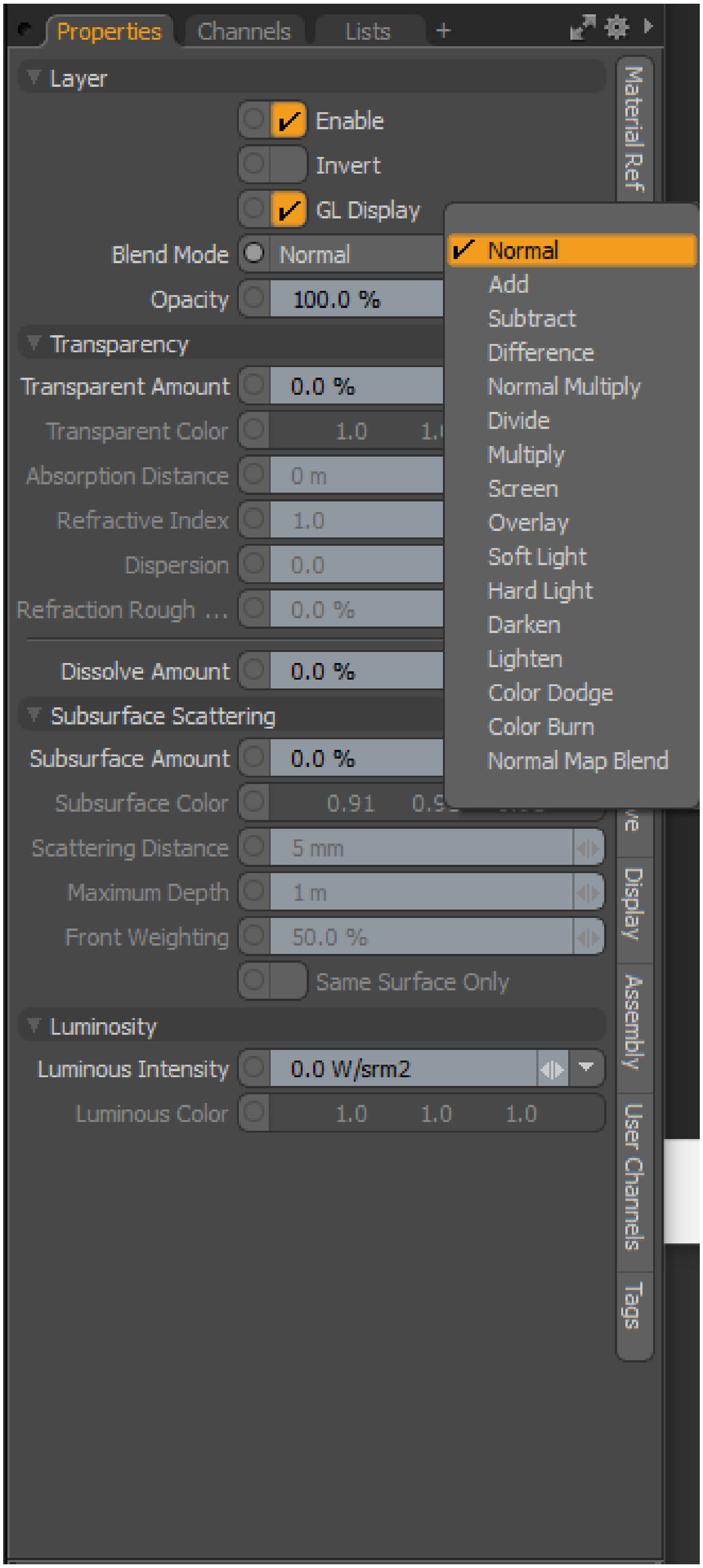
The **Falloff Value** option allows you to modulate the strength of a Falloff deformer using a Modo texture layer. In the example, a **Linear Falloff Deformer** is applied to a plane on the left, shearing it. On the right, a **Noise** procedural modulates the **Falloff** strength.


Perspective
Shaded
Ray GL: OFF
No Items
Channels: 0
Deformers: ON
GL: 127008
200 mm
Action Center Auto [Snap : OFF]
Perspective
Shaded
Ray GL: OFF
No Items
Channels: 0
Deformers: ON
GL: 127008
200 mm
Action Center Auto [Snap : OFF]

Layer Blend Modes

The flexibility and power of the Shader Tree is extended with the use of **Blend Modes**, a means of determining how multiple layers interact with each other. The key point here is that there must be more than one layer in order for blending to work. The Shader Tree's layered approach to shading surfaces works in a fashion similar to layers in an image editor such as Photoshop; any layer above another obscures the layers below, unless one (or more) of three conditions is true:

- A mask is applied, either as a layer or a surface tag, limiting the effect of the layer exclusively to areas inside the mask. This reveals lower layers outside the masked region.
- The transparency of the layer is changed to reveal the lower layers.
- A **Blend Mode** other than **Normal** is specified, changing the way the layer interacts with the lower layers.


Properties
Channels
Lists
+
Material Ref
Display
Assembly
User Channels
Tags
✓ Normal
Add
Subtract
Difference
Normal Multiply
Divide
Multiply
Screen
Overlay
Soft Light
Hard Light
Darken
Lighten
Color Dodge
Color Burn
Normal Map Blend

Layer

☑ Enable

☐ Invert

☑ GL Display

Blend Mode: Normal

Opacity: 100.0 %

Transparency

Transparent Amount: 0.0 %

Transparent Color: 1.0 1.

Absorption Distance: 0 m

Refractive Index: 1.0

Dispersion: 0.0

Refraction Rough …: 0.0 %

Dissolve Amount: 0.0 %

Subsurface Scattering

Subsurface Amount: 0.0 %

Subsurface Color: 0.91 0.9

Scattering Distance: 5 mm

Maximum Depth: 1 m

Front Weighting: 50.0 %

☐ Same Surface Only

Luminosity

Luminous Intensity: 0.0 W/srm2

Luminous Color: 1.0 1.0 1.0

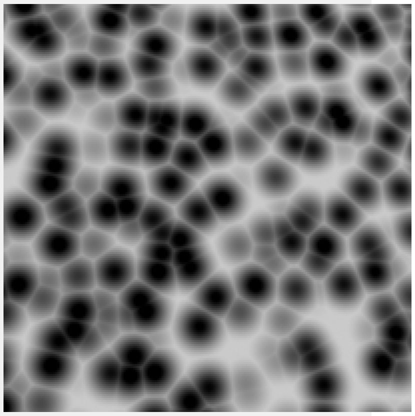
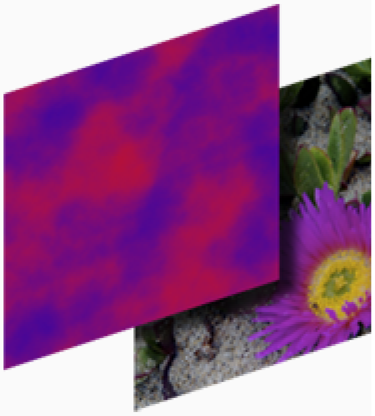
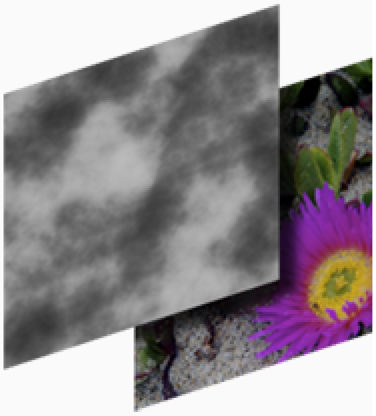
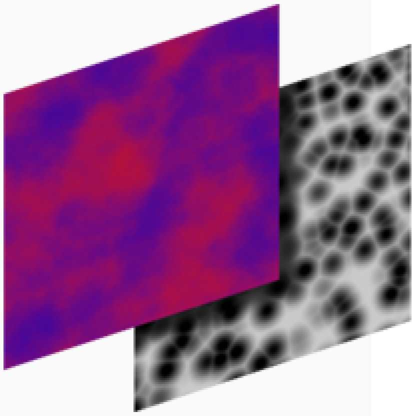
Most item layers in the Shader Tree offer a **Blend Mode** option to combine two or more layers in various ways using mathematical operations. The mode is specified by left-clicking on the **Blend Mode** button to open a contextual menu, and then selecting the mode from the available options.

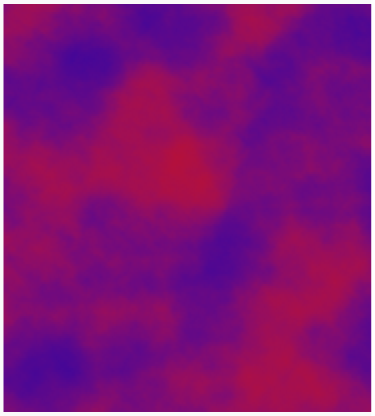
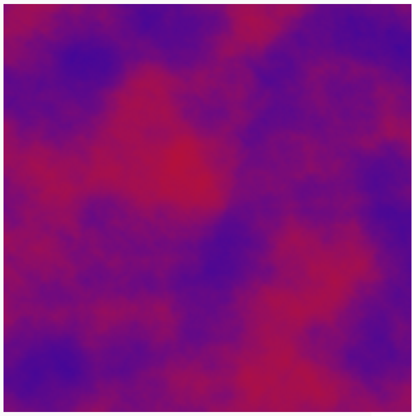
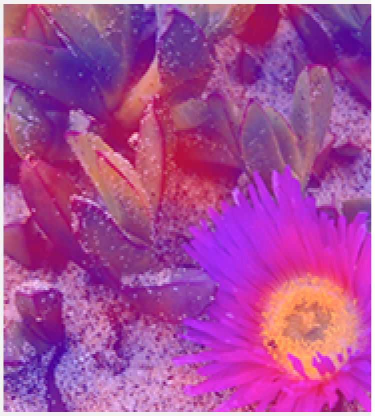
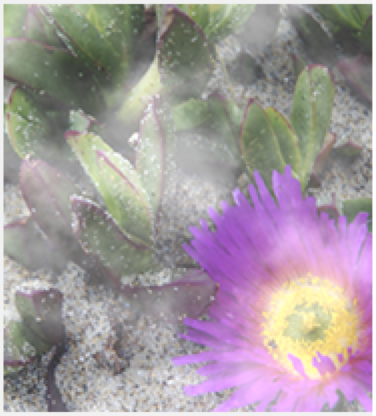
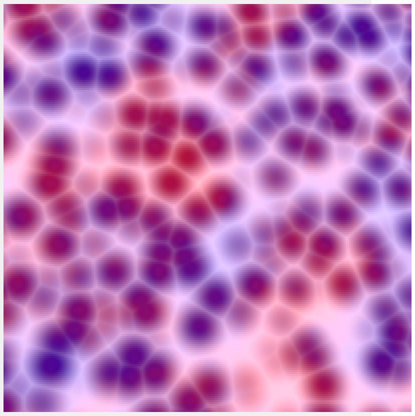
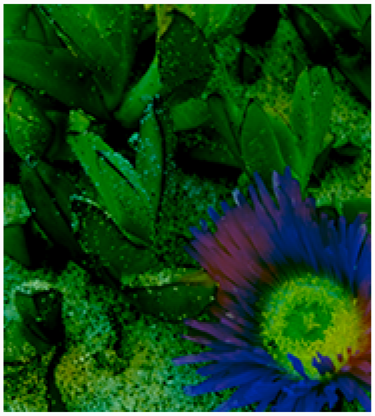
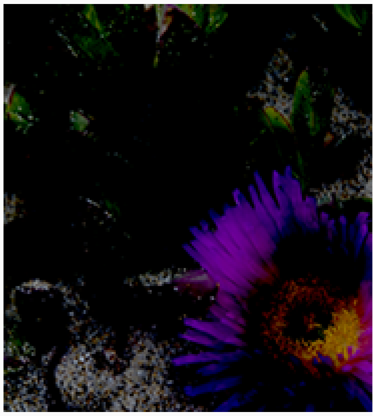
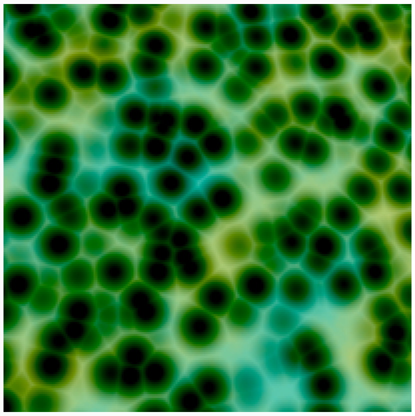
Note: Blend Mode results are only visible in the final render or in the **Preview** viewport.

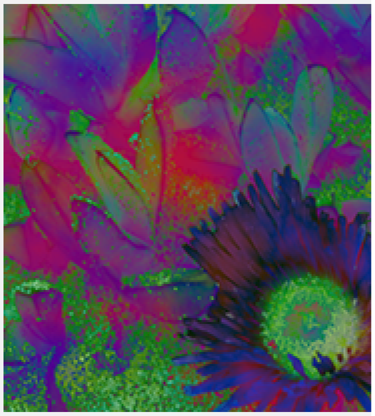
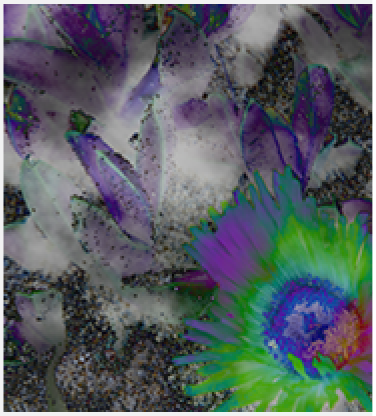
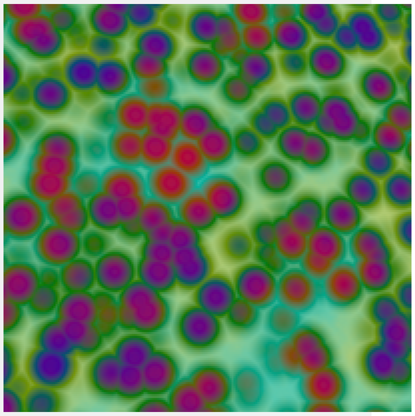
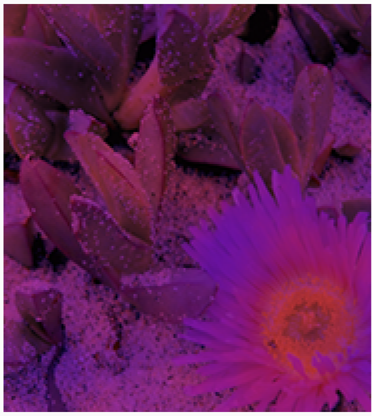
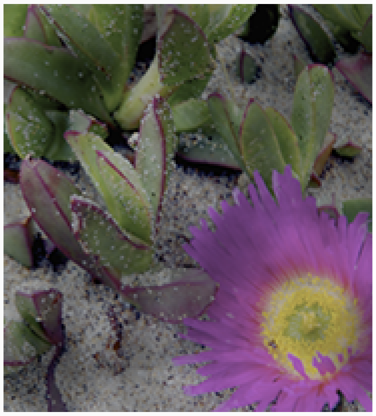
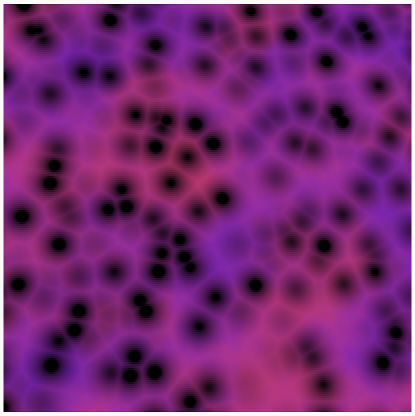
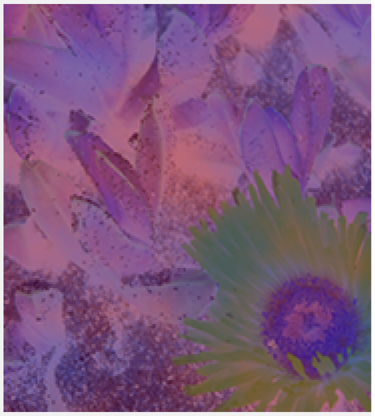
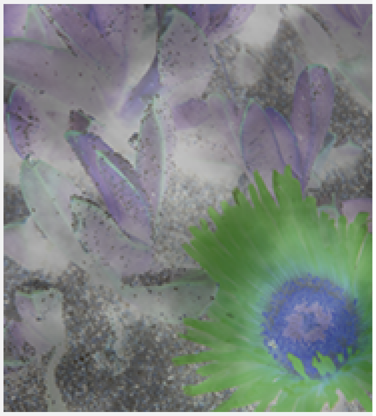
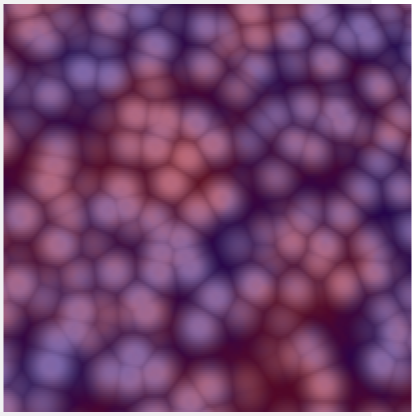
Blend Modes affect the RGB color values (even if you just have black and white specified) and have a couple of rules that need to be adhered to:

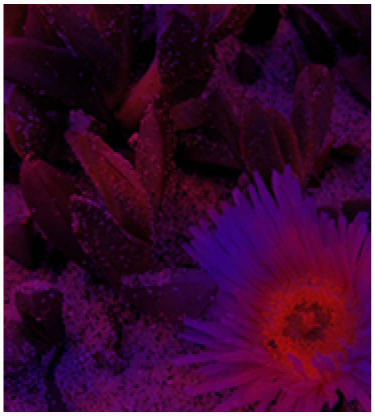
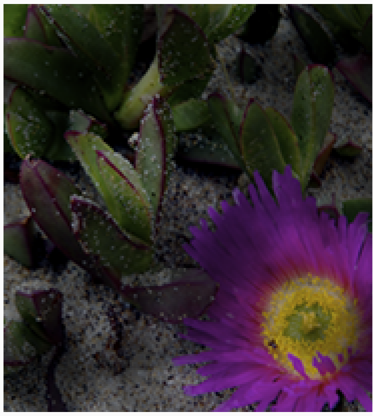
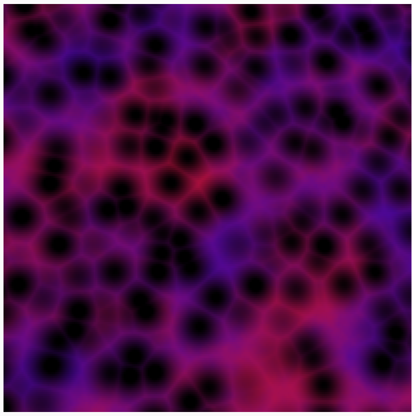
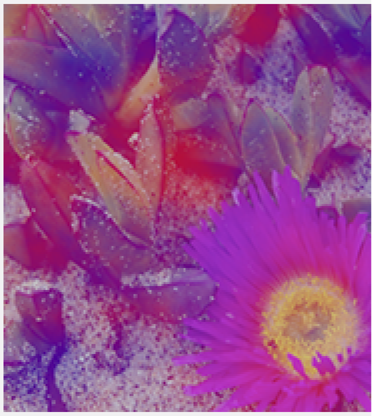
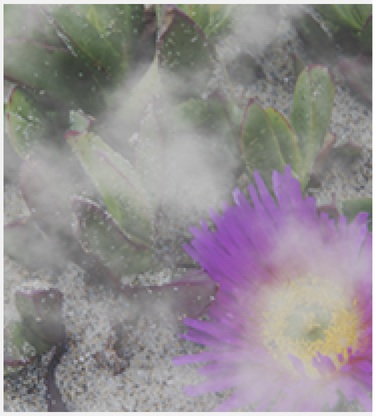
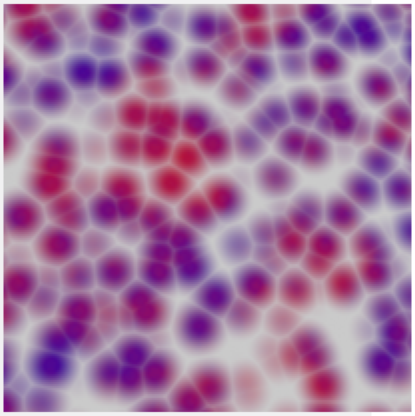
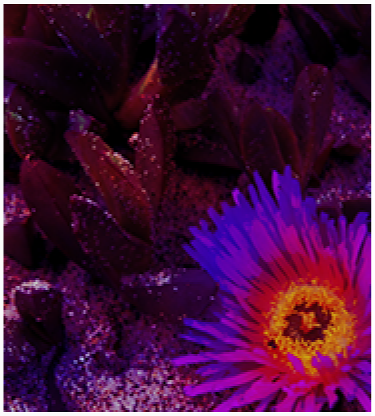
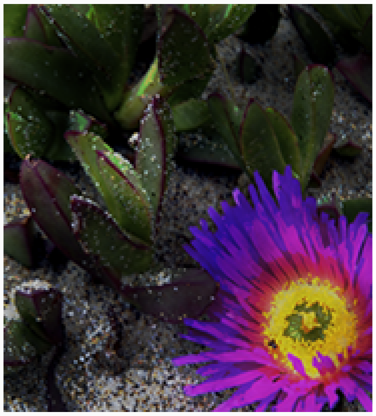
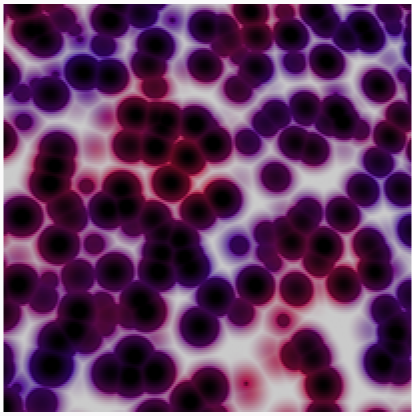
- There must be more than one layer of the same **Effect** type. For example, a **Diffuse Color** layer can only use the **Multiply** mode with another layer specified as **Diffuse Color**.
- A layer with a **Blend Mode** applied only affects other layers below it in the Shader Tree.

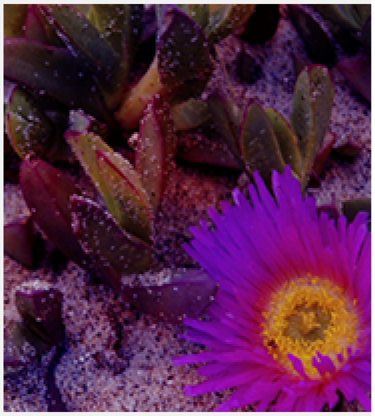
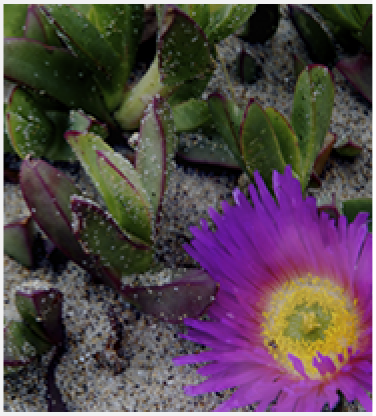
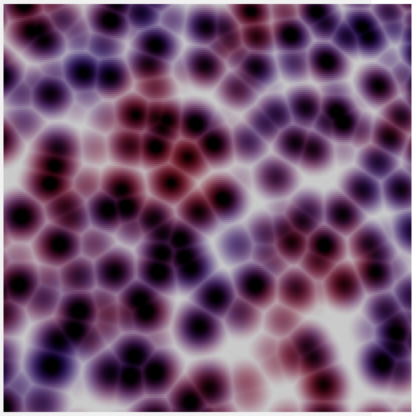
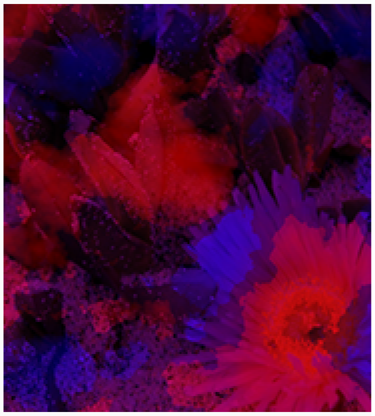
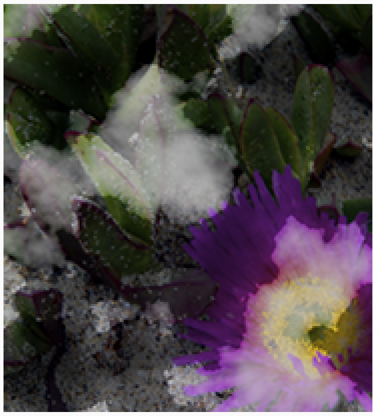
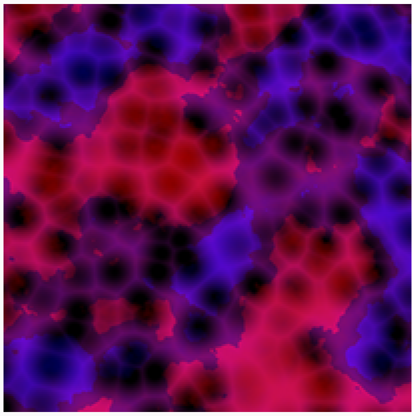
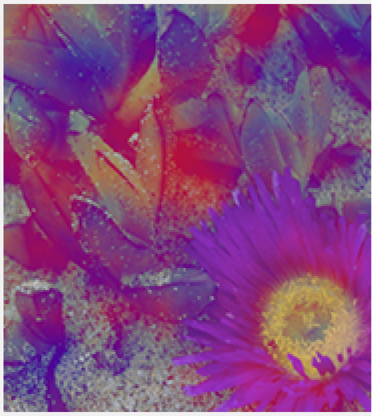
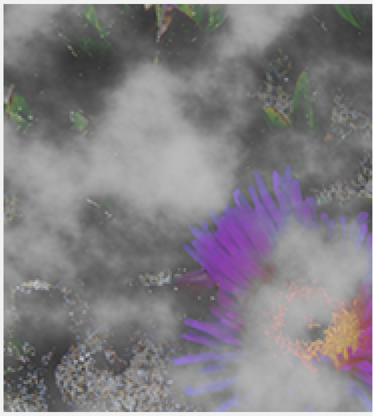
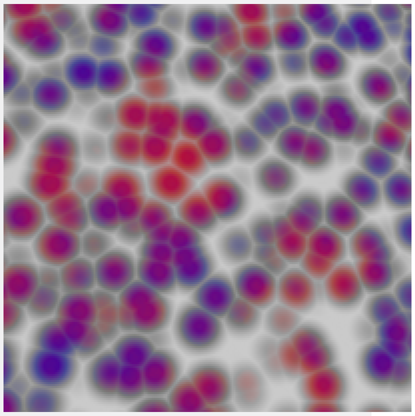
The following table show the various **Blend Modes**, with samples to illustrate the effect.

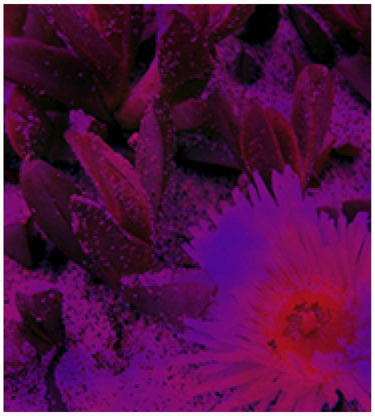
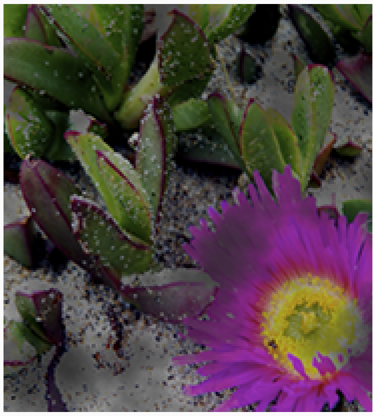
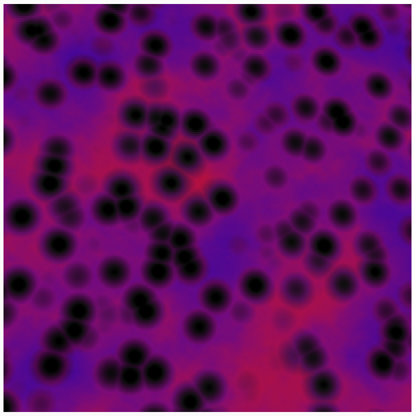
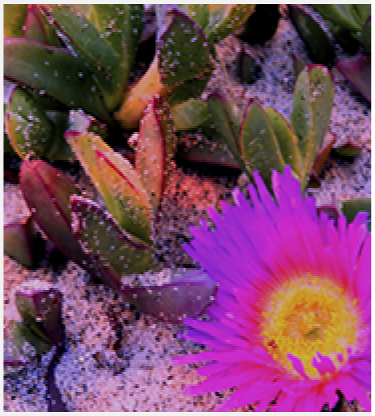
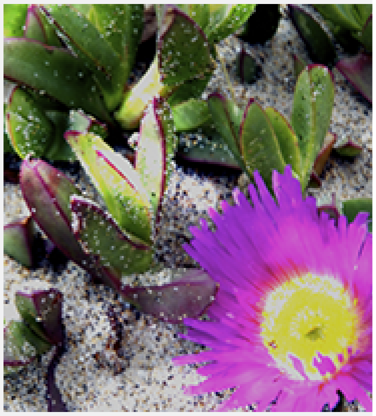
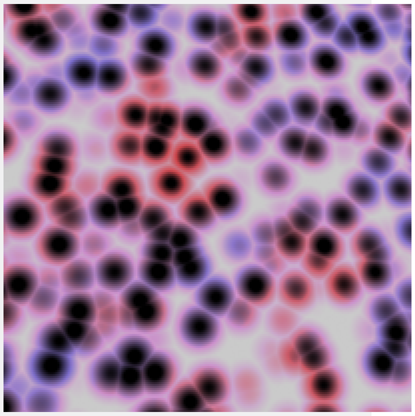
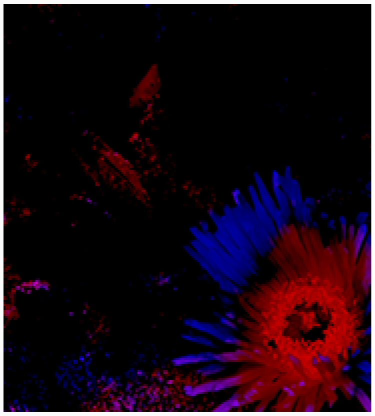
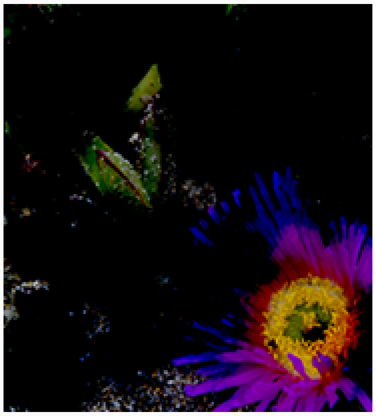
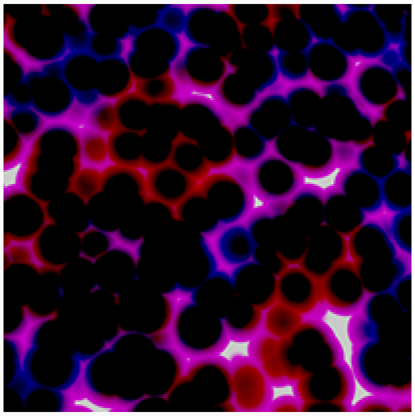
Original Images			
2 Layers - Fractal Noise and Original Images			
Blend	**Results**		

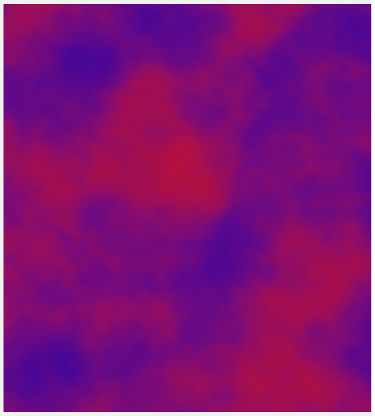
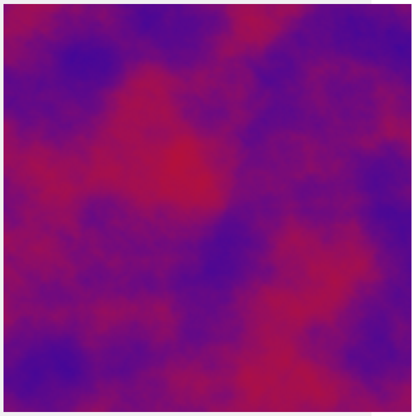
Mode			
Normal			
Add			
Subtract			

Difference			
Normal Multiply			
Divide			

Multiply			
Screen			
Overlay			

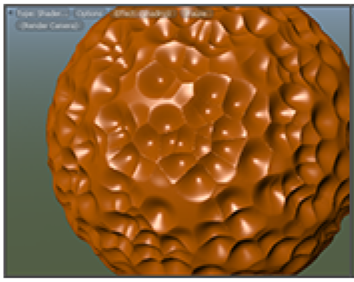
Soft Light			
Hard Light			
Lighten			

Darken			
Color Dodge			
Color Burn			

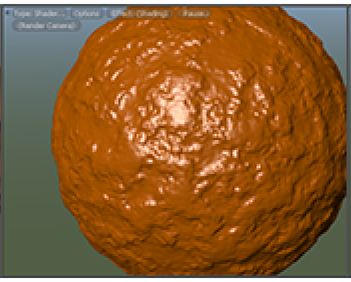
Normal Map Blend			

Doing the Math

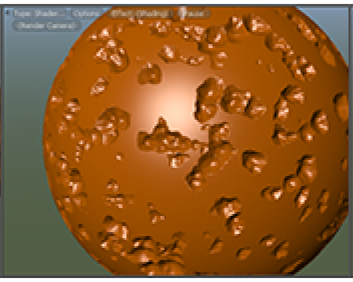
Blend Modes work by calculating the math (as specified) on the individual RGB channels. **Add**, for instance, takes the values from the two layers and adds them together for the result. Depending on the texture and the blend mode, it may produce values beyond 100%. To reduce or eliminate clipping for certain effects that can't be driven higher than 100% (such as **Displacement**), you may also need to adjust a layer's transparency. The following example shows how this works.



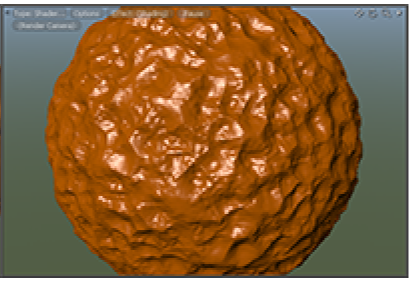
1.) Cellular displacement layer



2.) Noise layer overwrites the Cellular layer



3.) Noise layer set to Add produces clipped values



4.) Transparency of both layers set to 50%

1. A Cellular procedural layer applied as a displacement map.
2. In the Shader Tree, a **Noise** layer is added above it the Cellular layer and set as displacement, superseding the lower layer in the tree (image 1).
3. To produce a combination of both, the Noise layer's Blend Mode is set to **Add**, but this produces the clipped values shown in image 3. This is because the math produces values beyond 100% (75% + 75% = 150%), but there is no value higher than 100% (meaning all the way displaced), so that value is truncated to 100.
4. By changing the **Opacity** value of both layers to 50%, each layer now contributes equally to the displacement.

This can be biased in either direction by adjusting the **Opacity** values. As long as the sum is equal to 100%, you won't produce any clipped values (although, that may be the look you are going for). In some instances, certain effects can be driven higher than 100%, but in those cases it won't be necessary to adjust the transparency.

Tip: A single **Material** item produces values for multiple effects, such as diffuse color and amount, reflection, transparency, and so on. Single texture layers in the Shader Tree can effectively blend with the values produced by the material item without the need for an additional matching layer.

Advancing the Shader Tree

The Shader Tree is both immensely powerful and tremendously flexible. It is this flexibility that can lead to it feeling unwieldy at times. Based on user feedback, it was a design requirement to allow materials and textures to be applied at any level in the scene, even globally if necessary, but as the saying goes, "With great power comes great responsibility".

As a result of this flexibility, diligence is required to keep the Shader Tree organized, and the effort necessary increases as the complexity of a project increases. Understanding how to work with layers in the Shader Tree and what each contributes is an important start in wielding the power that the Shader Tree provides. The basics are covered in Shader Tree, Shader Tree Item Layers, and Layer Effects. Beyond the basics covered on these pages, it was important to add some additional information going over some more advanced aspects of shading surfaces in Modo.

Groups and Groups

The number one most important item in the Shader Tree is the Material Group item. In the **Add Layer** function of the Shader Tree, it is simply referred to as a Group, but here we're calling it a Material Group to avoid any confusion with actual groups. The Material Group serves multiple purposes and understanding its few settings are extremely important in getting the most out of them.

When you add a Material Group item to the tree, they are simply empty containers. These containers can hold any number of layers, including collections of other groups. In this case, they are strictly organizational devices, providing hierarchy to a scene. For example, you might associate similar materials to an item in the scene, making it easier to navigate a complex tree. This can be illustrated by selecting a few layers in the tree and pressing the **Ctrl**+**G** keyboard shortcut, grouping together similar materials under an undefined Material Group item.

The Material Group item also provides the principal means for applying materials to surfaces using masks. By selecting from named items and/or polygon tags in the Material Group item's **Properties** panel, any contents (or layers) within the Material Group can be limited to only affect geometry surfaces that carry that specific tag in the scene. This is done automatically every time you apply a Material Tag using the **M** keyboard command. Pressing **M** opens the **Polygon Set Material** dialog, allowing you to define a name and pressing **OK** generates a new Material Group item in the Shader Tree, automatically setting the **Tag Type** and **Tag Name** options to the particular name that was defined. It's a very simple procedure, but the implications of the power that the Material Group provide are immense.

Masking Surfaces

While the Material Group is an easy way to assign surfacing for geometry when rendering, its reliance on polygon tags limits what is possible with just Material Groups. Not all surface borders are hard-edged polygonal boundaries. Modo provides several ways to mask textures that provide much finer control. The masking functions combined with the layering possibilities of the Shader Tree makes it simple to produce very complex surfaces with ease. Whenever an alpha transparency mask is applied to a texture layer, the underlying layers are revealed.

Group Masks

The Group Mask is a Layer Effect option, defined for a layer in the **Effects** column of the Shader Tree. It allows the contents of an entire Material Group, regardless of complexity or number of layers, to be masked by using any single texture item. The texture can be a procedural texture such as a Noise Procedural layer, an image map, or even a weight map using the Vertex Map Texture layer. The use of image maps provides very fine pixel-level control. You can create these images externally using an image editor, or paint them directly in Modo using the built-in paint tools. The use of gradients opens up a number of interesting possibilities as well, where Material Groups can be masked by the numerous gradient input parameters, such as slope or incidence angle.

Note: You can render complicated material setups directly in the Advanced 3D viewport and make changes in real time. The Advanced 3D Viewport supports **Group Masks** allowing you to mask multiple material layers.
The Advanced 3D viewport supports group mask Blend Mode and Opacity changes, resulting, in most cases, in little difference between the representation of a material in Advanced 3D viewport and the representation of a material in Modo’s Preview Viewport.
For more information, see 3D (OpenGL) Viewport.

Layer Masks

The Layer Mask result is similar to the masking functionality of the Group Mask, but its application is limited to a single layer within the Shader Tree, providing even more granular control. There are two ways to apply a layer mask:

- Choose the **Layer Mask** effect and place the layer directly below the layer you want to mask, or
- Drag and drop the layer onto the target layer, automatically defining it as a layer mask effect.

You can modify the settings of the texture layer, toggling the visibility of this kind of mask using the **+** icon preceding the layer's name.

Alpha Channels

Modo opens and respects embedded alpha channels or layer transparency on imported image map layers. Their actual application can be a point of confusion, because some artists are surprised when they can't see through their surface in the transparent areas of the image map. This is because Modo respects the alpha channel for the image map as a texture layer (revealing any lower surface shading), but it doesn't serve double duty as **Transparency** (or **Stencil**/**Dissolve**) for the surface when set as the default **Diffuse Color** effect. To achieve these effects, you would also need to apply the image as a **Layer** or **Group Mask**, or additionally assign the Alpha channel on the image as **Transparency** or **Dissolve** using the **Alpha Only** option in the image's Properties. Alternatively, you can simply use the **RGBA** layer effect to designate both diffuse color and transparency (applied as **Stencil**) using a single texture layer.

By virtue of how Modo handles alpha transparency for a image maps, this makes it possible to layer any number of transparent images and combine the result into a single complex surface. Even when painting, using transparent image layers, Photoshop-like multi-layered images are possible by stacking several image map layers with Alpha channels. You can select the layer in the Shader Tree to designate which image receives the brush strokes while painting, leaving other layers unaffected.

Item Masks

Within the Material Group item's **Properties** panel, you can specify item masks in addition to the other settings. Item Masks can be extremely useful limiting the application of images or shading to a single Mesh Item layer, without resorting to a bunch of different polygon tags. This makes it possible to apply different shading or image maps to an instance simply by specifying the instanced item's name in the **Item** property of the Material Group item. Instances usually have the inherited parent items name followed by a number in parenthesis. A simple Item selection shows which Instance is the target for the mask. Item Masks also allow shading to be applied to individual Items that would otherwise overwrite other same named (tagged) surfaces in the scene.

Layering Surfaces

Not only can you layer images and textures in a single material, you can also layer different tags for interesting results. Polygons can only have a single part or material tag, but they can have both tags simultaneously as well as belonging to an unlimited number of selections sets. The tagged polygons do not have to be identical in their grouping, or even similar. It is therefore possible to create a selection set that spans polygons from two separate material surfaces, define it as a mask and apply an image that straddles the two surfaces. Use a straight material tag for the two regular surfaces and then assign the smaller areas to the selection set with custom UVs. It's also very useful to apply labels and such.

Cleaning Up the Shader Tree

On occasion, you may find the Shader Tree encumbered with unnecessary layers that need to be removed, but choosing which layers to delete can be difficult. These extra layers can be the result of experimentation or excessive copying of material presets.

The **Purge Unused Materials** command removes all Shader Tree material and/or texture layers that do not contribute to the final rendered image, including those that reference unassigned polygon tags. These could occur when geometry is deleted, or new tags are assigned, as polygons can only hold a single material or part tag. Moving item layers into a scene from another can also cause extra layers to be added.

To activate the purge, select the command from the menu bar, under **Texture** > **Purge Unused Materials**.

Targeted Control

One often confusing aspect of the Shader Tree is its Base Shader item that defaults to the top of every Shader Tree. The Base Shader controls visibility, shading rate, and many other aspects of the rendered image. In a way it is like a soldier at the gate controlling what information gets by the gate to the renderer and how much of that information is passed along for evaluation. Due to the global nature of the Shader Tree, with the Base Shader at the top and unmasked, it has meant that any shader items added below it have their settings overwritten by Base Shader settings unless the lower level shader was masked appropriately and placed higher up in the Shader Tree.

The juggling of Shader Tree layers and Group Material masks evaluating based on their position can get confusing, especially in complex scenes. Sometimes you are just looking for a simple way to make a single object invisible to the camera. To do this, Item Shaders have been introduced. They are in every way identical to the regular shader items except that they are the only Shader Tree layer that isn't represented in the Shader Tree. These are assigned and modified directly in the **Item List**. When any of their **Control** options are enabled, those settings always override those of the Shader Tree counterpart upon evaluation.

To assign an Item Shader, in the **Item List**, right-click on the target Mesh Item layer and select the **Create Item Shader** option from the menu. With the Item Shader selected, you can modify its attributes in the **Properties** panel. To target a specific control, such as **Visibility**, disable all the **Control** options except for **Control Visibility**, and then disable the single **Visible to Camera** option. This makes the targeted item invisible to the camera, yet it still casts shadows and shows up in reflections. **Item Shaders** simplify many aspect that used to be difficult to apply.

Some Additional Notes

It's easy to forget that texture items such as image maps and procedurals can be added to lights and environments as well. By adjusting their **Projection** types and **Effect** settings, a whole range of interesting effects are achievable. Texture layers can be used to mask any other types of layer, including Surface Generators and Fur. The only layer that cannot be used as a mask for Surface Generators or Fur is the Occlusion layer. This is because it requires rays to be fired to calculate and this only happens at render time, which is too late in the evaluation process to produce the intended result. In these cases, bake the occlusion layer out to a image map texture and then reapply the image map in place of the occlusion.

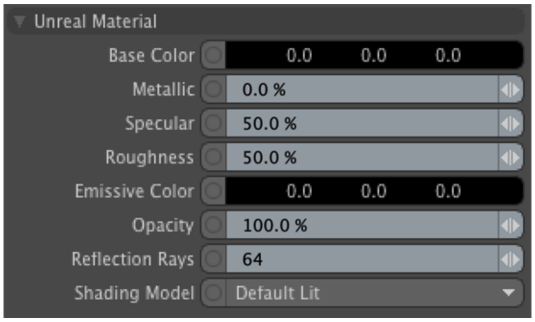
PBR Textures

Physically based rendering, or PBR, is a method of producing materials in 3D applications which simulates how light interacts with similar materials in the real world. The goal of physically based rendering is to produce an end product in your 3D rendering engine that provides photorealistic results on your object's surfaces.

Modo has several base materials available which use the PBR model, such as the Principled Shader,Unity Material, and Unreal Material.

Materials that support the PBR model offer a series of inputs which control how light and surface attributes are evaluated for final rendering. Specular Amount, Roughness, and Metallic are some of the inputs you might expect to see on a PBR material.

You can control these inputs through a series of UI controls and sliders within the material in the Shader Tree, or you can import textures into the Shader Tree, where each texture map is assigned to a specific input on the material. The value of each pixel within the imported texture map controls the intensity, color, or value of the specific input.



Unreal Material

Base Color 0.0 0.0 0.0

Metallic 0.0 %

Specular 50.0 %

Roughness 50.0 %

Emissive Color 0.0 0.0 0.0

Opacity 100.0 %

Reflection Rays 64

Shading Model Default Lit

The **Unreal Material** input sliders.

PBR Loader

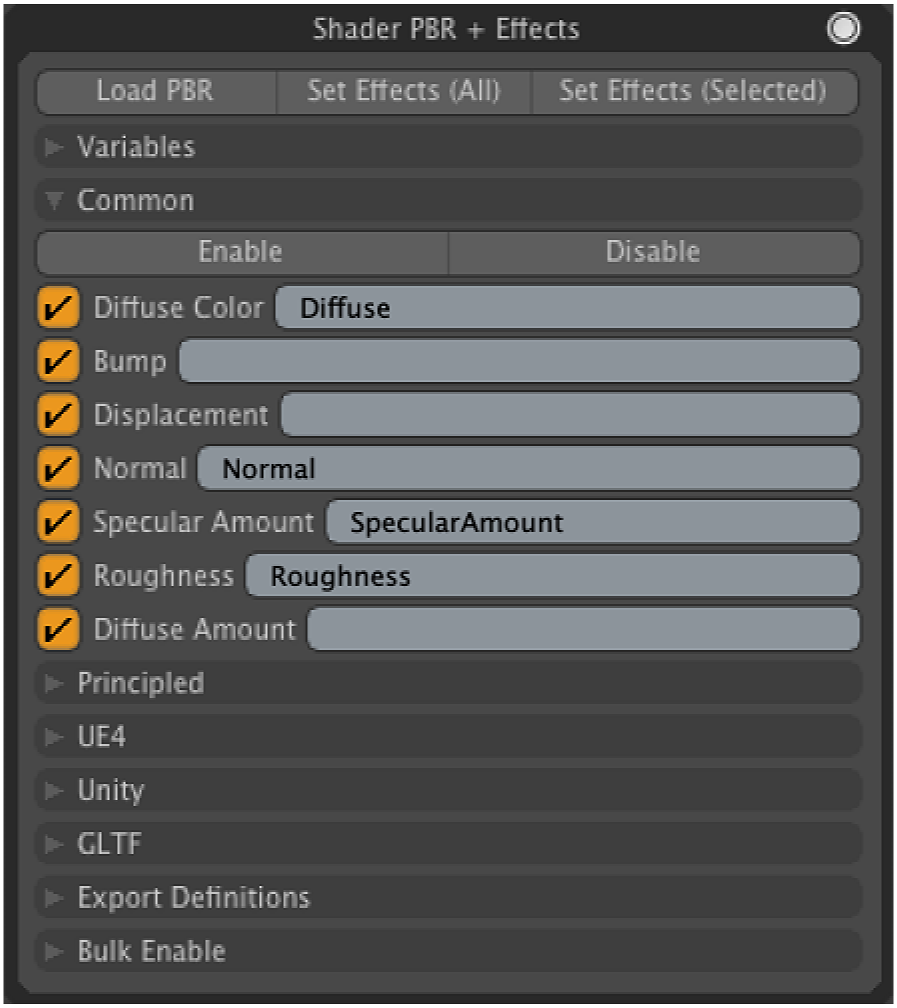
You can use the Modo PBR loader to import textures for use in PBR materials. Click **Texture** > **PBR** > **Load PBR**, or click the PBR icon in the **Shading** tab above the Shader Tree in the **Modo** layout.

PBR Textures Filename Conventions

Make sure your textures follow naming conventions when importing them for PBR materials. The PBR loader in Modo assumes that the name and input type of the image are separated by an underscore ("_"). For example, **myImageName_inputType.tga**, **inputType_myImageName.tga** are both valid conventions.

The PBR loader searches the image name string and looks for any matches defined in the **Shader PBR + Effects** popover. You can enter definitions in the **Shader PBR + Effects** popover to tell Modo which **Effect** should be assigned to imported textures in the Shader Tree.

For example, an artist enters **Diffuse** in the **Diffuse Color** property of the **Shader PBR + Effects** popover and then imports a texture named **myImageName_2k_Diffuse.tga**. Modo identifies the **Diffuse** input type in the filename and assigns the imported texture a **Diffuse Color** effect in the Shader Tree.



Shader PBR + Effects
Load PBR
Set Effects (All)
Set Effects (Selected)
Variables
Common
Enable
Disable

☑ Diffuse Color: Diffuse

☑ Bump:

☑ Displacement:

☑ Normal: Normal

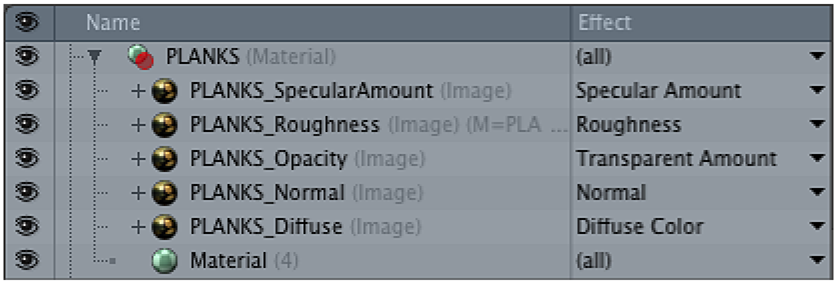
☑ Specular Amount: SpecularAmount

☑ Roughness: Roughness

☑ Diffuse Amount:

Principled
UE4
Unity
GLTF
Export Definitions
Bulk Enable

Defining common input types in the **Shader PBR + Effects** popover.



Name	Effect
PLANKS (Material)	(all)
PLANKS_SpecularAmount (Image)	Specular Amount
PLANKS_Roughness (Image) (M=PLA ...	Roughness
PLANKS_Opacity (Image)	Transparent Amount
PLANKS_Normal (Image)	Normal
PLANKS_Diffuse (Image)	Diffuse Color
Material (4)	(all)

Imported PBR textures with an **Effect** assigned to each in the Shader Tree based on the definitions in the **Shader PBR + Effects** popover.

Note: **Effect** in the Shader Tree tells Modo what kind of effect the selected layer should have on the object's surface. See Layer Effects for more information.

Note: You can assign type names for material inputs in the **Shader PBR + Effects** popover. These definitions are not case sensitive. See PBR Textures for more information.

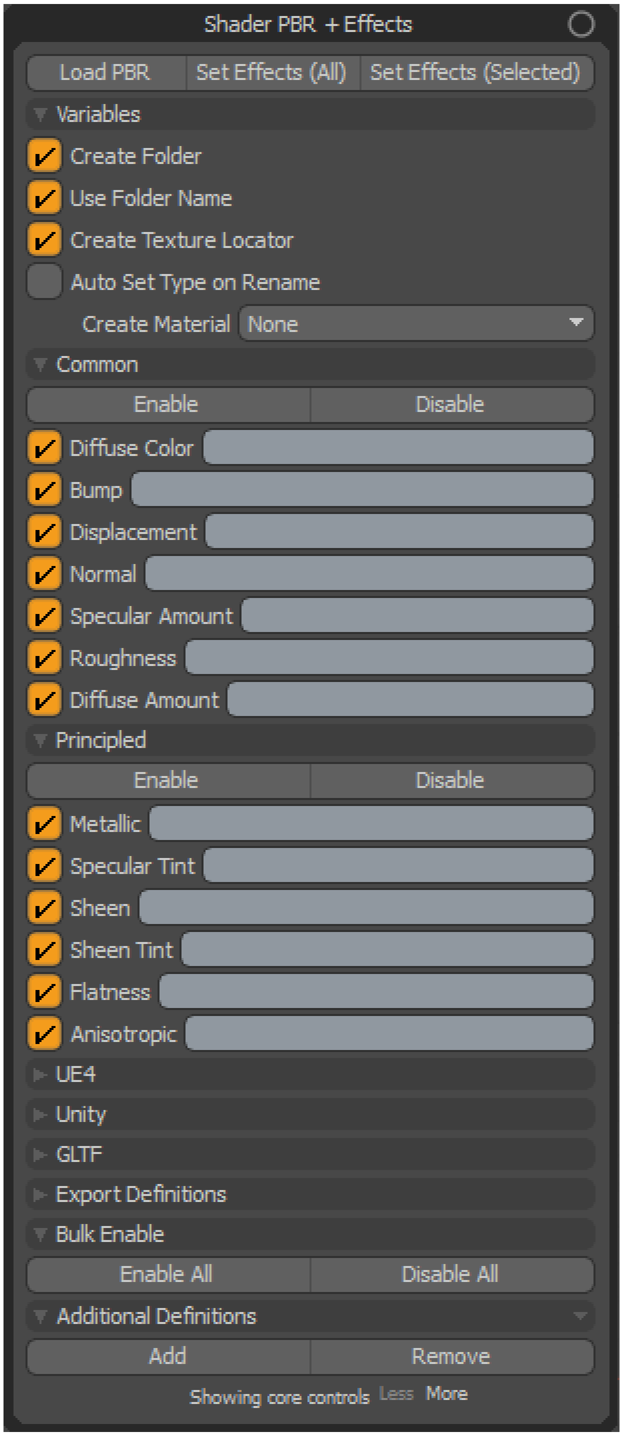
Importing Unreal PBR Textures with the PBR Loader

1. From the **Texture** > **PBR** menu, open the **Preferences Popover**.

   The **Shader PBR + Effects** popover opens.

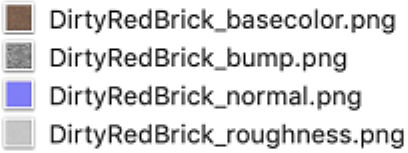
2. Expand the **UE4** menu and click **Enable**.

   All definitions under the **UE4** menu are enabled.


Shader PBR + Effects
Load PBR
Set Effects (All)
Set Effects (Selected)
Variables
☑ Create Folder
☑ Use Folder Name
☑ Create Texture Locator
☐ Auto Set Type on Rename
Create Material None
Common
Enable
Disable
☑ Diffuse Color
☑ Bump
☑ Displacement
☑ Normal
☑ Specular Amount
☑ Roughness
☑ Diffuse Amount
Principled
Enable
Disable
☑ Metallic
☑ Specular Tint
☑ Sheen
☑ Sheen Tint
☑ Flatness
☑ Anisotropic
UE4
Unity
GLTF
Export Definitions
Bulk Enable
Enable All
Disable All
Additional Definitions
Add
Remove
Showing core controls Less More

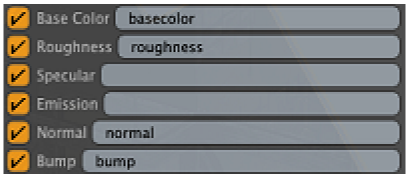
3. Under the **UE4** menu, enter a string that Modo can search for during import next to each of the texture map types you want to import.

   For example, if you use the word **basecolor** in your diffuse map filenames, type **basecolor** into the **Base Color** property, and if you use the word **normal** in your normal map filenames, type **normal** into the **Normal** property.



DirtyRedBrick_basecolor.png
DirtyRedBrick_bump.png
DirtyRedBrick_normal.png
DirtyRedBrick_roughness.png

The texture files on disk.



☑ Base Color basecolor
☑ Roughness roughness
☑ Specular
☑ Emission
☑ Normal normal
☑ Bump bump

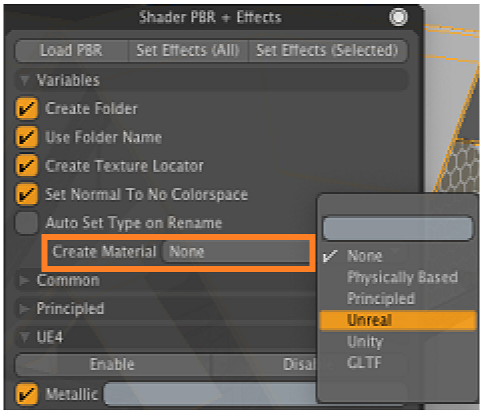
Definitions entered into the UE4 texture map types based on the texture filenames on disk.

Note: These terms do not need to be case sensitive, but they need to match the exact term in the filename which has to be separated by an underscore from the rest of the filename, for example **myTextureMap_basecolor.png**.

Tip: If you use several terms to define your texture maps, you can use multiple definitions for one input by separating each definition with a semi-colon. For example, **Albedo;Base;Col**.

4. Under the **Variables** menu, select **Unreal** from the **Create Material** dropdown.

   An Unreal material is created for your PBR textures once you import them.

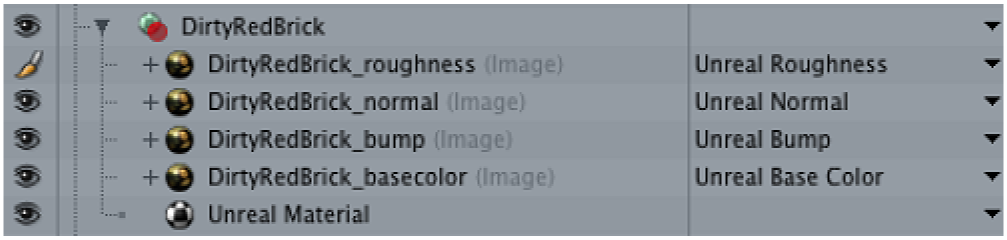


Shader PBR + Effects
Load PBR | Set Effects (All) | Set Effects (Selected)
Variables
☑ Create Folder
☑ Use Folder Name
☑ Create Texture Locator
☑ Set Normal To No Colorspace
☐ Auto Set Type on Rename
Create Material None
Common
Principled
UE4
Enable | Disal
☑ Metallic

5. Under the **Variables** menu, ensure **Create Folder** and **Use Folder Name** are enabled, then close the **Shader PBR + Effects** popover.

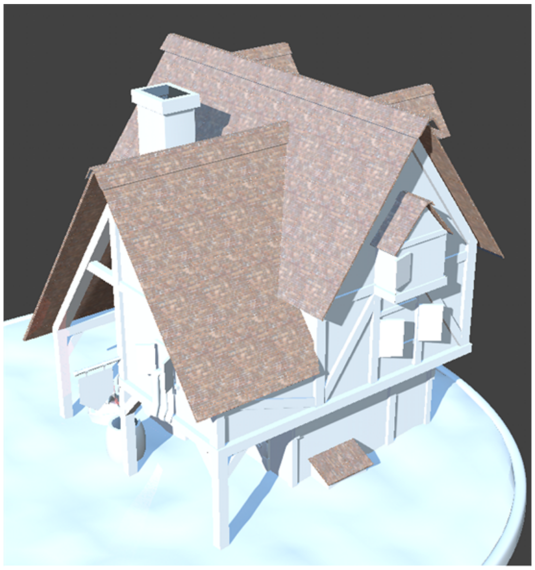
   A folder layer is created in the Shader Tree for your PBR textures once textures are imported, and the folder layer is named after the folder containing the textures on disk.
6. From the **Texture** > **PBR** menu, click **Load PBR**.
7. Select all the textures you want to import for your Unreal material and click **Open**.

The textures are imported to the Shader Tree and each assigned an **Effect** based on the definitions created in step 3.



DirtyRedBrick	
DirtyRedBrick_roughness (Image)	Unreal Roughness
DirtyRedBrick_normal (Image)	Unreal Normal
DirtyRedBrick_bump (Image)	Unreal Bump
DirtyRedBrick_basecolor (Image)	Unreal Base Color
Unreal Material	

Tip: Select the material folder layer and use the **Polygon Tag** dropdown in **Properties** > **Texture Layers** to assign the material to a set of polygons.



The Unreal material assigned to the roof of the asset.

Adding Custom Definitions

If there are no definitions available for your preferred workflow in the **Shader PBR + Effects** popover, you can add your own custom definitions. The following example walks you through how to add a custom definition for diffuse color textures that you want to use as ambient occlusion maps. Diffuse color maps can be used as ambient occlusion maps as a quick way to obtain ambient occlusion:

1. From the **Texture** > **PBR** menu, click **Preferences Popover**.
2. Under **Additional Definitions**, click **Add**.

   A new definition is added to the popover and options are exposed.

3. Enter the filename convention for your ambient occlusion maps in **Search String**, for example: **AO**.

   To learn more about the PBR loader filename conventions, see PBR Textures Filename Conventions. The PBR loader searches the imported texture or material name for this string, and assigns the **RGBA** effect.
4. In the **RGBA** dropdown, select **Diffuse Color**.

   The Diffuse Color effect is applied to textures imported with **AO** in the filename.
5. In the **Blend Mode** dropdown, select **Multiply**.

   The Multiply blend mode is assigned so darks and lights within the texture are multiplied on top of textures below this layer, creating the effect of shadows and highlights.

Adding Definitions for Packed Maps

Packed texture maps allow you to have a single texture which packs different map types into each RGBA channel. For example, you could have a single texture file named OcclusionRoughnessMetallic where the red channel contains the ambient occlusion map, the green channel contains the roughness map, and the blue channel contains the metallic map.

The PBR loader supports the loading of packed maps, and assigns each channel the desired effect.

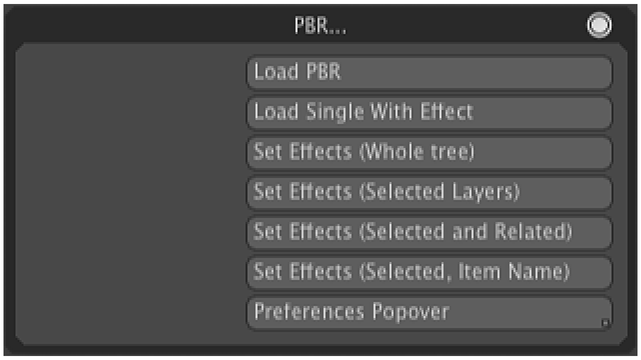
To define effects for packed maps and their channels:

1. From the **Texture** > **PBR** menu, click **Preferences Popover**.
2. Under **Additional Definitions**, click **Add**.
3. Enter the filename convention for your packed maps in **Search String**, for example: **packed**.
4. Enable **Packed**.
5. Select the desired effect for the red channel from the **Red** dropdown.

   For example, if your red channels contain the roughness map, select **Roughness** from the dropdown.
6. Select the desired effects for the **Green**, **Blue**, and **Alpha** channels. Set **(none)** if you don't want an effect defined for that channel.

PBR Loader Options

Click **Texture** > **PBR** or click the PBR button in the **Shading** tab above the Shader Tree in the **Modo** layout to access the PBR loader options.


PBR...
Load PBR
Load Single With Effect
Set Effects (Whole tree)
Set Effects (Selected Layers)
Set Effects (Selected and Related)
Set Effects (Selected, Item Name)
Preferences Popover

The PBR loader options.

Loading PBRs

You can load PBR textures by clicking the **Load PBR** option. This options allow you to select an image or multiple images to import. Once an image is selected, the loader imports the image and any image in the same folder that has the same name into the Shader Tree. If multiple files are selected, Modo loads just those in without searching for others.

For example, the PBR loader assumes **myimagename_2k** is the name of the texture, and looks for other files in the same folder with **myimagename_2k** in the filename: **myimagename_2k_diffuse.tga**.

If you only want to import a single PBR texture, click **Load Single With Effect**. This option does not search for other textures in the same folder with the same name.

Each imported image is then assigned an **Effect** in the Shader Tree based on the input type defined in the file name. In the following example, the PBR loader assumes **diffuse** is the input type: **myimagename_2k_diffuse.tga**.

Note: You can define input types in the PBR Textures.

Setting Effects

You can assign **Effects** to existing layers in the Shader Tree using the Set Effects options. These layers do not have to contain image clips, and depending on which Set Effect option you select, Modo searches either the layer name or the image clip name for the name and PBR definition.

To set effects to existing layers in the Shader Tree:

1. Click **Texture** > **PBR** > **Preferences Popover** and make sure the **Shader PBR + Effects** popover is populated with the PBR definitions you require.

   See Importing Unreal PBR Textures with the PBR Loader and PBR Textures for more information.

2. Select a layer in the Shader Tree you would like to automatically assign an **Effect** to.

> **Note:** The image clip in the layer or the layer name must use the PBR naming conventions based on the definitions you set in the **Shader PBR + Effects** popover. See PBR Textures Filename Conventions.

3. Click **Texture** > **PBR** > **Set Effects (Selected Layers)** if you'd like to automatically assign an effect to only the selected layer based on the layer's image clip, OR

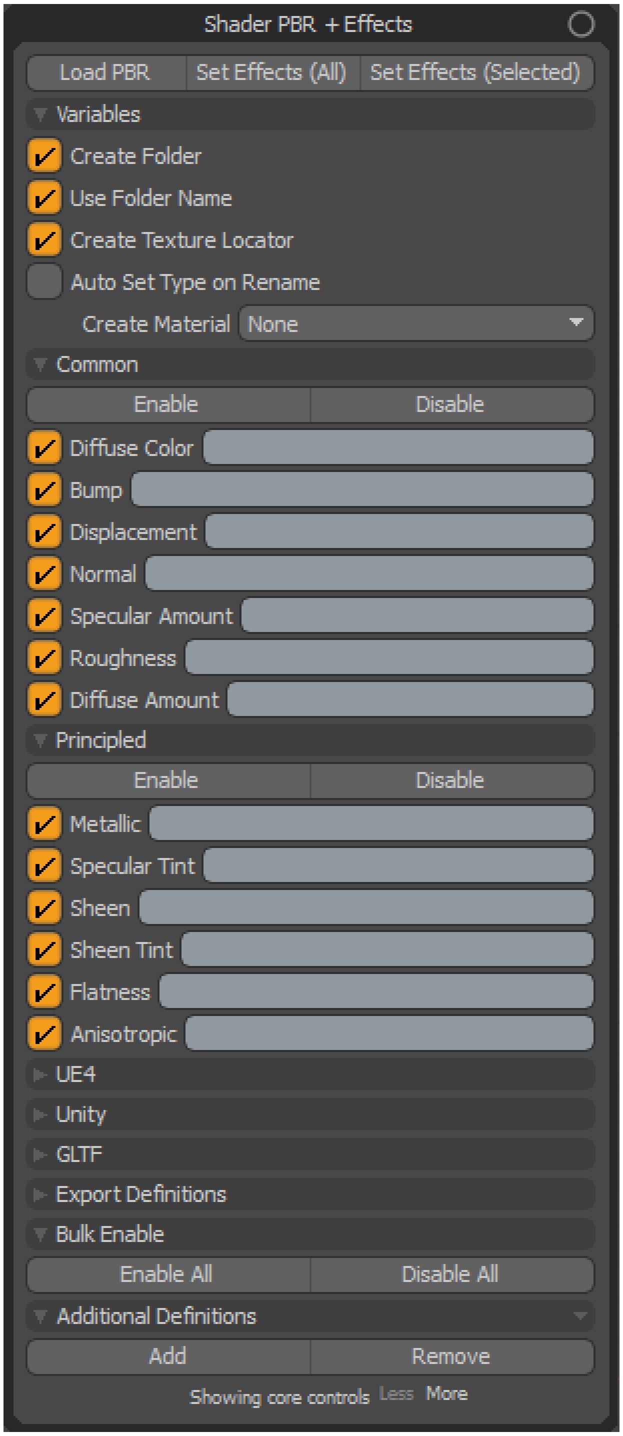
   Click **Set Effects (Selected and Related)** if you'd like to automatically assign effects to the selected layer and all layers sharing the same image clip name, OR

   Click **Set Effects (Whole Tree)** if you'd like to automatically assign effects to all layers in the Shader Tree based on each of the layer names, regardless of whether the layer contains an image clip or not, OR

   Click **Set Effects (Selected, Item Name)** if you'd like to automatically assign effects to the selected layer and all layers sharing the same layer name, regardless of whether the layer contains an image clip or not.

PBR Loader Preferences

Click **Texture** > **PBR** > **Preferences Popover**, or click the PBR button > **Preferences Popover** in the **Shading** tab above the Shader Tree in the **Modo** layout to access the PBR Loader preferences. This opens the **Shader PBR + Effects** popover.


Shader PBR + Effects
Load PBR
Set Effects (All)
Set Effects (Selected)
Variables
Common
Enable
Disable
Principled
Enable
Disable
UE4
Unity
GLTF
Export Definitions
Bulk Enable
Enable All
Disable All
Additional Definitions
Add
Remove
Showing core controls
Less
More

☑ Create Folder
☑ Use Folder Name
☑ Create Texture Locator
☐ Auto Set Type on Rename
Create Material: None

☑ Diffuse Color
☑ Bump
☑ Displacement
☑ Normal
☑ Specular Amount
☑ Roughness
☑ Diffuse Amount

☑ Metallic
☑ Specular Tint
☑ Sheen
☑ Sheen Tint
☑ Flatness
☑ Anisotropic

The **Shader PBR + Effects** popover.

Variables

Variables settings affect the way PBR textures are imported into the Shader Tree.

Create Folder	When enabled, a single folder layer is created in the Shader Tree for the textures imported with the PBR loader.
Use Folder Name	When enabled, the folder created with **Create Folder** is renamed to match the name of the folder on disk to which the imported textures belong to.
Create Texture Locator	When enabled, a single texture locator is created and each imported texture is assigned to it. When disabled, each imported texture is assigned to its own texture locator.
Set Normal To No Colorspace	When enabled, any normal maps imported with the PBR loader have their **Colorspace** property set to **None**. When disabled, the **Colorspace** of imported normal maps is set to **default**.
Auto Set Type on Rename	When enabled, a listener is added to listen for item rename operations, and when a texture layer is renamed in the Shader Tree, the appropriate **Effect** is assigned to the item in the Shader Tree. For example, if a texture layer is renamed to **myImageLayer_Normal**, the **Effect** is set to whatever **Normal** is defined as in the Preferences Popover.
Create Material	Select one of the following options from the **Create Material** dropdown to add the selected material to the bottom of the group of imported textures in the Shader Tree: • **None** - Adds no material layer. • **Physically Based** - Adds a **Material** layer with the **Shading Model** set to **Physically Based**. • **Principled** - Adds a **Material** layer with the **Shading Model** set to **Principled**. • **Unreal** - Adds an **Unreal Material** layer. • **Unity** - Adds a **Unity Material** layer. • **GLTF** - Adds a **glTF Material** layer.

Common

Common definitions add PBR effects commonly used by shaders such as diffuse maps to the imported textures. These effects can usually be found in the **Basic Channels** menu in the **Effects** column of the Shader Tree.

Note: Definition names are not case sensitive.

Enable	Enable all **Common** definitions. **Common** definitions are used during the import of texture files through the PBR loader.
Disable	Disable all **Common** definitions. No **Common** definitions are used during the import of texture files through the PBR loader.
Diffuse Color	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Diffuse Color** effect to the imported texture if it is found. For example, entering **diffuse** into the property means any texture imported using the PBR loader with diffuse within the filename, separated by an underscore ("_") are assigned the **Diffuse Color** effect in the Shader Tree.
Bump	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Bump** effect to the imported texture if it is found. For example, entering **bump** into the property means any texture imported using the PBR loader with diffuse within the filename, separated by an underscore ("_") are assigned the **Bump** effect in the Shader Tree.
Displacement	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Displacement** effect to the imported texture if it is found. For example, entering **displacement** into the property means any texture imported using the PBR loader with displacement within the filename, separated by an underscore ("_") are assigned the **Displacement** effect in the Shader Tree.
Normal	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Normal** effect to the imported texture if it is found. For example, entering **normal** into the property means any texture imported using the PBR loader with normal within the filename, separated by an underscore ("_") are assigned the **Normal** effect in the Shader Tree.
Specular Amount	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Specular Amount** effect to the imported texture if it is found. For example, entering **specAmt** into the property means any texture imported using the PBR loader with specAmt within the filename, separated by an underscore ("_") are assigned the **Specular Amount** effect in the Shader Tree.
Roughness	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Roughness** effect to the imported texture if it is found.

	For example, entering **rgh** into the property means any texture imported using the PBR loader with rgh within the filename, separated by an underscore ("_") are assigned the **Roughness** effect in the Shader Tree.
Diffuse Amount	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Diffuse Amount** effect to the imported texture if it is found. For example, entering **diffAmt** into the property means any texture imported using the PBR loader with diffAmt within the filename, separated by an underscore ("_") are assigned the **Diffuse Amount** effect in the Shader Tree.

Principled

Principled definitions add Principled PBR effects to the imported textures. These effects can usually be found in the **Principled** menu in the **Effects** column of the Shader Tree.

Enable	Enable all **Principled** definitions. **Principled** definitions are used during the import of texture files through the PBR loader.
Disable	Disable all **Principled** definitions. **Principled** definitions are not used during the import of texture files through the PBR loader.
Metallic	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Principled** > **Metallic** effect to the imported texture if it is found. For example, entering **mtl** into the property means any texture imported using the PBR loader with mtl within the filename, separated by an underscore ("_") are assigned the **Principled** > **Metallic** effect in the Shader Tree.
Specular Tint	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Principled** > **Specular Tint** effect to the imported texture if it is found. For example, entering **specTint** into the property means any texture imported using the PBR loader with specTint within the filename, separated by an underscore ("_") are assigned the **Principled** > **Specular Tint** effect in the Shader Tree.
Sheen	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Principled** > **Sheen** effect to the imported texture if it is found. For example, entering **sheen** into the property means any texture imported using the PBR

	loader with sheen within the filename, separated by an underscore ("_") are assigned the **Principled** > **Sheen** effect in the Shader Tree.
Sheen Tint	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Principled** > **Sheen Tint** effect to the imported texture if it is found. For example, entering **shnTint** into the property means any texture imported using the PBR loader with shnTint within the filename, separated by an underscore ("_") are assigned the **Principled** > **Sheen Tint** effect in the Shader Tree.
Flatness	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Principled** > **Flatness** effect to the imported texture if it is found. For example, entering **flat** into the property means any texture imported using the PBR loader with flat within the filename, separated by an underscore ("_") are assigned the **Principled** > **Flatness** effect in the Shader Tree.
Anisotropic	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Principled** > **Anisotropic** effect to the imported texture if it is found. For example, entering **anis** into the property means any texture imported using the PBR loader with anis within the filename, separated by an underscore ("_") are assigned the **Special Effects** > **Anisotropic** effect in the Shader Tree.

UE4

UE4 definitions add Unreal PBR effects to the imported textures. These effects can usually be found in the **Unreal** menu in the **Effects** column of the Shader Tree.

Enable	Enable all **UE4** definitions. **UE4** definitions are used during the import of texture files through the PBR loader.
Disable	Disable all **UE4** definitions. **UE4** definitions are not used during the import of texture files through the PBR loader.
Metallic	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Metallic** effect to the imported texture if it is found. For example, entering **mtl** into the property means any texture imported using the PBR loader with mtl within the filename, separated by an underscore ("_") are assigned the **Unreal Metallic** effect in the Shader Tree.

Base Color	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Base Color** effect to the imported texture if it is found. For example, entering **color** into the property means any texture imported using the PBR loader with color within the filename, separated by an underscore ("_") are assigned the **Unreal Base Color** effect in the Shader Tree.
Roughness	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Roughness** effect to the imported texture if it is found. For example, entering **rgh** into the property means any texture imported using the PBR loader with rgh within the filename, separated by an underscore ("_") are assigned the **Unreal Roughness** effect in the Shader Tree.
Specular	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Specular** effect to the imported texture if it is found. For example, entering **spec** into the property means any texture imported using the PBR loader with spec within the filename, separated by an underscore ("_") are assigned the **Unreal Specular** effect in the Shader Tree.
Emission	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Emissive Color** effect to the imported texture if it is found. For example, entering **emisCol** into the property means any texture imported using the PBR loader with emisCol within the filename, separated by an underscore ("_") are assigned the **Unreal Emissive Color** effect in the Shader Tree.
Normal	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Normal** effect to the imported texture if it is found. For example, entering **norm** into the property means any texture imported using the PBR loader with norm within the filename, separated by an underscore ("_") are assigned the **Unreal Normal** effect in the Shader Tree.
Bump	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Bump** effect to the imported texture if it is found. For example, entering **bump** into the property means any texture imported using the PBR loader with bump within the filename, separated by an underscore ("_") are assigned the **Unreal Bump** effect in the Shader Tree.
Ambient	The PBR loader looks for the name entered here in the filename of imported textures and

Occlusion	assigns the **Unreal Ambient Occlusion** effect to the imported texture if it is found. For example, entering **AO** into the property means any texture imported using the PBR loader with AO within the filename, separated by an underscore ("_") are assigned the **Unreal Ambient Occlusion** effect in the Shader Tree.
Subsurface Color	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Subsurface Color** effect to the imported texture if it is found. For example, entering **srfCol** into the property means any texture imported using the PBR loader with srfCol at the end of the file name are assigned the **Unreal Subsurface Color** effect in the Shader Tree.
Clearcoat	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Clearcoat Amount** effect to the imported texture if it is found. For example, entering **clr** into the property means any texture imported using the PBR loader with clr within the filename, separated by an underscore ("_") are assigned the **Unreal Clearcoat Amount** effect in the Shader Tree.
Clearcoat Roughness	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Clearcoat Roughness** effect to the imported texture if it is found. For example, entering **clrRgh** into the property means any texture imported using the PBR loader with clrRgh within the filename, separated by an underscore ("_") are assigned the **Unreal Clearcoat Roughness** effect in the Shader Tree.
Opacity	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unreal Opacity** effect to the imported texture if it is found. For example, entering **opacity** into the property means any texture imported using the PBR loader with opacity within the filename, separated by an underscore ("_") are assigned the **Unreal Opacity** effect in the Shader Tree.

Unity

Unity definitions add Unity PBR effects to the imported textures. These effects can usually be found in the **Unity** menu in the **Effects** column of the Shader Tree.

Enable	Enable all **Unity** definitions. **Unity** definitions are used during the import of texture files

	through the PBR loader.
Disable	Disable all **Unity** definitions. **Unity** definitions are not used during the import of texture files through the PBR loader.
Metallic	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Metallic** effect to the imported texture if it is found. For example, entering **mtl** into the property means any texture imported using the PBR loader with mtl within the filename, separated by an underscore ("_") are assigned the **Unity Metallic** effect in the Shader Tree.
Albedo	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Albedo** effect to the imported texture if it is found. For example, entering **albedo** into the property means any texture imported using the PBR loader with albedo within the filename, separated by an underscore ("_") are assigned the **Unity Albedo** effect in the Shader Tree.
Smoothness	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Smoothness** effect to the imported texture if it is found. For example, entering **smooth** into the property means any texture imported using the PBR loader with smooth within the filename, separated by an underscore ("_") are assigned the **Unity Smoothness** effect in the Shader Tree.
Emission	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Emission** effect to the imported texture if it is found. For example, entering **emis** into the property means any texture imported using the PBR loader with emis within the filename, separated by an underscore ("_") are assigned the **Unity Emission** effect in the Shader Tree.
Normal	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Normal** effect to the imported texture if it is found. For example, entering **normal** into the property means any texture imported using the PBR loader with normal within the filename, separated by an underscore ("_") are assigned the **Unity Normal** effect in the Shader Tree.
Bump	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Bump** effect to the imported texture if it is found.

	For example, entering **bump** into the property means any texture imported using the PBR loader with bump within the filename, separated by an underscore ("_") are assigned the **Unity Bump** effect in the Shader Tree.
Ambient Occlusion	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Ambient Occlusion** effect to the imported texture if it is found. For example, entering **AO** into the property means any texture imported using the PBR loader with AO within the filename, separated by an underscore ("_") are assigned the **Unity Ambient Occlusion** effect in the Shader Tree.
Detail Mask	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Detail Mask** effect to the imported texture if it is found. For example, entering **detailMask** into the property means any texture imported using the PBR loader with detailMask within the filename, separated by an underscore ("_") are assigned the **Unity Detail Mask** effect in the Shader Tree.
Detail Albedo x2	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Detail Albedo x2** effect to the imported texture if it is found. For example, entering **detailAlbedo** into the property means any texture imported using the PBR loader with detailAlbedo within the filename, separated by an underscore ("_") are assigned the **Unity Detail Albedo x2** effect in the Shader Tree.
Detail Normal	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **Unity Detail Normal** effect to the imported texture if it is found. For example, entering **detailNormal** into the property means any texture imported using the PBR loader with detailNormal within the filename, separated by an underscore ("_") are assigned the **Unity Detail Normal** effect in the Shader Tree.

GLTF

GLTF definitions add glTF PBR effects to the imported textures. These effects can usually be found in the **glTF** menu in the **Effects** column of the Shader Tree.

Enable	Enable all **GLTF** definitions. **GLTF** definitions are used during the import of texture files through the PBR loader.

Disable	Disable all **GLTF** definitions. **GLTF** definitions are not used during the import of texture files through the PBR loader.
Metallic	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **glTF Metallic** effect to the imported texture if it is found. For example, entering **mtl** into the property means any texture imported using the PBR loader with mtl within the filename, separated by an underscore ("_") are assigned the **glTF Metallic** effect in the Shader Tree.
Base Color	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **glTF Base Color** effect to the imported texture if it is found. For example, entering **color** into the property means any texture imported using the PBR loader with color within the filename, separated by an underscore ("_") are assigned the **glTF Base Color** effect in the Shader Tree.
Roughness	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **glTF Roughness** effect to the imported texture if it is found. For example, entering **rgh** into the property means any texture imported using the PBR loader with rgh within the filename, separated by an underscore ("_") are assigned the **glTF Roughness** effect in the Shader Tree.
Emission	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **glTF Emission** effect to the imported texture if it is found. For example, entering **emis** into the property means any texture imported using the PBR loader with emis within the filename, separated by an underscore ("_") are assigned the **glTF Emission** effect in the Shader Tree.
Normal	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **glTF Normal** effect to the imported texture if it is found. For example, entering **normal** into the property means any texture imported using the PBR loader with normal within the filename, separated by an underscore ("_") are assigned the **glTF Normal** effect in the Shader Tree.
Ambient Occlusion	The PBR loader looks for the name entered here in the filename of imported textures and assigns the **glTF Ambient Occlusion** effect to the imported texture if it is found. For example, entering **AO** into the property means any texture imported using the PBR loader with AO within the filename, separated by an underscore ("_") are assigned the **glTF Ambient Occlusion** effect in the Shader Tree.

Export Definitions

Export Definitions	Opens a file explorer for you to pick a save location to store a configuration (**.cfg**) file. The configuration file stores all the material definitions in the **Shader PBR + Effects** popover. This is useful if you need to share definitions with other Modo users.
Clear Definitions	Removes all existing definitions in the **Shader PBR + Effects** popover.

Bulk Enable

Disable All	Disables all definitions in the **Shader PBR + Effects** popover.
Enable All	Enables all definitions in the **Shader PBR + Effects** popover.

Additional Definitions

The additional definitions allow you to add custom definitions and effects for materials and imported textures in the Shader Tree. You can also add definitions for packed maps with these options.

Add	Adds a new custom definition to the list. Click to display the following options: • **Enabled** - enables the definition. • **Search String** - the PBR loader looks for the name entered here in the filename of imported textures or materials in the Shader Tree and assigns the effect defined in **RGBA** to the imported texture if it is found. • **RGBA** - choose an effect to assign to textures or materials in the Shader Tree which contain the string defined in **Search String**. See Layer Effects to learn about all the available effects. • **Colorspace** - automatically assigns a colorspace to the imported texture or material defined in this definition. If you are defining a map that uses scalar data (such as roughness, normal, or bump maps) select **none**. Otherwise, select **default**. • **Invert** - enables **Invert** on the textures or materials in the Shader Tree which contain the string defined in **Search String**.

	See Invert to learn more. - **Blend Mode** - assigns the selected blend mode to the textures or materials in the Shader Tree which contain the string defined in **Search String**. See Layer Blend Modes to learn more. - **Packed** - when enabled, the RGBA channels are exposed to allow you to assign an effect to each channel in a packed map. See Layer Effects to learn about all the available effects. **Note:** Each time you click **Add**, a new definition is added to the list.
Remove	Removes the last custom definition in the list.

Additional Options

Click **More** at the bottom of the **Shader PBR + Effects** popover to expose the following common options in each menu:

Colorspace	Automatically assigns a colorspace to the imported texture or material defined in this definition. If you are defining a map that uses scalar data (such as roughness, normal, or bump maps) select **none**. Otherwise, select **default**. See Colorspace on Image Maps/Sequences for more information.
Invert	Enables **Invert** on the textures or materials in the Shader Tree for this definition. See Invert to learn more.
Blending Mode	Assigns the selected blend mode to the textures or materials in the Shader Tree defined in this definition. See Layer Blend Modes to learn more.

Nodal Shading

Historically in Modo, all material shading for a surface was defined through the Shader Tree exclusively, using a stack order paradigm. Layers higher in the list order obscured the lower layers unless a mask was applied or its blending mode was changed. This method of working is incredibly intuitive and flexible, however it did make certain types of setups difficult to manage, or certain effects impossible altogether.

With the introduction of Nodal Shading, you can control nearly any surface attribute on a per sample basis, using a variety of driving influences, producing effects that were simply not possible in earlier versions.

Nodal Shading is named so because it uses a network of nodes, simple visual representations of a specific function. These functions could be representative of an item and its channels, a mathematical operation, or even some type of instruction to evaluate data a certain way. These various nodes are then linked together creating a network that defines the flow of information and controls how it is evaluated, the resulting network is called a graph.

Working in this way provides a sort of visual programming methodology for creating the shading instructions, where each node applies its contribution and then passes on the information to the next node. The flow of information through the network is left to right, where the left side link connections provide inputs for the information, which is evaluated and then returned to the right side connections to produce the output. Once a network is set up and assigned to a surface, the associated graph is evaluated at render time to produce the final shaded results.

Tip: The **Schematic** viewport, used most often for rigging and particles, is the control interface where the shading graphs are defined. It is important to understand first that nodal shading is not a replacement for the Shader Tree, which is still a requirement to apply the shading to the actual surfaces for rendering. Each graph **must** contain at least one Shader Tree specific item. So a basic workflow when defining nodal shading is to add an item to the Shader Tree (placed into the appropriate target Material Group) and then that texture layer is added to the **Schematic** viewport for further network development.

Graph Building Basics

Working on a node graph is made more efficient by having multiple viewports available at once. You can easily navigate between the Shader Tree, the **Schematic** viewport and the **Channels** list within the default **Render** workspace layout. It is accessible in the menu bar under **Layout** > **Layouts** > **Render** (or simply the

Render tab, in the tabbed workspace). The **Schematic** panel, not visible by default, is located in the tabbed viewport group directly under the **Preview** window, by selecting the **Schematic** specific tab.

Any network graph needs at least two nodes. Since it is necessary to have at least one Shader Tree specific item in a node network, it is often easiest to start any new shading graph by adding the initial item there. If you are not familiar with the Shader Tree, it is highly recommended to read Shader Tree. This explains the adding of items to the tree and setting **Effects**. Once the item is added (and if necessary, positioned appropriately in the hierarchy), it can be placed into the **Schematic** viewport. This is done a couple of ways, either by drag and drop or by simply selecting the item layer in the Shader Tree and using the **Add Selected** button found in the **Schematic** viewport.

Additional nodes can then be added directly in the **Schematic** viewport using its **Add** button. Left-clicking the button opens a menu containing all possible nodes, though you'll mostly be sticking to the **Channel Modifiers**, **Shading Inputs**, and **Shader Nodes** menu options when working with nodal shading. Selecting any item in the menu adds it to the **Schematic** viewport. The resulting node can be repositioned by left-clicking and dragging on the header of the node to the new location, and releasing the mouse button. It is easiest to see the flow of the data working left to right, so when adding nodes in the **Schematic** viewport, it is best to arrange them in this order, with the final destination node positioned on the furthest right-hand side of the viewport.

Newly added nodes come in with either common channels intact, or with no channels at all, depending on the item. Channels are the actual operators in the node graph, and which ones used is dependent on the desired outcome. When a specific channel is required, it can be added to a node by first selecting the target node in the **Schematic**, then selecting one or more of its related channels in the **Channels** viewport. Using the same drag-and-drop action or **Add Selected** method outlined above, adds the channel(s) to **Schematic**, attaching them to their associated item.

The final step in creating a node graph is the connecting of the channels. This represents the actual data flow between items. There are two types of basic connections, an input or an output. Some channels have only one or the other, and others have both. Channels are connected from the right side of the node, the output, into the left side of another node, the input. The channel connections are designated by the small circle on the sides of the node. Clicking and dragging creates a noodle, a line drawn in the viewport that visually represents the connection, and dragging the noodle from an output to an input (or vice versa) and releasing the mouse button defines the actual connection, fixing the line between the channels. Even if the node is moved, the noodles adjust, ensuring the connection representation is retained.

Warning: This is the way a basic node graph is built, adding nodes, adding channels where necessary, and then making connections between those channels. There are also various rules governing the connections of different channel types that can be very helpful to understand when building up complex graphs. The full details of working with **Items** and **Channels** in the Schematic viewport are covered under Schematic Viewport.

Nodal Shading Basics

Knowing how to connect a graph is the easy part, but doesn't do much good if there is no knowledge of the underlying architecture. At its most basic level, **Nodal Shading** is all about driving values on a per-sample basis. To understand that better, let's break it up into its two parts:

- **Driving values** - A value is simply a number, but what is being driven? Each item in the Shader Tree has a number of attributes, these are the settings that appear in the **Properties** panel for each item when selected. So, for a **Material** item, this would be things like the **Diffuse Color** or **Reflection Amount** controls. Each of these controls is actually a channel on the item, where a **channel** is simply any attribute that can be animated, or in the case of nodal shading, controlled.

  In a schematic context, taking the RGB color output of one node and connecting it in to the RGB color input of another, say the Material's **Diffuse Color** channel, causes that incoming value to control, or drive, the channel it is connected to. The color defined in the Material setting itself no longer applies; its values are being overwritten by those of the incoming connection. So basically, any channel that is connected by its input is being driven by the incoming value. Driven channels are signified in the **Channels** viewport by the small gear icon with an arrow.

- **Per sample** - What makes up a sample? When rendering an image, if we focus on just a single pixel for now, this one pixel is made up from many multiples of samples to arrive at the final value. For example, a single ray trace operation fires a ray from the camera out into the scene in a straight line. If the ray hits a surface, at that location in 3D space the hit surface is sampled to find out what it looks like. Answering questions like what is the surfaces color, is it shiny, reflective, is it in shadow? all of these different possibilities are individual samples. Nodal shading gives you access to the sample engine at this sub-pixel level.

  Normally these samples are all derived directly from settings defined in the Shader Tree, but with nodal shading, things get really interesting because you can utilize the information from those samples directly, modify their results mid-stream, and then apply those results to different attributes altogether. The samples can be from a number of sources, and this is where the Shader Inputs come into play. Shader Inputs are information collecting nodes, outputting the results of the various samples taken. Each allows you to collect specific information about the scene and use what is collected to control something else.

  For example, the Illuminate Node evaluates if light is shining on a surface and how much, the Raycast Node allows you to create sampling rays that collect information about surfaces the rays interact with. The Ray Type Node separates shading information based on the actual type of ray used to compute it, and the Shader Inputs provide the actual information about anything that is being shaded, whether it is a surface, a particle, a volumetric, or a texture.

Nodal shading doesn't have to be about complex graphs, rays, or samples. Sometimes simple operations are all that are required. In the Shader Tree alone, you are limited to the available **Effect** settings that can be assigned for each layer. If every possible permutation of effect were available, the resulting list would be impossible to manage. Much more control is available with nodal shading. It is easy to, for example, modulate a grids **Line Width** with a texture, or even with the surfaces angle of incidence, or amount of illumination. Driving values on a per sample basis is a fairly simple concept, but also incredibly powerful when used to its fullest potential.

Building a Procedural Texture:

You can also drive render outputs nodally through input channels for the color and value of each render output. For more information on how to do this, see Driving Render Outputs Nodally.

Shading Nodes

The Shader Input nodes provide the specific sampling functionality used in Nodal Shading, providing values based on various scene-related parameters. The resulting samples can be fed into texture channels to produce unique effects not possible with the Shader Tree alone. The shader Channel Modifiers provide some additional functionality to control the colors in intuitive ways. These elements combined with the various Shader Tree Item Layers and the Channel Modifiers in a schematic graph provide a very powerful way to wield Modo's shading power.

Shader inputs		
	Illuminate Node	The Illuminate node is used to sample the light that falls onto a given surface.
	Raycast Node	The Raycast node adds the ability to fire rays based on user defined criteria, gather information from where the ray hits or doesn't hit, and then use this information to drive other channel values.
	Ray Type Node	When rendering an image, the first rays fired into the scene are Camera Rays. These are used to evaluate what the camera sees.
	Shader Inputs	The Shader Inputs make accessible the internal sample vector used for rendering.

Color Correct	
Color Blend	The Color Blend node combines two different inputs together, resulting in a single evaluated output.
Color Gamma	The Color Gamma node changes the gamma of an input.
Color Invert	The Color Invert node reverses the incoming color values, providing a negative image type of effect.
Color Kelvin	The Channel Color Kelvin node outputs RGB values based on the Kelvin thermal dynamics color-temperature color model.
Color Correct	The Color Correct node provides an alternate method to modify the values of a color input.
Color HSV	The Channel Color HSV node outputs RGB values based on the intuitive **Hue**, **Saturation** and **Value** color input controls.

Illuminate Node

The Illuminate node is used to sample the light that falls onto a given surface. For each sample, the result is either illuminated or in shadow. The sample results can be derived from specific light sources, or be a combination of all lights in the scene. This can be used, for example, to mask surface attributes to appear only in areas of illumination or shadow, or to use the illumination amount to drive a specific channel, for example, using it to scale the scale/size channel of a procedural texture.

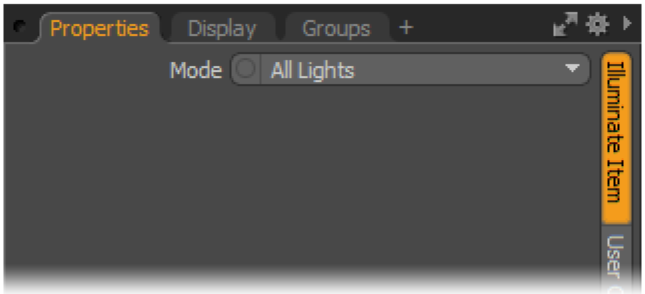
The Illuminate node is added to the Schematic Viewport using the **Add** function, found in the pop-up menu under **Shader Inputs** > **Illuminate**. With the node selected in the Schematic viewport, you can define the Mode, controlling if all lights in a scene are used to evaluate illumination, or if only the lights connected to the nodes **Light** input are evaluated. When working with connected lights the relationship output connector from the light is connected to the **Light** input on the node, you can connect any number of lights. Once connected, the results of the illumination are evaluated per sample and the value returned to its specific output.

Note: For information on working with node graphs, see Schematic Viewport

Illuminate Node Example:

Illuminate Position Example:

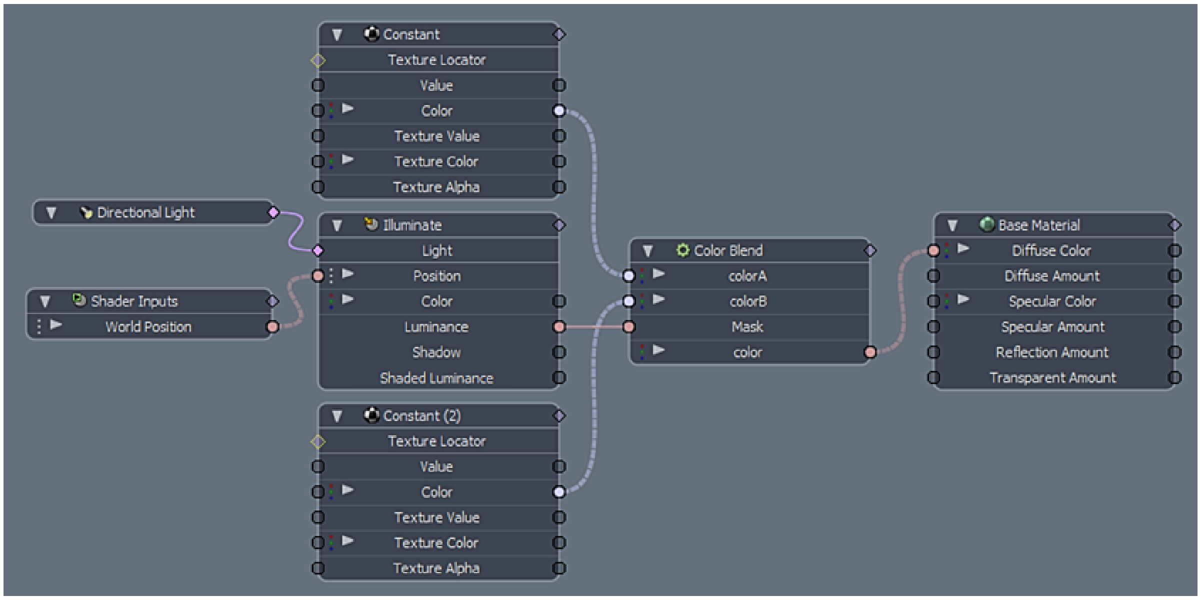
The following **Illuminate Item** options are available for the Illuminate node:


Properties
Display
Groups
+
Illuminate Item
User C

Mode: All Lights

- **Mode** - The **Mode** option determines sampling based on using specific lights in the scene with the **Connected Lights** option or using **All Lights** in the scene.

Schematic Node: Illuminate


Constant
Texture Locator
Value
Color
Texture Value
Texture Color
Texture Alpha
Directional Light
Illuminate
Light
Position
Color
Luminance
Shadow
Shaded Luminance
Shader Inputs
World Position
Color Blend
colorA
colorB
Mask
color
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount
Constant (2)
Texture Locator
Value
Color
Texture Value
Texture Color
Texture Alpha

Option	Description
Light Input	When the Illuminate items Mode option is set to **Connected Light** only the Light items specifically connected to the input are sampled as part of the node evaluation.

Option	Description
Position Input	The **Position** input defines the location of the sampling target. By default this is the World Position, a shader input is connected automatically, see Shader Inputs. The **World Position** setting means that the illumination sampling of the node takes place in the same location where the initial shading sample is derived. You can disconnect the **Shader** Input and connect a specific XYZ location if you want to sample the illumination of a specific point (like a light detector), or use any of the numerous vector math nodes to warp the sampling locations, to produce interesting results.
Color Output	Outputs the color of the connected light on a per sample basis.
Luminance Output	The output value represents where the illumination of the lights is present on the surface.
Shadow Output	The output value represents where the illumination of the lights is hidden in shadow on the surface.
Shaded Luminance Output	The output value represents the Diffuse amount (reflected light) from the illuminated surface.

Raycast Node

The Raycast node adds the ability to fire rays based on user defined criteria, gather information from where the ray hits or doesn't hit, and then use this information to drive other channel values. For example, the Raycast node makes it possible to make a virtual camera for a scene, what it sees, the results of the ray evaluations, would be displayed on another surface elsewhere in the scene, determined by the output connections.

To create a ray, you need to have both a position and a direction. These can be a specific defined vector value or be representative of shading values within the scene, such as a position and normal sample from a surface using a shader input connection, see Shader Inputs. The inputs are connected to the Origin and Direction, and represent the ray that is fired for sampling, and are evaluated per sample. If nothing is connected to these inputs, the current sample position and/or ray direction is used.

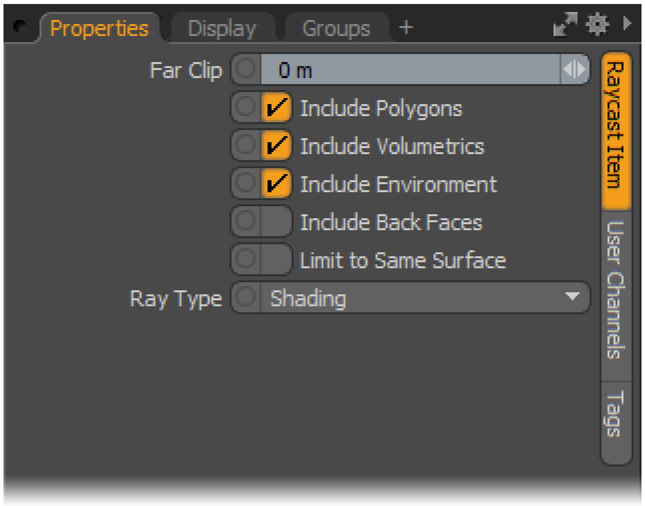
Once the sampling ray is defined it is also necessary to specify what the ray should see in the scene, either polygons, volumetrics, and so on. These options are defined in the **Properties** panel when the node is selected. Once the target for sampling has been hit, it needs to be determined what is sampled from the hit- Shading, Bump/Normal and/or Opacity. The results of what the ray samples are then returned to the various outputs on the node. These values can then be connected to the input of another channel, driving its value.

The Ray Cast node is added to the **Schematic** viewport using the **Add** function, found in the pop-up menu under **Shader Inputs** > **Ray Type**. There are no specific inputs for the node, but each type of selected ray is evaluated automatically for each sample and returned to its specific output.

Note: For information on working with node graphs, see Schematic Viewport

Video: Watch the Raycast and Raycast Vector example videos.

The following **Raycast Item** options are available for the Raycast node:



Far Clip 0 m

☑ Include Polygons

☑ Include Volumetrics

☑ Include Environment

☐ Include Back Faces

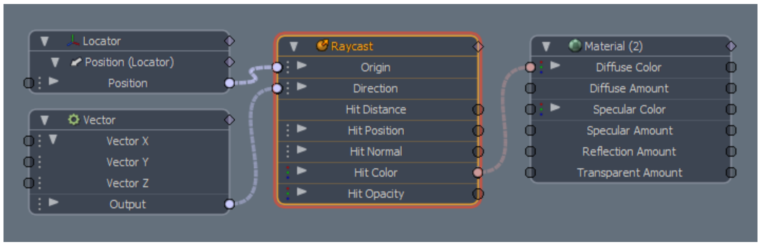
☐ Limit to Same Surface

Ray Type Shading

Option	Description
Far Clip	The **Far Clip** option controls the maximum sampling ray length. When set to 0, the ray is not clipped (infinite ray).
Include Polygons	This toggle determines if **Polygons** are sampled by the defined ray, this includes all the geometry in the scene.
Include Volumetrics	This toggle determines if **Volumetrics** are sampled by the defined ray, including Blobs, Volumes and Sprites.
Include	This toggle determines if the '**Environment** is sampled by the defined ray (The

Option	Description
Environment	environment is the result of not hitting anything else in the scene).
Include Back Faces	This toggle determines if **Back Faces** of polygons are sampled by the defined ray.
Limit to Same Surface	This toggle limits the sample made by the Raycast node to the same surface the ray is cast from (adjoining surfaces would be ignored, per surface of course).
Ray Type	The Ray Type option determines the type of sampling ray that is generated by the node.
Shading	Samples the fully shaded position, including Diffuse, Specular, and Reflection and the illumination on the surface.
Shadow	Samples if position is illuminated or in **Shadow**.
Material	Samples material Diffuse Color value only, ignoring the additional shading contributions.
Normal	Samples the shading normal, including perturbations of the ray caused by Bump Maps, Normals Maps, and Displacement.

Schematic Node: Raycast


Locator
Position (Locator)
Position
Vector
Vector X
Vector Y
Vector Z
Output
Raycast
Origin
Direction
Hit Distance
Hit Position
Hit Normal
Hit Color
Hit Opacity
Material (2)
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount

Option	Description
Origin Input	This input channel determines the origin of the sampling ray that is fired for scene evaluation.
Direction Input	This input channel determines the ray direction of the sampling ray that is fired for scene evaluation.
Hit Distance	This output channel represents the length of the ray that is fired, defined by its

Option	Description
Output	sample origin position to the hit surface. This doesn't include the distance from the camera to the sampling position.
Hit Position Output	This output channel represents the position in 3D space where the ray hit occurs.
Hit Normal Output	This output channel represent the evaluated normal direction of the hit surface.
Hit Color Output	This output channel represent the evaluated color of the hit surface.
Hit Opacity Output	This output channel represent the evaluated transparency of the hit surface.

Ray Type Node

When rendering an image, the first rays fired into the scene are Camera Rays. These are used to evaluate what the camera sees. When a particular ray hits a surface, the shading of that surface is evaluated to produce the final pixel. For simple surfaces this may be the only ray fired, but in most cases additional rays are created based on the surface settings, such as when hitting a mirrored surface a Reflection ray is created, or with global illumination Indirect (light bounce) rays are generated. All the results of these various ray types are combined to make the final image. The Ray Type node allows you to be able to selectively control attributes based on the type of ray used to compute the value.

Each of the Ray Types work as boolean value, they either are that type, or they are not, returning what would essentially look like a fully black or white image. This allows for all kinds of interesting combinations, such as changing values on a surface when it reflected off of another surface, a sphere appearing red, but reflecting green for instance.

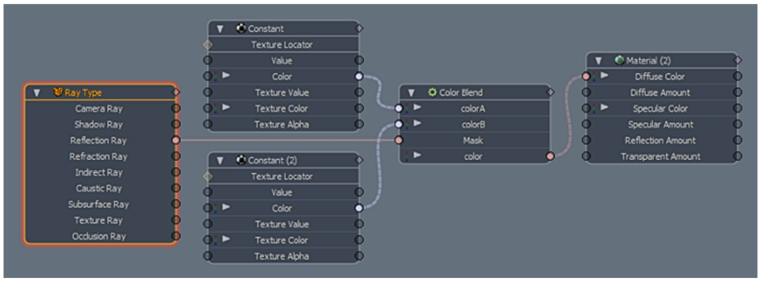
Usage

The Ray Type node is added to the **Schematic** viewport using the **Add** function, found in the pop-up menu under **Shader Inputs** > **Ray Type**. There are no specific inputs for the node, but each type of selected ray is evaluated automatically for each sample and returned to its specific output.

Note: For information on working with node graphs, see Schematic Viewport

Video: Watch this Ray Type example video.

Schematic Node: Ray Type


Ray Type
Camera Ray
Shadow Ray
Reflection Ray
Refraction Ray
Indirect Ray
Caustic Ray
Subsurface Ray
Texture Ray
Occlusion Ray
Constant
Texture Locator
Value
Color
Texture Value
Texture Color
Texture Alpha
Constant (2)
Texture Locator
Value
Color
Texture Value
Texture Color
Texture Alpha
Color Blend
colorA
colorB
Mask
color
Material (2)
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount

Option	Description
Camera Ray Output	A **Camera Ray** would be a ray that leaves the rendering camera for evaluation and intersects a geometric surface.
Shadow Ray Output	A **Shadow Ray** is generated from a surface to determine if it is lit by a direct light source or blocked by an object.
Reflection Ray Output	A **Reflection Ray** is a ray generated bouncing off of a reflective surface.
Refraction Ray Output	A **Refraction Ray** is a ray that is traced through a transparent/translucent surface.
Indirect Ray Output	An **Indirect Ray** is generated from a surface to determine if it is illuminated by surrounding objects.
Caustic Ray Output	A **Caustic Ray** is a light ray that is focused through a transparent surface or reflected off a mirrored surface.
Subsurface Ray Output	A **Subsurface Ray** is a light ray that is bounced through a surface and blended with its diffuse color value.
Texture Ray Output	A **Texture Ray** is a geometric surface with a Texture layer applied to it to control its shading.
Occlusion Ray Output	An **Occlusion Ray** is generated in semi-random directions from a surface to determine how much of the environment is visible, evaluating if the position is blocked by surrounding surfaces or not.

Shader Inputs

The Shader Inputs make accessible the internal sample vector used for rendering. This information is essentially the data that is used to describe any given shading sample. This provides access to computed values at the shader level, on a per sample basis.

Usage

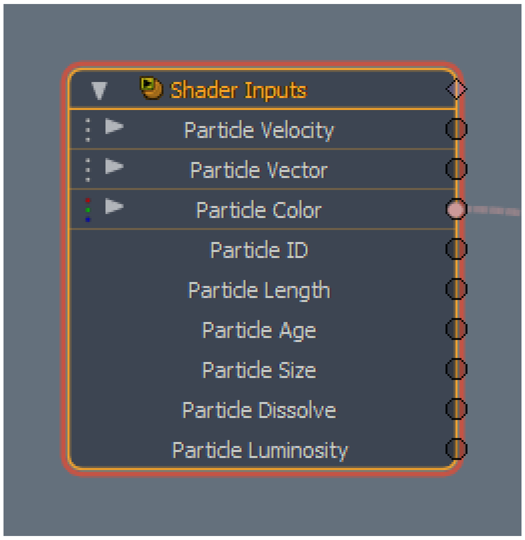
The Shader Input node is added to the **Schematic** viewport using the **Add** function, found in the pop-up menu under **Shader Inputs** > **Shader Inputs**. The Shader Inputs node is unique, in that it comes in without any specific channel associations. This is because there are many channels that can be used by the node, and having them all present at once would be overwhelming. Evaluation channels can be added as packages on an as-needed basis. This is done by right-clicking on the resulting node in the schematic and choosing the appropriate option from the **Add Shader Input** menu option. This can also be used to add additional Shading Input packages to existing Shader Input nodes.

For ease of use, it is also possible to add a Shader Input and channel packages in one step, using the additional options found in the **Schematic** viewports **Add** menu. In the **Shader Inputs** sub-menu, there are options for each Package type available: Particle Sample, Sample Position, Sample Ray, and Surface Normal.

There are no specific inputs for the node, but each sample type is evaluated automatically for every position and returned to its specific output.

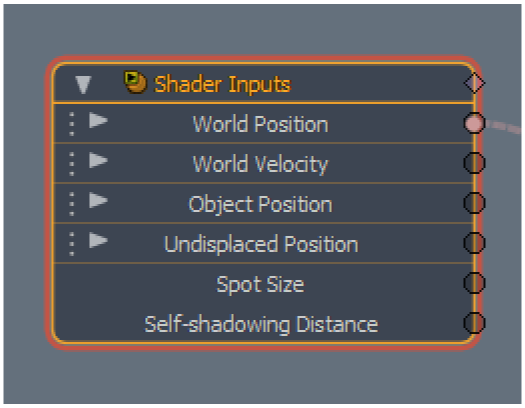
Note: For information on working with node graphs, see Schematic Viewport.

Schematic Node: Shader Inputs


Shader Inputs
Particle Velocity
Particle Vector
Particle Color
Particle ID
Particle Length
Particle Age
Particle Size
Particle Dissolve
Particle Luminosity

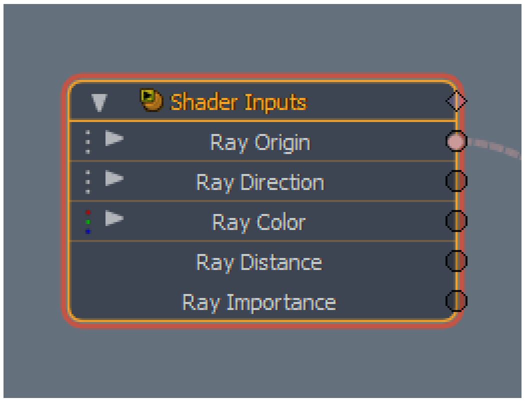
Particle Samples

Option	Description
Particle Velocity	Outputs the computed Particle Velocity values.
Particle Vector	Outputs the computed Particle Vector values.
Particle Color	Outputs the computed Particle Color values.
Particle ID	Outputs the computed Particle ID value.
Particle Length	Outputs the computed length of the particles.
Particle Age	Outputs the computed age of the particles since birth.
Particle Size	Outputs the computed size of the particles.
Particle Dissolve	Outputs the computed transparency of the particles.
Particle Luminosity	Outputs the computed luminosity of the particles.


Shader Inputs
World Position
World Velocity
Object Position
Undisplaced Position
Spot Size
Self-shadowing Distance

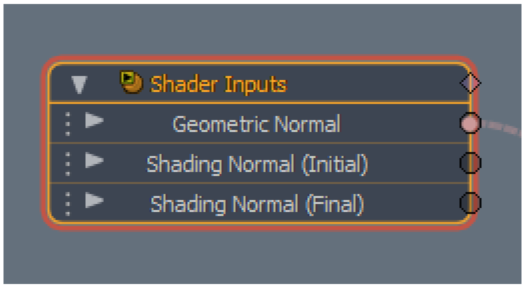
Sample Position

Option	Description
World Position	Outputs the computed position of a given sample in world space coordinates X, Y, and Z.
World Velocity	Outputs the computed world velocity of a given sample X, Y, and Z.
Object Position	Outputs the position of a given sample in Object Space coordinates X, Y, and Z.
Undisplaced Position	Outputs the undisplaced Position of a given sample on a geometric surface in Object Space coordinates X, Y, and Z.
Spot Size	
Self-shadowing Distance	


Shader Inputs
Ray Origin
Ray Direction
Ray Color
Ray Distance
Ray Importance

Sample Ray

Option	Description
Ray Origin	Outputs the computed sample **Ray Origin** position for a given sample.
Ray Direction	Outputs the computed sample **Ray Direction** for a given sample.
Ray Color	Outputs the computed **Ray Color** for a given sample.
Ray Distance	Outputs the computed ray length for a given sample.
Ray Importance	Outputs the Rays computed Importance value for a given sample.



Surface Normal

Option	Description
Geometric Normal	Outputs the geometric normal of a given sample.
Shading Normal (Initial)	Outputs the unperturbed normal of a given sample.
Shading Normal (Final)	Outputs the perturbed normal of a given sample.

Color Blend

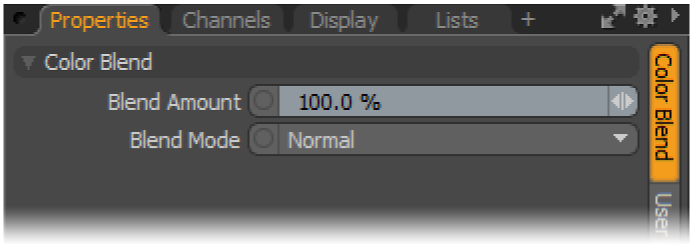
The Color Blend node combines two different inputs together, resulting in a single evaluated output. A third input masks the operation, allowing its input value to drive the blending amount. The Color Blend attributes also allow you to choose the type and amount of blending to perform, adding additional flexibility to the blend operation.

Usage

The Color Blend node is added to the Schematic Viewport using the **Add** function, found in the pop-up menu under **Channel Modifiers** > **Color** > **Color Blend**. Once added, the RGB color (or Value) outputs of other nodes can be connected to the **A** or **B** inputs. Additionally, a single channel **Mask** Value can be specified to control the blending between the two values texturally or per sample, based on the input (this is essentially like controlling the **Blend Amount** with the mask values). The result of the blend evaluations are then sent through the **Color** output for connecting to another RGB or value input.

Note: For information on working with node graphs, see Schematic Viewport

When the node is selected, attributes appear in the **Properties** panel.


Properties
Channels
Display
Lists
+
Color Blend
User

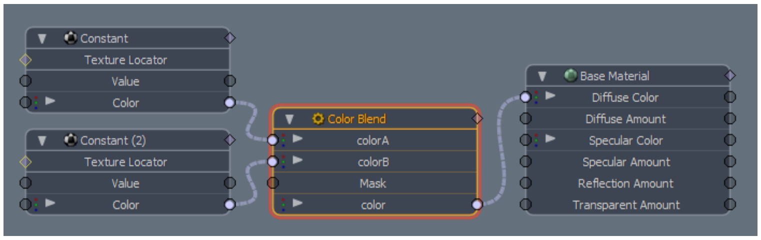
Color Blend

Blend Amount 100.0 %

Blend Mode Normal

Color Blend

Option	Description
Blend Amount	The **Blend Amount** essentially changes the opacity of the **A** input value, where 100 is fully the **A** input, 50% is equal parts **A** and **B**, and 0% is fully the **B** input value.
Blend Mode	The **Blend Mode** options determine an additional way to calculate the way the **A** and **B** values combine into the final result, defined by various mathematical functions that are applied to the numerical values of each input. **Note:** For more information on **Blend Modes**, see Layer Blend Modes


Constant
Texture Locator
Value
Color
Constant (2)
Texture Locator
Value
Color
Color Blend
colorA
colorB
Mask
color
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount

Schematic Node: Color Blend

Option	Description
Color A/B Input	These inputs read RBG values from the output of another node. Input values can be simple RGB values or complex combinations of multiple nodes. The final representation for the blending between the two inputs is determined by the **Blend Amount**, the **Blend Mode**, and the **Mask** Input.
Mask Input	The **Mask Input** controls the transparency of the **A** input value, revealing the **B** values in the masked areas. White shows the **A** input fully, attenuating toward black which shows the **B** input value.
Color Output	The **Color** output field represents the final results of the evaluated blend.

Color Correct

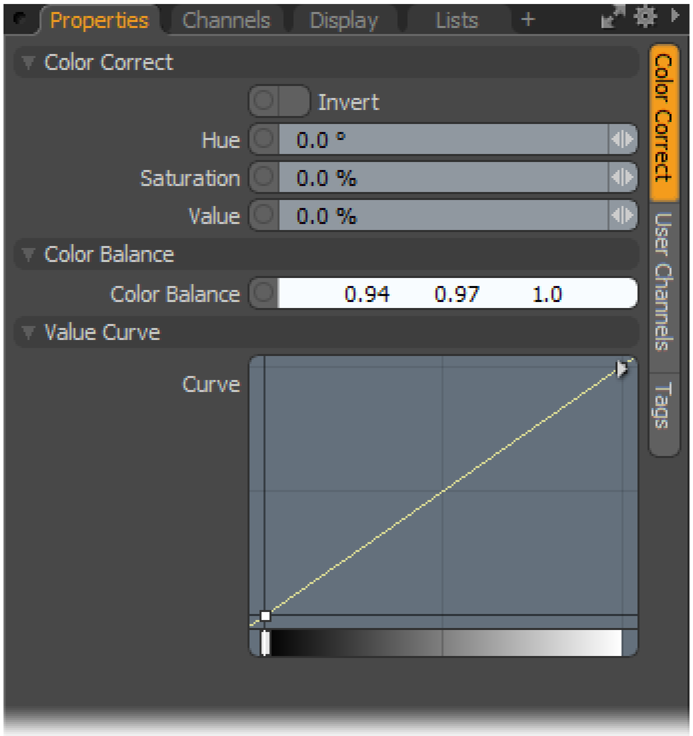
The Color Correct node provides an alternate method to modify the values of a color input. It can be used to manipulate the Hue (H), Saturation (S), or Value (V) components of a color individually, invert the colors and use a curve graph to non-linearly manipulate the value (brightness) component of the input. A mask input also allows the application amount to be controlled with a texture value.

Usage

The Color Correct node is added to the Schematic Viewport using the **Add** function, found in the pop-up menu under **Channel Modifiers** > **Color** > **Color Correct**. Once added, an RGB color output from another node can be connected to the **Color** Input of the Color Correct node. The result of the corrections are then sent through the **Color** output for connecting to another RGB or value input.

Note: For information on working with node graphs, see Schematic Viewport

When the node is selected, attributes appear in the **Properties** panel.



Color Correct

Invert

Hue 0.0 °

Saturation 0.0 %

Value 0.0 %

Color Balance

Color Balance 0.94 0.97 1.0

Value Curve

Curve

Color Correct

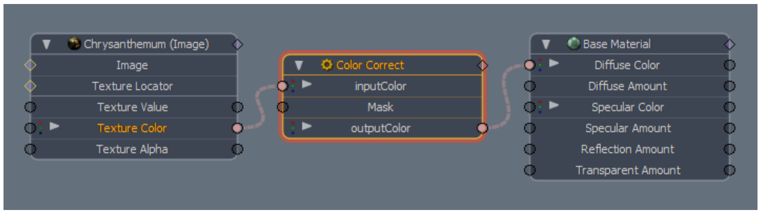
Option	Description
Invert	When the **Invert** option is enabled, the incoming color values are reversed, providing a negative image type of effect.
Hue	The **Hue** setting allows you to offset the hue (color) component of the incoming RGB color values. Uses an angle input amount offsetting 180° in either direction.
Saturation	The **Saturation** setting allow you to control the strength component of the colorization, either to increase vividness in a muted input with positive values up to 100%, or to dull overly strong colors with negative values down to -100%.
Value	The **Value** setting allows you to control the brightness component of the colorization, either to lighten the image with positive values up to 100%, or to darken it with negative values down to -100%.

Color Balance

Option	Description
Color Balance	The **Color Balance** setting provides you a way to tint the image as a way to remove slight color casts, used to white balance an image. For example when an image has a slight blue coloring, an orange tint can be added (orange being the opposite color from blue) to counteract the blue colorization.

Value Curve

Option	Description
Curve	Using an in-line graph editor, you can modify the value component of the color input in a non-linear fashion. The graph itself represent the brightness range of the color input, from dark values (black) on the left attenuating toward the light values (white) on the right. Middle-clicking on the graph line inserts a key into the graph (or use **Ctrl**+**Alt**+click). You can then click and drag to move the line adjust the brightness value at that location, up to increase brightness, and down to decrease. You can add as many keys as necessary to reach the desired results. Graph keys are removed by clicking to select and then pressing the **Del** key on the keyboard.


Chrysanthemum (Image)
Image
Texture Locator
Texture Value
Texture Color
Texture Alpha
Color Correct
inputColor
Mask
outputColor
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount

Schematic Node: Color Correct

Option	Description
Color Input	These inputs read RBG values from the output of another node. Input values can be simple RGB values or a texture node, or complex combinations of multiple nodes. The input is modified using the various controls found in the **Properties** panel.
Mask	The **Mask** input controls the strength of the Color Correction across the image using an input value. White shows the corrected input fully, attenuating toward black which shows no correction.
Color Output	The **Color** output field represents the final results of the evaluated color correction.

Color Gamma

The Color Gamma node changes the gamma of an input. Like the other color channel nodes, this operation can also be masked with a texture or value.

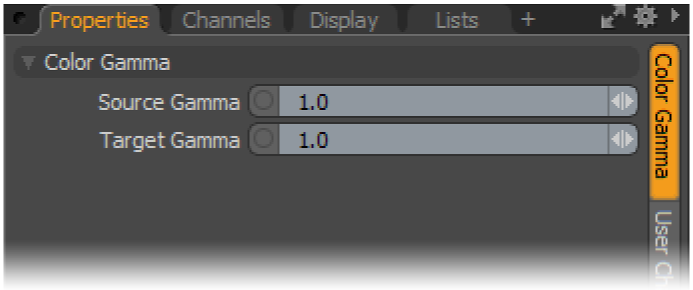
Gamma is the measure of total contrast in an image and represents the relationship of an image input to an image output. A gamma value of 1 is said to be linear, the value input is identical to the value output. Modifying the **Gamma** value changes how the image is represented. Values higher than 1 lighten the image, while values lower than 1 darken it. Adjustments to the gamma value are not linear, which means they don't change all the pixels in an image the same amount, explaining why it is often referred to as a gamma curve.

Usage

The Color Gamma node is added to the Schematic Viewport using the **Add** function, found in the pop-up menu under **Channel Modifiers** > **Color** > **Color Gamma**. Once added, an RGB color output from another node can be connected to the **Color** Input of the Color Gamma node. The result of the adjustments are then sent through the **Color** output for connecting to another RGB or value input.

Note: For information on working with node graphs, see Schematic Viewport

When the node is selected, the following attributes also appear in the **Properties** panel.



Color Gamma

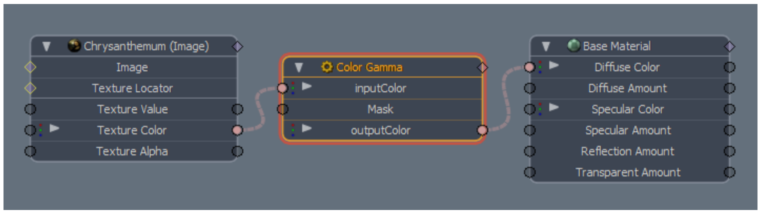
Source Gamma 1.0

Target Gamma 1.0

Color Gamma

Option	Description
Source Gamma	The **Source Gamma** value represents the gamma as a correction to the incoming RGB inputs (if necessary, from an uncorrected source file).

Option	Description
Target Gamma	The **Target Gamma** represents the gamma correction applied to the Source input. The adjusted results are then sent to the **Color** output.


Chrysanthemum (Image)
Image
Texture Locator
Texture Value
Texture Color
Texture Alpha
Color Gamma
inputColor
Mask
outputColor
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount

Schematic Node: Color Gamma

Option	Description
Color Input	These inputs read RBG values from the output of another node. Input values can be simple RGB values or a texture node, or complex combinations of multiple nodes. The input is modified using the gamma controls found in the properties panel.
Mask	The **Mask** input controls the strength of the Gamma application across the image using an input value. White shows the adjusted input fully, attenuating toward black which shows no adjustment.
Color Output	The **Color** output field represents the final results of the evaluated gamma application.

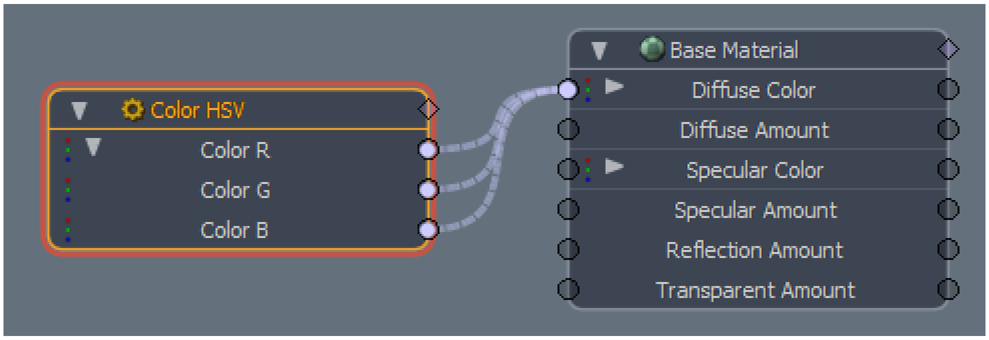
Color HSV

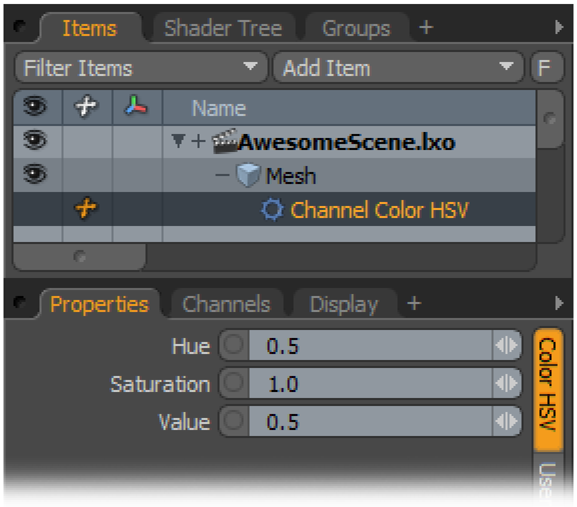
The Channel Color HSV node outputs RGB values based on the intuitive **Hue**, **Saturation** and **Value** color input controls. This allows you to smoothly animate colors by hue, saturation, or value, with correct interpolation, rather than making excessive keyframes for red, green, and blue channels individually.

Usage

The Color HSV node is added to the Schematic Viewport using the **Add** function, found in the pop-up menu under **Channel Modifiers** > **Color** > **Color HSV**. Once added, you can select the node and define a color in the **Properties** panel, based on the intuitive **Hue**, **Saturation** and **Value** color model. Values can either be static or animated. The corresponding color is converted to RGB values internally and output of the three R, G, and B channels of the node.

Note: For information on working with node graphs, see Schematic Viewport.


Color HSV
Color R
Color G
Color B
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Channel Color HSV
Properties
Channels
Display
Color HSV
Use

Hue 0.5
Saturation 1.0
Value 0.5

Channel Color HSV

Option	Description
Hue/Saturation/Value	These options determine the color value using the HSV color model. Their RGB output values can be linked to a materials RGB value input channels to control the materials colors with the modifier.

Color Invert

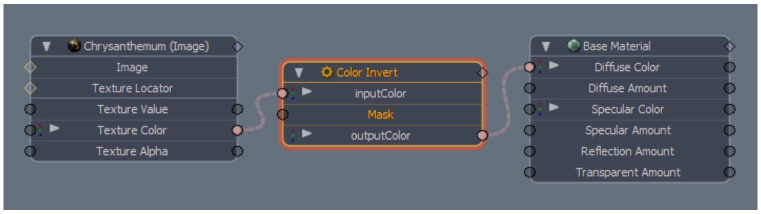
The Color Invert node reverses the incoming color values, providing a negative image type of effect.

Usage

The Color Invert node is added to the Schematic Viewport using the **Add** function, found in the pop-up menu under **Channel Modifiers** > **Color** > **Color Invert**. Once added, an RGB color output from another node can be connected to the **Color** Input of the Color Invert node. The result of the adjustments are then sent through the **Color** output for connecting to another RGB or value input.

Note: For information on working with node graphs, see Schematic Viewport

The color invert node has no attributes that appear in the **Properties** panel.


Chrysanthemum (Image)
Image
Texture Locator
Texture Value
Texture Color
Texture Alpha
Color Invert
inputColor
Mask
outputColor
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount

Schematic Node: Color Invert

Option	Description
Color Input	These inputs read RBG values from the output of another node. Input values can be simple RGB values or a texture node, or complex combinations of multiple nodes.
Mask	The **Mask** input controls the strength of the invert application across the image using an input value. White shows the adjusted input fully, attenuating toward black which shows no adjustment.
Color Output	The **Color** output field represents the final results of the invert values.

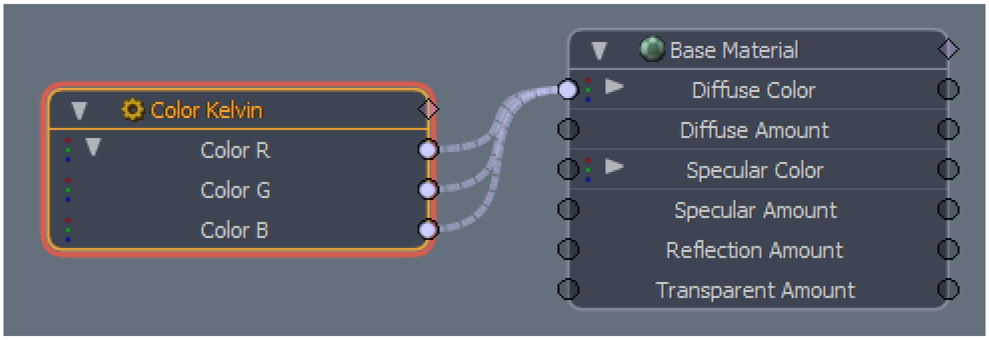
Color Kelvin

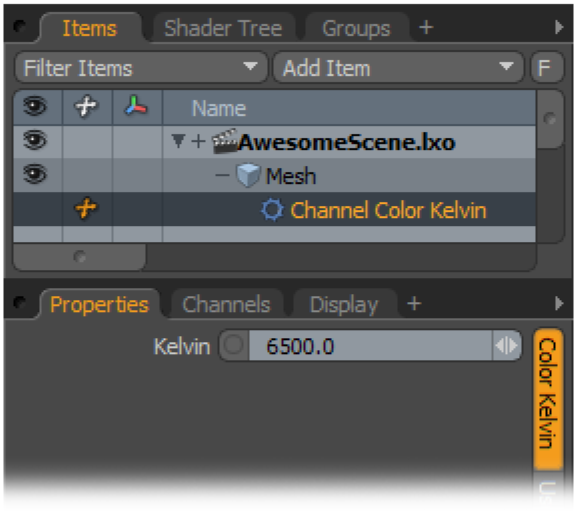
The Channel Color Kelvin node outputs RGB values based on the Kelvin thermal dynamics color-temperature color model. This allows you to smoothly animate colors across the kelvin range, with correct interpolation, rather than making excessive keyframes for red, green, and blue channels individually. This would be useful to animate, for instance, a distant light for a sunrise type of effect when the color would change across the kelvin spectrum as the light attenuates through the atmosphere.

Usage

The Color Kelvin node is added to the Schematic Viewport using the **Add** function, found in the pop-up menu under **Channel Modifiers** > **Color** > **Color Kelvin**. Once added, you can select the node and define a color temperature value in the **Properties** panel, based on the Kelvin color-temperature color model. Values can either be static or animated. The corresponding color is converted to RGB values internally and output of the three R, G, and B channels of the node.

Note: For information on working with node graphs, see Schematic Viewport


Color Kelvin
Color R
Color G
Color B
Base Material
Diffuse Color
Diffuse Amount
Specular Color
Specular Amount
Reflection Amount
Transparent Amount


Items
Shader Tree
Groups
Filter Items
Add Item
F
Name
AwesomeScene.lxo
Mesh
Channel Color Kelvin
Properties
Channels
Display
Color Kelvin

Kelvin 6500.0

Channel Color Kelvin

Option	Description
Kelvin	These options determine the color value using the Kelvin color model. Their RGB output values can be linked to a materials RGB value input channels to control the materials colors with the modifier.

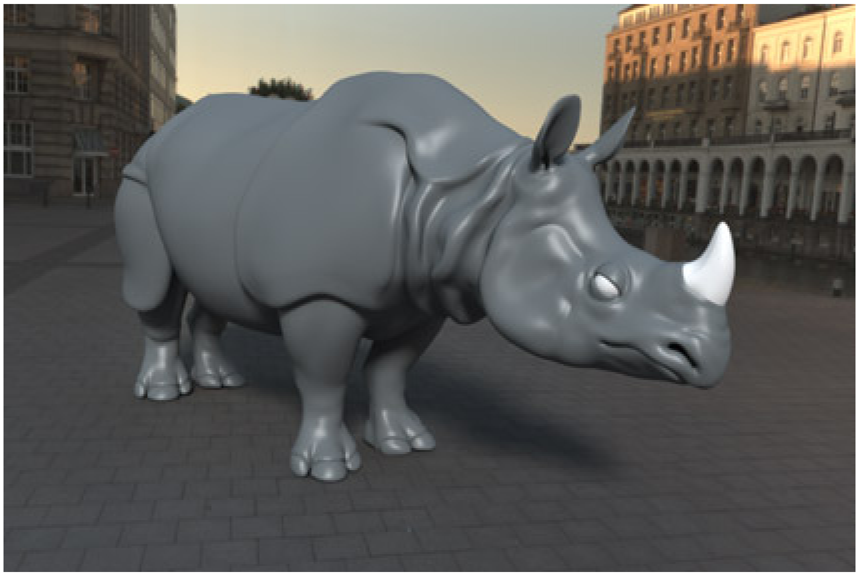
Environments

Environments in Modo offer a quick way to fill your scene's background with colors, gradients and image maps. You can also add lighting to your scene by adding an HDR image and global illumination. The following Environment Itemtopic provides details on how you can add and work with an Environment item. The Environment Material topic describes how you can use an Environment material to add color to your background. Finally, Image Based Lighting explains how you can light your scene using Environments.

Environment Item

The **Environment Item** allows you to set options related to the environment's visibility and intensity in a Modo scene. When selected, its associated attributes can be edited within the Properties viewport. Environments offer a quick, flexible way to fill a Modo background with colors, gradients, and image maps. With the addition of an HDR image and global illumination, environments can also light a scene providing an easy path to create stunningly realistic results. For more information on setting up Image Based Lighting (IBL), see Image Based Lighting.

A rendered rhino object. The background area is solid black as no geometry is visible, nor has an environment been inserted.



Environments can give the impression of a much larger scene without adding additional geometry.

Also, keep in mind that since Environments are Shader Tree items, they can have additional Shader Tree item layers added to them. For example, lightprobe images can be added producing full 360° environments, or camera projected images can provide plates allowing you to incorporate 3D elements into a filmed scene by simply adjusting the settings of the Texture Locator.

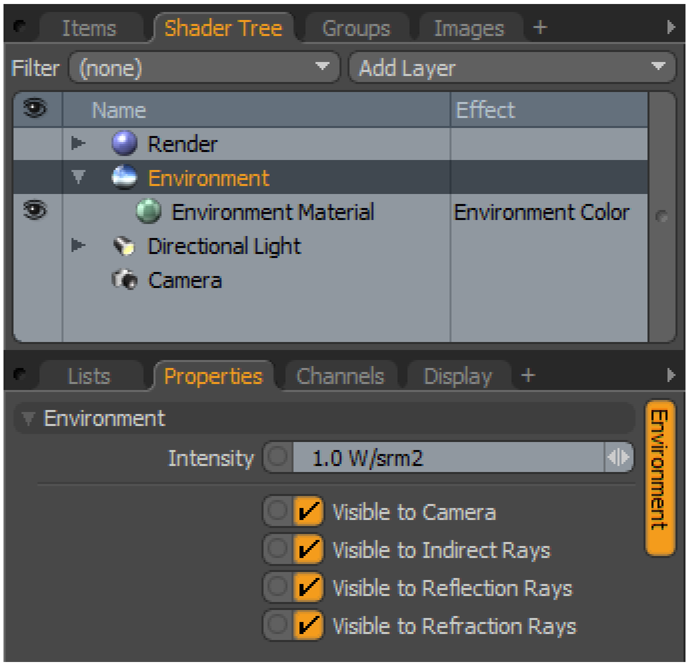
Procedural layers can be added and layered producing interesting effects. By right-clicking on the item's effect column, different options can be selected, producing a variety of different and interesting blending effects as materials are layered. For more information, see Effect - Environment Item.

Modo also allows for multiple Environments. Additional Environment Items can be added to the Shader Tree by selecting **Item** > **Create Environment** from the menu bar or by right-clicking the item itself and choosing **Duplicate**. This allows you to specify an environment for reflections, refractions, and the background itself, with a completely different version of the environment for lighting the scene.

Tip: You can drag and drop an Environment Preset from the Preset Browser into the **Environment** folder on the Shader Tree. A contextual menu appears, allowing you to either replace or add your selected Environment Preset. For more information, see Adding and Modifying Shader Tree Item Layers.

In addition, you can drag and drop an Environment Preset from the Preset Browser onto the n. For more information, see Changing Environments .

Environment


Items
Shader Tree
Groups
Images
Filter
(none)
Add Layer
Name
Effect
Render
Environment
Environment Material
Environment Color
Directional Light
Camera
Lists
Properties
Channels
Display
Environment
Environment

Intensity 1.0 W/srm2

☑ Visible to Camera

☑ Visible to Indirect Rays

☑ Visible to Reflection Rays

☑ Visible to Refraction Rays

Intensity - Acts as a multiplier, providing you with a convenient way to control and modify the overall brightness of the selected Environment. This setting has the greatest effect when global illumination is enabled, controlling to what degree the overall environment contributes to the lighting of the scene.

Visible to Camera - Toggles whether or not the selected Environment is visible to the camera when rendered. This setting is independent to the others, even when not visible to the camera, it can still contribute to the lighting of the scene providing you with the ability to specify alternate images or textures for the background in an additional environment item.

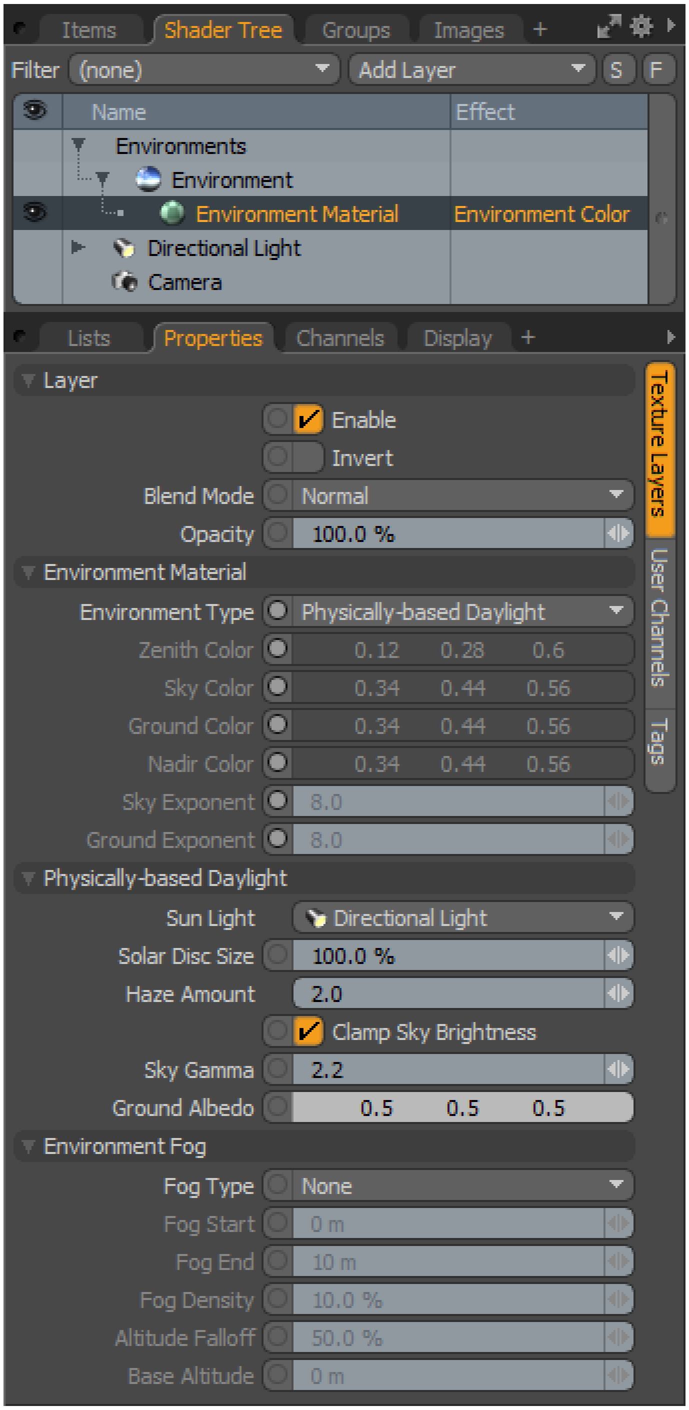
Visible to Indirect Rays - Toggles whether or not the environment contributes lighting to global illumination. This setting only affects scenes that have Global Illumination enabled.

Visible to Reflection Rays - Toggles whether or not environments have an effect on any reflective surfaces within a scene.
Visible to Refraction Rays - Toggles whether or not environments have an effect on any refractive surfaces within a scene.

Environment Material

The **Environment Material** provides a simple method for adding a basic color or color gradient to the background of a scene as well as applying some atmospheric fog. Each scene file by default includes a single environment item with an associated environment material. You can add additional material layers to an environment in the **Shader Tree** viewport, by clicking **Add Layer** > **Environment Material**. Any environment material, when combined with Modo's global illumination functionality, uses the colors of the environment to illuminate the scene, producing an incredibly realistic lighting simulation. For information on enabling this function, see Render Item: Global Illumination.

When you click on the environment material, the following options are visible in the item's **Properties** panel.


Items
Shader Tree
Groups
Images
+
Filter (none)
Add Layer
S
F
Name
Effect
Environments
Environment
Environment Material
Environment Color
Directional Light
Camera
Lists
Properties
Channels
Display
+
Texture Layers
User Channels
Tags

Layer
☑ Enable
☐ Invert
Blend Mode: Normal
Opacity: 100.0 %

Environment Material
Environment Type: Physically-based Daylight
Zenith Color: 0.12 0.28 0.6
Sky Color: 0.34 0.44 0.56
Ground Color: 0.34 0.44 0.56
Nadir Color: 0.34 0.44 0.56
Sky Exponent: 8.0
Ground Exponent: 8.0

Physically-based Daylight
Sun Light: Directional Light
Solar Disc Size: 100.0 %
Haze Amount: 2.0
☑ Clamp Sky Brightness
Sky Gamma: 2.2
Ground Albedo: 0.5 0.5 0.5

Environment Fog
Fog Type: None
Fog Start: 0 m
Fog End: 10 m
Fog Density: 10.0 %
Altitude Falloff: 50.0 %
Base Altitude: 0 m

Layer

Enable: Allows you to temporarily enable/disable a layer without losing stored values. Disabled layers are saved with the scene file. This is the same as toggling the eye icon "👁" in the visibility column of the **Shader Tree** viewport.

Invert: This setting inverts any RGB color values specified in the environment material.

Blend Mode: Affects blending between material layers allowing you to stack several layers for different effects. For more on blending, see the Layer Blend Modes topic.

Opacity: At 100% a layer is completely opaque, values below 100% change the transparency of the current material layer allowing other lower layers to become visible, ramping toward completely transparent at 0% essentially disabling the layer. If only one environment material is present, modifying the opacity fades the brightness of the background and affects how much it contributes to global illumination.

Environment Material

Environment Type: Specifies colors and gradients that appear in the background of rendered images. Once selected, the appropriate options for **Zenith**, **Sky**, **Ground** and **Nadir** colors are available as well as the **Sky** and **Ground** exponent values that adjust the bias of the gradients between the **Zenith**/**Sky** and **Ground**/**Nadir** respectively.

- **4 Color Gradient**: This easy to use gradient allows you to set 4 color values which are used to simulate the sky and ground. The **Zenith** color is the color that is directly overhead and this ramps into the **Sky** color which starts at the horizon and ramps upward. The **Nadir** color is directly below the camera and ramps upwards into the **Ground** color which terminates at the horizon when it meets the **Sky** color. There is a soft blend between the **Zenith** and **Sky** colors and a soft blend between **Ground** and **Nadir**. The boundary between **Ground** and **Sky** is hard edged to give the illusion of a distant horizon. When using a **4 Color Gradient**, the **Sky** and **Ground Exponents** are used to compress or expand the gradation between **Zenith** and **Sky** and the **Nadir** and **Ground**. Higher values push the gradient transition closer to the horizon, while lower values spread the gradient further across the available spectrum.
- **2 Color Gradient**: This option reduces the colors to **Zenith** and **Nadir** and creates a soft ramp between the two.
- **Constant**: Uses only the **Zenith** color for the entire background.
- **CIE Overcast Sky**: This option provides two user defined color values; **Zenith** and **Nadir**. The **Nadir** color is used in without any ramp effect for everything below the horizon. The **Zenith** color starts directly above the camera and then has a slight ramp to a somewhat darker color. By definition the CIE overcast sky is three times brighter at the **Zenith** than it is at the horizon, so the **Zenith** color is reduced in brightness by that amount by the time it reaches the horizon. When this setting is applied with global illumination, you can re-create the traditional overcast render look popular in images that want to show off an untextured model's surface detailing.
- **Physically-based Daylight**: This option works in tandem with global illumination to create renders lit by an incredibly realistic daylight simulation. Especially useful for architectural exterior and interior renders or in any scene that needs nice outdoor lighting. When this option is selected, the following **Physically Based Daylight** option can be used to further control the simulation.

Physically-based Daylight

When **Physically-based Daylight** is selected as the **Environment Type**, a few additional options become available, allowing you to fine-tune the look of the lighting simulation. Global illumination needs to be enabled in order for the **Physically-based Daylight** function to work.

Sun Light: This setting allows you to set a distant light (other light types don't show up here) that are considered the sun in the rendered images. Additional properties specific to the light itself are available in the Directional Light's **Physical Sun** settings.

Solar Disc Size: A small disk is added to the rendered image, representing the actual sun, above the horizon. It also appears in reflections, refractions and so on. The default value of 100% is correct to the scale of what you see in the real world, however, you can change the size depending on your particular scene needs using this setting.

Haze Amount: In the real world, as the sun nears the horizon, light has to pass through a greater amount of atmosphere to reach our eyes, additionally, moisture, dust and pollutants in the air further affect the light. The further the light has to pass through these particles, the more shorter wavelengths of light are absorbed, tending the light toward warmer hues. Increasing the **Haze** setting in Modo pushes the horizon colors toward oranges and yellows for midday lighting scenarios and deeper reds (sunsets) in evening times. Time of day can be controlled in the distant light's **Physical Sun** settings.

Clamp Sky Brightness: The **Physically-based Daylight** function is a highly accurate lighting simulation generating light intensities outside of normal render display settings. The **Clamp Sky Brightness** toggle limits the overall brightness of the sky, producing pleasing results more easily. When disabled, you might think your scene is completely 'blown out' and overexposed, but actually, since Modo renders in full floating point accuracy, you can bring back all that detail using a combination of white level, gamma and tone mapping. For more information, see Tone Mapping.

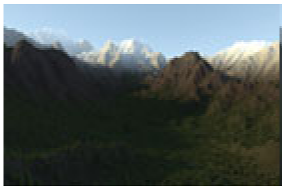
Sky Gamma: Adjusting gamma is the ability to adjust the overall contrast of a target in a non-linear fashion. The **Sky Gamma** option allows you to adjust the values of the procedurally generated **Physically-based Sky** lightening or darkening it for artistic refinement.

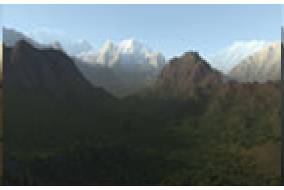
Ground Albedo: This option represents the color of any light reflected from the ground upwards in the overall environmental light simulation. You can specify an appropriate scene-specific color using the RGB color navigator. The Albedo is color only, having no brightness in the calculation, so RGB values between 0-1 (1-255) are the most appropriate.

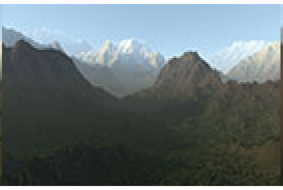
Disc In-Scatter: Mixes in light that has been deflected or scattered out of the direct path from the sun, then gets scattered back in. This results in a reddish cast around the edge of the solar disc (known as limb-darkening), and producing a reddening toward the bottom at low elevations (near sunrise or sunset). This is increasingly more noticeable at higher haze levels. A setting of 100% shows the physically correct result attenuating toward 0%, which disables the effect, making the solar disc appear as it did in previous versions.

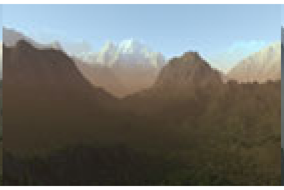
Environment Fog

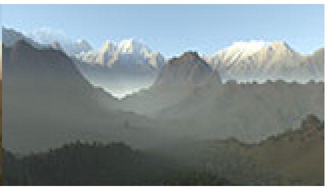
The **Environment Fog** option produces a simulated (non-volumetric) fog effect by applying the environment color to relevant materials with increased intensity as the geometry is further from the camera. This is also a great option to fade objects into a background, increasing the apparent depth (and scale) of a scene. Fog can also be specified in the Shader item with similar options. The **Environment Fog** option differs in that it always uses the the environment color, whereas for the **Shader Fog**, any color can be specified.


No Fog


Linear Fog


Exponential Fog


Underwater Fog


Layered Fog

Fog Type: Allows you to choose from the different fog types.

- **None**: Disables **Environment Fog** from rendering.
- **Linear**: When selected, fog renders in the scene based on the **Start** and **End** distances, where there is no fog up to the starting distance, then surfaces attenuate toward a solid environment color and maximum fog density at the end distance.
- **Exponential**: When selected, fog renders starting at the camera's position, increasing in strength the further a surface is from the camera. Based on exponential values, the strength increases non-linearly and is directly influenced by the **Fog Density** setting (so for very large scenes, you should use a very small value).
- **Underwater**: This option is basically the same as the **Exponential** function, except that the fog has a tendency to absorb the environment color more readily, producing an environment colored cast. It also reduces color saturation with distance. The example above is a late afternoon environment, so the reds of the setting sun influence the fog color. For a true underwater look, set the environment to a 2 or 4 color gradient with a blueish or teal color, or apply Shader Fog and define the Fog color independent of the environment.
- **Layered**: Produces a ground fog-like effect where the fog density increases with both depth and height.

Fog Start/End Distance: When the **Fog type** is set to **Linear**, you can specify specific starting and ending distances for the fog effect.

Fog Density: This percentage value determines how thick the fog is just in front of the camera. The default value of 10% indicates that geometry just in front of the camera has a 10% blend of the fog color. The density of the fog increases as it recedes further from the camera.

Altitude Falloff: This option determines the maximum height of the fog when using the **Layered Fog** option. Fog attenuates from the **Base Altitude**, fading from maximum density at the base to fully transparent at the defined **Altitude Falloff** value.

Base Altitude: Defines the bottom of the layered fog. Anything below has fog applied up to the maximum density (based on distance from camera), and above this value fully attenuates up to the **Altitude Falloff**.

Create Unclamped Physical Sky

The command **Create Unclamped Physical Sky** creates a **Physical Sky** and disables any related clamping. It also specifies the distant light as the sun, or creates one if it doesn’t exist, disabling the clamping on the light item as well. Lastly, it automatically adjusts the exposure values to make up for the unclamped environment settings. The command can be found in the menu bar under **Render** > **Create Unclamped Physical Sky**.

Image Based Lighting

Any environment in Modo can be used to light a scene with the addition of Global Illumination, whether it is an image map, a gradient, or even a solid color or procedural texture layer. Individually, layers don't have specific brightness or luminosity settings, but their contribution to a scene can be adjusted by modifying the **Opacity** value of the layer or globally with the **Environment Item** > **Intensity** setting producing brighter, more evenly illuminated scenes or darker scenes, depending on how the settings are adjusted.

When global illumination is combined with High Dynamic Range (HDR) images, incredibly realistic simulations of real world light interaction can be produced, a technique known as image based lighting (IBL). HDR imagery is generated in such a way as to not only capture the colors of a scene but faithfully capture the brightness values as well, creating a dynamic range that far exceeds what most monitors are capable of displaying.

A full 360° panoramic HDR image captures any visible lights, exterior lighting (including the sun), the sky's contribution, and most importantly, bounced light. Modo in turn is able to utilize this extended information and illuminate the scene. Because of HDRs ability to capture such subtleties in a scene, scenes utilizing image based lighting are often indistinguishable from photographs. As a result, they have become extremely popular amongst visual effects artists and art directors wishing to absolutely control every aspect of an advertisers image, and the CAD visualization crowd seeking to produce ever more realistic simulations for planning and marketing purposes.

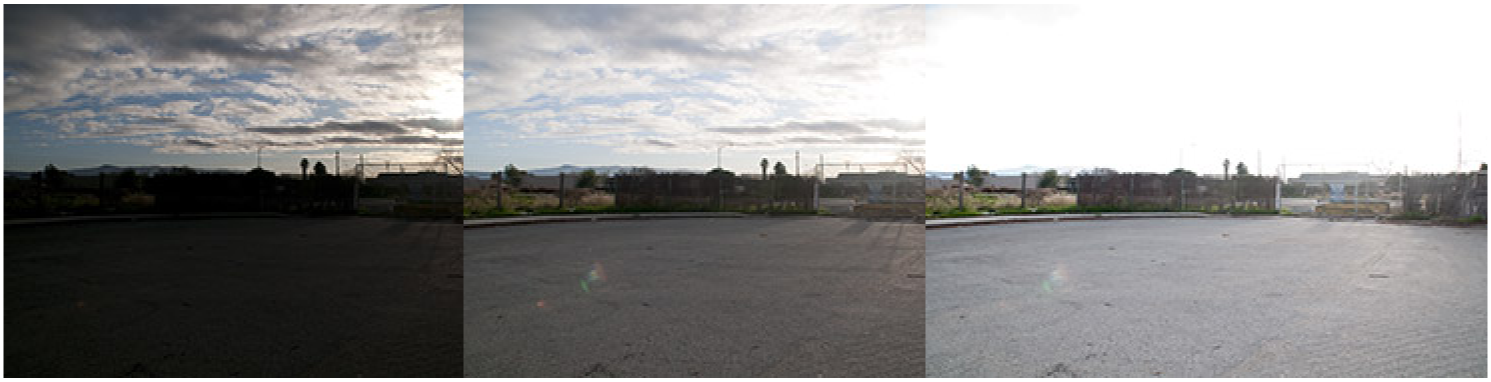
The following images show a series of IBL renders utilizing the same settings, only the environment image has changed.

Robot model images provided by John Hayes.

Where Do HDRIs Come From?

HDR images used for IBL are often comprised of a series of photographs taken of the same scene, with each having a slightly different exposure (length of time the shutter is open). This sequence of images, ranging from extremely dark to extremely bright, are then combined in such a way that for every pixel, a color value and a brightness value can be calculated.

Tip: Photoshop has an automation command **Merge to HDR** that is capable of doing this; several free tools are available that are capable of doing this as well. See HDR Labs for more info.



Sky exposure (-2 EV) Default Exposure (0 EV) Foreground Exposure (+2 EV)

The resulting HDR image can then be saved in a format that retains this information, such a **.exr** or **.hdr**. When loaded into Modo, the brightness information is then extracted and used to illuminate the scene producing an incredibly realistic simulation of the lighting from the original scene.

Note: Formats that support only 16-bit and lower can be used, but they don't retain the brightness values necessary to properly illuminate the scene.

Setting Modo for IBL

A few settings need to be in order first for image based lighting to work in Modo. But don't worry, it's all very easy to set up.

Tip: Don't happen to have any HDR images available? A great alternative that produces equally impressive results is the **Physically Based Sky** feature. This function can be enabled in the Environment Material item; use the **Environment Type** dropdown to select it.

Lights

Modo offers a variety of different light types that simulate real world sources of illumination. You can add a light to the scene from menu bar, by selecting **Item** > **Create Light** and then selecting the type of light you wish to add. Lights can also be added from the Items (Scenes) List by clicking **Add Item** and selecting one of the available light types. Settings related to the lights' illumination are set within the Properties Panel when light items are selected, material-specific properties are available through the light material item, selectable within the Shader Tree. Select any of the lights types listed below to be directed to the appropriate page of the documentation.

Tip: If you have multiple lights in the scene that you want to adjust, you can disable **Light Illumination** in the 3D viewport properties on the **Visibility** tab. You can still see the light widgets in the viewport, but the illumination they provide is disabled so you can move the light positions quickly and easily.

Direct Light Types

Area Light

Cylinder Light

Directional Light

Dome Light

Photometric Light

Point Light

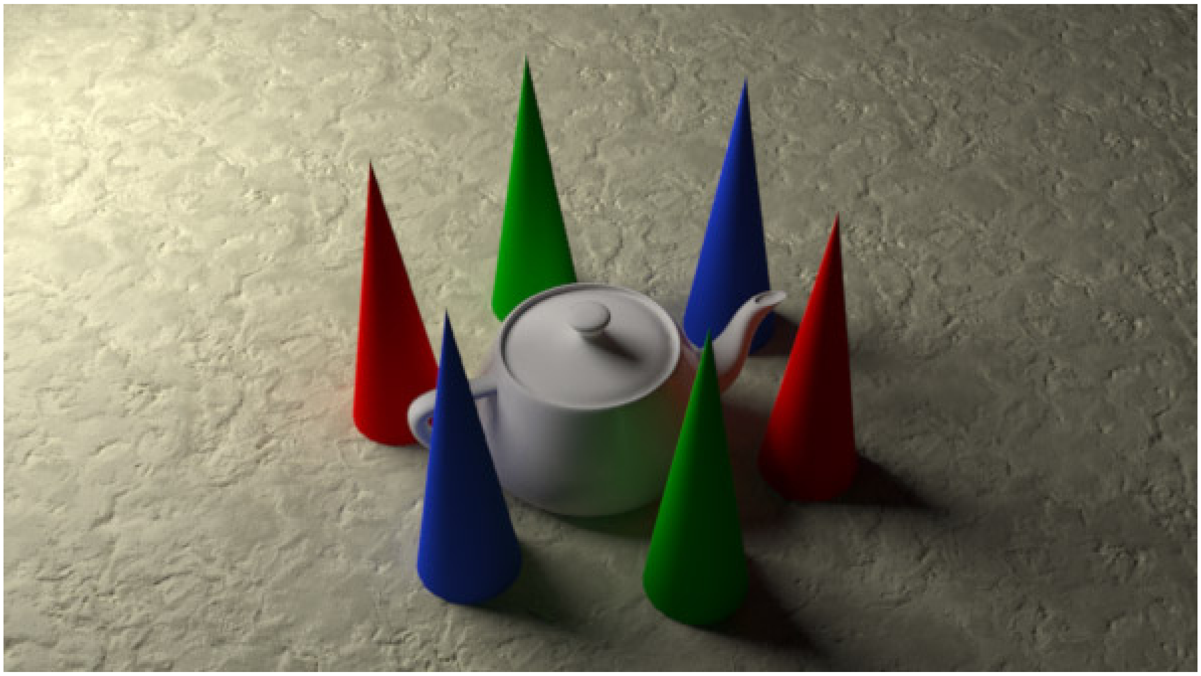
Portal

Spot Light

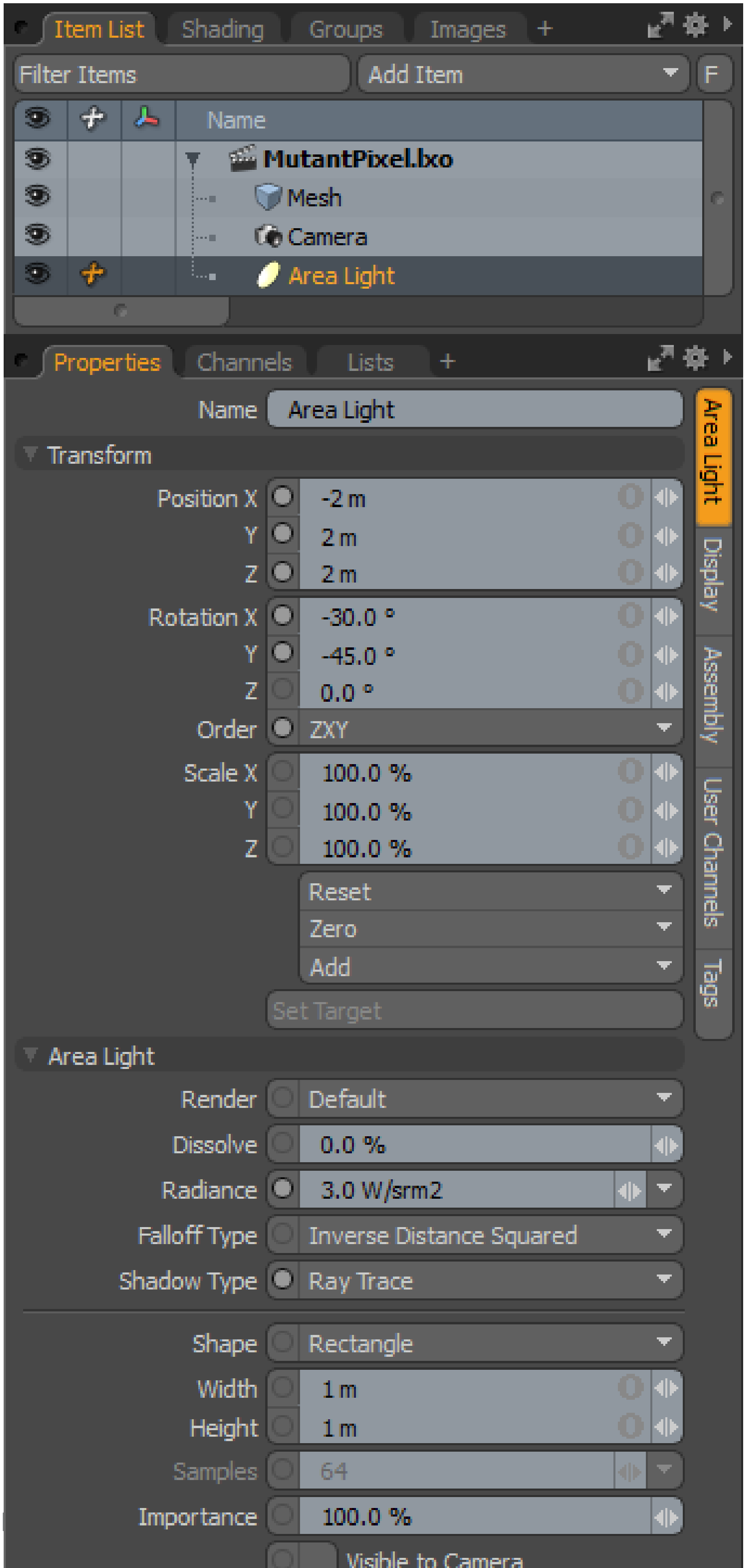
Mesh Light

Area Light

The **Area Light** is similar to a studio soft box style light, often used in professional photography because of the soft shadows they produce. Light comes from a square or round plane depending on the light settings. Area lights are excellent for creating photo-realistic lighting effects as they create a very natural light emission and smooth shading. They are typically more expensive, render time-wise, than other light types because they require more samples due to their size, but their natural tendency to create soft edged shadows is worth the trade-off in many situations.



Area Lights use a physically accurate light model so the intensity of the light falls off with the inverse square of the distance to the light. For example, a surface is half as bright when the light is twice as far away. As a physically based light, the intensity is also increased as the volume (size) of the light increases. Increasing the size of the light relative to the model it affects also increases the amount of shadow spread. When adjusting the **Area Light** in the 3D view be sure to use the **Item Transform** tool. This provides additional handles so you can adjust the length and width of the light directly in the 3D viewport. To control the light's color, shadow color, and other material-related properties, refer to the Light Material topic.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Camera
Area Light
Properties
Channels
Lists
+
Area Light
Display
Assembly
User Channels
Tags
Reset
Zero
Add
Set Target

Name: Area Light

Transform

Position X: -2 m
Y: 2 m
Z: 2 m

Rotation X: -30.0 °
Y: -45.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Area Light

Render: Default
Dissolve: 0.0 %
Radiance: 3.0 W/srm2
Falloff Type: Inverse Distance Squared
Shadow Type: Ray Trace

Shape: Rectangle
Width: 1 m
Height: 1 m
Samples: 64
Importance: 100.0 %
Visible to Camera

- **Name** - this data field displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available for the **Area Light**:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that the rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This scale transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. Transform items are the channel groups that store the actual transform values, controlling any item's position, rotation, and/or scale. You can add as many transform items as you want for any transform property desired. Adding additional **Transform** items produces an additive effect where each transform group is evaluated before the next, and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

Note: It should be noted that by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.

There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the master scene, they must first exist.

- **Set Target** - by selecting the light item, and a single additional item in the item list, then pressing **Set Target**, this function allows you to target specific items in a scene, automating the rotation of an item, so that it continuously points towards the targeted item. Once activated, additional options appear:
  - **Remove Target** - removes the target link between the two items.
  - **Enable** - toggling this option off temporarily disables the targeting function while retaining the link between the items.

  - **Set Focus Distance** - sets the focal distance for DOF when targeting camera items to other items.
  - **Roll** - offset the item's rotation angle away from the target.
  - **Time Offset** - set time offset, by a number of frames; to set how the light follows the target item. It can either be delayed, behind it with a negative value, or run ahead of it with a positive value.

The following **Area Light** options are available for the **Area Light**:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and, when invisible, it does not. On some instances you may prefer to fix this state, setting the light as **On** (enabled) or **Off** (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.
- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiance** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Area Light** and decreasing the value reduces the light intensity. With area lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.
- **Falloff Type** - light in the real world isn't a uniform brightness; its intensity diminishes with distance. Photographers may be familiar with the concept that a light is a quarter as bright at twice the distance away, known as the Inverse Square Law. Modo lights default to this setting, providing a realistic way to light a scene. However, there are times when you may not wish to have this behavior, so three falloff type options are provided:
  - **None** - no falloff; light is consistently bright across its distance.
  - **Inverse Distance** - light is half as bright at twice the distance.
  - **Inverse Distance Squared** - default behavior; light is a quarter as bright at twice the distance.
- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.
- **Shape** - Area lights can be specified as **Rectangular** or **Elliptical**. The shape of the light slightly impacts total distribution of light.
- **Width/Height** - these values set the size for the light as an absolute value for width and height and is based on the current units. By default, the unit system is Metric so the width and height would be set directly as meters. When the light item is selected in the 3D viewport, additional handles are available allowing you to interactively scale the area light item. Keep in mind that the size of the light affects the apparent intensity as mentioned above.

- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

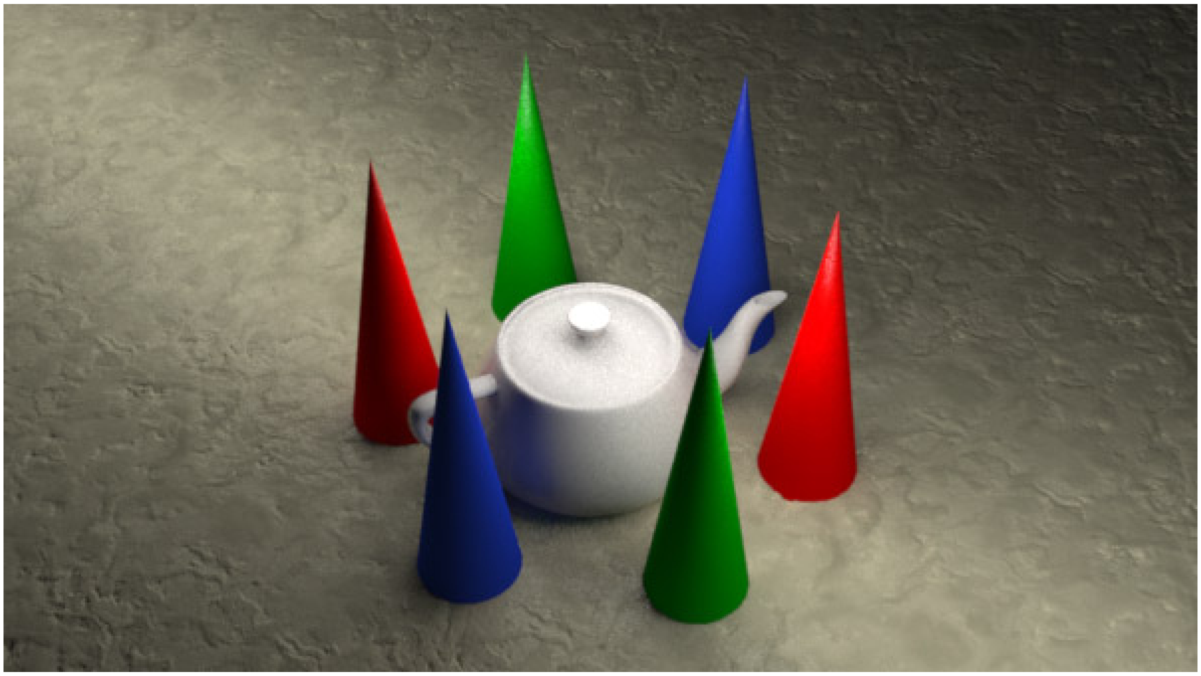
  As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.
- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.
- **Visible to Camera** - you can enable this option to make the light itself visible to cameras. The size of the visible light is determined by the **Spread Angle** attribute.
- **Visible to Reflection/Refraction Rays** - you can enable either or both of these options to make the light itself visible to reflective and/or transparent surfaces. The size of the visible light is determined by the **Spread Angle** attribute.
- **LPE Label** - you can provide a label name here, which appears as the label for any Light Path Expression output you generate for this light. See Rendering with mPath.

The following **Light linking** options are available for the **Area Light**:

- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the shader, it's possible to control a light's effect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the shader, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.
- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option and click **OK** to accept. Once the group is defined, select the named group here.
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

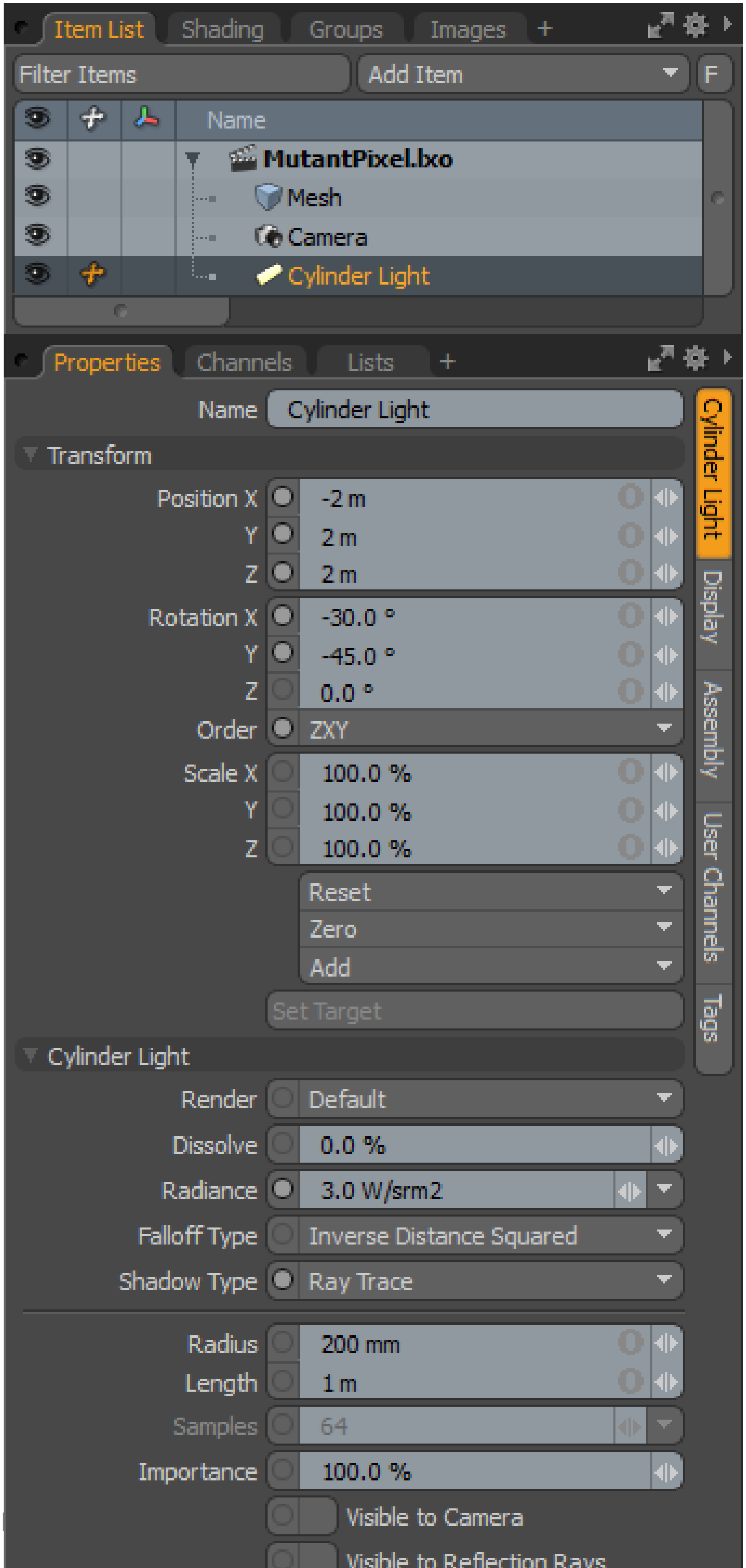
Cylinder Light

The **Cylinder Light** is similar to a common fluorescent tube, in that it emits light in a radial fashion outward from a cylindrical shape. Cylinder lights use a physically accurate light model so the intensity of the light falls off with the inverse square of the distance to the light. For example, a surface is half as bright when the light is twice as far away. As a physically based light, the intensity is also increased as the volume (size) of the light increases. Increasing the size of the light relative to the model it affects also increases the amount of shadow spread.



The light has **Position** and **Rotation** controls as well as **Radiance**, **Shadow Type**, **Samples**, **Cylinder Radius**, and **Length**. When adjusting the **Cylinder Light** in the 3D view, be sure to use the **Item Transform** tool. This provides additional handles so you can adjust the length and diameter of the **Cylinder Light** directly in the 3D viewport. To control the light's color, shadow color, and other material-related properties, refer to the Light Material topic.

Cylinder Light Properites


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Camera
Cylinder Light
Properties
Channels
Lists
+
Cylinder Light
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Set Target
Cylinder Light

Name Cylinder Light

Position X -2 m
Y 2 m
Z 2 m

Rotation X -30.0 °
Y -45.0 °
Z 0.0 °

Order ZXY

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Render Default

Dissolve 0.0 %

Radiance 3.0 W/srm2

Falloff Type Inverse Distance Squared

Shadow Type Ray Trace

Radius 200 mm

Length 1 m

Samples 64

Importance 100.0 %

Visible to Camera

Visible to Reflection Rays

- **Name** - this data field displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that the rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This scale transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. Transform items are the channel groups that store the actual transform values, controlling any item's position, rotation, and/or scale. You can add as many transform items as you want for any transform property desired. Adding additional **Transform** items produces an additive effect where each transform group is evaluated before the next, and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

Note: It should be noted that by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.

There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the master scene, they must first exist.

- **Set Target** - by selecting the light item, and a single additional item in the item list, then pressing **Set Target**, this function allows you to target specific items in a scene, automating the rotation of an item, so that it continuously points towards the targeted item. Once activated, additional options appear:
  - **Remove Target** - removes the target link between the two items.
  - **Enable** - toggling this option off temporarily disables the targeting function while retaining the link between the items.

  - **Set Focus Distance** - sets the focal distance for DOF when targeting camera items to other items.
  - **Roll** - offset the item's rotation angle away from the target.
  - **Time Offset** - set time offset, by a number of frames; to set how the light follows the target item. It can either be delayed, behind it with a negative value, or run ahead of it with a positive value.

The following **Cylinder Light** options are available:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and, when invisible, it does not. On some instances you may prefer to fix this state, setting the light as **On** (enabled) or **Off** (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.
- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiance** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Cylinder Light** and decreasing the value reduces the light intensity. With cylinder lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.
- **Falloff Type** - light in the real world isn't a uniform brightness; its intensity diminishes with distance. Photographers may be familiar with the concept that a light is a quarter as bright at twice the distance away, known as the Inverse Square Law. Modo lights default to this setting, providing a realistic way to light a scene. However, there are times when you may not wish to have this behavior, so three falloff type options are provided:
  - **None** - no falloff; light is consistently bright across its distance.
  - **Inverse Distance** - light is half as bright at twice the distance.
  - **Inverse Distance Squared** - default behavior; light is a quarter as bright at twice the distance.
- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.
- **Radius/Length** - these values set the size for the light as an absolute value for **Radius** and **Length** and is based on the current units. By default, the unit system is Metric so the radius and length would be set directly as meters. The size of the light affects the apparent intensity as mentioned above.
- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When

the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.

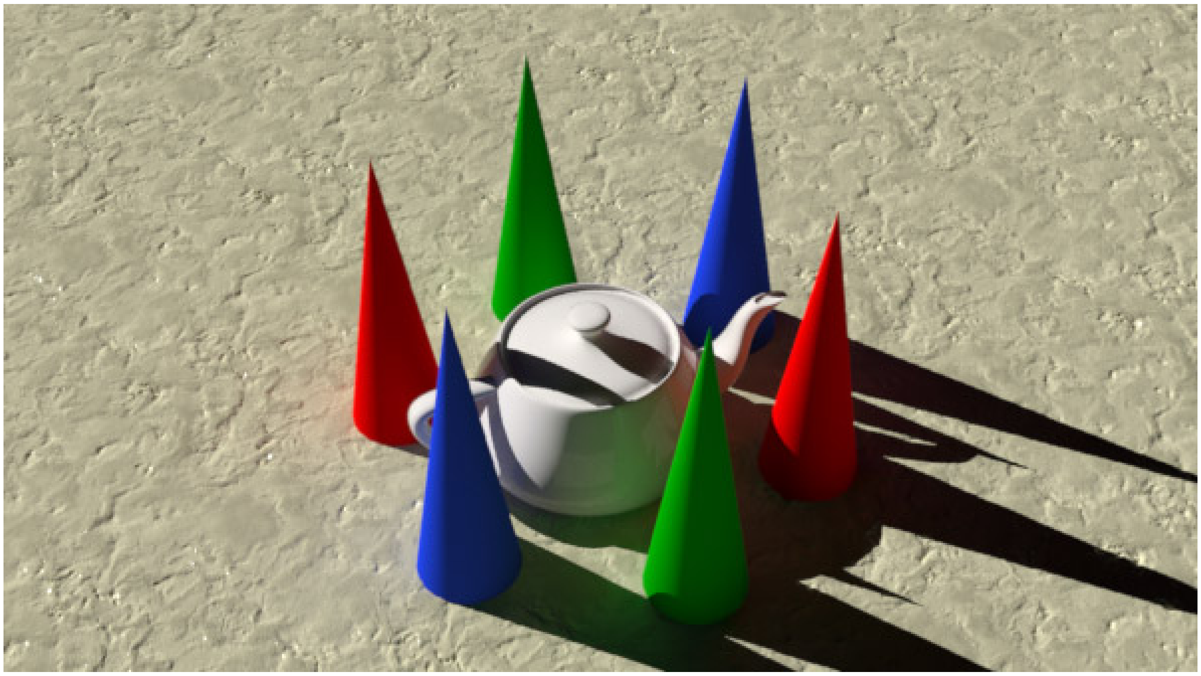
- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.
- **Visible to Camera** - you can enable this option to make the light itself visible to cameras. The size of the visible light is determined by the **Spread Angle** attribute.
- **Visible to Reflection/Refraction Rays** - you can enable either or both of these options to make the light itself visible to reflective and/or transparent surfaces. The size of the visible light is determined by the **Spread Angle** attribute.
- **LPE Label** - you can provide a label name here, which appears as the label for any Light Path Expression output you generate for this light. See Rendering with mPath.

The following **Light Linking** options are available:

- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the shader, it's possible to control a light's effect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the shader, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.
- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option and click **OK** to accept. Once the group is defined, select the named group here.
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

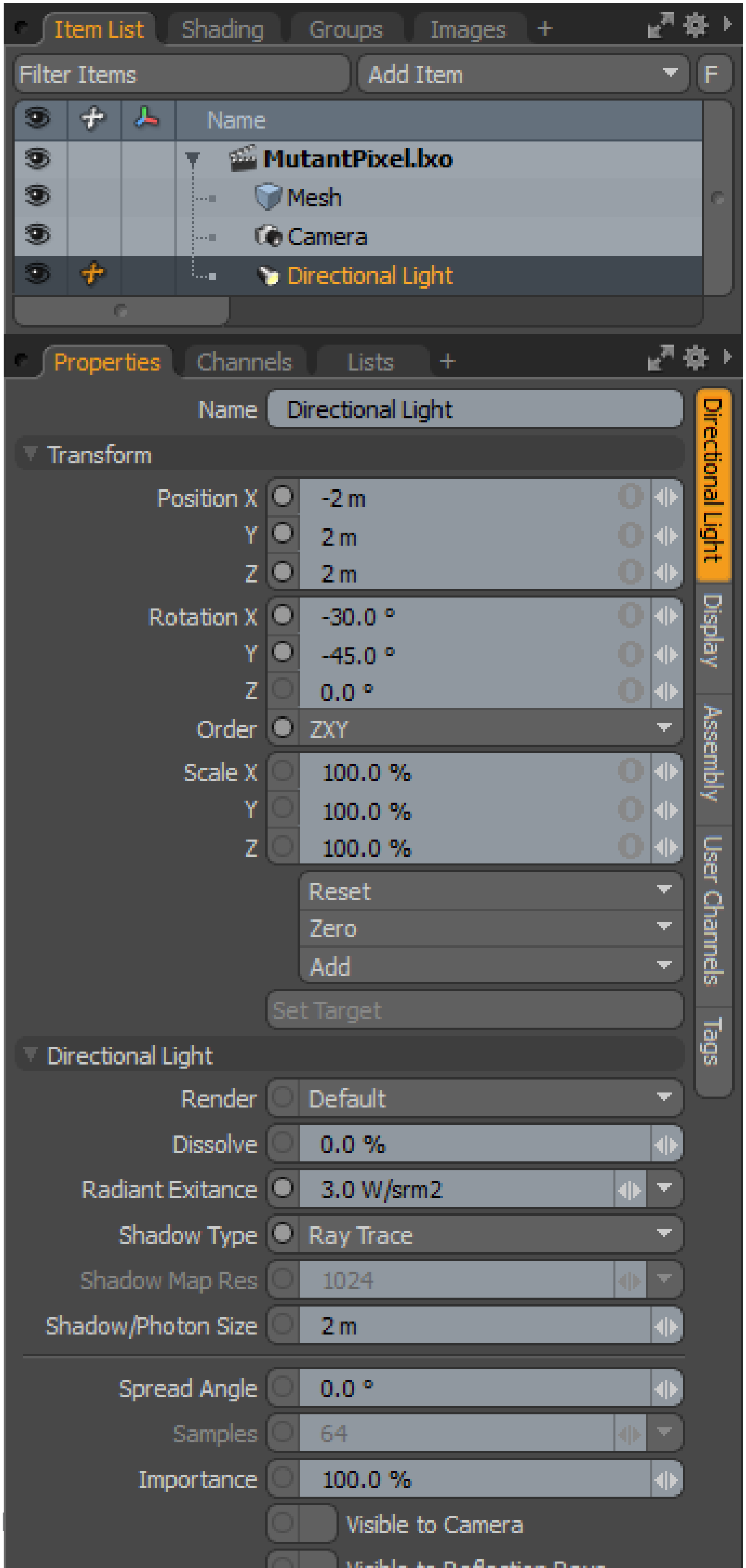
Directional Light

The **Directional Light** is most similar to the Sun. This light type emits a wall of parallel rays in the direction that the light is facing. The position of a **Directional Light** is mostly irrelevant as the rays travel infinitely along their path through the scene and, as such, only the **Rotation** is used to set the direction of the light's rays. Also, as a result of the infinite nature of the rays, there is no falloff for a **Directional Light**.



As with all items, the **Directional Light** controls are located on the **Properties** tab, when the light is selected. Directional lights have **Position** and **Rotation** options as well as controls for the intensity of the light, shadow type, and other settings covered fully below. To control the light's color, shadow color, and other material-related properties, refer to the Light Material topic.

Directional Light Properties


Item List
Shading
Groups
Images
Filter Items
Add Item
F
Name
Properties
Channels
Lists
Directional Light
Display
Assembly
User Channels
Tags
Reset
Zero
Add
Set Target

MutantPixel.lxo
- Mesh
- Camera
- Directional Light

Name: Directional Light

Transform

Position X: -2 m
Y: 2 m
Z: 2 m
Rotation X: -30.0 °
Y: -45.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Directional Light

Render: Default
Dissolve: 0.0 %
Radiant Exitance: 3.0 W/srm2
Shadow Type: Ray Trace
Shadow Map Res: 1024
Shadow/Photon Size: 2 m
Spread Angle: 0.0 °
Samples: 64
Importance: 100.0 %
Visible to Camera

- **Name** - this data field displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that the rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This scale transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. Transform items are the channel groups that store the actual transform values, controlling any item's position, rotation, and/or scale. You can add as many transform items as you want for any transform property desired. Adding additional **Transform** items produces an additive effect where each transform group is evaluated before the next, and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

Note: It should be noted that by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.

There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the master scene, they must first exist.

- **Set Target** - by selecting the light item, and a single additional item in the item list, then pressing **Set Target**, this function allows you to target specific items in a scene, automating the rotation of an item, so that it continuously points towards the targeted item. Once activated, additional options appear:
  - **Remove Target** - removes the target link between the two items.
  - **Enable** - toggling this option off temporarily disables the targeting function while retaining the link between the items.

  - **Set Focus Distance** - sets the focal distance for DOF when targeting camera items to other items.
  - **Roll** - offset the item's rotation angle away from the target.
  - **Time Offset** - set time offset, by a number of frames; to set how the light follows the target item. It can either be delayed, behind it with a negative value, or run ahead of it with a positive value.

The following **Directional Light** options are available:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and, when invisible, it does not. On some instances you may prefer to fix this state, setting the light as **On** (enabled) or **Off** (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.
- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiant Exitance** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Directional Light** and decreasing the value reduces the light intensity. With directional lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.
- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.
- **Shadow Map Res** - sets the resolution for the calculated deep shadow map in pixels. This option is only available when **Deep Shadow Map** is selected.
- **Shadow/Photon Size** - since directional lights are infinite, **Shadow Map Size** allows you to specify the area that the shadow map affects. Think of it like a rectangular panel aligned with the light item's position in the 3D viewport (an outline appears around the light signifying the map size). Shadow map "rays" fire forward from this panel and objects within this space have shadows. As the shadow map size increases, you may find you need to increase the resolution to reduce block-like artifacts. When generating caustics with **Directional Lights**, adjustments to the **Shadow/Photon** size also effect the photon emit area.
- **Spread Angle** - the default value of 0° results in crisp hard-edged shadows. If you wish to produce a more realistic soft-edged shadow you may need to increase this value. Since the **Directional Light** is most often used to simulate the Sun, a very small amount of **Spread Angle** is recommended. Values between 2 and 5 degrees give a subtle, soft edge to the shadow similar to sunlight. Keep in mind that increasing this value too high results in noisy shadows unless you also increase the number of Direct Illumination **Light Samples** in the **Render** item.

- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

  As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.
- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.
- **Visible to Camera** - you can enable this option to make the light itself visible to cameras. The size of the visible light is determined by the **Spread Angle** attribute.
- **Visible to Reflection/Refraction Rays** - you can enable either or both of these options to make the light itself visible to reflective and/or transparent surfaces. The size of the visible light is determined by the **Spread Angle** attribute.
- **LPE Label** - you can provide a label name here, which appears as the label for any Light Path Expression output you generate for this light. See Rendering with mPath.

The following **Light Linking** options are available:

- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the shader, it's possible to control a light's effect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the shader, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.
- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option and click **OK** to accept. Once the group is defined, select the named group here.
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

The following **Physical Sun** options are available:

The **Physical Sun** function works in tandem with the **Physically-based Daylight** setting under the **Environment Material** properties. It is in that dialog box you enable the Daylight lighting function and select which light acts as the sun. Here in the directional light properties, you can activate that light as the sun and use the time and day values to set the sun's position. You should note that the Physical Sun's **Position** is not animated or computed for times when the sun is not visible (meaning night-time).

- **Physical Sun** - toggles the directional light as the sun in the scene. You'll want to enable this option for Physically-based Daylight renders.
- **Time** - sets the time of day for your lighting, and effects the position and angle of the sun, as well as the colors in the atmosphere. For example, afternoons offer cooler lighting and shorter shadows and evening times tend toward warmer oranges and reds, and longer shadows.
- **Day** - the sun rises and sets at different times during the year, based on the date supplied. This setting positions the sun precisely in your sky, based on the location specified below.
- **Daylight Savings Time** - when enabled, the **Time** value is adjusted one hour, to make up for the time change during daylight savings time periods.
- **Haze** - in the real world, as the sun nears the horizon, light has to pass through a greater amount of atmosphere to reach our eyes. Additionally, moisture, dust, and pollutants in the air (collectively described as haze) further affect the light. The further the light has to pass through these particles, the shorter the wavelengths of light are scattered. The longer wavelengths pass through, tending the light toward warmer hues. Increasing the **Haze** setting in Modo pushes the horizon colors toward oranges and yellows for midday lighting scenarios, and deeper reds (sunsets) in evening times. This is the same setting as in the Physically-based Daylight properties.
- **Clamp Intensity** - the Physically-based daylight function is a highly accurate lighting simulation generating light intensities outside of normal render display settings. The **Clamp Intensity** control can be used to limit the overall brightness of the sun, producing pleasing results more easily. There are three options: when set to **none** (disabled), it might seem like the scene is completely blown out and overexposed but, actually, since Modo renders in full floating point accuracy, it's possible to bring back all that detail using a combination of white level, gamma, and tone mapping.

  When set to **Clamp**, the value is truncated at a level that produces pleasing, though not accurate, results without having to resort to any manual tone mapping adjustments. The **Replace** option simply replaces the lighting value with the current setting of the directional light's **Radiant Exitance** value (basically its brightness).
- **Gamma** - the **Gamma** option is an additional tweaking control that basically affects the brightness and coloration of the lighting/environment for when the given solution isn't quite to your expectation. This control allows for some additional adjustment.
- **North Offset** - you may have modeled your scene relative to the XYZ planes in Modo, and might not have taken into consideration whether it was facing north or south. You can use the **North Offset** to make it

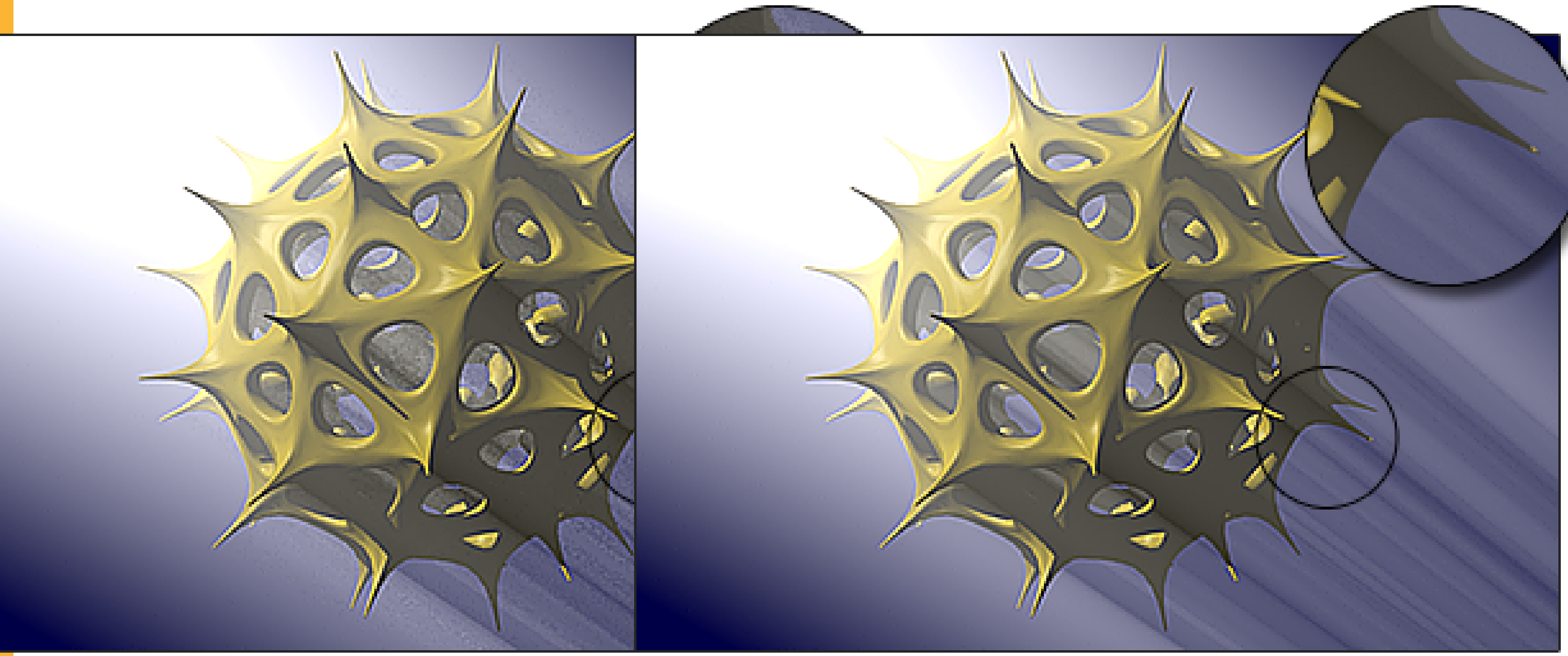
face the proper direction without having to resort to item rotations. Your model can stay in its comfortable XYZ position and the world revolves around it.

- **Location** - the sun appears in different positions in the sky from different locations on earth. The **Location** setting allows you to set the location for your scene so the Sun appears in the proper position for your simulated sky. A number of well-known locations are provided, but you can also specify the latitude and longitude positions for your custom location, as well as save it in the configuration file for future use.

The following **Volumetrics** options are available:

When light shines on a particular matter suspended in the air, such as water vapor, smoke, or even pollution, the light rays become visible. In computer graphics, this effect is known as "volumetric lights". Their affect can be subtle or pronounced and, either way, it can add a good amount of believability and atmosphere to renders. Additional settings related to volumetrics can be found in the Light Material item.

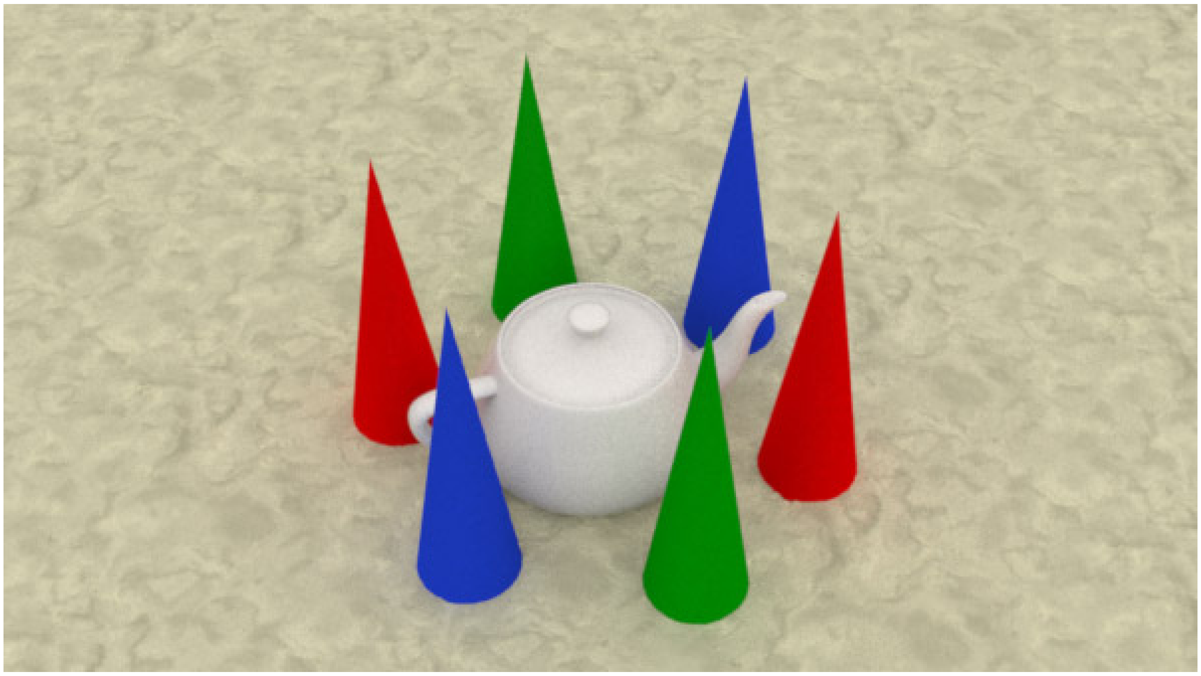
- **Volumetrics** - toggles the effect on or off. When enabled, the **Samples**, **Height**, and **Base** options become available.
- **Samples** - any objects that intersect the light beam volume cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light, sometimes referred to as "god rays". The **Sample** setting defines the degree of accuracy that Modo uses to calculate these shadows through the volume; higher numbers of samples give smoother and more accurate results, while lower values render more quickly but become increasingly grainy. The example images below demonstrate the difference between 32 samples, on the left, and 256 samples on the right.



- **Height** - this is the depth of the volume, as measured extending out from the front of the light. A setting of 20m extends the rays 20m from the light including falloff.
- **Radius** - sets the circular region around the light where volume effect originates from.

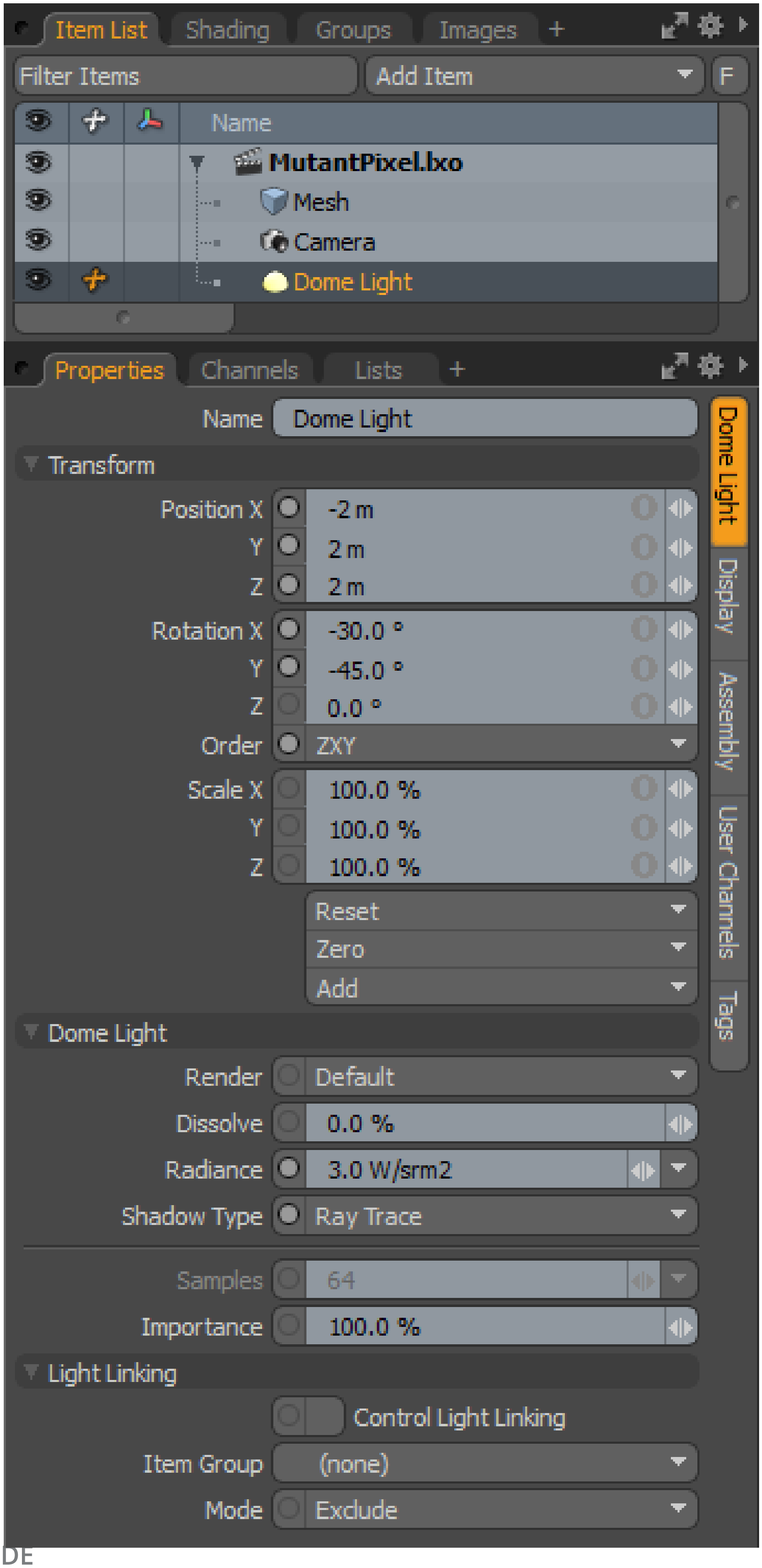
Dome Light

The **Dome Light** creates a virtual dome of lights around the scene producing a flat, even illumination over surfaces similar to photos taken on an overcast day. Dome lights have often been used to simulate Global Illumination; however, with the Modo rendering engine, Global Illumination is just as fast, if not faster, than dome lights and generally yields more pleasing results.



This light has **Position** and **Rotation** controls as well as **Radiance**, **Shadow Type**, and **Samples**.

Note: There is no size or scale value as the dome always scales itself to encompass the entire scene, regardless of size. To control the light's color, shadow color, and other material-related properties, refer to the Light Material topic.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Camera
Dome Light
Properties
Channels
Lists
+
Dome Light
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Dome Light
Light Linking

Name Dome Light

Position X -2 m
Y 2 m
Z 2 m
Rotation X -30.0 °
Y -45.0 °
Z 0.0 °
Order ZXY
Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Render Default
Dissolve 0.0 %
Radiance 3.0 W/srm2
Shadow Type Ray Trace
Samples 64
Importance 100.0 %

☐ Control Light Linking
Item Group (none)
Mode Exclude

- **Name** - this data field displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available for the **Dome Light**:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that the rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This scale transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. Transform items are the channel groups that store the actual transform values, controlling any item's position, rotation, and/or scale. You can add as many transform items as you want for any transform property desired. Adding additional **Transform** items produces an additive effect where each transform group is evaluated before the next, and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

Note: It should be noted that by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.

There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the master scene, they must first exist.

The following **Dome Light** options are available for the **Dome Light**:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and, when invisible, it does not. On some instances you may prefer to fix this state, setting the light as **On** (enabled) or **Off** (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.

- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiance** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Dome Light** and decreasing the value reduces the light intensity. With dome lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.
- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.
- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

  As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.
- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.

The following **Light Linking** options are available for the **Dome Light**:

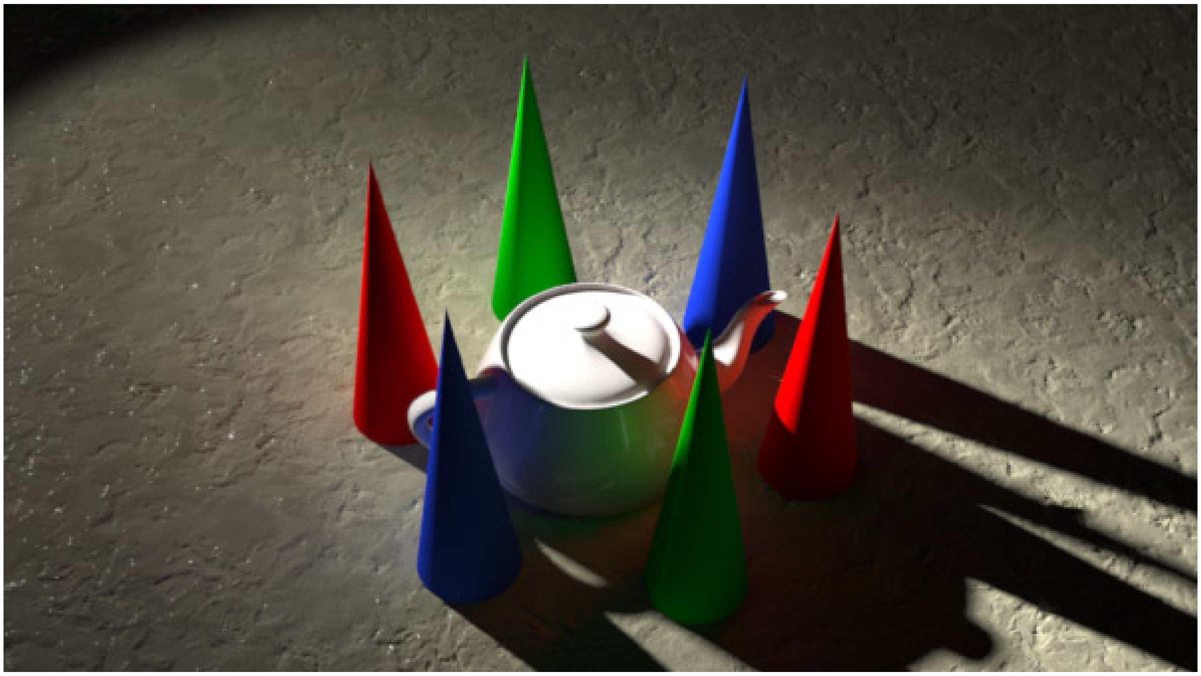
- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the shader, it's possible to control a light's effect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the shader, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.
- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting

the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option and click **OK** to accept. Once the group is defined, select the named group here.

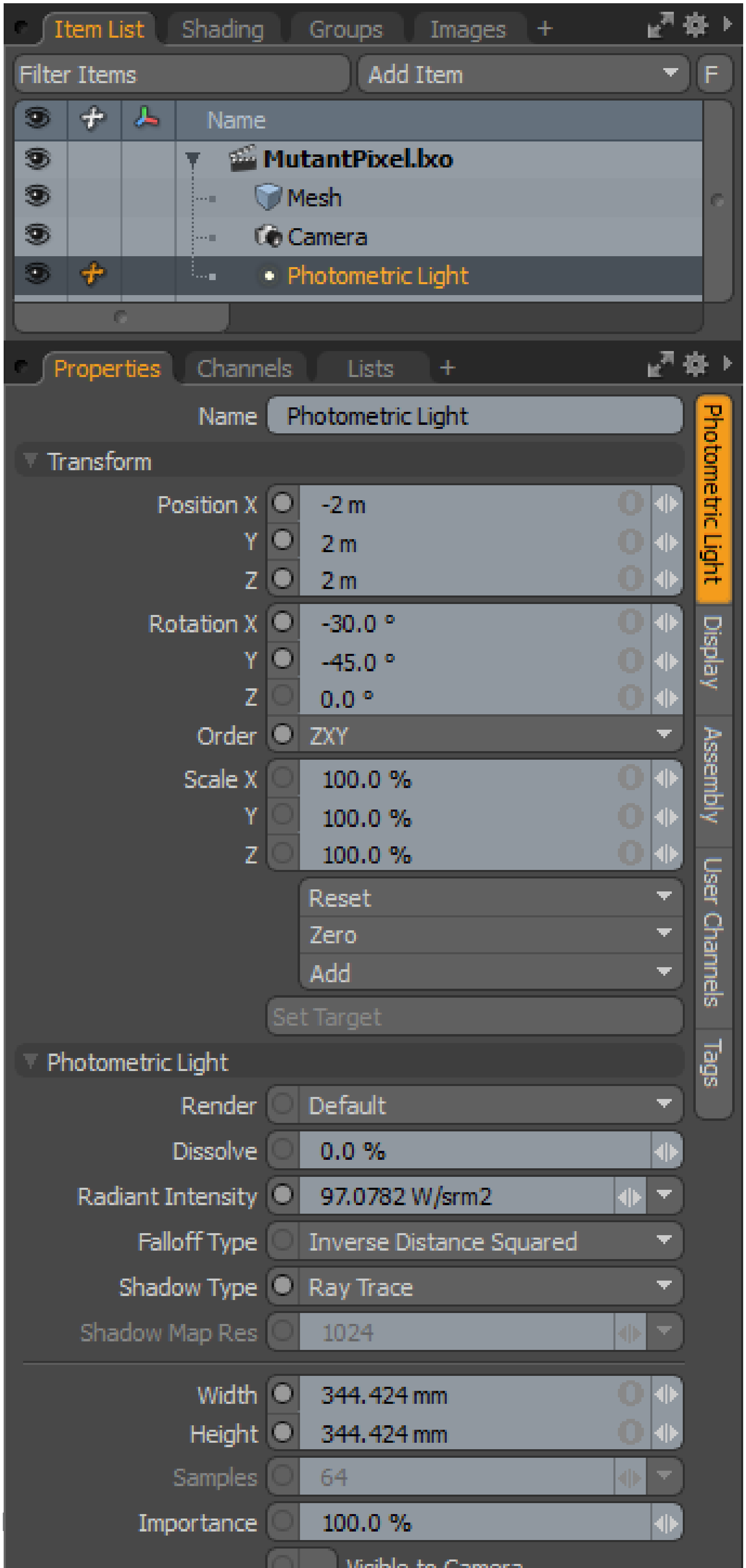
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

Photometric Light

Photometric lights use data provided by lighting manufacturers in the form of IES or EULUMDAT format photometry files. These files contain "photometric" measurements of a light's intensity in different directions, as well as information about the size of the actual luminous surface.



Modo uses this directional intensity information to shape the distribution of light and match real-world "luminaires" (light sources or fixtures). These photometric files also describe a luminaire's real-world intensity, and this value is extracted from the lighting distribution and displayed in the light's **Intensity** channel, where you can edit it to suit your scene. Keep in mind, editing the intensity means that it is no longer an accurate simulation of real lighting. The photometry files remain external to the Modo scene files and, like images, must be available when loading the scene.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Camera
Photometric Light
Properties
Channels
Lists
+
Photometric Light
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Set Target
Photometric Light

Name Photometric Light

Position X -2 m

Y 2 m

Z 2 m

Rotation X -30.0 °

Y -45.0 °

Z 0.0 °

Order ZXY

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

Render Default

Dissolve 0.0 %

Radiant Intensity 97.0782 W/srm2

Falloff Type Inverse Distance Squared

Shadow Type Ray Trace

Shadow Map Res 1024

Width 344.424 mm

Height 344.424 mm

Samples 64

Importance 100.0 %

Visible to Camera

- **Name** - this data field displays the current item name. You may easily change it by clicking within the field and typing the new name.

The following **Transform** options are available for the **Photometric Light**:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This **Scale** transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's **Channels** with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. **Transform** items are the channel groups that store the actual transform values controlling any item's position, rotation, and/or scale. You can add as many **Transform** items as desired for any transform property desired. Adding additional **Transform** items produce an additive effect, where each transform group is evaluated before the next and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

> **Note:** It should be noted that, by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.
>
> There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Because of this fact, you may need to specifically create item transforms when **Referencing** because, in order to override the channels in the master scene, they must first exist.

- **Set Target** - by selecting the light item, and a single additional item in the item list, then pressing **Set Target**, this function allows you to target specific items in a scene, automating the rotation of an item, so that it continuously points towards the targeted item. Once activated, additional options appear:
  - **Remove Target** - removes the target link between the two items.
  - **Enable** - toggling this option off temporarily disables the targeting function while retaining the link between the items.

  - **Set Focus Distance** - sets the focal distance for DOF when targeting camera items to other items.
  - **Roll** - offset the item's rotation angle away from the target.
  - **Time Offset** - set time offset, by a number of frames; to set how the light follows the target item. It can either be delayed, behind it with a negative value, or run ahead of it with a positive value.

The following **Photometric Light** options are available for the **Photometric Light**:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it does not. In some instances, you may prefer to fix this state, setting the **Photometric Light** as **On** (enabled) or **Off** (disabled), regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.
- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiant Intensity** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Photometric Light** and decreasing the value reduces the light intensity. With photometric lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.

> **Tip:** Keep in mind that a light's intensities and falloffs adhere to real world scale values. Scenes that do not adhere to a real world scale factor likely require intensities to be adjusted appropriately to achieve appropriate illumination.

- **Falloff Type** - light in the real world isn't a uniform brightness; its intensity diminishes with distance. Photographers may be familiar with the concept that a light is a quarter as bright at twice the distance away, known as the Inverse Square Law. Modo lights default to this setting, providing a realistic way to light a scene. However, there are times when you may not wish to have this behavior, so three falloff type options are provided:
  - **None** - no falloff; light is consistently bright across its distance.
  - **Inverse Distance** - light is half as bright at twice the distance.
  - **Inverse Distance Squared** - default behavior; light is a quarter as bright at twice the distance.
- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.

- **Shadow Map Res** - sets the resolution for the calculated deep shadow map in pixels. This option is only available when **Deep Shadow Maps** is selected. Increasing this value increases the quality and accuracy of the generated map, while increasing renter times as well.
- **Width/Height** - defines the size of the illumination area for the **Photometric Light**, much like it does for an area light. Larger values produce more light, while smaller values produce less. The size value are extracted from the photometry file. Modifications to this value would no longer produce an accurate simulation of the light.
- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

  As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.
- **Width/Height**: The 'Width/Height' setting defines the size of the illumination area for the Photometric light, much like it does for an area light. Larger values produce more light while smaller values produce less. The size value are extracted from the photometry file, modifications to this value would no longer produce an accurate simulation of the light.
- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

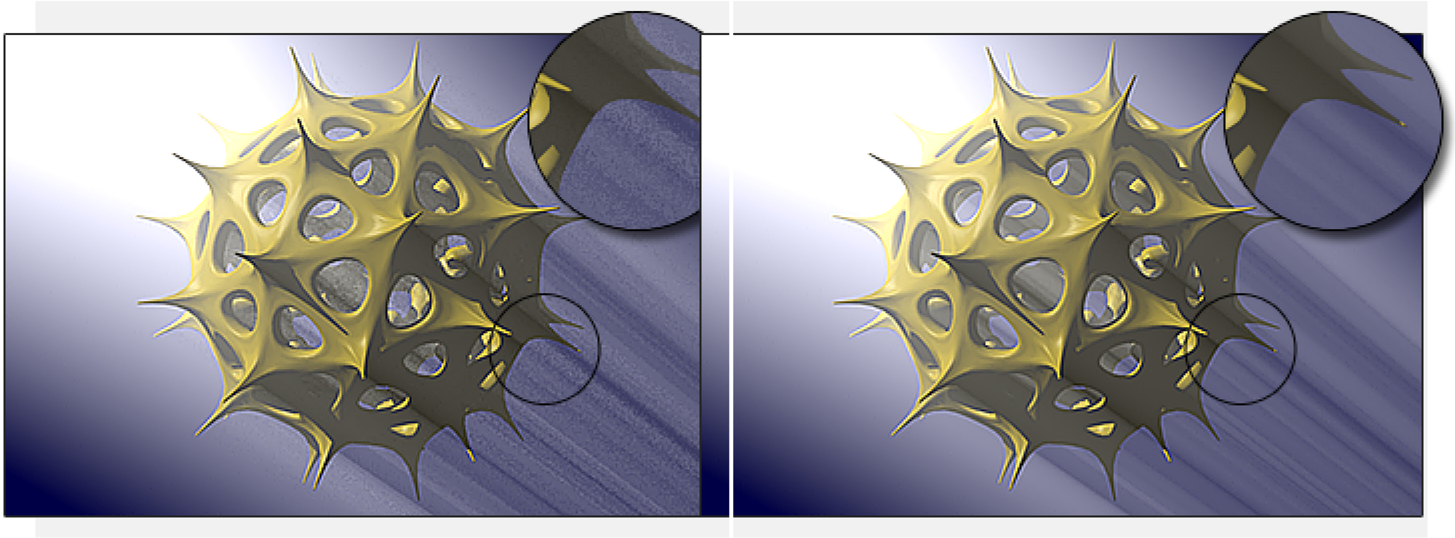
  As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.
- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.

- **Visible to Camera** - you can enable this option to make the light itself visible to cameras. The size of the visible light is determined by the **Radius** attribute.
- **Visible to Reflection/Refraction Rays** - you can enable either or both of these options to make the light itself visible to reflective and/or transparent surfaces. The size of the visible light is determined by the **Radius** attribute.
- **LPE Label** - you can provide a label name here, which appears as the label for any Light Path Expression output you generate for this light. See Rendering with mPath.
- **Cone Angle** - photometric lights emit their light in a cone shape that originates from the light's position toward the default Z direction. The **Cone Angle** defines the shape of the cone that is generated; larger values produce wider conical shapes and therefore larger areas of illumination, while smaller values produce tighter cones with smaller areas of illumination. A setting of 360° produces light similar to a point light radiating in all directions equally. This value can also be set interactively with the **Transform** tool in the 3D viewport.
- **Soft Edge** - defines an illumination falloff inside the edge of the photmetric lights cone producing a more natural look. The soft edge is contained within the **Cone Angle**, and attenuates illumination at the edges. A 40° **Cone Angle** with a 10° falloff would really be a 20° cone angle of full illumination that would falloff 10° across the outer edges of the cone. This value can also be set interactively with the **Transform** tool in the 3D viewport.
- **Photometry File** - the browse function opens a file dialog window that allows you to locate and load a **Photometry File** in **.ldt** or **.ies** formats. Like image maps, photometry files are external and need to be available to Modo each time a scene is loaded. When moving files, make sure to include the photometric files as well.

The following **Volumetrics** options are available for the **Photometric Light**:

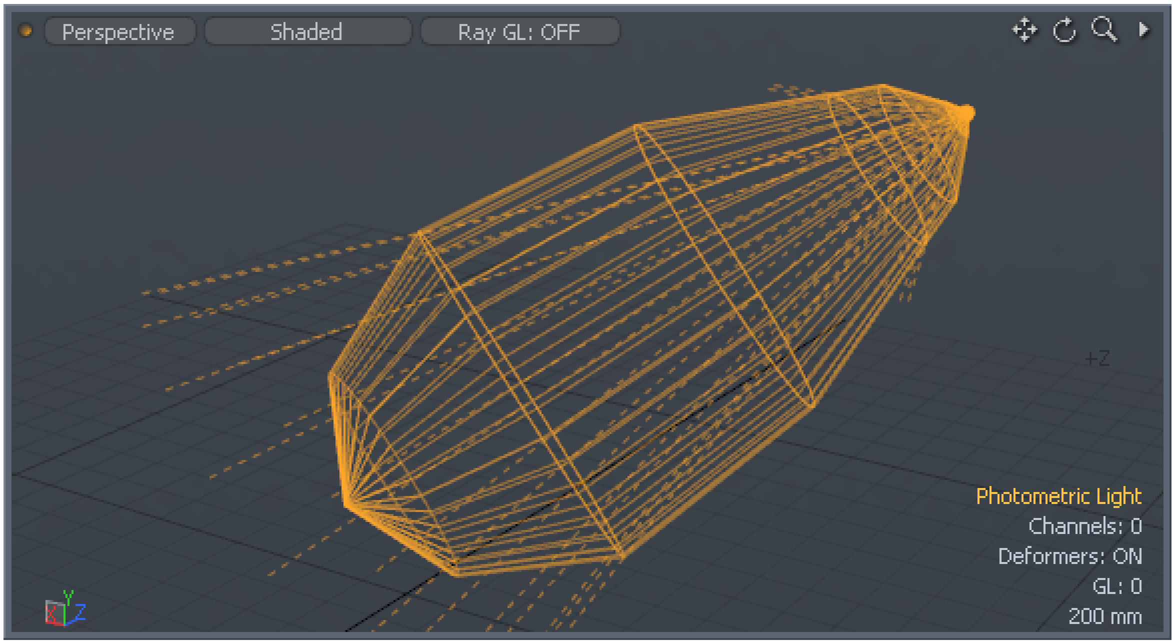
When light shines on a particular matter suspended in the air, such as water vapor, smoke, or even pollution, the light rays become visible. In computer graphics, this effect is known as "volumetric lights". Their affect can be subtle or pronounced and, either way, it can add a good amount of believability and atmosphere to renders. Additional settings related to volumetrics can be found in the Light Material item.

- **Volumetrics** - toggles the effect on or off. When enabled, the **Samples**, **Height**, and **Base** options become available.
- **Samples** - any objects that intersect the light beam volume cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light, sometimes referred to as "god rays". The **Sample** setting defines the degree of accuracy that Modo uses to calculate these shadows through the volume; higher numbers of samples give smoother and more accurate results, while lower values render more quickly but become increasingly grainy. The example images below demonstrate the difference between 32 samples, on the left, and 256 samples on the right.



- **Sphere Radius** - this is the size of the volume overall as measured from the light's center position, extending out in all directions around the source. A setting of 10m produces a sphere with an overall radius of 20m including falloff. The actual falloff within the spheres volume is dictated by the supplied photometric information.

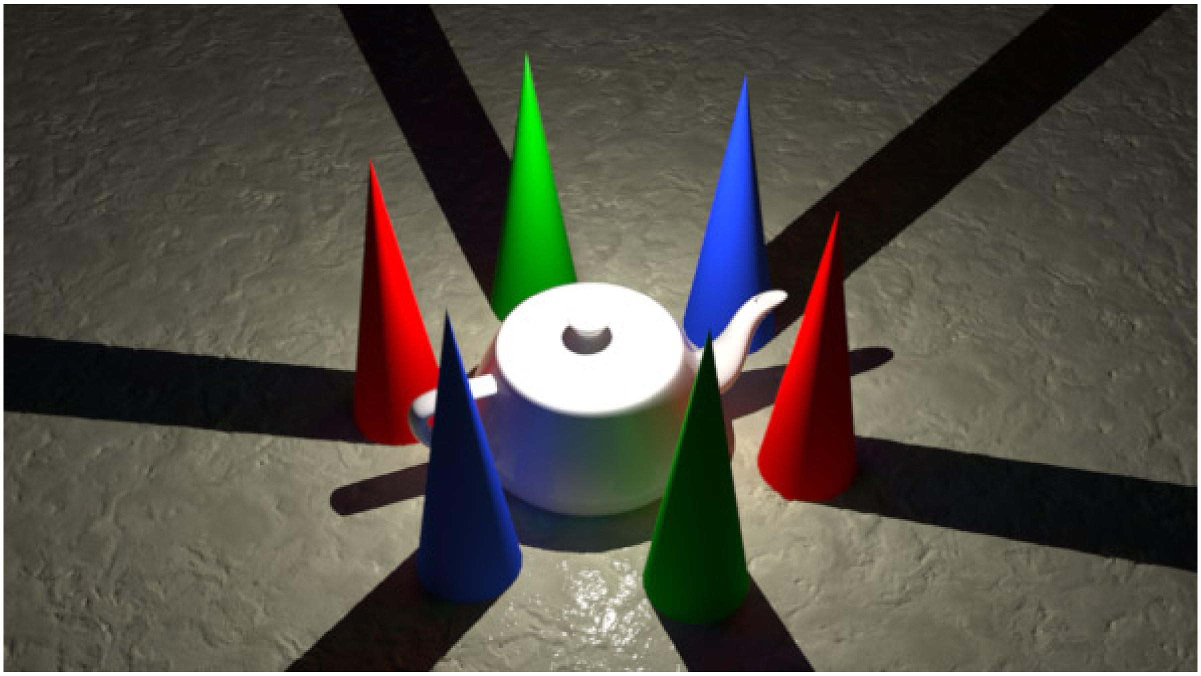
Photometric Light Shape Preview


Perspective
Shaded
Ray GL: OFF
+Z
Photometric Light
Channels: 0
Deformers: ON
GL: 0
200 mm

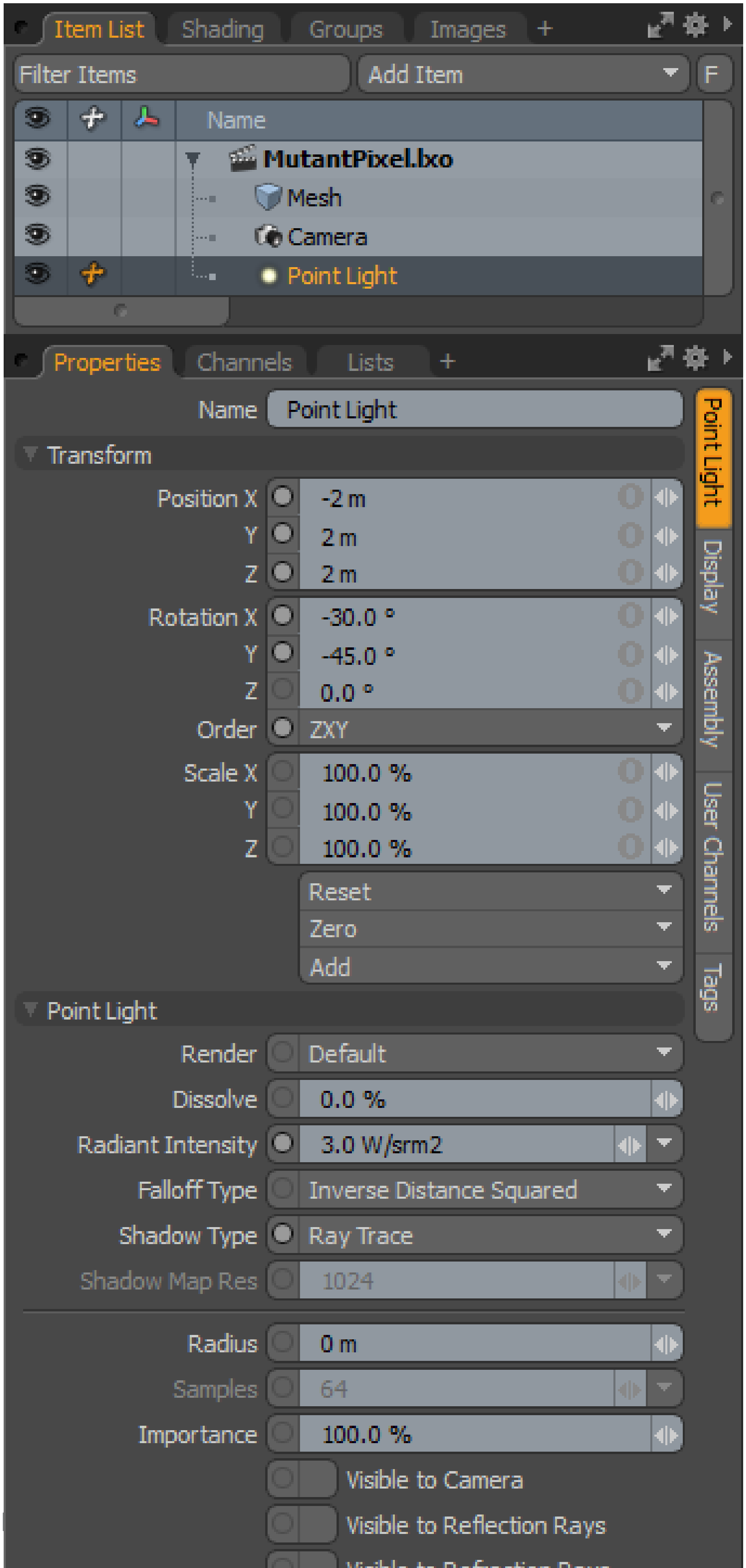
Photometric lights display a displaced wireframe sphere at the lights position with the displacement radius based on the intensity of the photometric data file (IES or EULUMDAT). You can adjust the size of the preview by adjusting the light's 'Size' value in the 'Display' sub-tab.

Point Light

The **Point Light** emits light equally in all directions, similar to a light bulb or a candle flame as all rays originate from a single point.



Point lights obey the physical law of inverse square falloff for intensity. As with all items, the **Point Light** controls are located on the **Properties** tab when the light is selected. Point lights have **Position** and **Rotation** options as well as controls for the intensity of the light, shadow type, and other settings covered fully below. To control the lights color, shadow color, and other material-related properties, refer to the Light Material topic.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Camera
Point Light
Properties
Channels
Lists
+
Point Light
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Point Light

Name: Point Light

Position X: -2 m
Y: 2 m
Z: 2 m

Rotation X: -30.0 °
Y: -45.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Render: Default
Dissolve: 0.0 %
Radiant Intensity: 3.0 W/srm2
Falloff Type: Inverse Distance Squared
Shadow Type: Ray Trace
Shadow Map Res: 1024

Radius: 0 m
Samples: 64
Importance: 100.0 %

☐ Visible to Camera
☐ Visible to Reflection Rays

- **Name** - this data field displays the current item name. You may easily change it by clicking within the field and typing the new name.

The following **Transform** options are available for the **Point Light**:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This **Scale** transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's **Channels** with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. **Transform** items are the channel groups that store the actual transform values controlling any item's position, rotation, and/or scale. You can add as many **Transform** items as desired for any transform property desired. Adding additional **Transform** items produce an additive effect, where each transform group is evaluated before the next and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

Note: It should be noted that, by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.

There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Because of this fact, you may need to specifically create item transforms when **Referencing** because, in order to override the channels in the master scene, they must first exist.

The following **Point Light** options are available for the **Point Light**:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and when invisible, it does not. In some instances, you may prefer to fix this state, setting the **Point Light** as **On** (enabled) or **Off** (disabled), regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.

- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiant Intensity** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Point Light** and decreasing the value reduces the light intensity. With point lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.

Tip: Keep in mind that a light's intensities and falloffs adhere to real world scale values. Scenes that do not adhere to a real world scale factor likely require intensities to be adjusted appropriately to achieve appropriate illumination.

- **Falloff Type** - light in the real world isn't a uniform brightness; its intensity diminishes with distance. Photographers may be familiar with the concept that a light is a quarter as bright at twice the distance away, known as the Inverse Square Law. Modo lights default to this setting, providing a realistic way to light a scene. However, there are times when you may not wish to have this behavior, so three falloff type options are provided:
  - **None** - no falloff; light is consistently bright across its distance.
  - **Inverse Distance** - light is half as bright at twice the distance.
  - **Inverse Distance Squared** - default behavior; light is a quarter as bright at twice the distance.
- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.
- **Shadow Map Res** - sets the resolution for the calculated deep shadow map in pixels. This option is only available when **Deep Shadow Maps** is selected.
- **Radius** - when above '0', gives the direct light some size. The default value of 0m results in crisp hard-edged shadows. An increased **Radius** produces a soft-edged result; the larger the **Radius**, the softer the specular highlight and shadow is. Keep in mind that increasing this value too high results in noisy shadows unless you also increases the number of Direct Illumination **Light Samples** in the **Render** item, or the light's **Samples** value when adaptive sampling is disabled.
- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.

- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.
- **Visible to Camera** - you can enable this option to make the light itself visible to cameras. The size of the visible light is determined by the **Radius** attribute.
- **Visible to Reflection/Refraction Rays** - you can enable either or both of these options to make the light itself visible to reflective and/or transparent surfaces. The size of the visible light is determined by the **Radius** attribute.
- **LPE Label** - you can provide a label name here, which appears as the label for any Light Path Expression output you generate for this light. See Rendering with mPath.

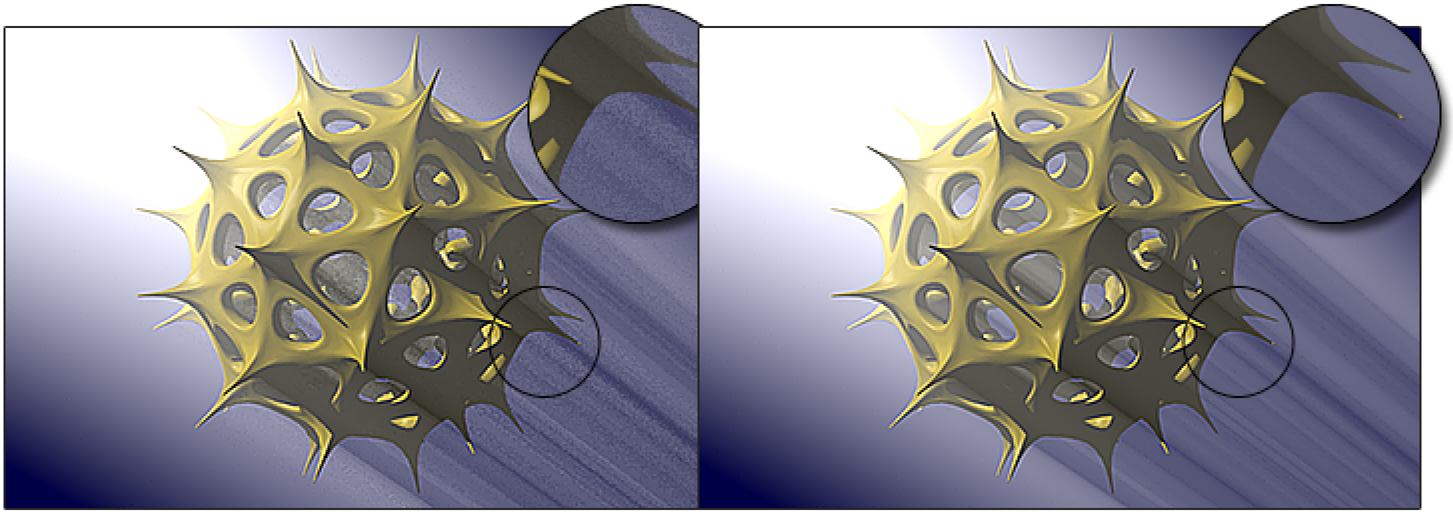
The following **Light Linking** options are available for the **Point Light**:

- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the shader, it's possible to control a light's effect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the shader, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.
- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option and click **OK** to accept. Once the group is defined, select the named group here.
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

The following **Volumetrics** options are available for the **Point Light**:

When light shines on particular matter suspended in the air, such as water vapor, smoke, or even pollution, the light rays become visible. In computer graphics this effect is known as "volumetric lights". Their affect can be subtle or pronounced and, either way, it can add a good amount of believability and atmosphere to renders. Additional settings related to volumetrics can be found in the Light Material item.

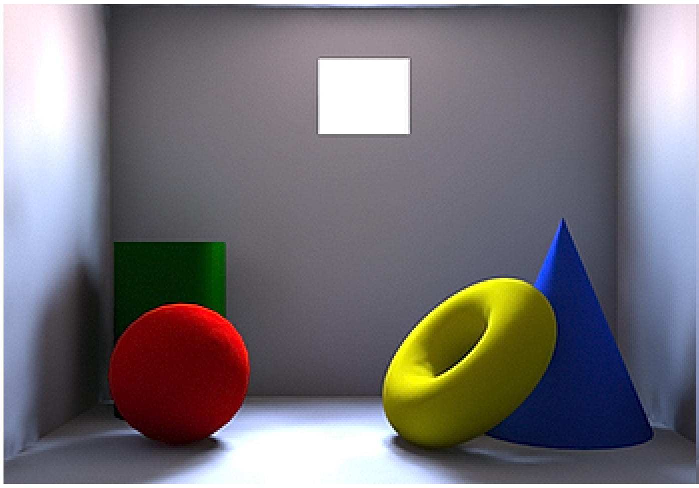
- **Volumetrics Checkbox** - toggles the effect on or off. When enabled, the **Samples**, **Height**, and **Base** options become available.
- **Samples** - any objects that intersect the light beam volume cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light, sometimes referred to as "god rays". The **Sample** setting defines the degree of accuracy that Modo uses to calculate these shadows through the volume; higher numbers of samples give smoother and more accurate results, while lower values render more quickly but become increasingly grainy. The example images below demonstrate the difference between 32 samples, on the left, and 256 samples on the right.



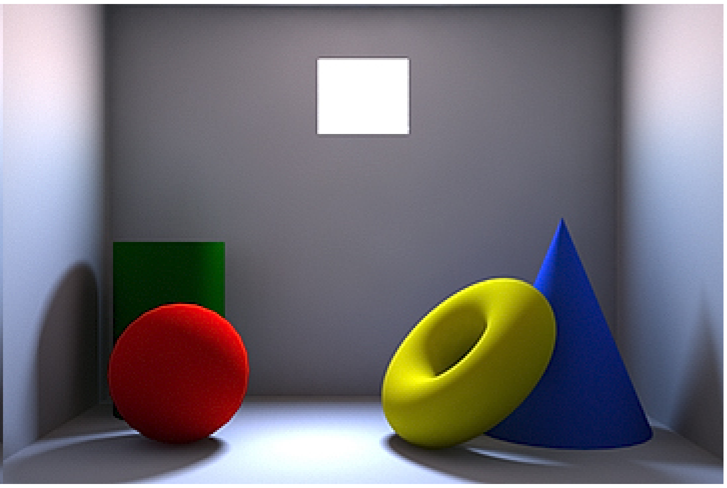
- **Sphere Radius** - the size of the volume overall as measured from the light's center position, extending out in all directions around the source. A setting of 10m produces a sphere with an overall radius of 20m including falloff.

Portal

Portals are not lights in the traditional sense, as they produce no illumination by themselves. They are meant more to control lighting, especially that of interior renders that rely mainly on Global Illumination. Think of them as gathering rays and focusing them into a specific area. In this following example, the interior is illuminated solely by the environment, no lights are being used whatsoever. At the default settings, the image on the left exhibits some chattering and splotches, especially in the area of shadow transitions. In the second image on the right, a **Portal** has been added to the scene and set with 512 light samples. This produces a much smoother result with no other setting or value changes.

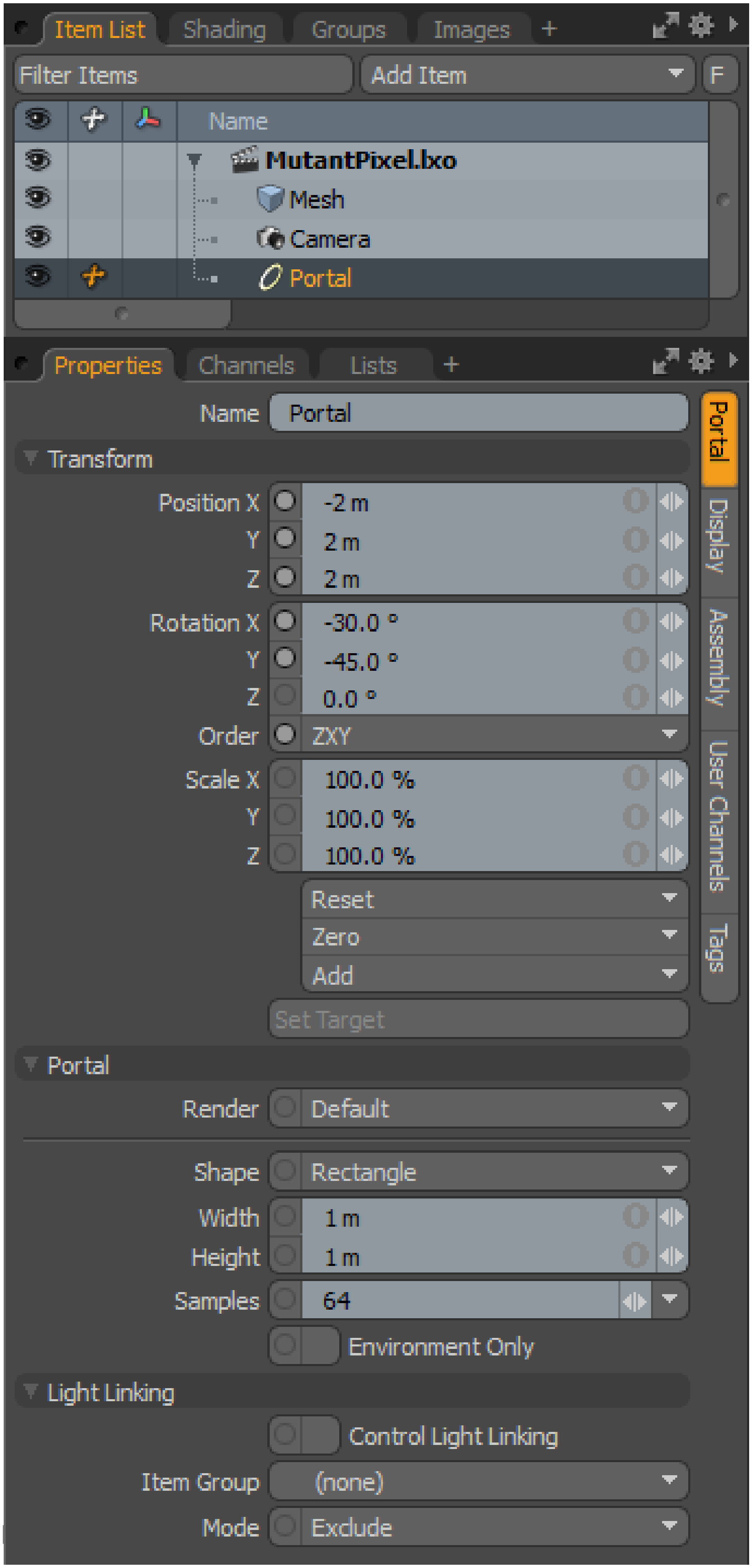


No Portal, 512 IC Samples



Portal Added with 512 Samples

The main use for a **Portal** is not, surprisingly, to be placed at openings; anywhere light would enter into an interior, such as a window, door, or skylight. For best results you should position the portal directly outside the opening itself (or inside depending on the shape of the opening) and scale it to be slightly larger. You can also adjust the shape of the portal to better represent the shape of the window, either as round or rectangular. You should also note it's better to use fewer portals than more, as this can add significantly to render calculation times. For example, when representing a multi-paned window, using a single **Portal** scaled to the over-all window size would be better than individual portals for each pane. You should also note that portals take their color from the environment behind them and therefore have no light material setting, like those of other direct light sources.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Camera
Portal
Properties
Channels
Lists
+
Portal
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Set Target
Portal
Light Linking

Name: Portal

Position X: -2 m
Y: 2 m
Z: 2 m

Rotation X: -30.0 °
Y: -45.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Render: Default

Shape: Rectangle

Width: 1 m

Height: 1 m

Samples: 64

Environment Only

Control Light Linking

Item Group: (none)

Mode: Exclude

- **Name** - this data field displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available for **Portal**:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This scale transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist it simply adds them. **Transform** items are the channel groups that store the actual transform values controlling any items position, rotation, and/or scale. You can add as many transform items as desired for any transform property desired. Adding additional **Transform** items produces an additive effect where each transform group is evaluated before the next and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

  It should be noted that, by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are created on an as-needed basis, reducing scene overhead. There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform items to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the master scene, they must first exist.
- **Set Target** - by selecting the light item, and a single additional item in the item list, and then pressing **Set Target**, this function allows you the ability to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. Once activated, additional options appear.
  - **Remove Target** - removes the target link between the two items.
  - **Enable** - toggling this option off temporarily disables the targeting function while retaining the link between the items.
  - **Set Focus Distance** - sets the focal distance for DOF when targeting camera items to other item.
  - **Roll** - offset the item's rotation angle away from the target.

- **Time Offset** - time the offset, by a number of frames, to set how the light follows the target item. It can either be delayed behind it with a negative value, or run ahead of it with a positive value.

The following **Portal** options are available for **Portal**:

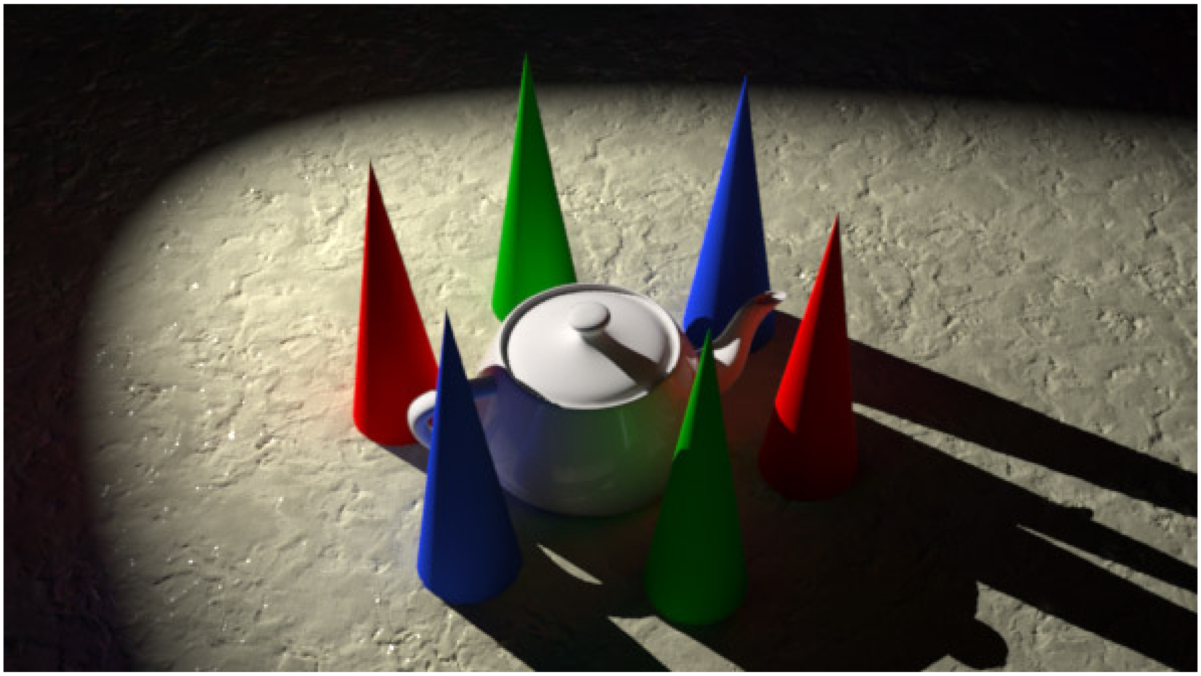
- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable portals using the visibility function 👁 of the item list. When the **Portal** is visible, it contributes to the final rendered scene and when invisible, it does not. In some instances, you may prefer to fix this state, setting the **Portal** as **On** (enabled) or **Off** (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling portals for test renders.
- **Shape** - determines the final shape of the **Portal**, set as either **Rectangular** or **Elliptical**. You can then adjust the **Width** and **Height** options to further define square, rectangular, circular, or oval type shapes.
- **Width/Height** - determines the actual size of the **Portal**. You should size the **Portal** to approximate the overall size of the opening it's representing.
- **Samples** - determines the portal's light gathering ability, having greater numbers of **Samples** increases the overall accuracy of the final rendered image, but also keeps in mind that additional rays also adversely affect render times, so you should adjust the rays to find a balance of quality versus render time.
- **Environment Only** - when this option is enabled, all sampling is taken from the **Environment** definition and any direct lights in the scene are ignored.

The following **Light Linking** options are available for **Portal**:

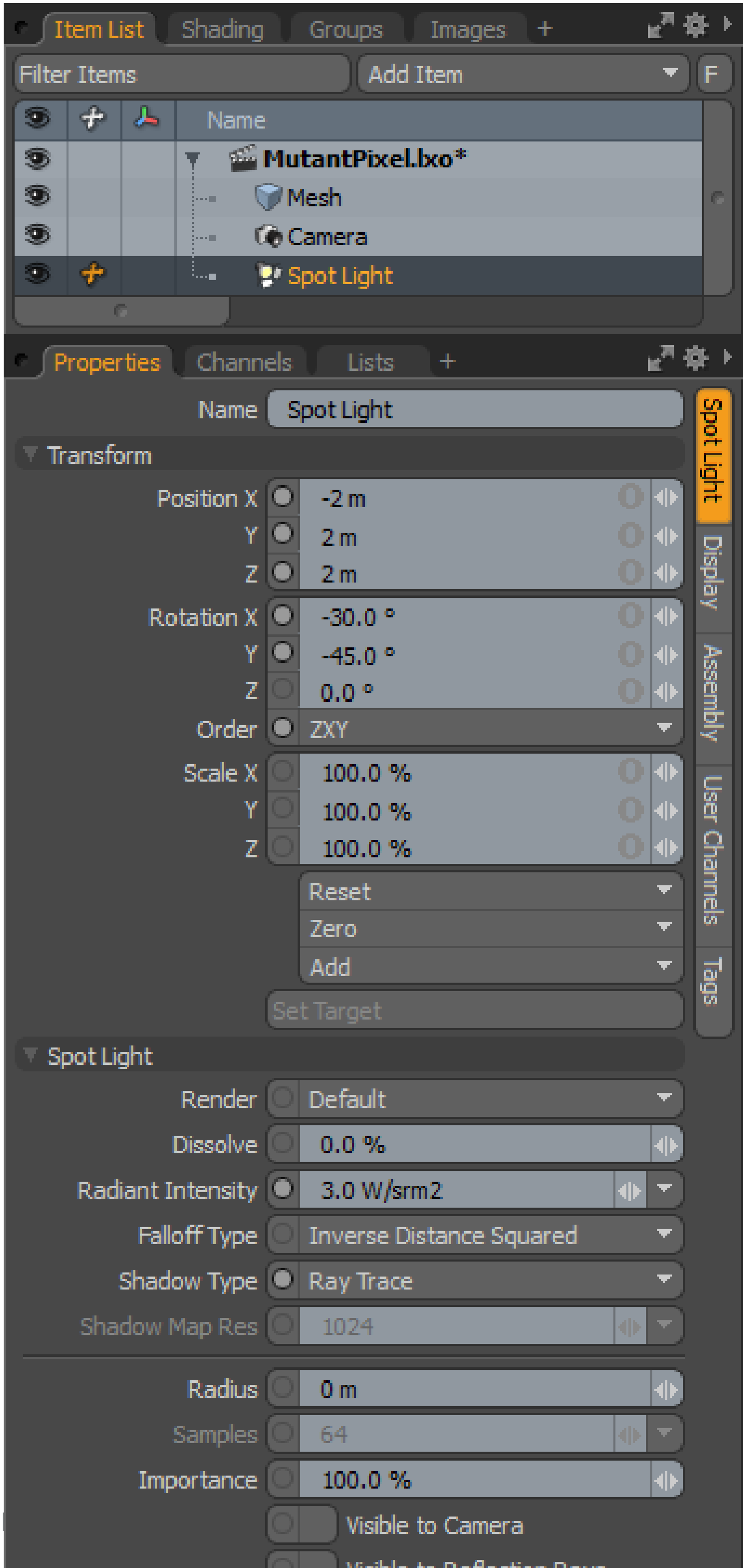
- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the **Shader**, it is possible to control a lights affect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific Items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the **Shader**, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.
- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option, then click **OK** to accept. Once the group is defined, select the named group here.
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

Spot Light

The **Spot Light** looks and acts like a real world spot light and obeys the physical law of inverse square falloff of intensity. Light radiates from a single point outward in a cone shape, widening the further the subject is from the light itself. As with all items, the spot light controls are located on the **Properties** tab when the light is selected. Spot lights have **Position** and **Rotation** options, as well as controls for the intensity of the light, shadow type, and other settings covered fully below. Using the **Transform** tool with the **Spot Light** selected provides interactive handles for editing position, rotation, cone, and soft edge angles.



To control the lights color, shadow color, and other material-related properties, refer to the Light Material topic.


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
Properties
Channels
Lists
Spot Light
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Set Target
Spot Light

MutantPixel.lxo*
Mesh
Camera
Spot Light

Name: Spot Light

Position X: -2 m
Y: 2 m
Z: 2 m

Rotation X: -30.0 °
Y: -45.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Render: Default
Dissolve: 0.0 %
Radiant Intensity: 3.0 W/srm2
Falloff Type: Inverse Distance Squared
Shadow Type: Ray Trace
Shadow Map Res: 1024

Radius: 0 m
Samples: 64
Importance: 100.0 %
Visible to Camera
Visible to Reflection Rays

- **Name** - this data field displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available for the **Spot Light**:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Order** - allows you to set the order that the rotations are applied to the light item. Changing the order that rotations are applied can sometimes help to reduce or eliminate "gimbal lock".
- **Scale** - allows you to numerically set the size of the light item. This scale transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. Transform items are the channel groups that store the actual transform values, controlling any item's position, rotation, and/or scale. You can add as many transform items as you want for any transform property desired. Adding additional **Transform** items produces an additive effect where each transform group is evaluated before the next, and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

Note: It should be noted that by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.

There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the master scene, they must first exist.

- **Set Target** - by selecting the **Light** item, and a single additional item in the item list, and then pressing **Set Target**, this function allows you to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. Once activated, additional options appear:
  - **Remove Target** - removes the target link between the two items.
  - **Enable** - toggling this option off temporarily disables the targeting function while retaining the link between the items.

  - **Set Focus Distance** - sets the focal distance for DOF when targeting camera items to other items.
  - **Roll** - offset the item's rotation angle away from the target.
  - **Time Offset** - time the offset, by a number of frames, to set how the light follows the target item. It can either be delayed behind it with a negative value, or run ahead of it with a positive value.

The following **Spot Light** options are available for the **Spot Light**:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and, when invisible, it does not. On some instances you may prefer to fix this state, setting the light as **On** (enabled) or **Off** (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.
- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiant Intensity** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Spot Light** and decreasing the value reduces the light intensity. With spot lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.

> **Tip:** Keep in mind that a light's intensities and falloffs adhere to real world scale values, scenes that do not adhere to a real world scale factor likely require intensities to be adjusted accordingly to achieve appropriate illumination.

- **Falloff Type** - light in the real world isn't a uniform brightness; its intensity diminishes with distance. Photographers may be familiar with the concept that a light is a quarter as bright at twice the distance away, known as the Inverse Square Law. Modo lights default to this setting, providing a realistic way to light a scene. However, there are times when you may not wish to have this behavior, so three falloff type options are provided:
  - **None** - no falloff; light is consistently bright across its distance.
  - **Inverse Distance** - light is half as bright at twice the distance.
  - **Inverse Distance Squared** - the default behavior; light is a quarter as bright at twice the distance.
- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.
- **Shadow Map Res** - sets the resolution for the calculated deep shadow map in pixels. This option is only available when **Deep Shadow Maps** is selected.

- **Radius** - when above '0', gives the direct light some size. The default value of 0m results in crisp hard-edged shadows. An increased **Radius** produces a soft-edged result; the larger the **Radius**, the softer the specular highlight and shadow is. Keep in mind that increasing this value too high results in noisy shadows unless you also increases the number of Direct Illumination **Light Samples** in the **Render** item, or the light's **Samples** value when adaptive sampling is disabled.
- **Samples** - when the **Direct Illumination** option is set above zero, the **Samples** value is grayed out as there is no need for you to adjust it. The number of samples are dynamically allocated using the adaptive light sampling method. This provides you a global control to adjust overall shading and shadow quality. When the **Direct Illumination** option is set to '0', this disables adaptive sampling and you can manually adjust this **Samples** value to control light and shadow quality per light.

  As the size increases, additional samples are required to smooth out the resulting soft specular shading and soft edge shadows. If they appear grainy, increasing this sample setting is the most likely solution. The default value is a good starting point for balancing speed and quality. As the size increases or the resolution of the image increases you can drive this number higher to account for any grain that appears. Keep in mind that increasing the number of samples also increases render times.
- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.
- **Visible to Camera** - you can enable this option to make the light itself visible to cameras. The size of the visible light is determined by the **Radius** attribute.
- **Visible to Reflection/Refraction Rays** - you can enable either or both of these options to make the light itself visible to reflective and/or transparent surfaces. The size of the visible light is determined by the **Radius** attribute.
- **LPE Label** - you can provide a label name here, which appears as the label for any Light Path Expression output you generate for this light. See Rendering with mPath.
- **Cone Angle** - spot lights emit their light in a cone shape that originates from the light's position toward the default Z direction. The **Cone Angle** defines the shape of the cone that is generated; larger values produce wider conical shapes and therefore larger areas of illumination, while smaller values produce tighter cones with smaller areas of illumination. A setting of 360° produces light similar to a point light radiating in all directions equally. This value can also be set interactively with the **Transform** tool in the 3D viewport.
- **Soft Edge** - defines an illumination falloff inside the edge of the spot lights cone producing a more natural look. The soft edge is contained within the **Cone Angle**, and attenuates illumination at the edges. A 40° **Cone Angle** with a 10° falloff would really be a 20° cone angle of full illumination that would falloff 10°

across the outer edges of the cone. This value can also be set interactively with the **Transform** tool in the 3D viewport.

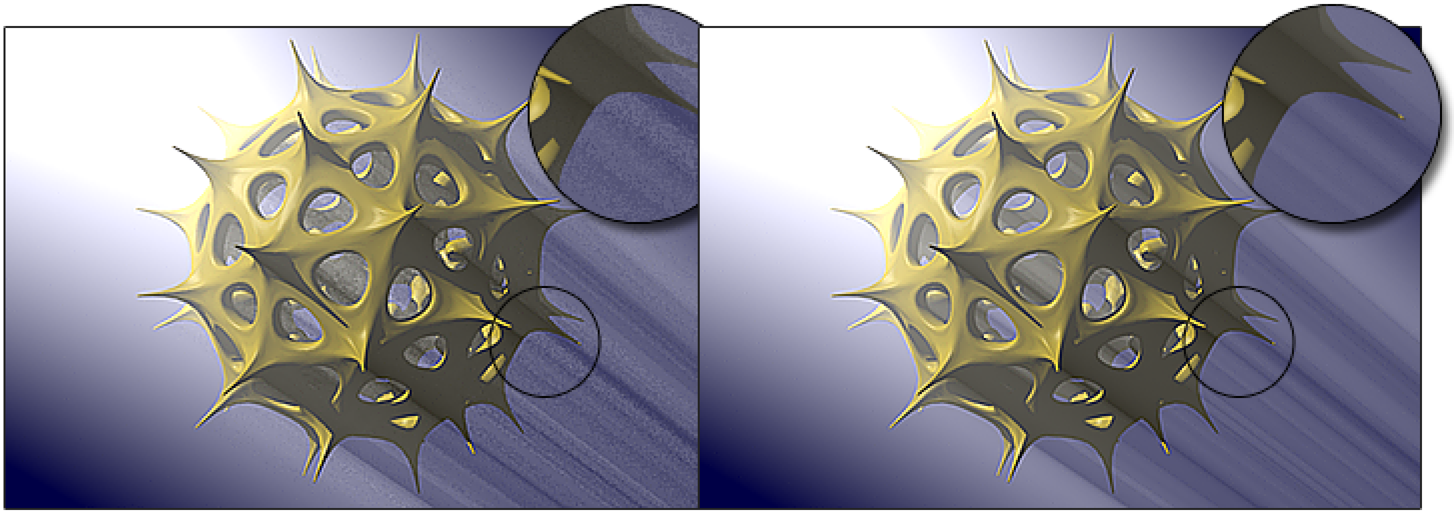
The following **Light Linking** options are available for the **Spot Light**:

- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the shader, it's possible to control a light's effect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the shader, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.
- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option and click **OK** to accept. Once the group is defined, select the named group here.
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

The following **Volumetrics** options are available for the **Spot Light**:

When light shines on a particular matter suspended in the air, such as water vapor, smoke, or even pollution, the light rays become visible. In computer graphics, this effect is known as "volumetric lights". Their affect can be subtle or pronounced and, either way, it can add a good amount of believability and atmosphere to renders. Additional settings related to volumetrics can be found in the Light Material item.

- **Volumetrics** - toggles the effect on or off. When enabled, the **Samples**, **Height**, and **Base** options become available.
- **Samples** - any objects that intersect the light beam volume cast shadows through it. You may have witnessed this when sunlight shining through the clouds creates streaks of light, sometimes referred to as "god rays". The **Sample** setting defines the degree of accuracy that Modo uses to calculate these shadows through the volume; higher numbers of samples give smoother and more accurate results, while lower values render more quickly but become increasingly grainy. The example images below demonstrate the difference between 32 samples, on the left, and 256 samples on the right.



- **Height** - this is the depth of the volume, as measured extending out from the front of the light. A setting of 20m extends the rays 20m from the light, including falloff.
- **Base** - sets the circular region around the light where volume effects originate from.

Mesh Light

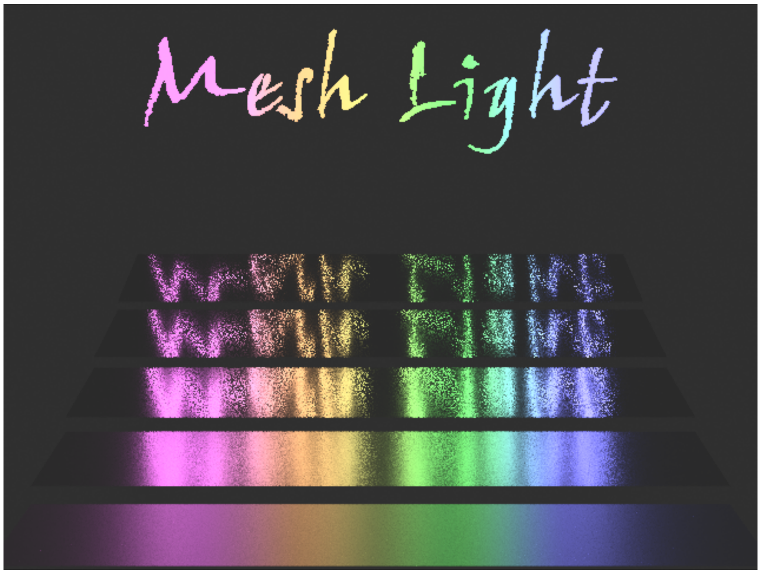
Mesh Lights are a type of direct light source whose shape comes from a mesh item in the scene. Like instances, multiple mesh lights with different positions and orientations can refer to the same mesh item. The prototype mesh itself is hidden during rendering. When direct illumination is calculated, the probability of sampling points within a particular triangle of the **Mesh Light** is proportional to the triangle's surface area, resulting in sample points being evenly distributed across the light.

The main advantage of Mesh Lights over luminous geometry is to reduced shading noise, especially if the light has thin areas and the surface being shaded is diffuse or rough. This is because luminous geometry relies on indirect (BRDF-based) sampling, while Mesh Lights can also use direct (area-based) sampling.

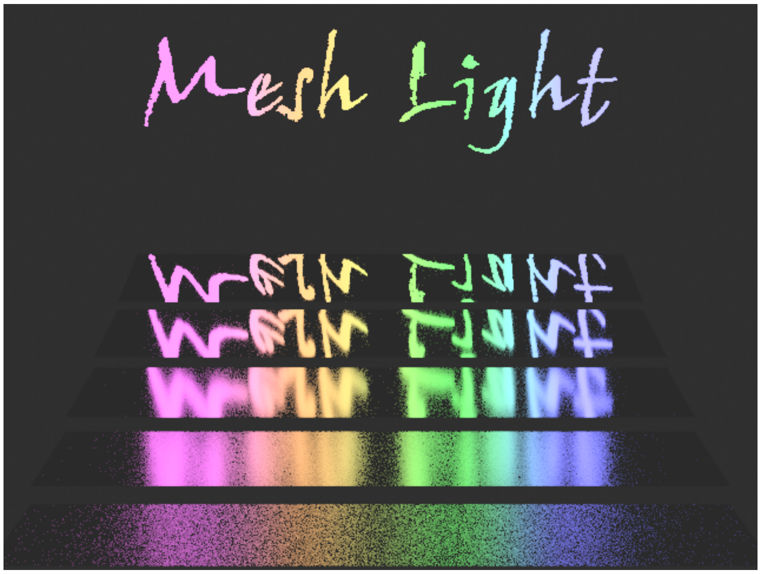
Note: Mesh Lights generate caustics like other direct light types. **Direct Caustics** need to be enabled in the Render Global Illumination > Caustics settings, and caustics only show in final renders, where photon tracing is calculated.

In the following example, the text was modeled and used as the shape of a **Mesh Light**. The rectangles are reflective with roughness values of 2%, 4%, 8%, 16% and 32%. Notice how the **Mesh Light** greatly reduces noise on the rough surface. Direct sampling is noisier on the smooth surfaces, but Multiple Importance Sampling (MIS) resolves that problem. Render times were roughly equal for all three images.

Mesh Light: Indirect Sampling Only:



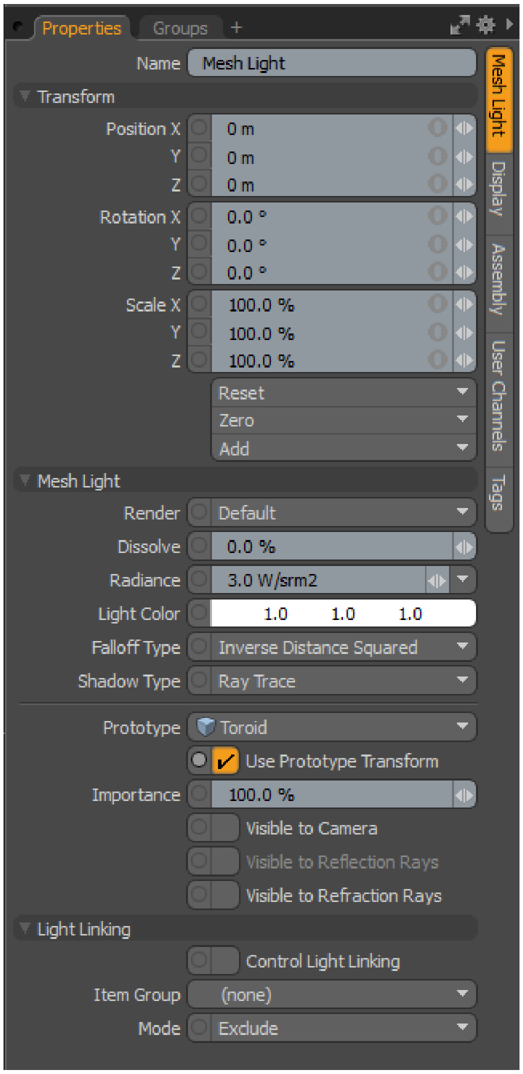
Mesh Light: Direct Sampling Only



Mesh Light: Multiple Importance Sampling (MIS)

To add a **Mesh Light** to your scene, open the **Item List**, click **Add Item** > **Lights** > **Mesh Light**. Once added, specify the item used as the Mesh Light shape in the **Mesh Light** > **Mesh** field and then set the other Rendering options.

Mesh Light Properties



Name: Mesh Light

Transform

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Mesh Light

Render: Default
Dissolve: 0.0 %
Radiance: 3.0 W/srm2
Light Color: 1.0 1.0 1.0
Falloff Type: Inverse Distance Squared
Shadow Type: Ray Trace
Prototype: Toroid
☑ Use Prototype Transform
Importance: 100.0 %
☐ Visible to Camera
☐ Visible to Reflection Rays
☐ Visible to Refraction Rays

Light Linking

☐ Control Light Linking
Item Group: (none)
Mode: Exclude

- **Name** - this data field displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available:

- **Position** - allows you to numerically position the light item in XYZ space.
- **Rotation** - allows you to numerically set the rotation of the light item.
- **Scale** - allows you to numerically set the size of the light item. This scale transform is a multiplier of the **Height** and **Width** options.
- **Reset** - resets the selected transform values to (0,0,0) returning the items back to their default state.
- **Zero** - resets the chosen transform property values to '0', leaving the **Center** position and **Mesh** position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted

version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.

- **Add** - adds additional **Transform** items to the associated item or, if they do not yet exist, it simply adds them. Transform items are the channel groups that store the actual transform values, controlling any item's position, rotation, and/or scale. You can add as many transform items as you want for any transform property desired. Adding additional **Transform** items produces an additive effect where each transform group is evaluated before the next, and so on. Additional item transforms are evaluated in their order in the **Channels** list, from the bottom upwards.

> **Note:** It should be noted that by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization, as only the necessary transforms are created on an as-needed basis, reducing scene overhead.
>
> There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the master scene, they must first exist.

The following **Mesh Light** options are available:

- **Render** - allows you to select from three choices: when set to **Default**, you can enable/disable lights using the visibility function 👁 of the item list. When the light is visible, it contributes to the final rendered scene and, when invisible, it does not. On some instances you may prefer to fix this state, setting the light as **On** (enabled) or **Off** (disabled) regardless of visibility. Also useful for workflows that auto toggle visibility, saving you from manually enabling lights for test renders.
- **Dissolve** - when the **Dissolve** function is set to any value above 0%, the light's overall effect on the scene attenuate as the value increases. When set to 100%, the light's effect on the scene is completely disabled. This function provides a convenient way to dim a light's effect within a scene.
- **Radiance** - controls the intensity of the light and uses the standard physically-based unit of Watts per volumetric meter. As you would expect, increasing this value increases the amount of light coming from the **Mesh Light** and decreasing the value reduces the light intensity. With Mesh Lights, the area of light generated in the scene is very small and therefore large values are likely necessary to illuminate the scene how you would expect.
- **Light Color** - change the color of the Mesh Light. For more information, see Color Picker Viewport.
- **Falloff Type** - light in the real world isn't a uniform brightness; its intensity diminishes with distance. Photographers may be familiar with the concept that a light is a quarter as bright at twice the distance away, known as the Inverse Square Law. Modo lights default to this setting, providing a realistic way to light

a scene. However, there are times when you may not wish to have this behavior, so three falloff type options are provided:

  - **None** - no falloff; light is consistently bright across its distance.
  - **Inverse Distance** - light is half as bright at twice the distance.
  - **Inverse Distance Squared** - default behavior; light is a quarter as bright at twice the distance.

- **Shadow Type** - offers options between **Ray Traced**, **None** and **Deep Shadow Maps**. In situations where you want a light to cast a shadow, **Ray Traced** gives the most accurate results. The traditional hard edge of ray traced shadows can easily be softened using **Spread Angle** value in any light item. **Deep Shadow Maps** are useful for volumetric lights and fur rendering, where a great deal of calculations are required to produce shadows; producing similar results to ray traced shadows while reducing the number of calculations.
- **Prototype** - this option lists the procedural meshes in your scene. Select which **Prototype** to use as the base Mesh Light transform source. The prototype mesh itself is hidden during rendering.
- **Use Prototype Transform** - enabling this option locks the **Mesh Light** transforms to the **Prototype** Mesh item. You can position the Mesh Light by positioning the Mesh item being used as its prototype. By default, this option is enabled. For more information on how this option works, see Using Prototype Transforms for a Mesh Light.

Note: Scenes created in previous versions of Modo load with this option disabled.

- **Importance** - Modo uses an adaptive light sampling method to evaluate lighting in a scene. This option allows you to control illumination sampling using a single global value, producing better results in less time. In rare cases, where adaptive sampling isn't producing the desired outcome, an individual light's **Importance** value can be adjusted, acting as a multiplier to the lights evaluated importance. Changes in **Importance** are relative to other lights in the scene, influencing it one way or the other as more or less important, in effect increasing or decreasing its sample allocation from the total number of possible **Light Samples**, all while remaining fully unbiased.
- **Visible to Camera** - you can enable this option to make the light itself visible to cameras. The size of the visible light is determined by the **Spread Angle** attribute.
- **Visible to Reflection/Refraction Rays** - you can enable either or both of these options to make the light itself visible to reflective and/or transparent surfaces. The size of the visible light is determined by the **Spread Angle** attribute.
- **LPE Label** - you can provide a label name here, which appears as the label for any Light Path Expression output you generate for this light. See Rendering with mPath.

The following **Light Linking** options are available:

- **Control Light Linking** - illumination on a surface is generally controlled by the **Shader** item in the **Shader Tree**. Within the shader, it's possible to control a light's effect on a surface with **Light Linking**. As its name describes, it links the illumination affects of a group of lights to specific items or material groups. When the **Control Light Linking** option is enabled on a light item, it acts as an individual light-specific override to the shader, allowing you to **Include** or **Exclude** a specific light's illumination on a group of items.

- **Item Group** - determines the specific group of item layers in the scene that is affected by the **Light Linking**. The group needs to be defined in the **Groups** viewport panel. This can be done easily by selecting the target items while in **Items** mode and then, in the **Groups** palette, click the **New Group** button. Define a name for the group in the pop-up dialog and choose the **From Selected Items** option and click **OK** to accept. Once the group is defined, select the named group here.
- **Mode** - determines whether the light is **Included**, meaning it only affects the items in the specified item group, and **Excluded** by all other surfaces, meaning it is ignored by any items in the specified item group.

Using Prototype Transforms for a Mesh Light

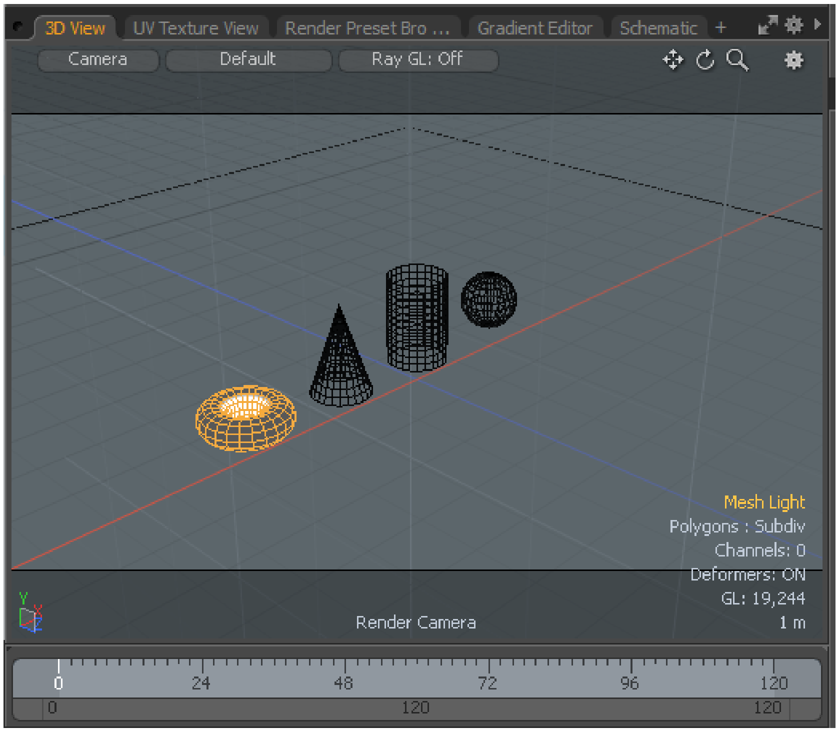
The following procedure is a simple example, which demonstrates how to lock the Mesh Light transforms to a selected mesh in your scene and to reposition the Mesh Light by moving the Mesh item.

To apply Prototype Transforms:

1. Open the **Render** layout.
2. Add a Ground Plane to your scene. Open **Item List**, click **Add Item**, expand **Volumes**, and then double-click **Ground Plane**.
3. In the **Properties** panel, open the **Ground Plane** tab and set the **Transform** > **Scale** values for **X**, **Y**, and **Z** to **2431.0%**
4. Add a number of other procedural meshes to your scene. Open the **Mesh Ops** tab, click **Add Operator**, expand **Mesh Operations** > **Create** and then double-click **Toroid**.

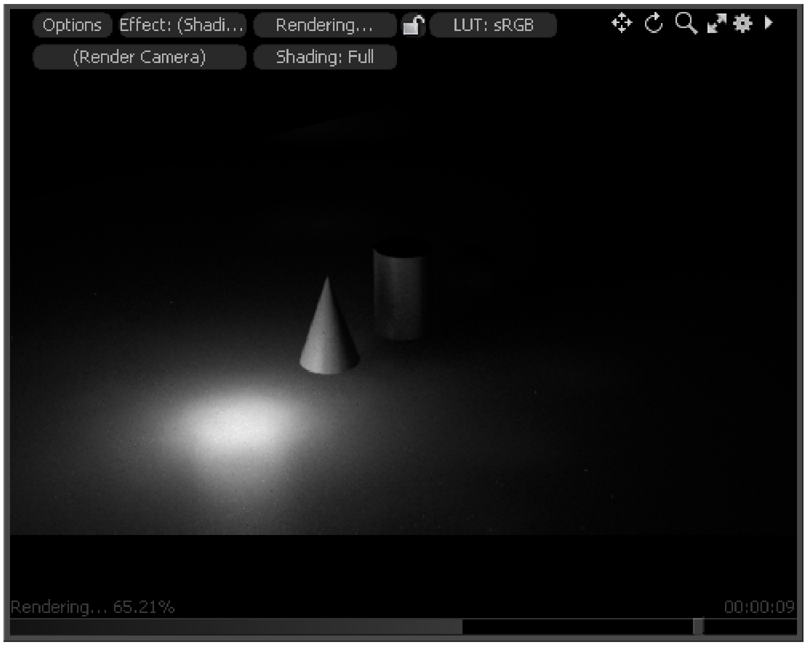
Tip: If you do not see the **Mesh Ops** tab, click the **+** icon and then click **Data Lists** > **Mesh Ops** to add the tab.

5. Repeat **Step 4** and add the following procedural meshes:
   - **Sphere**
   - **Cone**
   - **Cylinder**
6. Align the procedural meshes to display in a line on the X axis. Select a **Mesh** item and then press **W**. Move the handles to reposition each mesh.


3D View
UV Texture View
Render Preset Bro ...
Gradient Editor
Schematic
Camera
Default
Ray GL: Off
Mesh Light
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 19,244
1 m
Render Camera
0
24
48
72
96
120
0
120
120

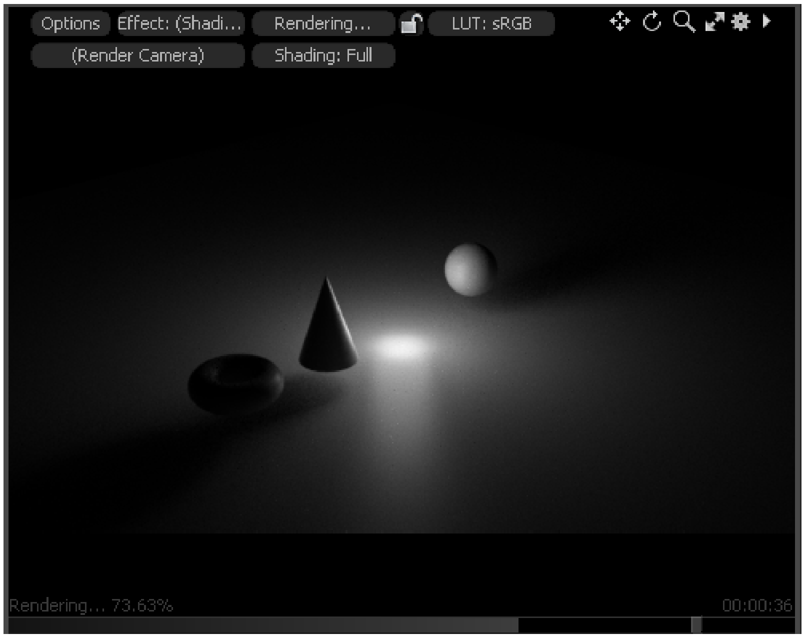
7. From the menu bar, click **Render** > **Open Preview** and then click the arrow to render your scene. Notice, the prototype mesh itself is hidden during rendering.

Tip: You can also open **Preview** by pressing **F8**.


Options
Effect: (Shadi...
Rendering...
LUT: sRGB
(Render Camera)
Shading: Full
Rendering... 65.21%
00:00:09

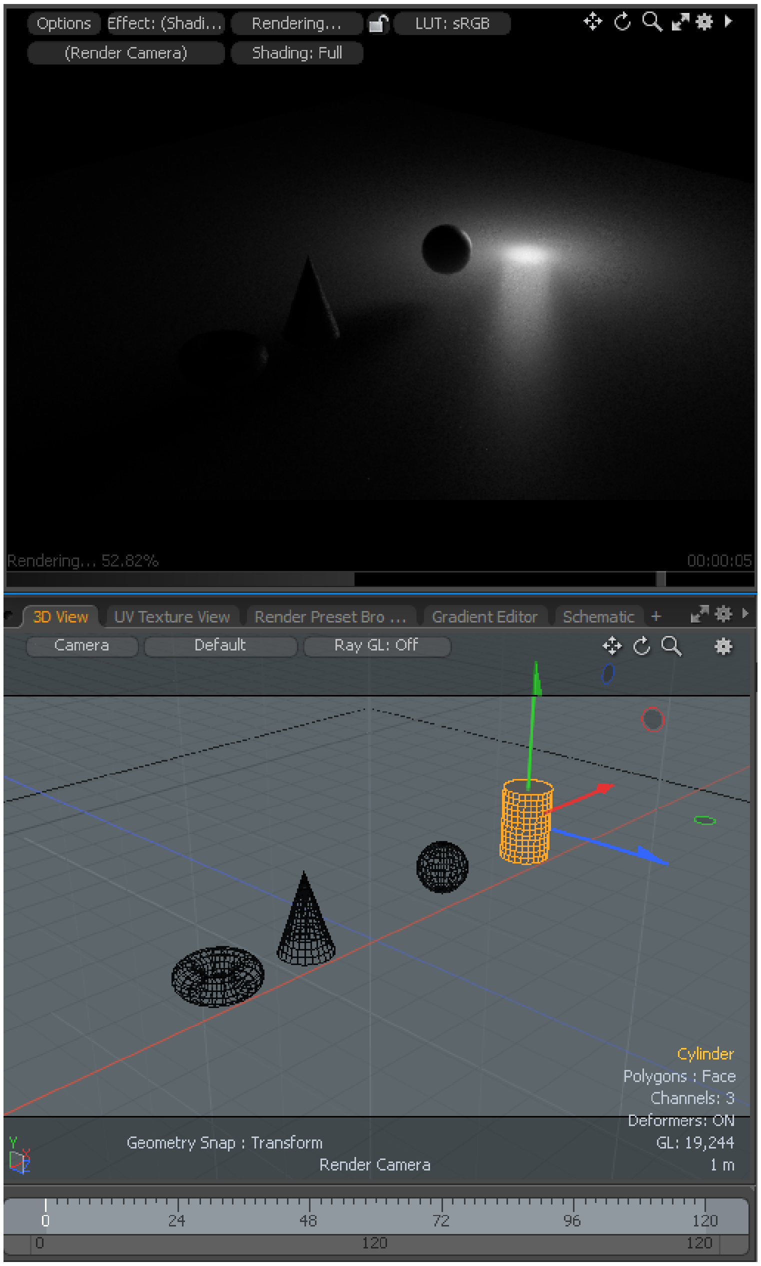
8. In the **Item List**, select the **Mesh Light** item.

9. In the **Properties** panel, open the **Mesh Light** tab, under the **Mesh Light** properties, set **Prototype** to use **Cylinder**. Notice, the Cylinder prototype mesh is hidden during rendering.


Options
Effect: (Shadi...
Rendering...
LUT: sRGB
(Render Camera)
Shading: Full
Rendering... 73.63%
00:00:36

10. In the **Item List**, select the **Cylinder**, press **W** and reposition it.

    The Mesh Light is repositioned to the location where you have moved the Prototype mesh item; **Cylinder**.

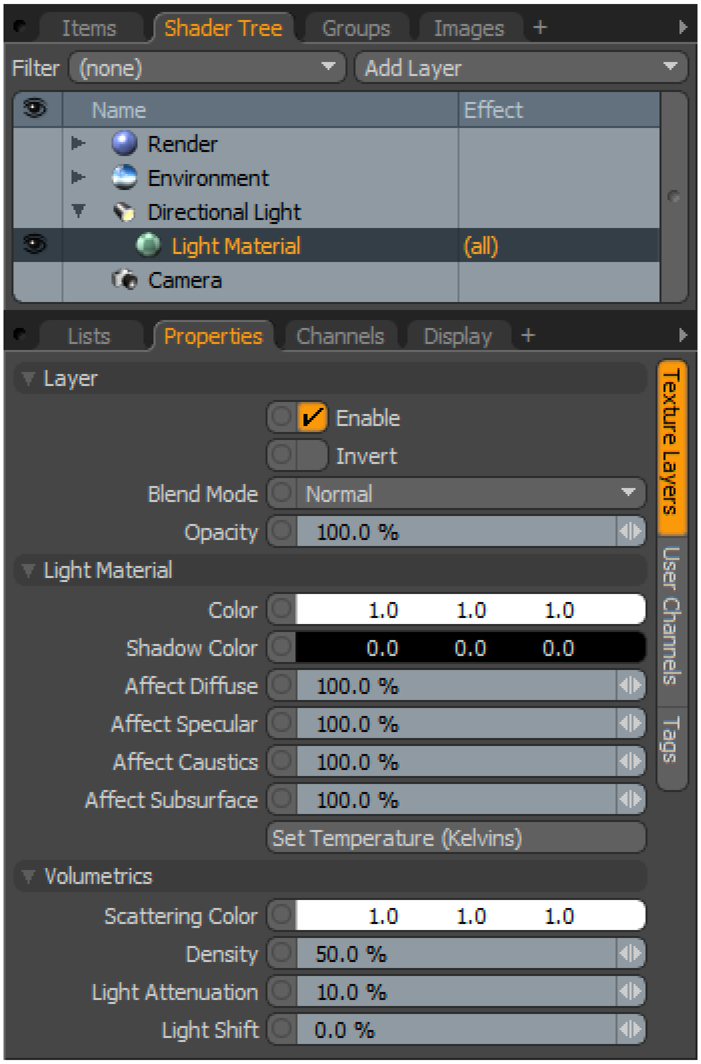

Options
Effect: (Shadi...
Rendering...
LUT: sRGB
(Render Camera)
Shading: Full
Rendering... 52.82%
00:00:05
3D View
UV Texture View
Render Preset Bro ...
Gradient Editor
Schematic
Camera
Default
Ray GL: Off
Cylinder
Polygons : Face
Channels: 3
Deformers: ON
GL: 19,244
1 m
Geometry Snap : Transform
Render Camera
0
24
48
72
96
120
0
120
120

Light Material

Lights derive settings for their illumination color, shadow color and how much they contribute to diffuse and specular shading from the light material item, as well as their volumetric settings as appropriate to the light.

While lights are located within the Items (Scenes) List, the light material item is located inside the Shader Tree; outside of the camera which has no material settings, Lights are the only items the have a dual identity in Modo existing in both places. The Light Material item can be found by clicking the small arrow widget next to the light item's name to reveal/hide the material item. Selecting the item displays its properties within the **Properties** viewport. All lights have a set of default values that can be easily modified, providing an intuitive and flexible method for customizing a light's effect. Additional settings related to the various light items can be found within the light's properties.

Also, keep in mind that since lights are **Shader Tree** items, they can have additional **Shader Tree** item layers added to them; images can be added to produce slide projector types of effects (by adjusting the settings of the Texture Locator), and by right-clicking on the item's **Effect** column, different options can be selected producing a variety of different and interesting effects. For more information on light effects, see the Effect - Light Item topic.


Items
Shader Tree
Groups
Images
+
Filter
(none)
Add Layer
Name
Effect
Render
Environment
Directional Light
Light Material
(all)
Camera
Lists
Properties
Channels
Display
+
Texture Layers
User Channels
Tags
Layer
Light Material
Volumetrics
Set Temperature (Kelvins)

☑ Enable
☐ Invert
Blend Mode: Normal
Opacity: 100.0 %
Color: 1.0 1.0 1.0
Shadow Color: 0.0 0.0 0.0
Affect Diffuse: 100.0 %
Affect Specular: 100.0 %
Affect Caustics: 100.0 %
Affect Subsurface: 100.0 %
Scattering Color: 1.0 1.0 1.0
Density: 50.0 %
Light Attenuation: 10.0 %
Light Shift: 0.0 %

Layer

Enable: Toggles the layer on and off. If there is only one material attributed to the light, this disables/enables the light itself.

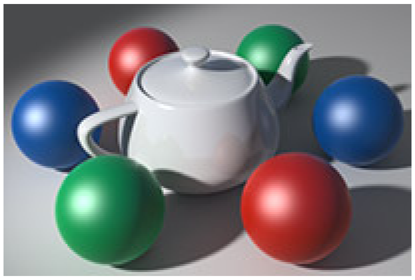
Invert: Inverts the RGB values for the lights material layer.

Blend Mode: Affects blending between material layers applied to the light allowing you to stack several layers for different effects. For more on blending, see Layer Blend Modes.

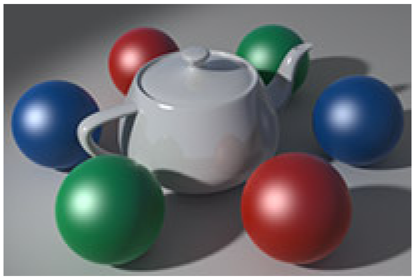
Opacity: Changes the transparency of the current material layer. If only one material is associated with your light, this also dims the light itself.

Light Material

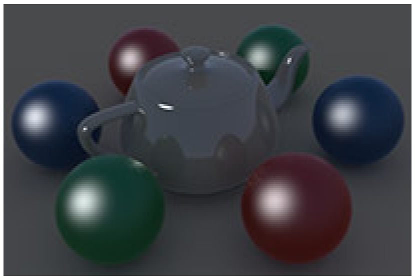
Color: The color control uses the standard Modo color component of three RGB values and provides direct control over the color of light emitted. Our brains are used to seeing pure white lights, but often times, there are many reasons to change the color for artistic affect. Also, when simulating outdoor lighting, setting a lights color to cooler values such as blue-ish tones tends to simulate mid to late day-time lighting where as warmer colors such as yellow or orange-tones is more like evening time.
Shadow Color: This setting tints the shadows produced by the associated light. In most cases this color is left as black, producing the 'absense-of-light' shadows we are all used to seeing, but other times, there can be creative reasons to modify the shadow color to either simulate a specific phenomenon, or simply for artistic effect. Some consider it good practice to tint the shadow with a complementary color to that used for the light's **Color** setting.



Affect Diffuse value of 100%

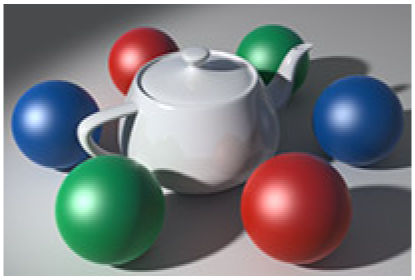


Affect Diffuse value of 66%

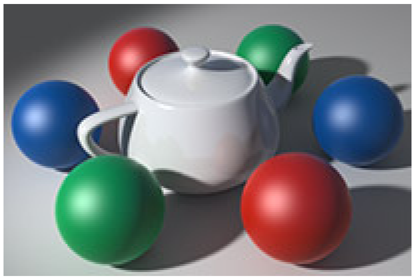


Affect Diffuse value of 0%

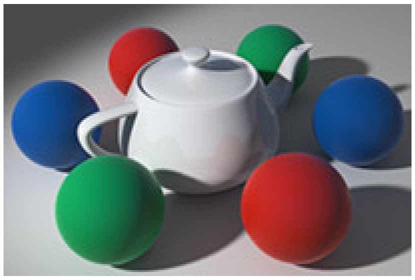
Affect Diffuse: The diffuse lighting is the light illuminating a surface that is only affected by position and angle of the light source, but is unaffected by the viewers position or angle. Light that behaves this way is described as being isotropic. In simpler terms, this is the normal shading you see over a surface. This percentage value provides the ability to control how much the light's illumination contributes to the overall diffuse shading within a scene on a per light basis. You can set this value to 0% if you want to disable the diffuse shading, allowing the creation of a "specular only" light source.



Affect Specular value of 100%

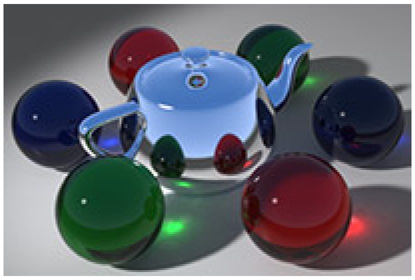


Affect Specular value of 66%

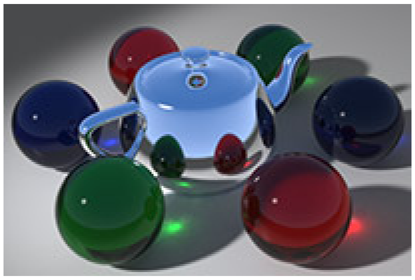


Affect Specular value of 0%

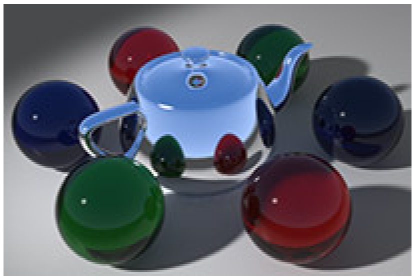
Affect Specular: The specular lighting is light that is dependent on the viewer's position and angle. Specular is meant to simulate the reflection of a bright light source upon a surface and is calculated as an angle vector between the camera and the light source itself. This percentage value provides the ability to control how much the light contributes to any specular shading within a scene on a per light basis. You can set this value to 0% if you want to eliminate specular shading produced by the light, allowing the creation of a "diffuse only" light source.



Affect Caustic value of 100%

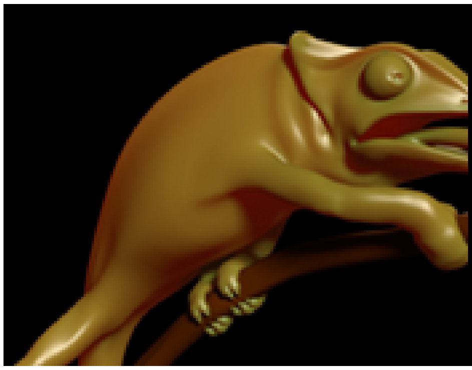


Affect Caustic value of 66%

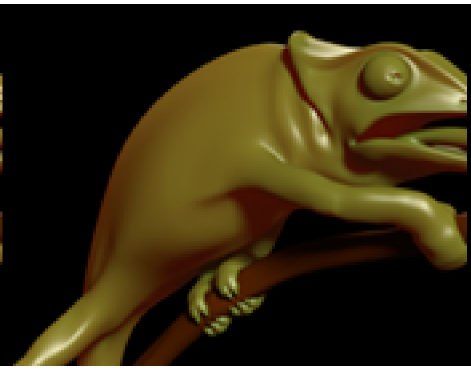


Affect Caustic value of 0%

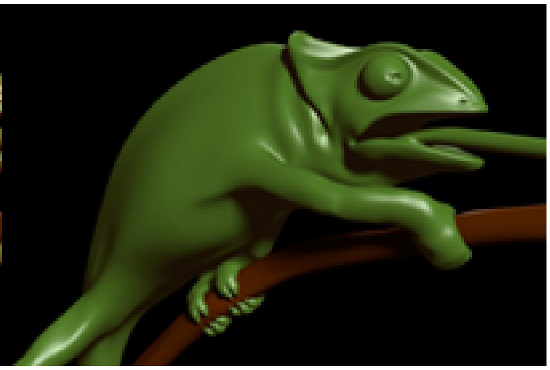
Affect Caustic: Caustics are the refraction or reflection of light, either through a transparent object or off of a reflective object and projected onto a surface. This percentage value provides the ability to control how much the lights illumination contributes to the overall caustic shading within a scene on a per light basis. You can set this value to 0% if you want to disable the light from contributing to generated caustics.



Affect Subsurface value of 100%



Affect Subsurface value of 66%



Affect Subsurface value of 0%

Affect Subsurface: Subsurface scattering is the appearance of light entering a translucent surface, diffusing (bouncing around) and then exiting. Typically when this happens, the exiting light is tinted by the interior of the surface producing a unique effect. The 'Affect Subsurface' percentage value provides the ability to control how much the lights illumination contributes to the overall subsurface scattering within a scene on a per light basis. You can set this value to 0% if you want to disable the light from contributing to subsurface scattering. Values beyond 100% boost the lights affect on the subsurface amount.

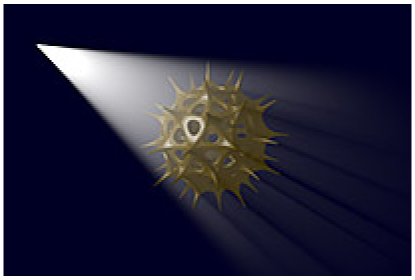
Set Temperature: This setting allows you to set a light's color based on the familiar kelvin scale for color temperature.

Some common values for K	
1500 K	Candlelight
2680 K	40 W incandescent lamp
3000 K	200 W incandescent lamp
3200 K	Sunrise/sunset
3400 K	Tungsten lamp
3400 K	1 hour from dusk/dawn
5000-4500 K	Xenon lamp/light arc
5500 K	Sunny daylight around noon
5500-5600 K	Electronic photo flash
6500-7500 K	Overcast sky
9000-12000 K	Blue sky

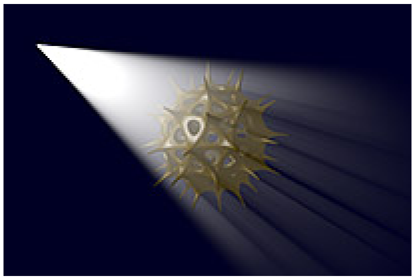
Volumetrics

When light shines on microscopic particles suspended in the air, such as water vapor, smoke or even pollution, the light rays become visible. In computer graphics this phenomenon is known as a volumetric light, their effect can be subtle or pronounced, either way it adds a good amount of believability and atmosphere to rendered images. For volumetrics to work, they must first be enabled in the light item's properties where additional settings specific to the volume itself are located.

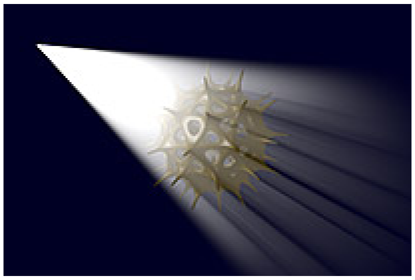
Scattering Color: This allows you to set an RGB color value for the light volume itself. You usually want this to be the same color as the light, however, there may be some creative reasons one might want to change it to something else entirely.



Density value of 25%

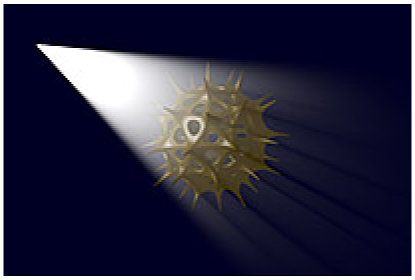


Density value of 50%

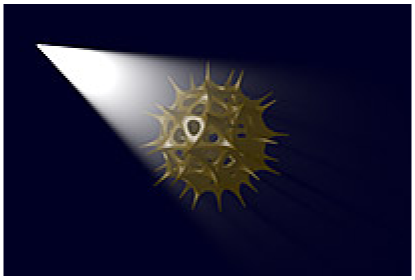


Density value of 100%

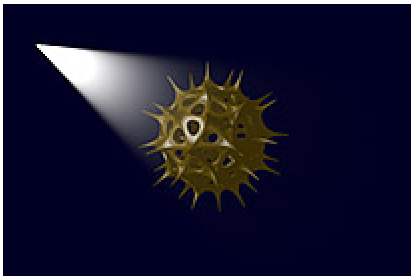
Density: This value sets the density or thickness for the volumetric effect, higher values make the volume thicker and more opaque and lower values make it appear more sparse and transparent.



Attenuation value of 25%



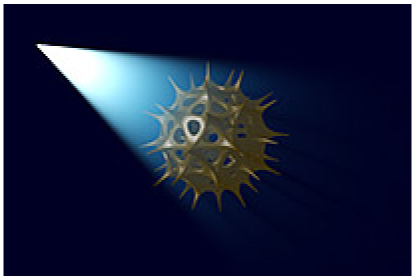
Attenuation value of 50%



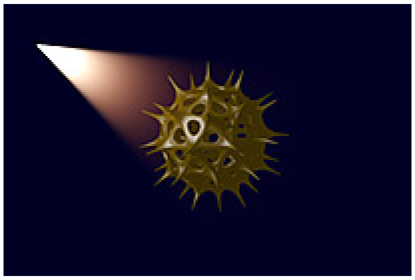
Attenuation value of 100%

Light Attenuation: Attenuation is the gradual loss of intensity when light is passing through any medium. With an attenuation setting of 0%, Modo produces a smooth gradual falloff along its length. as the **Light Attenuation** value increases, the volumetric effect gradually fades in overall intensity with a bias toward the light source.

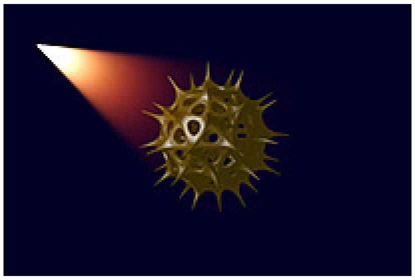
Tip: Combine with **Light Shift** to mimic large volumes of light scattering.



Light Shift value of -100%



Light Shift value of 50%



Light Shift value of 100%

Light Shift: As the light is increasingly attenuated, certain wavelengths are absorbed or scattered. The **Light Shift** function simulates this phenomenon. Negative values push the color of the volume increasingly toward cooler light blues, while positive values push the color toward warmer orange and red hues. The **Light Attenuation** value must be a number above 0% to see any effect

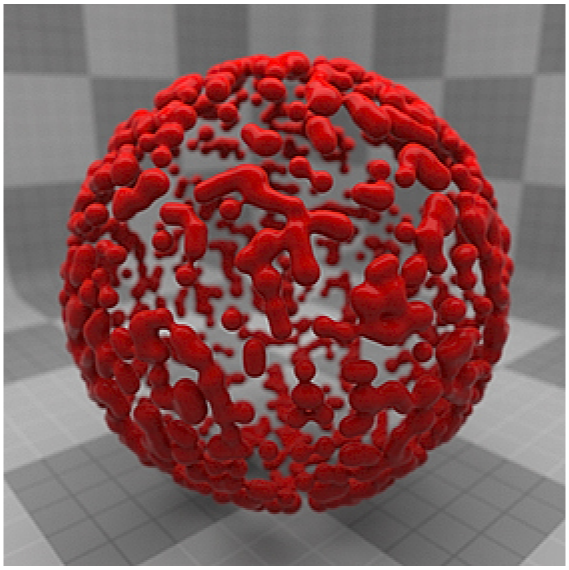
Volumes

This section provides details on how to use Blobs, Sprites, Volumes and Render Booleans in Modo.

Blobs

The **Blob** item provides a means to procedurally create blobby meta-ball like shapes; combinations of spherical forms with a smooth, interpolated curved surface covering them. The positions of individual blob shapes are determined by the vertices of a **Particle Source** item, where each vertex in the **Particle Source** produces a single blob (see **Particle Source** under Geometry for more information).

The form itself is dependent on the radius of the Blob particles and the distance between adjoining vertices, as the surfaces get closer together, they tend to join their surfaces (think like water droplets when they get close to each other). The resulting surface can be further controlled with the addition of textures added through the Shader Tree, producing complex surfaces very easily.



Adding Blobs

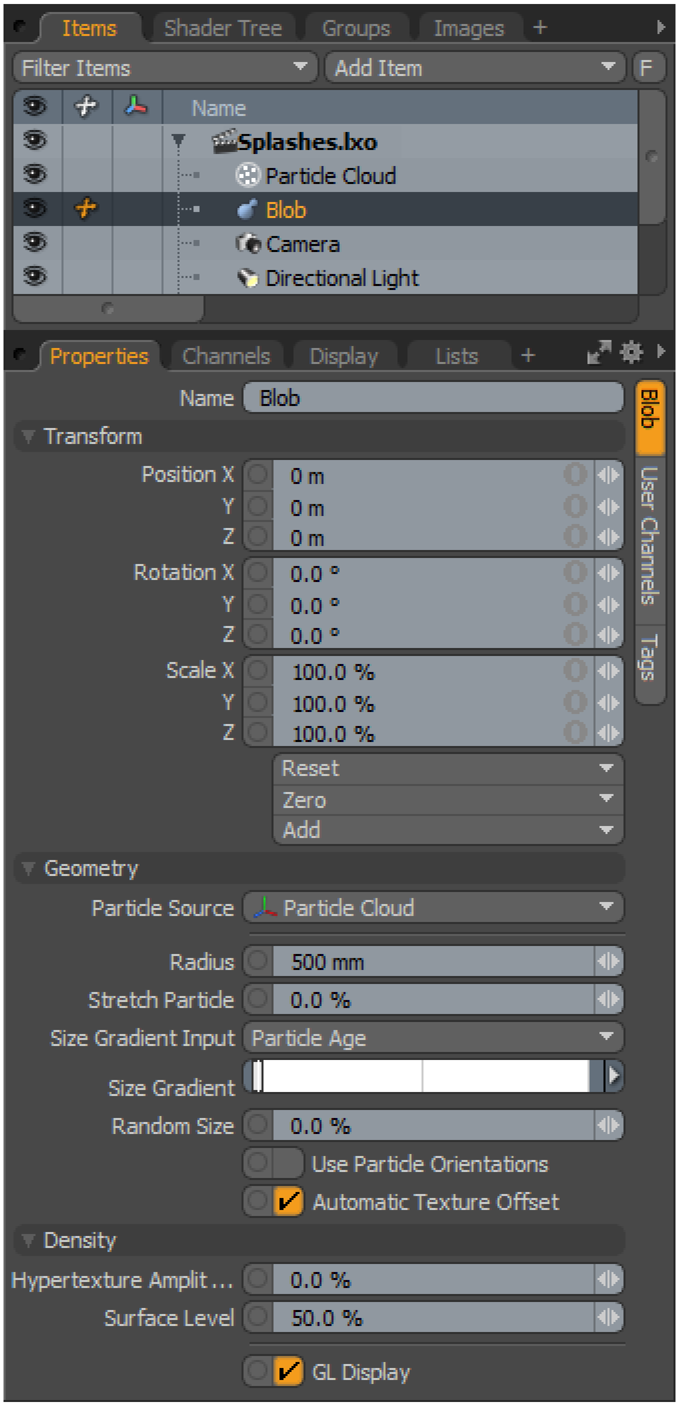
You can add a Blob item by clicking the **Add Item** function found in the Items (Scenes) List. Under **Volumes**, double-click **Blob**. Once added, the item appears at the bottom of the **Item List** viewport. When selected, a number of attributes controlling the shape appear in the Properties viewport.

Applying Materials

Normally, surfaces are applied through the use of polygon tags that define masks in the Shader Tree controlling material placement. Blobs, however, are a procedural surface created at render time and have no user selectable polygons to apply a tag to, so surfacing is applied to the **Blob** item through the use of an **Item Mask**.

You can quickly create an **Item Mask** by right-clicking on the **Blob** item in the **Items list** and choosing the **Create Item Mask** option from the contextual menu. You can then add **Materials** and other Shader Tree items, setting the appropriate **Effect** for each layer to build up surfacing attributes, just as you would for any other surface.

You can also control additional attributes of the **Particle Source** that are not surface specific, such as **Particle Size** and **Particle Density**, by creating an additional **Item Mask** and applying texture layers set to the appropriate **Effect** under the **FX Item** of the Shader Tree. For more information, see FX Item.


Items
Shader Tree
Groups
Images
+
Filter Items
Add Item
F
Name
Splashes.lxo
Particle Cloud
Blob
Camera
Directional Light
Properties
Channels
Display
Lists
+
Blob
User Channels
Tags
Transform
Reset
Zero
Add
Geometry
Density

Name: Blob

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Particle Source: Particle Cloud
Radius: 500 mm
Stretch Particle: 0.0 %
Size Gradient Input: Particle Age
Size Gradient
Random Size: 0.0 %
☐ Use Particle Orientations
☑ Automatic Texture Offset

Hypertexture Amplit…: 0.0 %
Surface Level: 50.0 %
☑ GL Display

Name - This data field displays the current item name. You can easily change it by left-clicking within the field and typing the new name.

Transform

Position - An Item transform that allows you to numerically position the item in XYZ space. By default, **Position** transforms originates from the Center position.

Rotation - An Item transform that allows you to numerically set the rotation of the item. By default, **Rotation** transforms originate from the Center position.

Order - Allows you to set the order that rotations are applied to the camera item. Changing the order in which rotations are applied can sometimes help to reduce, or eliminate, gimbal lock.

Note: The **Order** control is only displayed after you have selected a **Rotation**.

Scale - An Item transform that allows you to numerically set the size of the item. By default, **Scale** transforms originate from the Center position.

Reset - Click to reset the selected transform values to (0,0,0), returning the items back to their default state.

Zero - Click to reset the selected transform property values to 0, leaving the Center position and Mesh position intact.

Add - Transform Items are the channel groups associated to an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them, even though they are visible within the **Properties** panel. This is useful as an optimization, as the necessary transforms are added on an as-needed basis, reducing scene overhead.

There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values input fields). This action automatically adds the selected transform item to the **Channels** viewport list. The **Add** function can also be used to add the selected set of transforms to the **Channel** list, while keeping the default 0,0,0 values (a necessary step for **Referencing** - in order to override the channels, they must first exist).

Geometry

Particle Source - When a **Sprite** item is added to a scene, it is represented by a single locator type point at the World Origin, producing a single sprite image. To produce additional sprites, define a **Particle Source**, which can be any Mesh Item layer that contains vertices, where each vertex position is used to determine the position of the sprite particle. Any type of geometry can be used, such as single point polygons, vertices, two-point polygons, and regular polygons, however, only the vertex positions of any component are used.

Source layers can be normal geometric forms, point clouds that are imported or created using the Particle Cloud item, or any of the Particle Tools. They can be imported particle simulations controlled by way of an MDD deformer, RealFlow Particles, or an Alembic format data file.

Radius - This option determines the base size of the sprite image. From here, the size can be altered using the **Random Size** option, the addition of a Particle Size Vertex Map that allows you to paint the scale for individual particles interactively, or with the application of a texture layer, under the FX Item of the **Shader Tree**, set to control **Particle Size**.

Stretch Particles - This percentage controls the scale of particle stretching based on velocity. Values are additive to the base value, so a **Stretch Particle** value of 100% produces a particle that is stretched to twice its length (200% altogether).

Size Gradient Input - This control provides a means to define an input parameter in which to apply the **Size Gradient**, determining the scaling of the resulting volume, based on the chosen parameter. You can choose from a number of input parameters specific to **Volumes** and **Particles**, such as **Texture Value** (Density), **Distance**, and **Particle Velocity**.

Size Gradient - This is gradient slider allowing you to refine gradient value keys without having to leave the basic Modo interface. In much the same way as keyframes store transform values for items, gradient keys store values and softly attenuate between the defined key positions.

You can add keys by middle-clicking along the gradient. Keys are then adjusted by left-clicking and dragging to a new position. The gradient bar directly underneath provides a visual reference to the values being generated by the gradient itself. The defined values act as a multiplier of the final volume size on the preceding **Size Gradient Input** option.

Random Size - This option controls the amount of randomness added to the scale of each individual particle blob. The amount is determined by a scale percentage, so a setting of 25% adjusts the scale of the base radius size by an amount of 0%-25% randomly across all particles, 50% adjusts the size by 0%-50% randomly across all particles, and so on.

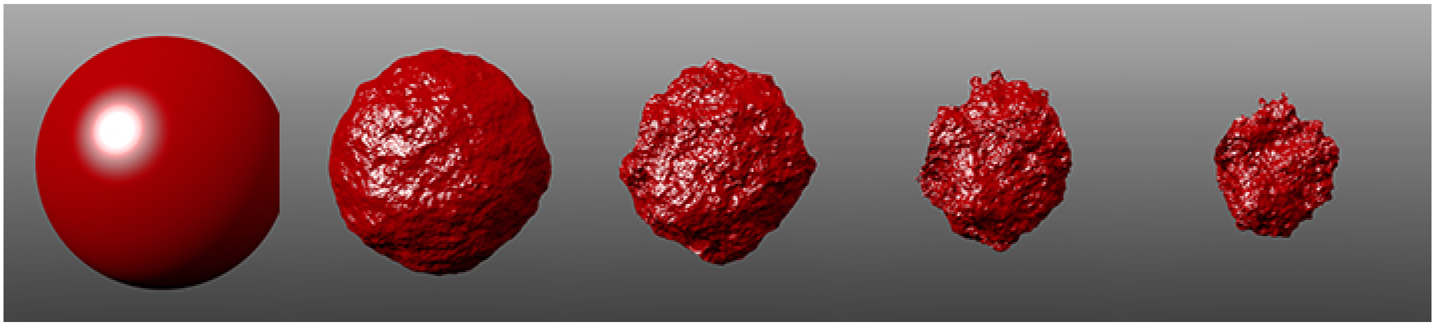
Use Particle Orientations - When enabled, aligns the particle to the direction vector from the particle source. The direction vector is usually the surface normal for surface particles, or the particle velocity vector for simulation particles (such as from RealFlow).

Note: In the case of blobs, this option is disabled by default as it adds to render times (an increase of ~30%), so don't use it if you don't need it.

Automatic Texture Offset - When enabled, adds a different offset to the hypertexture for each particle, so that individual particles look different.

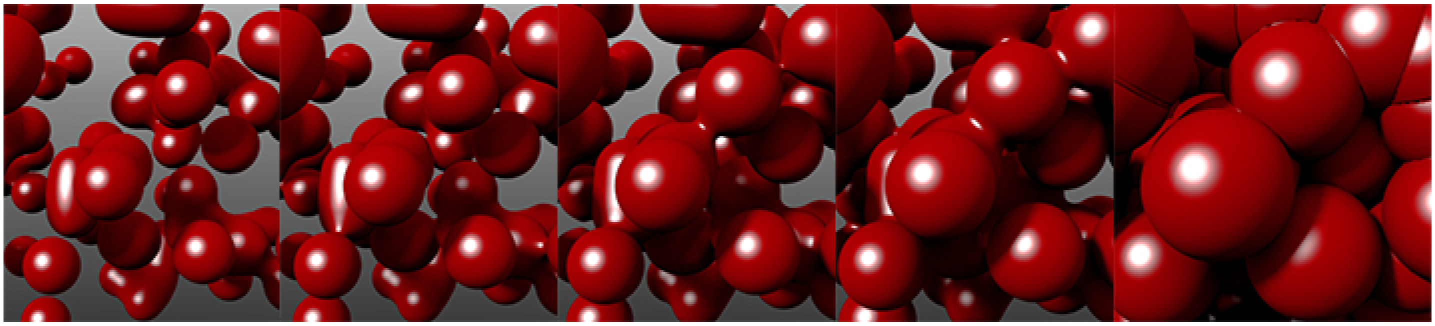
Density

Hypertexture Amplitude - Determines the amplitude (intensity) of a **Volume Density** texture layer applied to the blob volume. The higher the value, the more you see the effect of the density texture. It's similar in concept to displacement amplitude, but applies to volumes.



Amplitude 0% Amplitude 25% Amplitude 50% Amplitude 75% Amplitude 100%

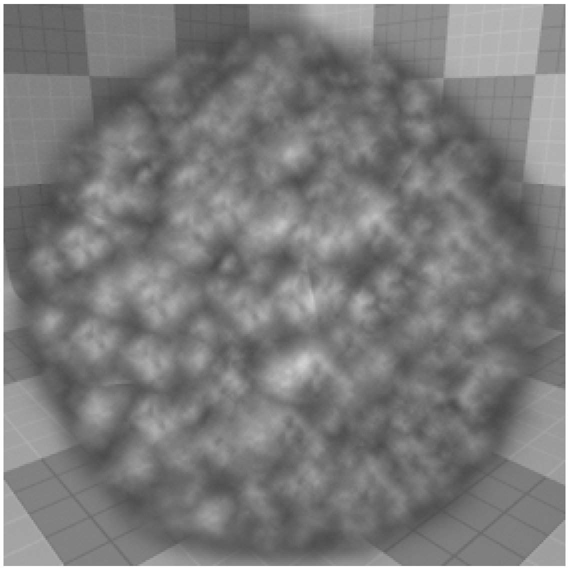
Surface Level - The surface level option determines the height of the limit surface, which in-turn controls how smooth or abrupt the transitions are between neighboring blobs. Low values produce smoother blobs that are slightly smaller, and higher values produce more spherical shapes that are slightly larger. At 100%, individual blobs no longer blend, but look like a collection of spheres instead.



Surface Level 0% Surface Level 25% Surface Level 50% Surface Level 75% Surface Level 100%

GL Display - This option toggles the display of the **Blob** proxy in the 3D OpenGL viewport. When enabled, you can see an approximation of how the blob particles render (the default state), but in heavy scenes you may find it useful to disable this option to increase scene interactivity or reduce overall visual clutter.

Sprites



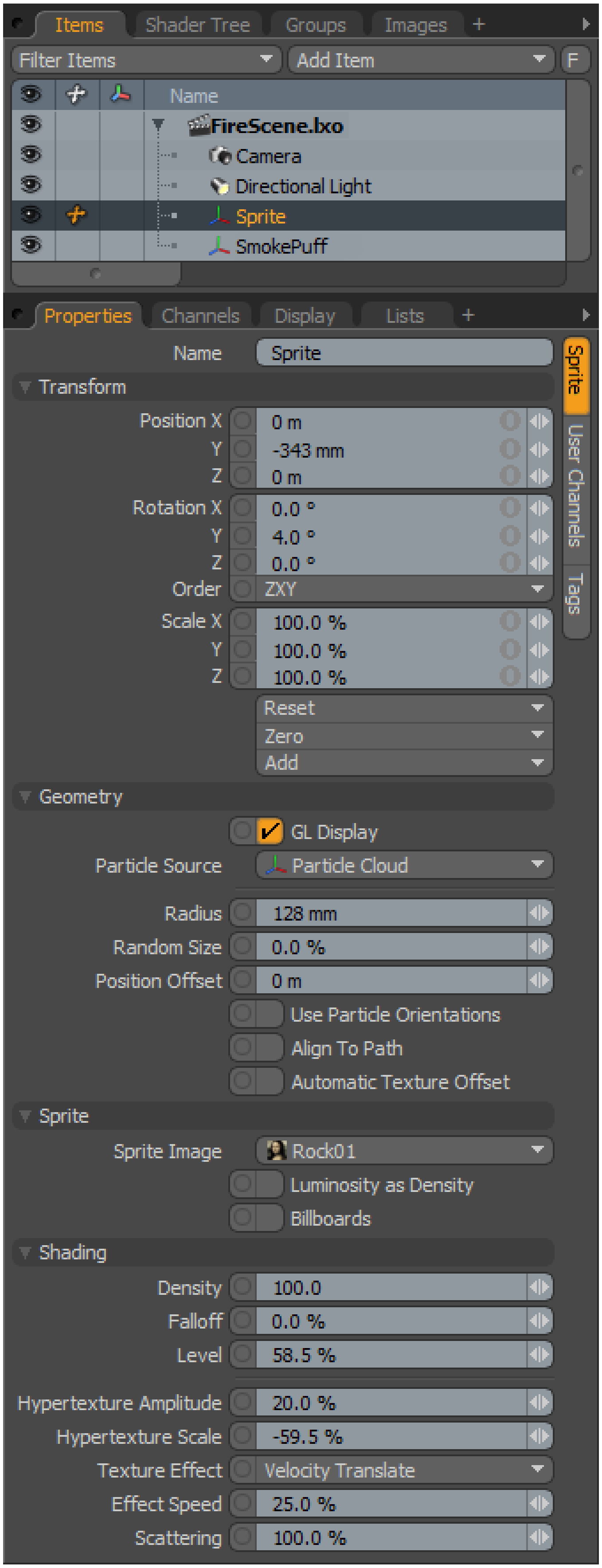
The sprite item provides useful functionality when using bitmap images to generate volume-like effects that render much more quickly than standard volume effects.

Adding Sprites

You can add a Sprite item in the **Items** list by clicking **Add Item** > **Volumes**, then double-clicking **Sprite**. Once added, the item appears at the bottom of the **Items** list. When selected, a number of attributes controlling the sprite item's appearance appear in the Properties Panel, this is where you specify the actual sprite texture and point source options.

Applying Materials

Normally, surfaces are applied using polygon tags that define masks in the **Shader Tree**, controlling material placement. However, sprites are procedural constructs created at render time and have no selectable polygons to apply a tag to, so surfacing is applied to the sprite item through the use of an item mask. You can quickly create an item mask by right-clicking on the sprite item in the **Items** list and choosing the **Create Item Mask** option from the contextual menu. You can then add materials and other **Shader Tree** items, setting the appropriate effect for each layer building up surfacing attributes just as you would for any other surface. The material's **Volumetric Scattering** and **Absorption** options are the main controlling factors in its appearance. You can also control additional attributes of the Particle Source that are not surface specific, this can be done by creating an additional item mask and applying texture layers set to the appropriate effect using the FX item of the Shader Tree. For more on using FX, see FX Item.


Items
Shader Tree
Groups
Images
Filter Items
Add Item
F
Name
FireScene.lxo
Camera
Directional Light
Sprite
SmokePuff
Properties
Channels
Display
Lists
Sprite
User Channels
Tags

Name: Sprite

Transform

Position X: 0 m
Y: -343 mm
Z: 0 m
Rotation X: 0.0 °
Y: 4.0 °
Z: 0.0 °
Order: ZXY
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %
Reset
Zero
Add

Geometry

☑ GL Display
Particle Source: Particle Cloud
Radius: 128 mm
Random Size: 0.0 %
Position Offset: 0 m
☐ Use Particle Orientations
☐ Align To Path
☐ Automatic Texture Offset

Sprite

Sprite Image: Rock01
☐ Luminosity as Density
☐ Billboards

Shading

Density: 100.0
Falloff: 0.0 %
Level: 58.5 %
Hypertexture Amplitude: 20.0 %
Hypertexture Scale: -59.5 %
Texture Effect: Velocity Translate
Effect Speed: 25.0 %
Scattering: 100.0 %

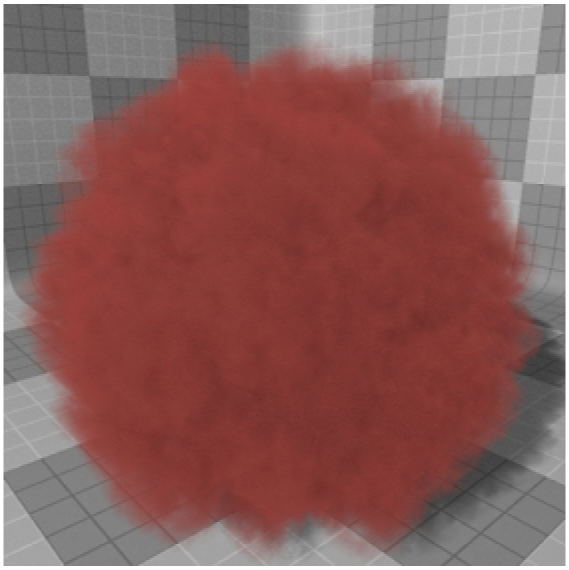
Option	Description
Name	Displays the current item name. You can change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space. By default, **Position** transforms originate from the center position.
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, **Rotation** transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and mesh position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values' input fields). This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the **Channel** list while keeping the default **0,0,0** values (a necessary step for referencing, in order to override the channels, they must first exist).
Geometry	
GL Display	Toggles the display of the sprite proxy in the 3D OpenGL viewport. Generally, this should be enabled to get an approximation of how the sprite particles render (the

Option	Description
	default state), but in heavy scenes it may be useful to disable this option to increase scene interactivity.
Particle Source	When a sprite item is created, it is represented by a single locator-type point at the World Origin, producing a single sprite image. To produce additional sprites or sprites as a volume, you need to define a particle source, which can be any item layer that contains vertices, where each vertex position is used to determine the position of the sprite particle (any type of geometry can be used- single point polygons, vertices, two-point polygons and regular polygons, however, only the vertex positions of any component are used). Source layers can be normal geometric forms, point clouds that are imported or created using the particle cloud item or any of the particle tools. They can be imported particle simulations controlled by way of a MDD deformer, RealFlow point cache or an Alembic format data file.
Radius	Specifies the base size of the blob. From here, the size can be altered with the **Random Size** option, the addition of a particle size Vertex Map that allows you to paint the scale for individual particles interactively, or with the application of a texture layer, under the FX item of the **Shader Tree**, set to control **Particle Size**.
Random Size	Controls the amount of randomness added to the scale of each individual particle sprite. The amount is determined by a scale percentage - a setting of 25% adjusts the scale of the base radius size by an amount of 0%-25% randomly across all particles, a setting of 50% adjusts the size by 0%-50% randomly across all particles and so on.
Position Offset	Adds an offset to the sprite position along the particle normal direction. This is mostly useful for surface particles when you want to detach them from the surface they originate from.
Use Particle Orientations	Aligns the particle to the direction vector from the particle source. The direction vector is usually the surface normal for surface particles or the particle velocity vector for simulation particles (like from RealFlow).
Align to Path	Aligns the particle to its local motion vector.
Automatic Texture Offset	Adds an offset to the hypertexture different for each particle so that each particle looks different.

Option	Description
Sprite	
Sprite Image	Determines the image used for the sprite particle. Load an image into the **Image List** to select it here. Multiple images may also be used when defined as a group. For more on creating groups, see the Groups Viewport topic.
Luminosity as Density	When enabled, the luminosity of the image is used to determine the density of individual sprite particles.
Billboards	When enabled, the assigned sprite image acts like individual single polygons with the image mapped to the surface that always face toward the camera. **Note:** Under **Shading**, set **Density** to 100% and **Falloff** to 0% for the full image effect.
Shading	
Density	Specifies the thickness of the individual sprite particles, controlling the apparent amount of particular matter that is visible in the overall volume.
Falloff	Specifies the attenuation of the individual sprite particles from their center point out toward their edges.
Level	Changes the level in the hypertexture.
Hypertexture Amplitude	Determines the amplitude (intensity) of a volume density texture layer applied to the sprite volume. The higher the value the more you see the effect of the density texture. It's similar in concept to displacement amplitude, but applies to volumes.
Hypertexture Scale	Changes the size of the hypertexture, it is the same things as changing the texture locator scale settings.
Texture Effect	Hypertextures can be animated on particles to create billowing or dissipation effects. The goal of these effects is to add secondary animation to the particles, which is important for explosions and smoke. The various texture effects are: • **None**: Disables the texture effect. • **Velocity Translate**: Moves the texture in the direction of the velocity vector. • **Expand**: The texture is scaled over time to give the impression that it expands. • **Expand and Dissolve**: Like the previous effect, but with the addition of a dissolve

Option	Description
	effect so as the texture expands it also dissolves. • **Pyroclastic**: A boiling motion where hot gas swirls around within the form adding a realistic secondary motion to the volume. Good for smoke.
Effect Speed	Controls the speed at which the texture effect animates.
Scattering	The overall scattering amount on the sprite volume, this is similar to the volume scattering amount in the material.

Volumes



The volume item is a very flexible, open-ended function that provides a means to create many unique and interesting forms that would be impossible to create otherwise, such as clouds, smoke, vapor clouds and dust clouds. They can also be further enhanced (and controlled) by Modo's numerous 3D procedural textures. There are several ways to create the volume effect, one being the single spherical volume that is created by the locator representation in the 3D scene, once the volume item is added. When using the **Geometry** option, the volume is confined to the inside of any completely closed, arbitrary geometric shape (closed meaning it is watertight with no holes). When using the **Particle Source** option, the source element's vertex positions determine multiple cloned volumes that combine to make the final overall shape (often overlapping). The shape of the volume particles are then controlled by the settings of the volume item.

Adding Volumes

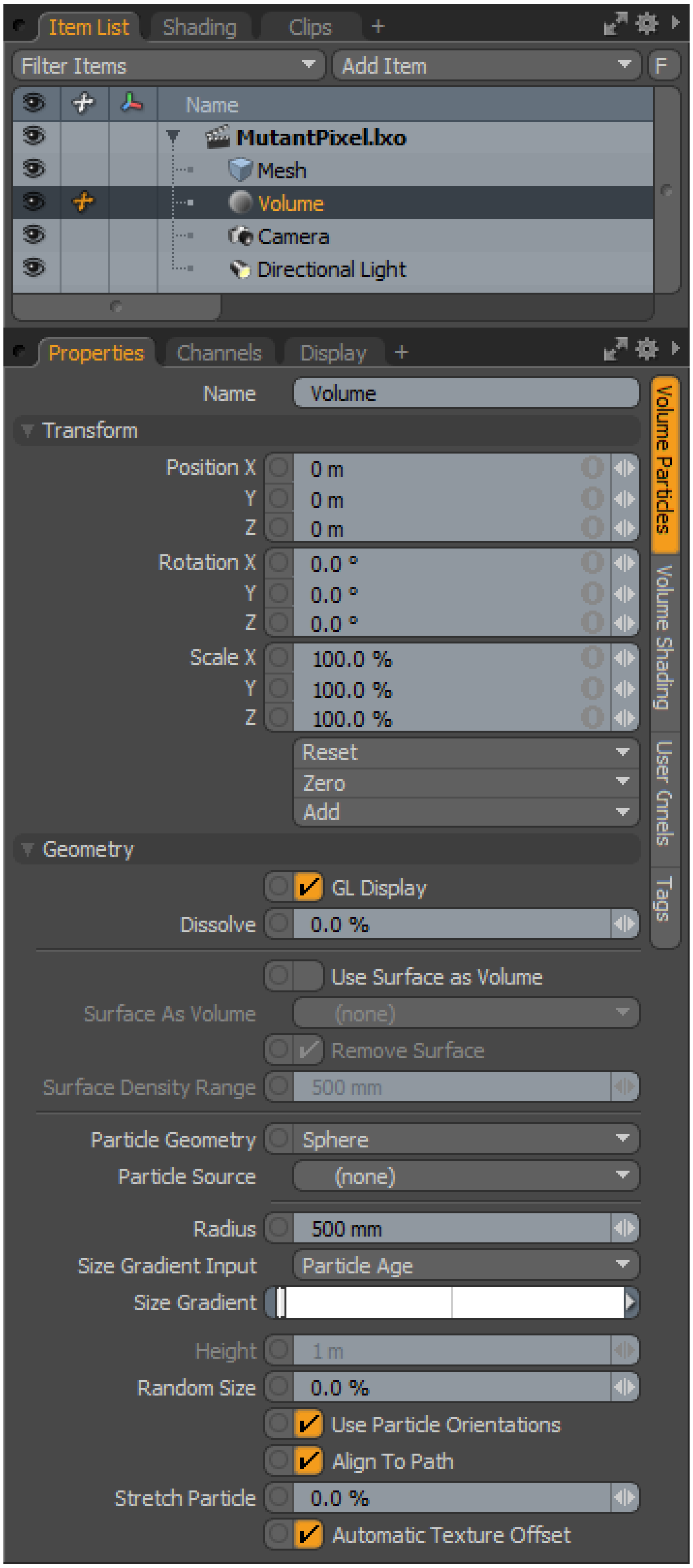
You can add a volume item to a scene in the **Items** list, by clicking **Add Item** > **Volumes**, then double-clicking **Volume**, creating the item in the scene at the origin. Once added, the item layer appears at the bottom of the **Items** list. When selected, a number of attributes controlling the volumetric form appear in the Properties Panel. This is where you can also define a **Surface as Volume** item, or a point source. Either can be a good way to control the overall shape of the rendered volume. When neither is defined, the volume simply renders as a spherical, cubic or cylindrical form, based on the **Radius** value of the **Geometry** section (where its base shape can also be defined).

Applying Materials

Normally, surfaces are applied through the use of polygon tags that define masks in the **Shader Tree** controlling material placement, however, volumes are procedural constructs created at render time and have no user-selectable polygons to apply a tag to, so surfacing is applied to the volume item itself through the use of an item mask. This type of mask links shading properties directly with an item layer. You can quickly create an item mask by right-clicking on the volume item in the **Items** list and choosing the **Create Item Mask** option from the contextual menu. This adds a properly defined Material Group item of the same name as the volume item to the **Shader Tree**, where you can then add materials and other **Shader Tree** items, setting the appropriate effect for each layer, building up surfacing attributes just like for any other surface. You can control additional attributes of the particle source that are not surface-specific, such as density, size and dissolve amount. This can be done by creating an additional item mask and applying texture layers set to the appropriate effect type using the FX Item of the **Shader Tree**.

Note: When a Mesh Item is defined in the **Surface as Volume** option, the resulting volume then picks up its surfacing from the target Mesh Item, so any surfacing meant for the volume needs to be applied in the material group specific to the geometry that is defined as the **Surface as Volume**.

Volume Particles


Item List
Shading
Clips
Filter Items
Add Item
F
Name
Properties
Channels
Display
Volume Particles
Volume Shading
User Chnels
Tags
Transform
Geometry

MutantPixel.lxo
Mesh
Volume
Camera
Directional Light

Name: Volume

Position X: 0 m
Y: 0 m
Z: 0 m
Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °
Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %
Reset
Zero
Add

☑ GL Display
Dissolve: 0.0 %
☐ Use Surface as Volume
Surface As Volume: (none)
☑ Remove Surface
Surface Density Range: 500 mm
Particle Geometry: Sphere
Particle Source: (none)
Radius: 500 mm
Size Gradient Input: Particle Age
Size Gradient
Height: 1 m
Random Size: 0.0 %
☑ Use Particle Orientations
☑ Align To Path
Stretch Particle: 0.0 %
☑ Automatic Texture Offset

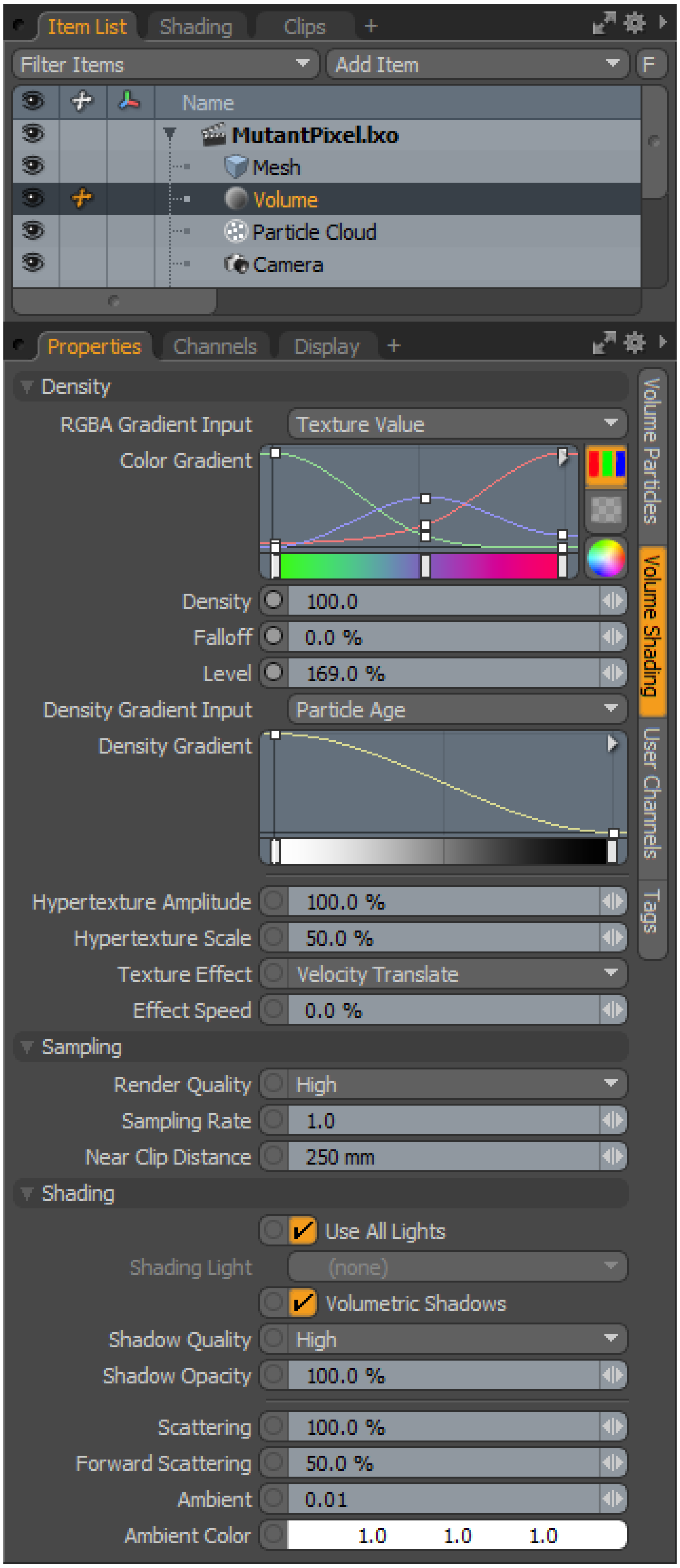
Option	Description
Name	Displays the current item name. You can change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space. By default, **Position** transforms originate from the center position.
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, **Rotation** transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and mesh position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values' input fields). This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the **Channel** list while keeping the default **0,0,0** values (a necessary step for referencing, in order to override the channels, they must first exist).
Geometry	
GL Display	Toggles the display of the volume proxy in the 3D OpenGL viewport. Generally, this should be enabled to get an approximation of how the volume renders (the default

Option	Description
	state), but in heavy scenes it may be useful to disable this option to increase scene interactivity.
Particle Source	When a sprite item is created, it is represented by a single locator-type point at the World Origin, producing a single sprite image. To produce additional sprites or sprites as a volume, you need to define a particle source, which can be any item layer that contains vertices, where each vertex position is used to determine the position of the blob particle (any type of geometry can be used- single point polygons, vertices, two-point polygons and regular polygons, however, only the vertex positions of any component are used). Source layers can be normal geometric forms, point clouds that are imported or created using the particle cloud item or any of the particle tools. They can be imported particle simulations controlled by way of a MDD deformer, RealFlow point cache or an Alembic format data file.
Use Surface as Volume	When enabled, you can specify an arbitrary mesh layer that is used to contain the volume effect, giving you ultimate control over the shape of the resulting rendered volume. The target Mesh Item needs to be one that is completely closed. Mesh volumes with open seams or non-manifold surfaces may produce undesirable results. To test an object for holes, in **Polygons Selection mode** under one of the **Modeling** tabs, press and hold the **Ctrl** key, changing the **Edges** button into one that says **Boundary** and click on it. Any highlighted edges represent holes within the mesh's surface.
Surface as Volume	Determines the particular mesh layer that is used for the volume effect when **Use Surface as Volume** is enabled. For proper results, target Mesh Item must be fully enclosed.
Remove Surface	When this option is enabled (the default behavior), the specified **Surface as Volume** mesh layer is hidden at render time, and only show the resulting volume effect.
Surface Density Range	Specifies the amount of attenuation from the surface of the volume inward.
Particle Geometry	When the **Use Surface as Volume** option is disabled, you can specify a **Particle Geometry** element. This creates a procedural geometric-shaped volume around each vertex of a particle source (or the volume locator itself when none is defined). Particles can be few or many, but the combination of the particle positions and the

Option	Description
	particle geometry shape determines the overall volume effect at render time. The possible shapes include sphere, cube, cylinder and cumulus, a half dome shape that, when textured, looks similar to the familiar, puffy cumulus clouds. **Note:** It is recommended to only have as many particles as are necessary to determining the final form, as having many overlapping Volumes can slow rendering calculations considerably.
Particle Source	When a volume item is created, it is represented by a single locator-type point at the World Origin, producing a single volume image. To produce additional particle geometry volumes, you need to define a particle source, which can be any item layer that contains vertices, where each vertex position is used to determine the position of the sprite particle (any type of geometry can be used- single point polygons, vertices, two-point polygons and regular polygons, however, only the vertex positions of any component are used). Source layers can be normal geometric forms, point clouds that are imported or created using the particle cloud item or any of the particle tools. They can be imported particle simulations controlled by way of a MDD deformer, RealFlow point cache or an Alembic format data file.
Radius/Height	Defines the outside dimensions of the particle geometry object. **Radius** refers to the overall dimensions of the sphere and cube shapes, and **Radius** and **Height** work together to define the dimensions of a cylinder. From there, the size can be altered via the 'Random Size' option, the addition of a particle size Vertex Map that allows you to paint the scale for individual particles interactively, or with the application of a texture layer, under the FX Item of the **Shader Tree**, set to control **Particle Size**.
Size Gradient Input	Provides a means to define an input parameter in which to apply the following **Size Gradient**, determining the scaling of the resulting volume based on the chosen parameter. You can choose from a number of input parameters specific to volumes and particles, such a **Texture Value (Density)**, **Distance**, and **Particle Velocity**.
Size Gradient	This is an inline mini gradient editor allowing you to define gradient value keys easily without having to leave the basic Modo interface. Much in the same way a keyframe stores transform values for items, gradient keys store values and softly attenuate between the defined key positions. Keys can be added by middle-

Option	Description
	clicking along the gradient and then clicking and dragging the keyup or down, adjusting the value at that position. The gradient bar directly underneath provides a visual reference to the values being generated by the gradient itself. The defined values act as a multiplier of the final volume size on the preceding **Size Gradient Input** option.
Random Size	Controls the amount of randomness added to the scale of each individual particle volume. The amount is determined by a scale percentage - a setting of 25% adjusts the scale of the base radius size by an amount of 0%-25% randomly across all particles, a setting of 50% adjusts the size by 0%-50% randomly across all particles and so on.
Use Particle Orientations	Aligns the particle to the direction vector from the particle source (direction of travel). The direction vector is usually the surface normal for surface particles or the particle velocity vector for simulation particles.
Align to Path	Aligns the particle to its local motion vector.
Stretch Particles	This percentage controls the scale of particle stretching based on velocity. Values are additive to the base value, so a **Stretch Particle** value of 100% produces a particle that was stretched twice the length (200% altogether).
Automatic Texture Offset	Adds an offset to the hypertexture different for each particle so that each particle looks different.

Volume Shading


Item List
Shading
Clips
+
Filter Items
Add Item
F
Name
MutantPixel.lxo
Mesh
Volume
Particle Cloud
Camera
Properties
Channels
Display
+
Volume Particles
Volume Shading
User Channels
Tags

Density

RGBA Gradient Input	Texture Value
Color Gradient	
Density	100.0
Falloff	0.0 %
Level	169.0 %
Density Gradient Input	Particle Age
Density Gradient	
Hypertexture Amplitude	100.0 %
Hypertexture Scale	50.0 %
Texture Effect	Velocity Translate
Effect Speed	0.0 %

Sampling

Render Quality	High
Sampling Rate	1.0
Near Clip Distance	250 mm

Shading

	☑ Use All Lights
Shading Light	(none)
	☑ Volumetric Shadows
Shadow Quality	High
Shadow Opacity	100.0 %
Scattering	100.0 %
Forward Scattering	50.0 %
Ambient	0.01
Ambient Color	1.0 1.0 1.0

]

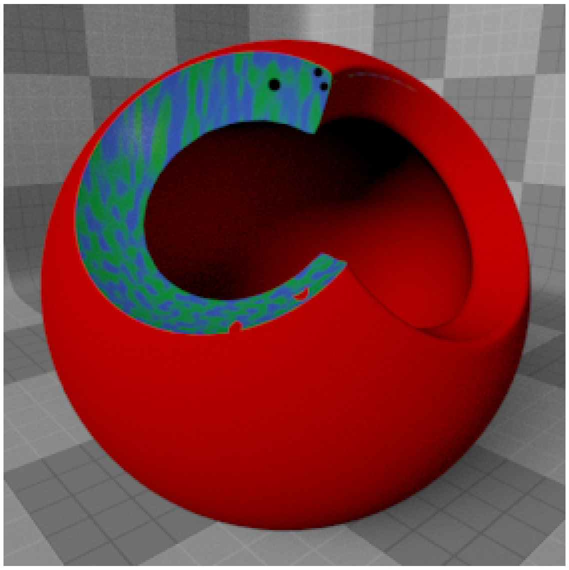
Option	Description
Density	
RGBA Gradient Input	Provides a means to define an input parameter in which to apply the following **Color Gradient**, determining the coloration of the resulting volume based on the chosen parameter. You can choose from a number of Input parameters specific to volumes and particles, such a **Texture Value (Density)**, **Distance**, and **Particle Velocity**.
Color Gradient	This is an inline mini gradient editor allowing you to define gradient value keys easily without having to leave the basic Modo interface. Much in the same way a keyframe stores transform values for items, gradient keys store colors or values and softly attenuate between the defined key positions. Keys can be added by middle-clicking along the gradient and then with the click, the key can be dragged up or down adjusting the value at that position. The gradient bar directly underneath provides a visual reference to the values being generated by the gradient itself. In the Color gradient, pressing the color wheel icon allows precise input of color values using the standard Modo color picker. The defined colors are applied to the final volume based on the preceding 'RGBA Gradient Input' option.
Density	Specifies the thickness or opacity of the volume, controlling the apparent amount of particular matter that is visible. This is a multiplier of the base value defined in the material item. For density samples, see Material.
Falloff	Biases the fading of the volume from its most dense areas to its least dense areas. A falloff of 0% produces a solid volume representing the particle geometry shape, a value of 50% produces smooth fading between the thickest areas leaving the thin areas transparent, and a value of 100% doesn't produce any volume at all as the volume has been biased all the way toward the transparent end of the spectrum. When no density texture is applied, the falloff controls the attenuation from the center of the volume to its outermost radius.
Level	Changes the maximum density level of the overall volume as a multiplier of the original volume with 100% being the maximum. At a level setting of 50%, what would have originally been the 50% density value is the 100%, producing a smaller particle volume.
Density Gradient Input	Provides a means to define an input parameter in which to apply the following **Density Gradient**, determining the thickness of the resulting volume based on the chosen parameter. You can choose from a number of input parameters specific to

Option	Description
	volumes and particles, such as **Texture Value (Density)**, **Distance**, **and Particle Velocity**.
Density Gradient	This is an inline mini gradient editor allowing you to define gradient value keys easily, without having to leave the basic Modo interface. Much in the same way, a keyframe stores transform values for items, gradient keys store values and softly attenuates between the defined key positions. Keys can be added by middle-clicking along the gradient and then clicking and dragging the key up or down, adjusting the value at that position. The gradient bar directly underneath provides a visual reference to the values being generated by the gradient itself. The defined values are applied to the final volume, based on the preceding **Density Gradient Input** option.
Hypertexture Amplitude	Determines the amplitude (intensity) of a volume density texture layer applied to the volume. The higher the value the more you see the effect of the density texture. It's similar in concept to displacement amplitude, but applies to volumes.
Hypertexture Scale	Changes the size of the hypertexture, it is the same things as changing the texture locator's **Scale** setting values.
Texture Effect	Hypertextures can be animated on particles to create billowing or dissipation effects. The goal of these effects is to add secondary animation to the particles, which is important for explosions and smoke. The various texture effects are: • **None**: Disables the **Texture Effect**. • **Velocity Translate**: moves the texture in the direction of the velocity vector. • **Expand**: the texture is scaled over time to give the impression that it expands. • **Expand and Dissolve**: like the previous effect, but with the addition of a dissolve effect so as the texture expands it also dissolves. • **Pyroclastic**: a boiling motion similar to hot gas swirling around within the volume, adding a realistic secondary motion. Good for smoke.
Effect Speed	Controls the speed at which the **Texture Effect** animates through the volume.
Sampling	
Render Quality	Determines the overall render quality of the resulting volume. The options are **Low**, **Med**, **High**, **Very High** and **Best**. As the quality increases, render times increase as well.

Option	Description
Sampling Rate	This is like the render quality option but can be fine-tuned in case the render quality is not enough. Like all rates in Modo smaller means better, but also slower. Use it with caution, using a rate of 0.5 instead of 1.0 greatly improves the sampling quality, but also multiplies volume rendering time by 4.
Near Clip Distance	Modo starts integrating the volume densities at a fixed distance from the camera. This also controls the initial integration step, so very small values lead to long render times if the camera is inside the volume. Think of it as a windshield around the camera. In general, this value only needs to be adjusted when you are inside a volume, like when you fly through a cloud.
Shading	
Use All Lights	By default, Modo uses all lights in the scene to compute the volume illumination. This can be very expensive however, if the scene has many lights, so that can be disabled.
Shading Light	When **Use all Lights** option is disabled, you can select which light or group of lights contribute to the volume's illumination. For information on creating group, see Groups Viewport.
Volumetric Shadows	When a light shines on a volume, the light is attenuated as it travels through the volume because of absorption. This is like self-shadowing on surfaces. For effects like smoke that don't need self shadowing, it is better to disable this option, as it saves on render time. Another way to optimize volumetric shadows is to use deep shadow maps on the lights, but that comes at the expense of precision and memory, though it can dramatically improve render times.
Shadow Quality	Volume self-shadowing is the most expensive effect to render, it has its own quality setting which is derived from the overall render quality. This is an important setting, as it usually plays a big role in controlling volume aliasing. **Note:** For very faint volumes it is not necessary to use a high shadow quality as the shadows are not really visible.
Shadow Opacity	Volume self shadowing also depends on the volume opacity (how much absorption is going on), this number can be tweaked with that setting to change the look of shadows without changing the overall volume opacity.

Option	Description
Scattering	This is the overall scattering amount on the item, this is similar to the volume scattering amount in the material. This is a multiplier of the base value defined in the material item. For density samples, see Material.
Forward Scattering	Light scattering usually scatters more strongly in the forward direction, that's why clouds are brighter when the light comes behind the cloud. This control allows you to tweak this behavior.
Ambient	Volume items have their own ambient lighting settings to approximate multiple scattering effects that naturally happen inside clouds: when a photon scatters inside a cloud it actually bounces many times before it reaches the viewer, each bounce adding more scattering. This effect is called multiple scattering, it is the equivalent of global illumination for surfaces. So one way to approximate it is to use a global ambient light setting, it is a crude approximation, but it looks correct for most volume renderings.
Ambient Color	When using an **Ambient** value above 0, the **Ambient Color** determines the actual color value that contributes to the added ambient calculations, affecting the overall volumes color.

Render Boolean



Booleans, as they relate to 3D computer graphics, are a technique of using geometry from one Mesh Item to modify the geometry of another; their intersection produces a subtractive trimmed surface that would be difficult to create otherwise. The render boolean item provides a means to trim away areas of an object or surface at render time, without needing to physically modify the mesh itself. The benefits are that the mesh remains fully intact, the results are fully animateable and it produces a texturable surface where the render boolean intersects with the trimmed mesh. Also, since the boolean is generated at render time on the final subdivided geometry, there is a less likely chance of producing the errors generally associated with boolean trimming of subdivision surfaces.

Adding Render Booleans

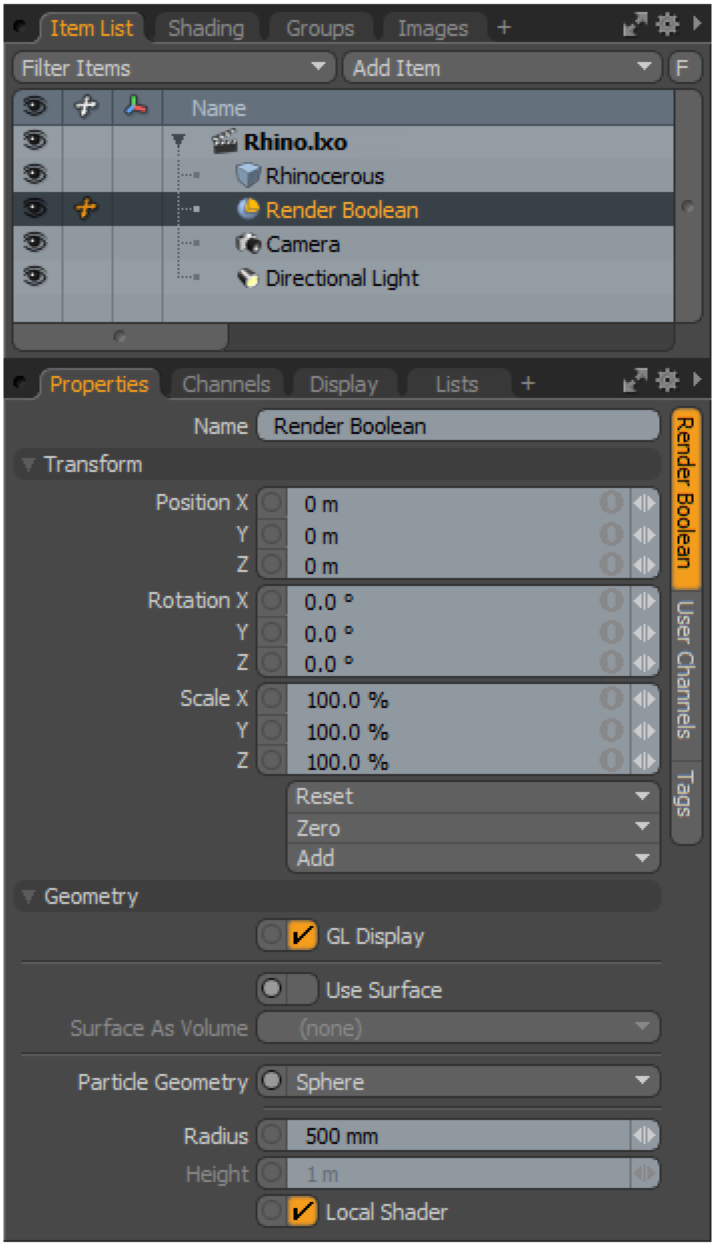
You can add a render boolean item to a scene in the **Items** list, by clicking **Add Item** > **Volumes**, then double-clicking **Render Boolean**. Once added, the item appears at the bottom of the **Items** list. When selected, a number of attributes controlling the boolean's functions appear in the Properties Panel.

Applying Materials

When utilizing the procedural geometry options of the render boolean, you can texture the resulting boolean surface with the use of an item mask. By right-clicking on the render boolean item in the **Items** list and selecting the **Create Item Mask** option, a new material group is added to the **Shader Tree**. You can then use the **Add Layer** option of the **Shader Tree** to add materials and texture layers that are then confined only to the areas where the booleans intersect with the target mesh. For more on working with layers in the **Shader Tree**, see Shader Tree. The render boolean item trims all items/surfaces in the scene, unless you explicitly disable the **Enable Surface Clipping** option of the Material item on a per surface basis (this option is enabled by default for all surfaces). When the **Use Surface** option is enabled for custom boolean meshes, the intersecting surface gets its surface shading from the Mesh Item itself. If you don't want this, you can disable the **Local Shader** option to have the target surface retain its own material attributes.

Animated Booleans

You can animate the render boolean operation by creating keyframes on the render boolean item when using the procedural shape option, or when a **Surface as Volume** mesh layer is specified. Animating the target mesh layer produces the animated boolean operation. For more on setting keyframes and animating in Modo, see Animating.


Item List
Shading
Groups
Images
Filter Items
Add Item
F
Name
Rhino.lxo
Rhinocerous
Render Boolean
Camera
Directional Light
Properties
Channels
Display
Lists
Render Boolean
User Channels
Tags
Transform
Reset
Zero
Add
Geometry

Name: Render Boolean

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

☑ GL Display

☐ Use Surface

Surface As Volume: (none)

Particle Geometry: Sphere

Radius: 500 mm

Height: 1 m

☑ Local Shader

Option	Description
Name	Displays the current item name. You can change it by clicking on the field and typing the new name.
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space. By default, **Position** transforms originate from the center position.

Option	Description
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, **Rotation** transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and mesh position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values' input fields). This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the **Channel** list while keeping the default **0,0,0** values (a necessary step for referencing, in order to override the channels, they must first exist).
Geometry	
GL Display	Toggles the display of the particle geometry proxy in the 3D OpenGL viewport. Generally, this should be enabled to get an approximation of how the boolean item renders (the default state), but in certain instances it may be useful to disable this option.
Use Surface	When enabled, you can select any arbitrary mesh layer from the scene and use it as a boolean volume. While render booleans work for nearly any geometry type, for best results, both should be fully closed watertight models, meaning they could virtually hold water without leaking.

Option	Description
Surface As Volume	Specifies the mesh layer that is used for the boolean operation, only when **Use Surface** is enabled.
Particle Geometry	The render boolean item also generates procedural geometry. You can choose between a cube, a sphere and a cylinder.
Radius/Height	Defines the outside dimensions of the particle geometry object. **Radius** refers to the overall dimensions of the sphere and cube shapes, and **Radius** and **Height** work together to define the dimensions of a cylinder.
Local Shader	This option is applied when using the procedural geometry of the render boolean. You can create an item mask for the render boolean item itself in the **Shader Tree** and assign materials and shading to it. Enabling the **Local Shader** option uses the defined shading of the item mask on the cut-away portions. When disabled, the shading for the item being trimmed is applied to the cut-away portions.

Tip: Render booleans aren't strictly for trimming geometry, they can also affect volume items such as Blobs, Sprites and Volumes.

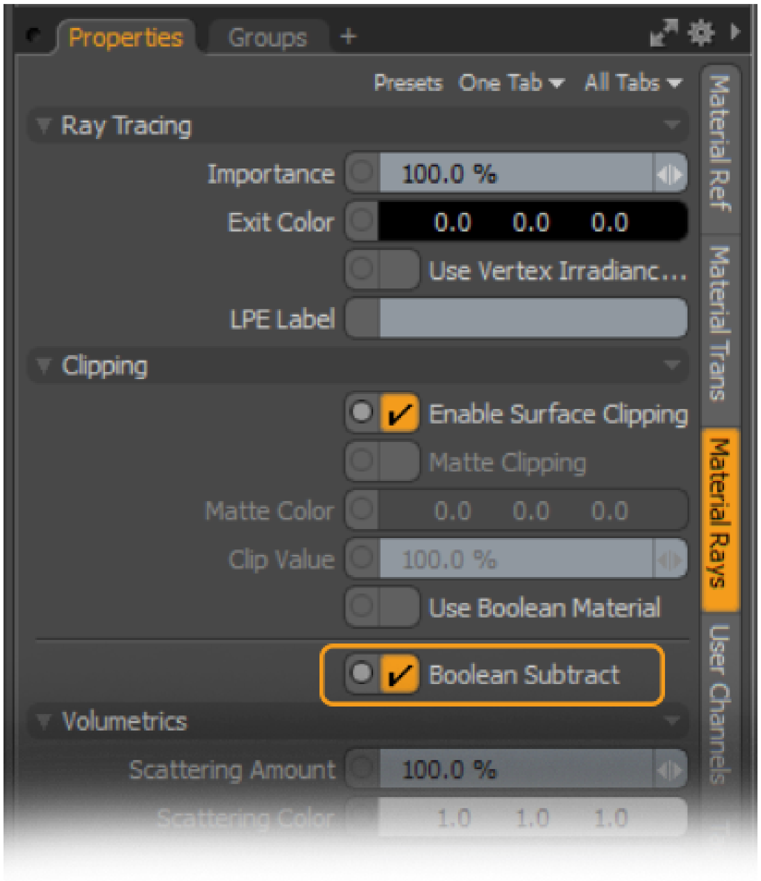
Create Boolean Effects at Render Time using mPath

Assigning a material as a boolean subtract in mPath allows you to create holes in your mesh at render time, without the need to create special items or physically alter your mesh. For the best results when using render booleans, both the subtractive objects and the objects being cut should consist of watertight enclosed volumes with outward facing normals.

Setting Up Render Booleans

To enable render booleans in mPath, you first need to mark your mesh as a **Subtract Boolean** within the mesh's material. To do this:

1. Select the material of the mesh you want to act as a render boolean in the shader tree
2. Once selected, navigate to the **Material Rays** tab of the material's Properties viewport
3. Within the **Material Rays** menu, enable **Boolean Subtract**


Properties
Groups
+
Presets
One Tab
All Tabs
Material Ref
Material Trans
Material Rays
User Channels

Ray Tracing

Importance 100.0 %

Exit Color 0.0 0.0 0.0

☐ Use Vertex Irradianc...

LPE Label

Clipping

☑ Enable Surface Clipping

☐ Matte Clipping

Matte Color 0.0 0.0 0.0

Clip Value 100.0 %

☐ Use Boolean Material

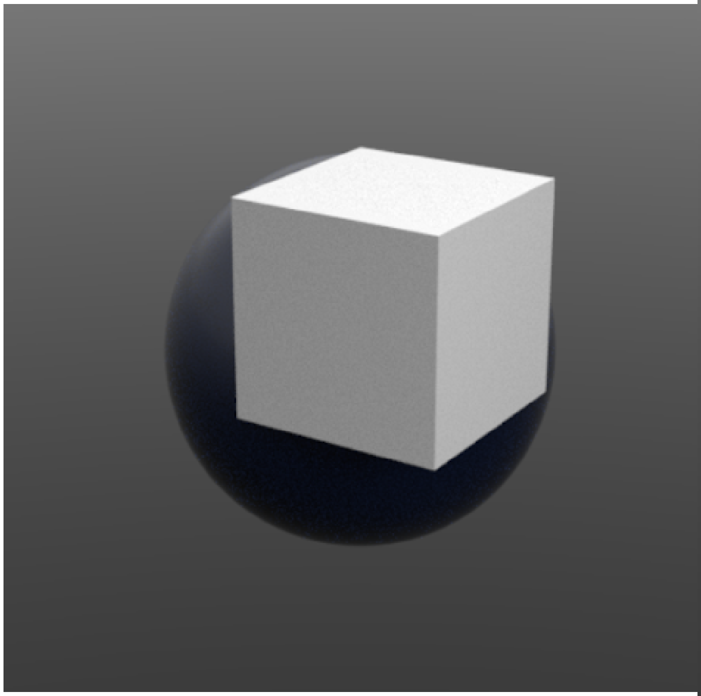
☑ Boolean Subtract

Volumetrics

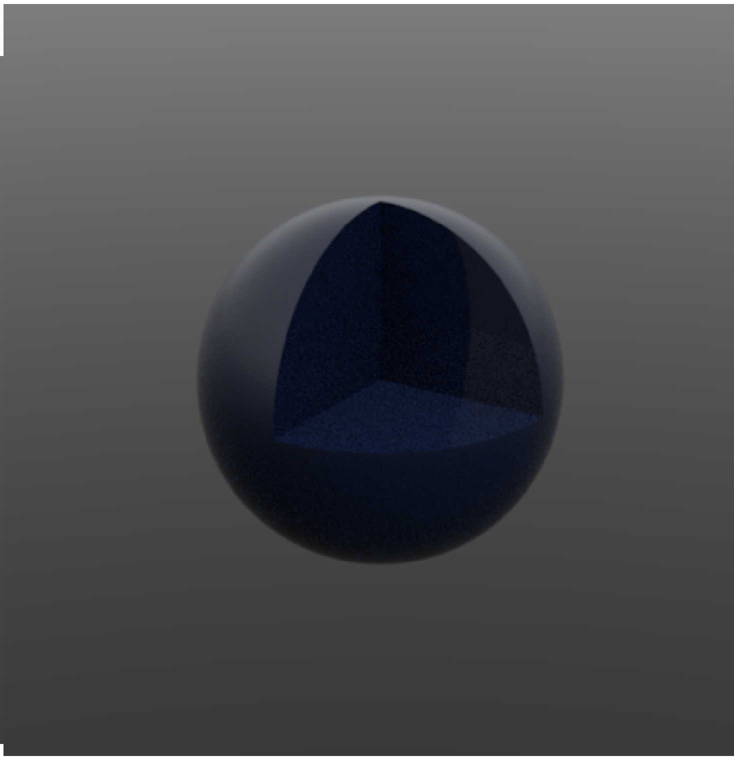
Scattering Amount 100.0 %

Scattering Color 1.0 1.0 1.0

4. The render boolean is now applied



Before marking the cube mesh as a

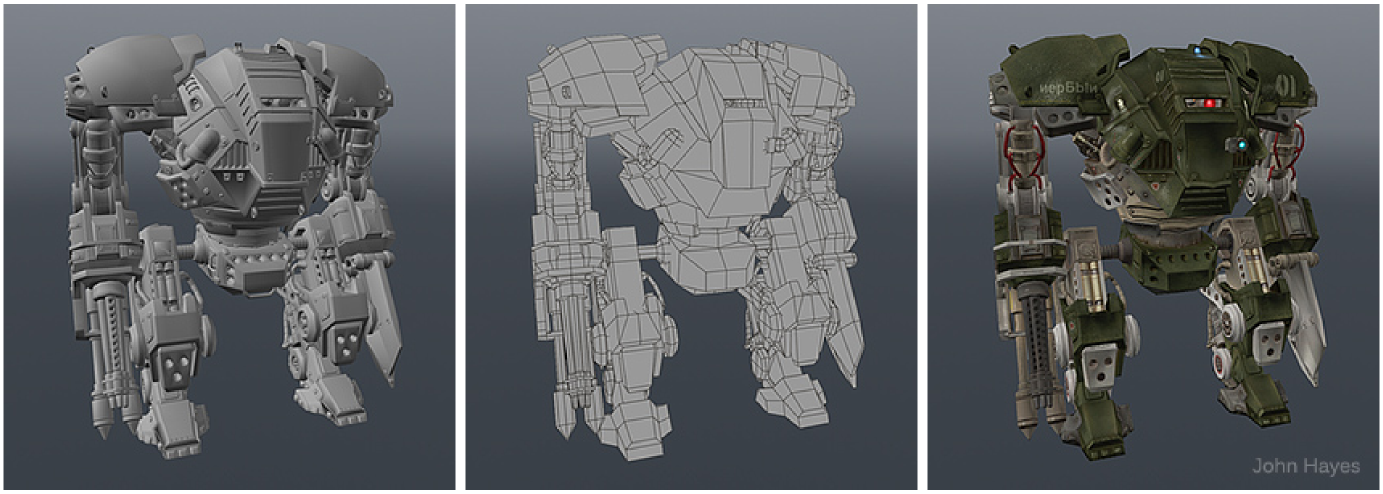


Boolean Subtract

After the cube is marked as a **Boolean Subtract**

Baking

Baking is the process used to pre-compute texture and lighting information. The result is often used for real-time applications (such as for games) or for exporting models into other software packages with complex shading intact, or simply as a way to eliminate rendering overhead in elaborate scenes. In this way, shaders and lighting are computed by the Modo rendering engine and baked directly into an image that maps perfectly to an existing UV texture.



John Hayes

Hi-res Geometry

Lo-res Geometry

Lo-res Baked Result

Depending on what your intended purpose is, there are a couple of methods for baking images. If your intent is to simply merge (flatten) multiple **Shader Tree** layers, then all that is needed is an image map added to the **Shader Tree** set to the appropriate **Effect** type before baking.

When baking maps such as Displacement and Normal maps, two layers of geometry are necessary, typically one is the high resolution model comprised of all the details of the final model, and a second low resolution model that approximates the shape of the high resolution model serving as the baking target. The low resolution geometry requires a proper UV map with a unique name (unique from any UV maps assigned to the high-res version). Additionally, you need to make sure the applicable layers are selected in the item list, and that the appropriate UV map is also selected. When Baking to Textures, suitable image maps must be placed and selected in the **Shader Tree** - don't worry, it only sounds complicated, each set-up is detailed below.

For information regarding adding layers and setting their Layer Effects, please see Shader Tree. Baking commands can be invoked from the menu bar within the **Render** menu and well as from the contextual menu in the **Shader Tree** layers.

Tip: For more information, review our Making Games Assets with Modo video series.

Bake Items and Baking with the Game Tools

Modo has a range of features to simplify the baking process, notably:

- **Bake Item** - Containers for all the various settings you may use. These enable quick re-triggering of the same bake.
- **Baking tools** - A set of shortcuts for creating bake components, including a **Baking Wizard**, **Baking** tools in the Games Tools layout Toolbox, and baking options in the **Mesh Operations** list.

If you are unfamiliar with baking, we recommend you familiarize yourself with the different baking operations. Once this is understood, refer to Working with Bake Items and Baking Tools for detailed information about these baking tools and how to use them.

Note: These tools offer an alternative workflow to the one described in this topic.

Tip: To reduce baking time and memory required to bake shaders, use the **Object Normal** texture effect. For more information, see Effect - Shader Item

Cages and Ray Distance

When baking, rays are fired at the high-resolution model, and when the rays intersect the surface, the detail is saved to a texture. Consequently, baking requires an offset distance to determine the ray origin. This offset distance should be adjusted to include all necessary detail, without a loss of accuracy. In Modo, this distance can be defined in one of two ways:

- **Cage** - A deformation of a mesh using a **Morph Map**. The rays are fired from the surface of the **Cage**.
- **Distance** - A distance offset from the mesh surface.

These represent the maximum ray travel amount, both positive and negative, from the baking surface and should be set to encompass all surface deviations between the two surfaces.

The **Cage** is a specially created **Morph Map** on the baking geometry, where the morph surface is typically offset from the limit surface, and represents the ray distance using the difference between the morph position and the undeformed position of the surface.

Note: When baking, if you specify a **Cage**, the **Distance** value should remain at **0** (zero).

The **Distance** value is a simple measurement from the mesh surface, and is used over the entire mesh.

Note: Black regions in the bake result represent areas beyond the defined **Distance**. You can increase the **Distance** to eliminate them, but setting the value too high can reduce accuracy and may introduce errors, depending on the geometry.

Bake to Render Outputs

The command **Bake to Render Outputs** allows you to use a UV map from a surface to generate a rendered image that can be applied back to the surface as a texture map.

The command uses the functionality of the **Shader Tree**'s Render Outputs, which allows the baking of:

- Ambient Occlusion specific texture maps,
- **Diffuse Coefficient** maps, or
- baking a **Final Color** map that fully computes all surfacing and shading of the target item.

Once rendered, images can be saved to disk from the Render Display viewport, then imported and applied back to the geometry.

Before baking, it is important to follow these steps:

1. Select the target item in the **Items** list or 3D viewport.
2. Select the target UV map in the Vertex Maps **Lists** viewport.

By default, render baking ignores any setting that may be applied to the Render Item's **Frame Width** or **Frame Height** settings, and instead uses the value set in the Render Item's **Bake Width** and **Bake Height** settings. The resolution set when a render item is first created is 1024x1024, meaning any baked renders are in a 1024x1024 format unless this setting is changed.

Note: If you're using Modo 12.2 or earlier, **Baking Width** and **Height** are accessed by clicking the **More** >> button at the bottom of the **Frame** tab in the **Properties** menu. If you're using Modo 13.0 or later, **Baking Width** and **Height** properties are accessed by clicking **More** at the bottom of the **Frame** tab in the **Properties** menu.

Additionally, an offset amount is applied to the final pixels to reduce, or eliminate, possible seams at UV borders edges.

Note: The amount of the border offset is determined by the **Bake UV Border Size** setting in the Preferences. See Bake UV Border for details.

In a default scene, the **Render Output** is set to **Final Color**, and you can right-click on the effect type in the **Effects** column of the **Shader Tree** and define an alternate render output style. Once the target items are selected, invoking the command **Render** > **Bake to Render Outputs** renders all active **Render Outputs** simultaneously.

Tip: The video below shows you how to bake to a **Render Output** using Modo 12.2 and earlier. For information on how to bake to a **Render Output** in Modo 13.0 and later, see the above note.

Open Progressive Baking

To quickly render progressive baking and set rendering options, on the application menu bar, click **Render** > **Open Progressive Baking**. The Preview Viewport displays.

Options
Effect: (None...
Paused
LUT: sRGB
(Render Item)
Shading: Full

You can preview your textures and lighting information in the render preview window and watch your changes as you refine their properties progressively. This results in faster iterations of your work, giving you a more interactive experience when baking diffuse and light maps, troubleshooting, and developing your art direction.

Video: Watch this video for a texture baking workflow to transfer lighting and shading details.

For more information about how to create and configure bake items, see Working with Bake Items and Baking Tools.

Tip: Preview saves it's full progressive render state when Auto-Save completes a save. You can load the previous render state at a later time.

Bake from Object to Render Outputs

The **Bake from Object to Render Outputs** command is similar to the **Bake to Render Outputs** command above, however it allows the additional ability to generate its results using an alternate mesh layer. It is essentially transferring the surfacing from one object to another.

Typically, this setup consists of:

- one layer with a detailed, high resolution mesh (the source mesh), and
- the target baking mesh layer, which is a lower resolution mesh, approximating the overall shape of the high resolution layer.

This feature is often used in games, where the detailed surfacing from the high-res asset is applied to the low-res assets to give the overall visual impression of higher resolution in the final model. Once rendered, images can be saved out from the Render Display viewport, then imported and applied to the target geometry.

Baking from an object requires the following steps:

1. Add a background mesh layer containing the high resolution model. This should visible in the **Items** list, but not selected.
2. Add a foreground mesh layer containing the low resolution model, and a uniquely named UV map. The layer should be visible and selected.
3. Select the foreground mesh UV map in the **Lists** viewport.
4. Select the command **Render** > **Bake from Object to Render Outputs**.

   This opens the bake options dialog.

5. Enter an offset **Distance** that should encompass all surface differences between the two mesh layers.

Note: You can also use a predefined **Cage** to determine the offset distance.

6. Click **OK**. Modo extracts the geometrical, shading, and lighting information from the background layer, and bakes this detail to the foreground mesh's UV map.

Note: Pressing **OK** renders all active **Render Outputs** simultaneously.

Bake to Selected Texture

The command **Bake to Selected Texture** can be found in the menu bar under **Render** > **Bake to Selected Texture**, but is more commonly used from the **Shader Tree** directly, using the right-click contextual menu (opened by clicking on the target layer) where it is called **Bake to Texture**.

This command generates a bitmap texture of one or more **Shader Tree** texture layers, when all layers share the same **Effect** type. The common uses for the function are:

- to convert a procedural texture layers to a bitmap for displacement freezing, or
- to combine complex, multiple **Shader Tree** layers into a single layer, or
- to pre-compute render intensive effects, such as occlusion, saving render time and eliminating any frame to frame variance that may occur.

To use this feature:

1. First define a new **Image Map** layer in the **Shader Tree**.The file save location is defined, as well as the **Resolution** and **Bit Depth**. For information on creating **Image Maps**, please see Add Images.

Note: The **Texture Locator** settings are used to determined the appropriate UV map for baking, so make sure the image is applied appropriately.

Note: The position of the layer within the **Shader Tree** is important, because only layers below the target are evaluated for baking.

2. Define the **Effect** for the layer, this must match the source layers.
3. Disable the layers visibility (by toggling the eye column) to ensure the target baking layer itself isn't evaluated as part of the bake.
4. Right-click on the layer, and select **Bake to Texture**.

This opens the **Render Display** window.

5. Save the image using **Save Image** button.

Bake to UDIM Tiles

If your UV map uses UDIMs for texture placement, you can bake the UV regions to a set of UV tiles using the **Use Clip UDIM** option in the **Texture Locator** tab of the **Shader Tree** properties. You can add a UDIM layer to the **Shader Tree** using **(new udims)**, to create UDIMs from scratch, or **(load udims)**, to use existing UDIMs.

To Bake To UDIMs

Note: The following steps create a UDIM folder from scratch using **(new udims)**. You can also add an existing folder using **(load udims)**. Any images in the folder can be overwritten with the baked images.

1. In the Shader Tree, add a UDIM layer using **Add Layer** > **Image Map** > **(new udims)**.

   This opens the folder browser.
2. Choose a folder and click **Select Folder**.

   The **UDIM Wizard** opens.
3. Edit the fields as required. See Creating UDIMs Using the UDIM Wizard for full details. It is important to generate the correct number of UDIMs so that they match the UDIM range in your UV map. The amount generated is determined by the **UDIM Range Start** and **UDIM Range End** fields. For example, if you set the range as 1001 to 1003, Modo will generate three UDIM tiles, placed at U=0, U=1, and U=2.

   Check **Overwrite Existing Images** so that the new UDIMs replace any existing UDIMs of the same name in the destination folder.
4. Click **OK** to generate the UDIMs.

   This will add a UDIM layer to the **Shader Tree**.
5. Depending on the position of the new layer in the **Shader Tree**, the new UDIMs may overlay existing layers and appear on the mesh. This isn't a problem as we don't need to see the new layer in order to bake to it. Hide the layer by clicking on the visibility icon in the left column of the **Shader Tree**.
6. Right-click on the new UDIM layer and select **Bake to Texture**.

   The baking window will open and each tile will be baked automatically.
7. The UDIMs need to be saved. Open the Images/Clips List and expand the folder containing your new UDIMs.

   The images have an asterisk by their names to indicate that they haven't been saved.

8. Select all the UDIM images, right-click and select **Save**.

   The asterisks disappear and the images are saved to the destination folder.

Bake from Object to Selected Texture

The command **Bake from Object to Selected Texture** can be used to generate two specific types of texture maps using the foreground and background mesh layer functionality described in Baking. Specifically, the function is used to generate **Displacement Maps** and **Normal Maps**.

A **Displacement Map** is a grayscale image that represents the difference in position between a low-resolution foreground mesh and a high-resolution background mesh. A **Normal Map** uses color to represent the difference in the surfaces normal facing direction. Both maps, once generated, can be applied back to the lower resolution geometry to give the impression of much greater resolution.

Note: With baking **Normal Maps**, you can either use a **Cage** or a **Distance** value. For **Displacement Maps**, use only a **Distance** value.

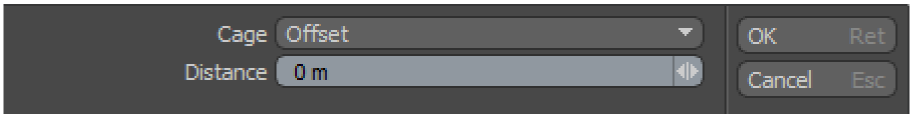
To use **Bake from Object to Selected Texture**:

1. Add a background mesh layer containing the high resolution model. This should be visible in the **Items** list, but not selected.
2. Add a foreground mesh layer containing the low resolution model, and a uniquely named UV map. The layer should be visible and selected.
3. Select the foreground mesh UV map in the **Lists** viewport.
4. Optionally, to use a **Cage**, add a **Morph Map**, which is offset from the surface.
5. Add an appropriate target bitmap texture layer to the **Shader Tree** with the required **Effect** type (such as **Normal**).

Note: The associated **Texture Locator** settings are used to determined the appropriate UV map for baking, so make sure the image is applied correctly.

6. Disable the texture layer's visibility (by toggling the eye column) to ensure the target baking layer isn't evaluated as part of the bake.
7. Right-click on the texture layer and select **Bake from Object to Selected Texture**.

   This opens the bake dialog.


Cage Offset
Distance 0 m

8. If you're using a **Cage** to generate a **Normal Map**, select the **Cage** and leave the **Distance** at **0 m**. Alternatively, for **Normal Maps** or **Displacement Maps**, specify the **Distance** value.
9. Click **OK**.

   This opens the **Render Display** and Modo bakes the detail to the texture layer.
10. Save the image using **Save Image** button.

Note: Make sure that the **Normal Map** layer in the **Shader Tree** is not in the same mask as the item you are trying to bake from.

Bake UV Border

Depending on the type of geometry, and the resolution of the baked image, UV border seams may be visible. These can be eliminated using the Bake UV Border Size option in **System** > **Preferences**, listed under **Rendering** > **Final Rendering**. This expands the UV border by the specified number of pixels to avoid distinct seams.

Note: The **Bake UV Border Size** must be greater than 0 to enable border expansion.

The **Bake UV Border** functionality requires an active Alpha **Render Output** with the same antialiasing settings as the render output. If no **Alpha Render Output** is found that fulfills this condition, Modo will automatically add one to the render window. This is added because Modo needs an alpha channel to compute the border offset.

Let's say you have an **Alpha Render Output** and a **Shading Normal Render Output** with **Antialiasing** set as follows:

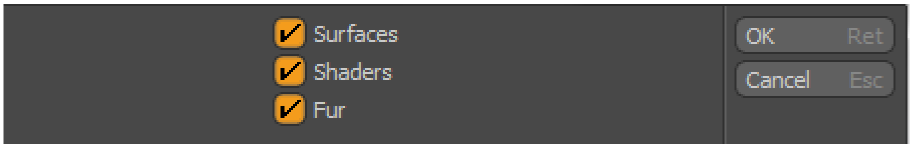
- **Antialiasing** is enabled on both - Modo does not add a new **Alpha Render Output** to the render window.
- **Antialiasing** is disabled on either one of them (but not both) - Modo generates an **Alpha Render Output** in the render window with **Antialiasing** set to match the **Shading Normal Render Output**.
- **Antialiasing** is disabled on both - no **Alpha Render Output** needs to be added to the render window.

Note: If an **Alpha Render Output** is added automatically, it is only visible in the render window, and does not appear in your scene.

Bake Geometry Cache

Modo has multiple features that work by way of virtual geometry. Virtual geometry describes geometry in the scene that you interact with, but is only a representation of the actual geometry used to render. Good examples of virtual geometry are Subdivision Surface models, Displacement, Render Curves, and Fur.

Selecting **Render** > **Render**, or pressing **F9**, causes Modo to take the instructions for each feature and convert them to real geometry for the final rendering. This procedurally generated geometry for rendering can be baked into real **Mesh** items in a scene, making it possible to apply complex procedural displacements or generate fur and bring it back to the scene as real geometry that can be directly edited or exported to other packages. Selecting **Render** > **Bake Geometry Cache** displays the following dialog.


☑ Surfaces
☑ Shaders
☑ Fur

Option	Description
Surfaces	When enabled, the Surface settings of the original object are cloned and added into new Material Groups assigned to the baked geometry.
Shaders	When enabled, the Shader settings per item of the original object are cloned and added to a Material Group assigned to the baked geometry.
Fur	When enabled, convert any Fur in the scene to geometry. If the Fur is set to **Strips**, it is converted to polygons. If the fur is set to **Tubes** it is converted to individual Curves, representing each individual fiber. **Note:** The Curves for Fur can be further converted to polygons using the 'Render Curves' command and invoking a second pass of the Bake command.)

Option	Description
	Running the command converts all **Mesh Item** layers in the scene, as well as **Instances** and the procedural effects **Fur**, **Render Curves**, and **Replicators**. Mesh **Density** is based on the settings for each individual item (exactly as it would tessellate for rendering). The resulting meshes are a separate mesh layer for each surface or effect in the scene. **Note:** Special procedural effects, such as **Volumes**, **Blobs**, **Render Booleans**, or **Sprites** cannot be converted to meshes.

Tip: Depending on the complexity of the scene, it is possible to create scenes of immense size using the **Bake Geometry Cache** command. The render statistics panel gives a readout of the total scene geometry during each render as a useful guideline.
If you only want to modify a single element, you can do so by saving individual mesh layers out by right-clicking on the layer in the **Items** list and selecting **Save Selected**, and then opening the resulting file and applying the **Bake Geometry Cache** command.

Working with Bake Items and Baking Tools

Modo has a set of tools to simplify baking located in the **Game** layout toolbox on the left panel, under the **Baking** sub-tab. For more information, see Creating Bake Items using the Toolbox Baking Tools.

In addition, a number of other baking tools are available in the **Mesh Operations** list from the **Bake** contextual menu. From this menu, you can bake occlusion, illumination, normals, displacements, texture effects, vertex shadings, vertex curvatures, and bake to an Alembic file. For more information, see Creating Baked Items using the Mesh Operations list Bake Tools.

Note: If you're unfamiliar with baking in Modo, it would help to read Baking to familiarize yourself with the concepts discussed in this section.

Understanding Bake Items

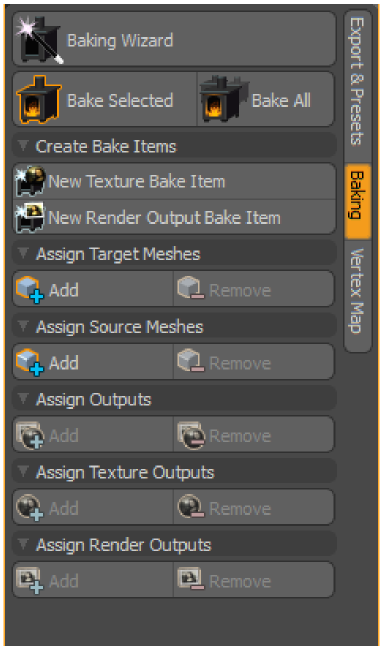
A **Bake Item** contains all the settings needed for a bake operation. By storing the settings in a single item, you can quickly edit and re-trigger a bake without having to reconstruct the bake from scratch.

The **Shader Tree** supports multiple bake items, allowing you to perform multiple bakes in sequence.

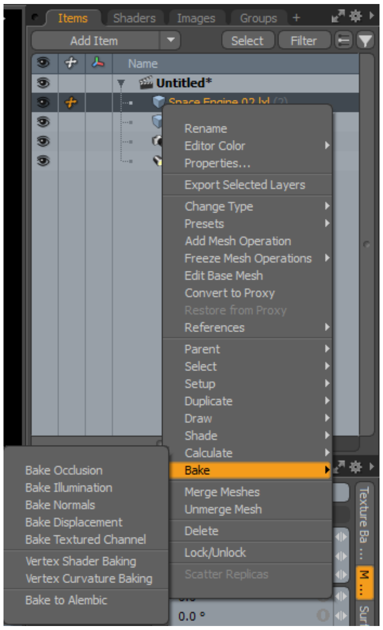
- You can add bake items using the Games Layout **Baking** tools in the Toolbox, namely the **New Texture Bake Item**, **New Render Output Bake Item** options, or by using the **Baking Wizard**. Each bake item is added to the **Shader Tree** in the **Bake Items List**.

  For more information, see:

  - Creating Bake Items using the Toolbox Baking Tools
  - Using the Bake Wizard


Baking Wizard
Bake Selected
Bake All
Create Bake Items
New Texture Bake Item
New Render Output Bake Item
Assign Target Meshes
Add
Remove
Assign Source Meshes
Add
Remove
Assign Outputs
Add
Remove
Assign Texture Outputs
Add
Remove
Assign Render Outputs
Add
Remove
Export & Presets
Baking
Vertex Map

- You can add bake items using the **Items List** contextual menu by selecting **Bake** and then specifying the type of bake you want to apply. A variety of bake options are available. For more information on how to create a bake item using this technique, seeCreating Baked Items using the Mesh Operations list Bake Tools.


Items
Shaders
Images
Groups
Add Item
Select
Filter
Name
Untitled*
Space Engine 02 lvl
Rename
Editor Color
Properties...
Export Selected Layers
Change Type
Presets
Add Mesh Operation
Freeze Mesh Operations
Edit Base Mesh
Convert to Proxy
Restore from Proxy
References
Parent
Select
Setup
Duplicate
Draw
Shade
Calculate
Bake
Merge Meshes
Unmerge Mesh
Delete
Lock/Unlock
Scatter Replicas
Bake Occlusion
Bake Illumination
Bake Normals
Bake Displacement
Bake Textured Channel
Vertex Shader Baking
Vertex Curvature Baking
Bake to Alembic
Texture Ba...
0.0 °

Applying a Basic Workflow

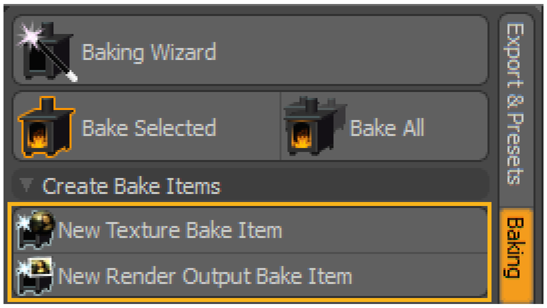
When using a **Bake Item**, follow these general steps:

1. Create a bake item of the preferred type.
2. Edit the **Bake Item Properties** according to the type of bake required.
3. Bake your scene, either locally or on a render farm.

   Once baked, you can return to step 2 to modify the bake properties if needed.

Creating Bake Items using the Toolbox Baking Tools

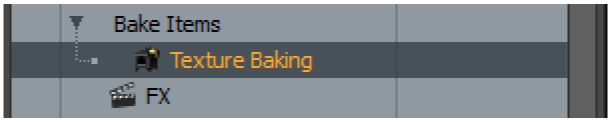
Bake Items can be made using the **Game** layout and applying the tools located in the toolbox's **Baking** sub-tab. To create a bake item, select one of the following:


Baking Wizard
Bake Selected
Bake All
Create Bake Items
New Texture Bake Item
New Render Output Bake Item
Export & Presets
Baking

Baking Tool	Description

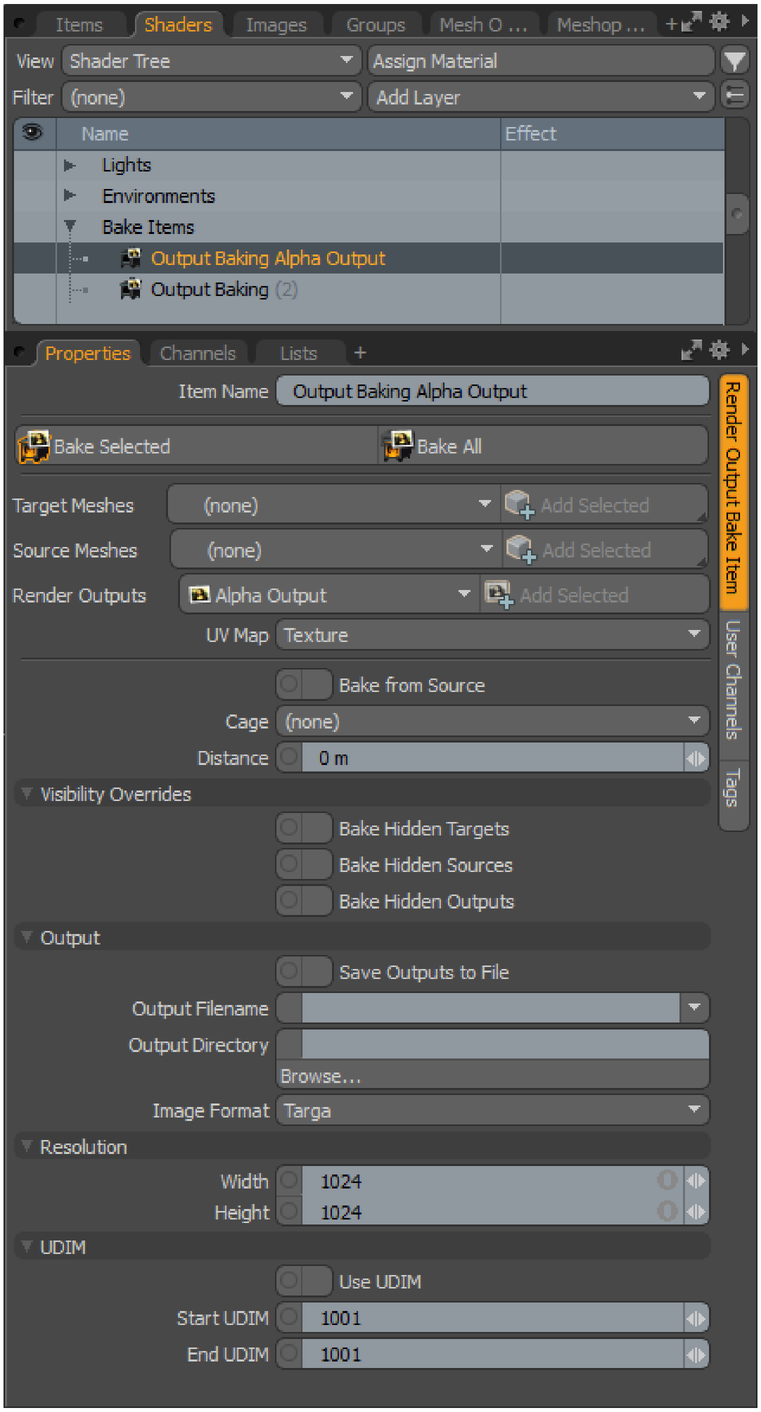
New Texture Bake Items	Creates a bitmap texture of one or more **Shader Tree** texture layers. These are baked to an Image Map in the Shader Tree.
New Render Output Bake Items	Creates a rendered image that can be saved and applied back to the surface as a texture map. These are baked to one of the Render Outputs. **Tip:** For more information on the baking output types, see Baking.

A new **Bake Item** is added to the **Shader Tree**, in the **Bake Items** section.



Note: You can remove a **Bake Item** from the **Shader Tree** using the **Delete** key.

Once added, select what is fired using either the Bake Selected or the Bake All option and configure the **Properties** as required in the right panel.


Items
Shaders
Images
Groups
Mesh O ...
Meshop ...
Assign Material
Add Layer
Name
Effect
Lights
Environments
Bake Items
Output Baking Alpha Output
Output Baking (2)
Properties
Channels
Lists
Bake Selected
Bake All
Add Selected
Add Selected
Add Selected
Visibility Overrides
Output
Browse...
Resolution
UDIM
Render Output Bake Item
User Channels
Tags

View: Shader Tree
Filter: (none)
Item Name: Output Baking Alpha Output
Target Meshes: (none)
Source Meshes: (none)
Render Outputs: Alpha Output
UV Map: Texture
Bake from Source: ○
Cage: (none)
Distance: 0 m
Bake Hidden Targets: ○
Bake Hidden Sources: ○
Bake Hidden Outputs: ○
Save Outputs to File: ○
Output Filename:
Output Directory:
Image Format: Targa
Width: 1024
Height: 1024
Use UDIM: ○
Start UDIM: 1001
End UDIM: 1001

Property Option	Description
Item Name	Depending on the baking option selected in the Game layout Baking Toolbox, the **Item Name** updates automatically and the appropriate options display.

	• **Texture Baking** - By selecting the **New Texture Bake Item** option, a **Texture Baking** item is listed in the **Texture Bake Items** tab of the **Properties** panel. • **Mesh Feature** - By selecting the **New Render Output Bake Item** option, an **Output Baking** item is listed in the **Render Output Bake Item** tab of the **Properties** panel.
Bake Selected	Fires bake items that are selected in the **Shader Tree**.
Bake All	Fires all bake items.
Target Meshes	Specifies the target meshes of the bake item.
Source Meshes	Specifies the source meshes of the bake item.
Render Outputs	Specifies the output textures of the bake item.
UV Map	Specifies the target UV map of the bake item. This option is not available for **Texture Bake Items**.
Bake from Source	Selects whether the outputs should be baked from the source meshes to the target meshes (an object-to-object bake). If this is enabled and no source meshes are set, then all meshes without the target UV map are automatically used as the source meshes for this bake. If disabled, then the outputs are baked directly from the target meshes back onto themselves. (for example, for baking procedural items down to textures).
Cage	Specifies the projection cage the baked item is baked with.
Distance	Specifies the distance that rays should be cast from the target mesh in order to find a source mesh to bake from. This value is overwritten by the bake cage, if one is selected.
Use Normal Map Preset	Selects whether the current normal map preset should be automatically applied to normal maps before baking (the preset applies tangent basis, sets up Invert Green channel and Tangent Type channels, as well as disabling color creation for the normal map). For more information, see Image Map and Editing Normal Map Presets. **Note:** Available for Texture Baking items only.

Visibility Overrides	
Bake Hidden Targets	When enabled, bakes to all connected target meshes, even if they are hidden.
Bake Hidden Sources	When enabled, bakes from all connected source meshes, even if they are hidden.
Bake Hidden Outputs	When enabled, bakes to all connected outputs, even if they are hidden.
Outputs	
Save Outputs to File	Selects whether baking results are saved directly to a file after baking. Render Output Bake items are found in the Render window while changes to the output images from Texture Bake items can be saved from the Images/Clips List. If this option is disabled, outputs must be saved manually.
Output Filename	Specifies the file name and pattern for outputting bake results.
Output Directory	Specifies the root folder for outputting bake results.
Image Format	Presents a dropdown menu to specify formats for Output Baking items only.
Bake to Channels	
Render Output Filter	Specifies the RGBA component(s) value used to filter the render output. The following options are available: **RGB**, **RGBA**, **Red**, **Green**, **Blue**, and **Alpha**. This option is disabled by default. Available for Texture Baking items only.
Texture Swizzling Override	Specifies the RGBA component(s) value to use to filter the texture output. The following options are available: **RGB**, **Red**, **Green**, **Blue**, and **Alpha**. This option is disabled by default. Available for Texture Baking items only.
Resolution	
Width	Specifies the width of the output image in pixels.

Height	Specifies the height of the output image in pixels.
UDIM	
Use UDIM	Allows you use UDIM if your UV map uses UDIMs for texture placement, you can bake the UV regions to a set of UV tiles.
Start UDIM	Allows you to specify the start value for the UDIM.
End UDIM	Allows you to specify the end value for the UDIM.

Configuring Bake Items

Bake Items are configured and fired using either the **Baking** tools panel or the item's **Properties** panel. The **Properties** panel provides a complete set of parameters and commands, whereas the toolbox panel provides basic baking commands.

You need to configure the bake item's parameters according to the type of bake that you're performing. For example, when baking **Displacement Maps** to a single texture, you only need to set the **Texture Outputs**. Also, when baking items for a Games workflow, use **Object Normal Maps** for static, non-deforming objects (environment, props, and so on) to render quickly in game engines.

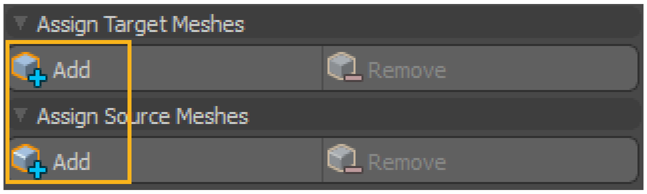
Modo provides shortcut buttons to populate the bake item parameters, including source meshes, target meshes, UV maps, and the different output types. These items are added to the bake item by first selecting the item(s) in the **Shader Tree**, and then using one of the following buttons:

- **Set** - Set the value of the parameter to the selected item.
- **Add** - Add the selected item to the parameter options list.
- **Remove** - Remove the selected item from the parameter options list.

You can either use the **Baking** panel or the bake item's **Properties** panel to assign the parameters.

Using the Baking Panel

The **Baking** panel offers a full set of **Add** and **Remove** buttons that operate on the selected bake item. Each button becomes active when the corresponding item type is selected in the **Items List** or **Shader Tree**. For example, if you add a new bake item and select a mesh (or meshes) in the **Items List**, the **Add** buttons become active for adding the mesh(es) as a source or target, as shown below:



If a selected item has already been added to the bake item, the **Remove** button becomes active.

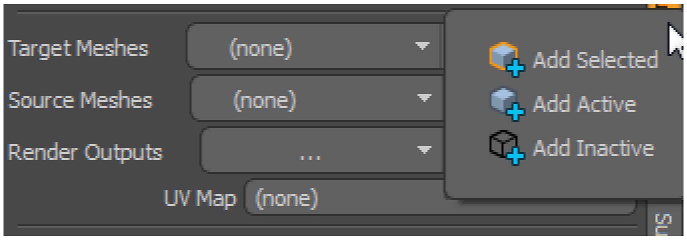
Using the Properties Panel

Unlike the **Baking** panel, the **Texture Bake Item** tab in **Bake Item Properties** panel contains only parameters and commands that are specific to the bake item type.

You can add, remove, or set each parameter as required. It's sometimes useful to apply these operations to certain items depending on their state. For this reason, Modo offers three options for add, remove, and set:

- **Selected** - Apply to only selected items.
- **Inactive** - Apply to all background items.
- **Active** - Apply to all foreground items.

Click on the corner of each button to access the three options.


Target Meshes (none)
Source Meshes (none)
Render Outputs ...
UV Map (none)

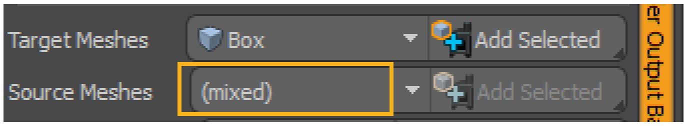
Set and remove values as follows:

- **Ctrl**+click the buttons to remove items.
- **Alt**+click the buttons to set the parameter to the selected item(s).

When selecting items from the **Shader Tree** to use in the bake item, be careful to keep the bake item selected by **Ctrl**+clicking the **Shader Tree** items. This way the **Texture Bake Item** remains available in the **Properties** panel.

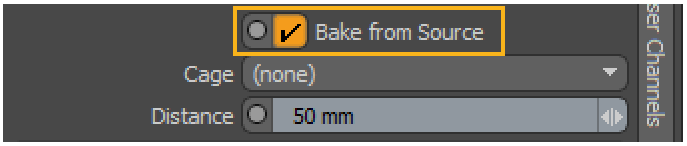
Multiple Selections

If you highlight and add multiple items, such as choosing multiple source/target meshes, or outputs, the items are combined and labeled as **(mixed)**, as shown below.


Target Meshes
Box
Add Selected
Source Meshes
(mixed)
Add Selected
er Output Ba

Bake from Source

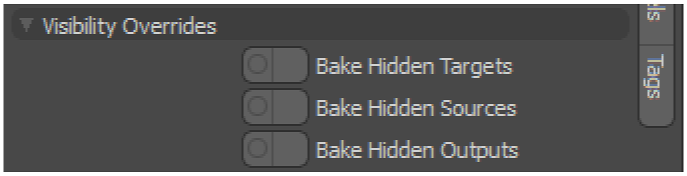
When baking from **Source Meshes**, you must check **Bake from Source**, and specify a bake distance or a **Morph Map** to be used as a **Cage**.


Bake from Source
Cage
(none)
Distance
50 mm
ser Channels

Note: You can bake from all meshes that are not identified as a target by checking **Bake from Source** and leaving **Source Meshes** set to **(none)**. If you also leave **Target Meshes** set to **(none)**, Modo automatically identifies the target by checking against the UV map that is associated with the output.

Visibility Overrides

The **Bake Item** offers three options for including hidden items in the bake. This allows you to configure the **Bake Item** and get the desired result from the bake, regardless of the visibility state of items in the scene. This eliminates the need to toggle foreground/background mesh visibility, and enable/disable render outputs, every time you want to bake.


Visibility Overrides
Bake Hidden Targets
Bake Hidden Sources
Bake Hidden Outputs
ls
Tags

- **Bake Hidden Targets** - Bake to hidden target meshes.
- **Bake Hidden Sources** - Bake from hidden source meshes.
- **Bake Hidden Outputs** - Bake to hidden **Render Outputs**.

Previewing Bake Items

Both the 3D (OpenGL) Viewport and the Preview Viewport allow you to preview bake results. See Ray GL and Preview Bake Mode

Texture Bake Items

To bake to a texture layer for inclusion in your scene, use a **Texture Bake Item**. Common uses for **Texture Bake Items** include **Displacement Maps** and **Normal Maps**.

Texture Bake Items use the following items:

- **Target Meshes** - One or more target meshes for the bake.
- **Source Meshes** - One or more source meshes to be used in the bake.
- **Texture Outputs** - A texture layer to receive the bake.

Baking to a Texture

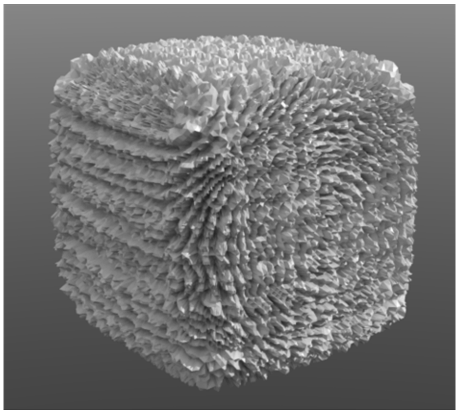
By baking to a selected texture, you can do the following:

- Generate a bitmap texture from one or more **Shader Tree** texture layers, such as combining several **Normal** layers.
- Convert procedural texture layers to a single layer for **Displacement Map** freezing.
- Pre-compute render-intensive effects, such as **Ambient Occlusion**.

To use a **Bake Item** to bake to a texture, you must choose one or more image texture layers from the **Shader Tree**. The baking process then chooses the correct items to bake from, and the UV map to bake to. The results are baked to the **Texture Outputs**.

Note: For more information on this type of baking, see Bake to Selected Texture.

Let's say you have a mesh with three **Displacement Maps**. In this case, **Wood**, **Weave**, and **Dots**. The mesh also has a UV map.



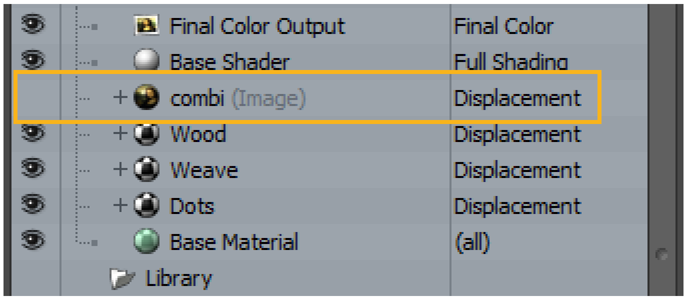
Use a **Bake Item** to merge the three layers.

1. Add a new, blank **Image Map** to the **Shader Tree** to receive the bake:

   In the Shader Tree, click **Add Layer**, and select **Image Map** > **(new image)**.

   In the file browser, select the image format and filename. Click **Save**.

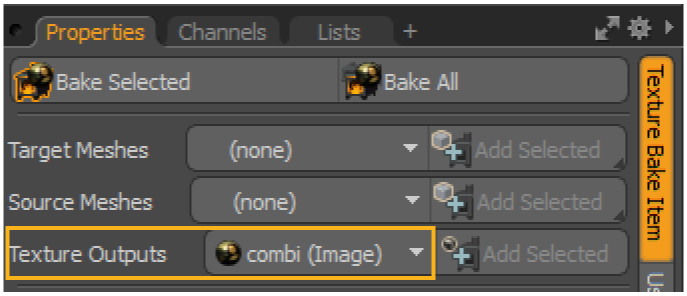
   On the **New Still** dialog, configure the image details and click **OK**.
2. Hide the new image by toggling the paintbrush in the left column.
3. Set the **Effect** of the new image to be the same as the layers that you're baking, in this case **Displacement**.



Final Color Output	Final Color
Base Shader	Full Shading
combi (Image)	Displacement
Wood	Displacement
Weave	Displacement
Dots	Displacement
Base Material	(all)
Library	

4. Click **New Texture Bake Item**.

   The **Bake Item** is added to the **Shader Tree**.
5. Select the **Bake Item** to open its **Properties**. Ensure the **Texture Bake Item** tab is selected.
6. Set the **Texture Outputs** to the new **Image Map**.



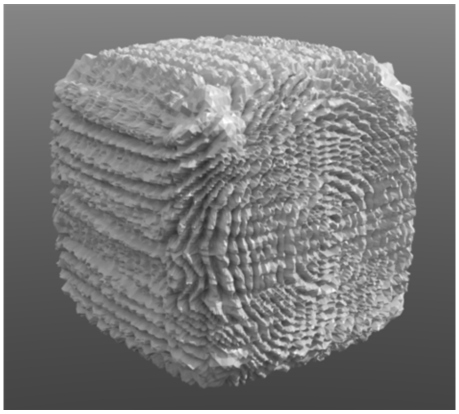
Target Meshes (none)

Source Meshes (none)

Texture Outputs combi (Image)

7. Click **Bake Selected** to fire the bake.

   This opens the **Render Display** and shows the bake process.
8. When the bake is finished you can save the image using the **Save Image** button in the **Render Display**.
9. Now that the image is baked, you can see the results by hiding the original **Displacement** layers, showing the new baked image in the **Shader Tree**, and re-rendering the mesh.



Note: The slight difference in detail is due to the relatively lower resolution of the target image layer.

Tip: Texture Bake Items have an option to use the current normal map preset, which means that the selected tangent basis is automatically applied and any generated normal maps are set up according to the current normal map preset's settings before baking happens. To enable this, in the Texture Bake Item's **Properties** panel, enable the **Use Normal Map Preset** checkbox.

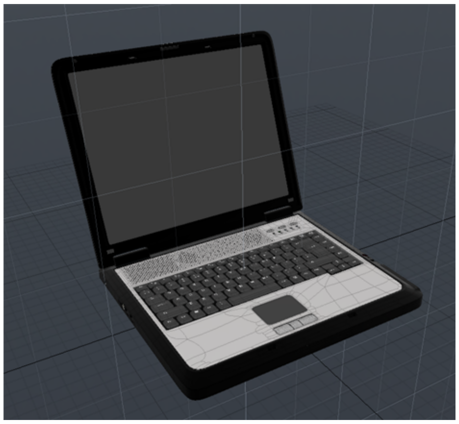
Baking from an Object to a Selected Texture

The **Texture Bake Item** can be used to generate **Displacement Maps** and **Normal Maps**, baked from a high-resolution source mesh (or meshes) to a low resolution target mesh (or meshes). You can also bake other map types as required. This is useful for creating assets for real-time environments (such as game engines) and transferring detail from objects to a texture for further modification.

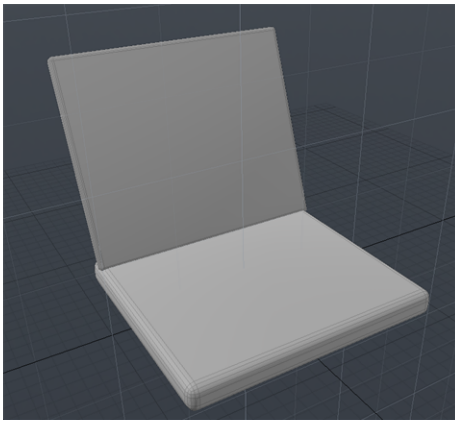
Note: For more information on this type of baking, see Bake from Object to Selected Texture.

Let's say we have a detailed mesh (pictured left) and we would like to bake a **Normal Map** and **Diffuse Color** texture to a low poly model (right). The low poly model has a UV with no overlapping islands.

Note: The order of the meshes in the item list does not affect the bake.

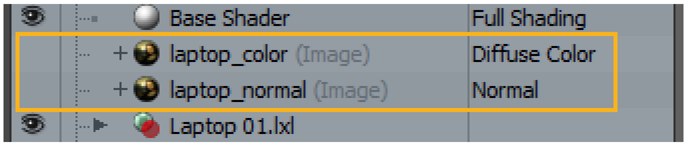


A high poly source mesh.



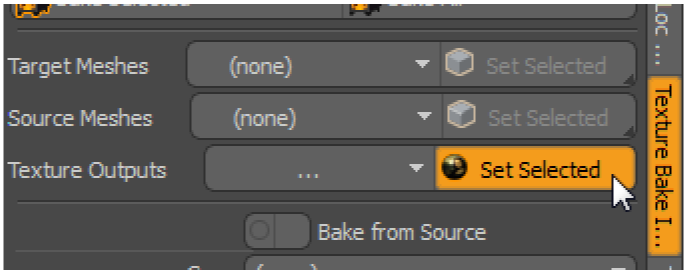
The low poly target mesh.

1. Add two new image layers, set the **Effect** of one to **Diffuse Color**, and the other to **Normal**.
2. Hide both layers.



Base Shader	Full Shading
laptop_color (Image)	Diffuse Color
laptop_normal (Image)	Normal
Laptop 01.lxl	

3. Click **New Texture Bake Item** to add a **Bake Item** to the **Shader Tree**.
4. Set **Target Meshes** to the low-resolution mesh.
5. Set **Source Meshes** to the high-resolution mesh.
6. With the **Bake Item** selected in the **Shader Tree**, **Ctrl**+click the two image layers to select them.
7. In the **Texture Bake Item** tab in the bake item's **Properties**, hold **Alt** and click **Set Selected** for the **Texture Outputs**.



Target Meshes (none)

Source Meshes (none)

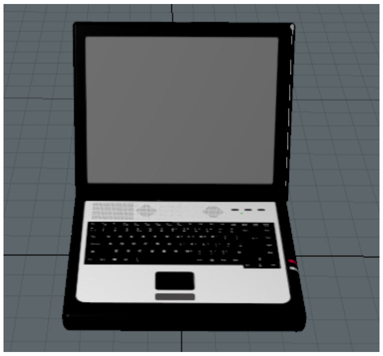
Texture Outputs ...

Texture Outputs is now set to **(mixed)** to indicate a multiple assignment.

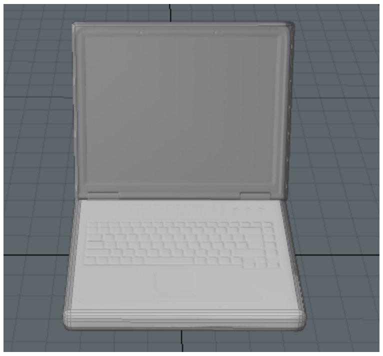
8. Check **Bake from Source**.
9. In this case we are not using a **Cage**, so set the **Distance** to a suitable length, say **200 mm**.
10. We are now ready to bake. Click **Bake Selected** in the **Properties**.

    The **Render Display** opens and the images are baked in sequence.

You can now save the images and apply them to the low-resolution mesh, as shown below:



Low poly mesh with baked **Diffuse Color** texture.



Low poly mesh with baked **Normal Map**.

Baking Render Output Bake Items

The second type of **Bake Item** is the **Render Output Bake Item**. This operates in a similar way to a **Texture Bake Item**, but it bakes to **Render Outputs**, which can be saved and re-applied to the mesh. Common uses include **Ambient Occlusion Maps** and **Final Color Maps**

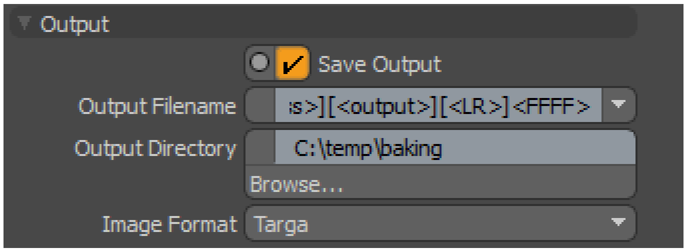
Render Output Bake Items use the following items:

- **Target Meshes** - One or more target meshes for the bake.
- **Source Meshes** - One or more source meshes to provide content for the bake.
- **Render Outputs** - A **Render Output** layer to receive the bake.
- **UV Map** - The target UV map.

Note: Unlike **Texture Bake Items**, **Render Output Bake Items** are baked in a single pass, and not in sequence. The results can be selected using the **Output** dropdown in the **Render Display**.

Output

You can choose an output filename and directory for the baked output. The bake is automatically saved to this directory.



Output

☑ Save Output

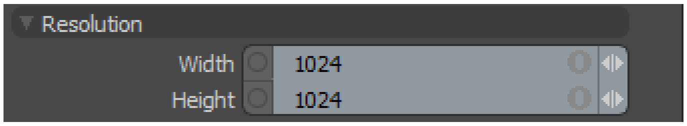
Output Filename: s>][<output>][<LR>]<FFFF>

Output Directory: C:\temp\baking

Image Format: Targa

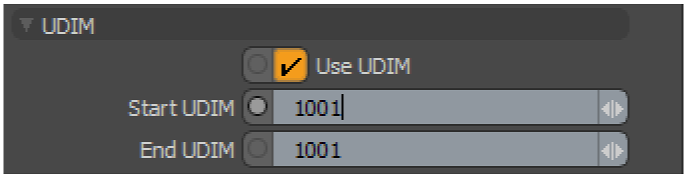
Render Output Bake Item: Output	
Save Output	Check this to save the output to the given filename and directory.
Output Filename	Choose a pattern for the filename.
Output Directory	The destination directory for the bake result.
Image Format	The image type.

Resolution


Resolution
Width 1024
Height 1024

This determines the pixel resolution of the output.

UDIM


UDIM
☑ Use UDIM
Start UDIM 1001
End UDIM 1001

If you're using UDIMs, you can bake multiple tiles by checking **Use UDIM**, and entering the range as defined by **Start UDIM** and **End UDIM**.

Note: For more information on using UDIMs, see UDIM Workflow

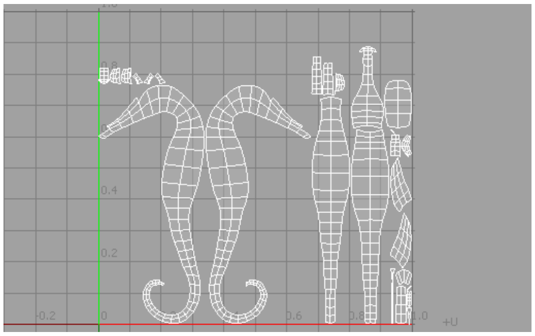
Baking to Render Outputs

You can use a **Render Output Bake Item** to bake surface texturing to one or more of the Render Outputs

You can choose one or more **Target Meshes**, one or more **Render Outputs**, and one or more **Source Meshes**. You must also choose a **UV Map** from the target(s).

Note: For more information on this type of baking, see Bake to Render Outputs.

Let's say we have a detailed mesh (the seahorse below) and we want to bake out the **Ambient Occlusion** and the **Final Color Output**. Let's also save the files using the **Bake Item**. The mesh UV map is shown on the right.

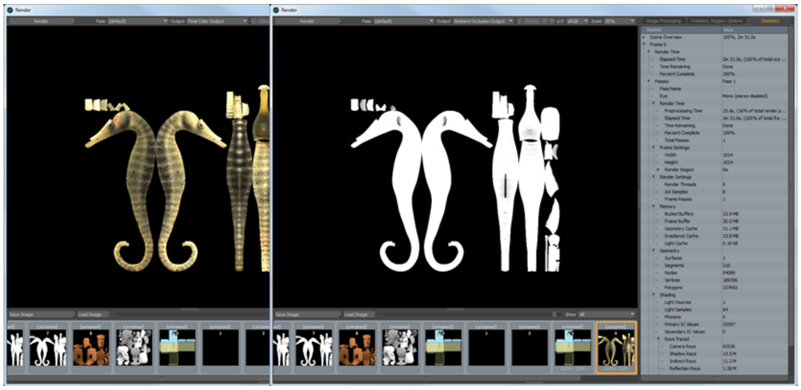


1. There must be a **Final Color Output** and an **Ambient Occlusion Render Output** in the **Shader Tree**. A **Final Color Output** is present by default. To add an **Ambient Occlusion** output, click **Add Layer** and select **Render Outputs** > **Lighting** > **Ambient Occlusion**.
2. Click **New Render Output Bake Item** to add a **Bake Item** to the **Shader Tree**.
3. With the **Bake Item** selected, **Ctrl**+click the **Final Color Output** and the **Ambient Occlusion Output** in **Shader Tree**.
4. In the **Texture Bake Item** tab in the bake item's **Properties**, hold **Alt** and click **Set Selected** for the **Render Outputs**.
5. Set the **UV Map** to use for the bake. This is the UV map pictured above.
6. Check **Save Output**.
7. To save both **Render Outputs** to separate files, set the **Output Filename** to **[<pass>][<output>][<LR>]<FFFF>]**.

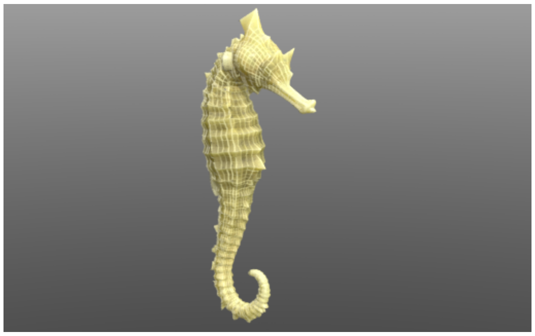
Note: The presence of the **[<output>]** tag ensures a different file is produced for each **Render Output** type.

8. Click **Browse...** and choose the **Output Directory**.
9. Click **Bake Selected** to fire the bake.

   The **Render Display** opens and the bake begins. When complete, the outputs are saved.



The image below shows the original mesh, frozen and textured with the output generated by the **Final Color Output**. This was inserted into the **Shader Tree** as an **Image Map**.



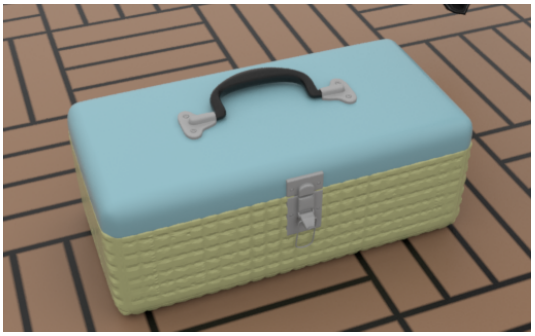
Bake from Object to Render Outputs

The **Render Output Bake Item** can be used to bake from high-resolution source meshes to low-resolution target meshes, transferring the surface from one object to another.

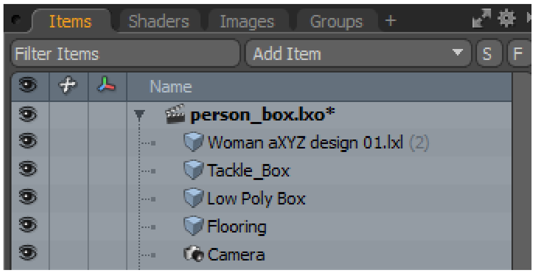
Note: For more information, see Bake to Render Outputs.

By default, the **Render Output Bake Item** bakes the entire scene to the target mesh. If you want to bake parts of the scene (such as a single character), you can assign each part to the **Source Meshes** parameter.

Let's say we have a scene with a human figure and a box, on a parquet floor as shown below. The box details is shown on the right.



The low poly box has a UV map to receive the bake. Each object is on a separate mesh layer:

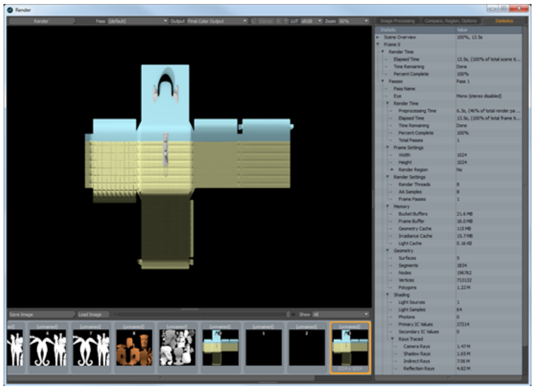


person_box.lxo*
Woman aXYZ design 01.lxl (2)
Tackle_Box
Low Poly Box
Flooring
Camera

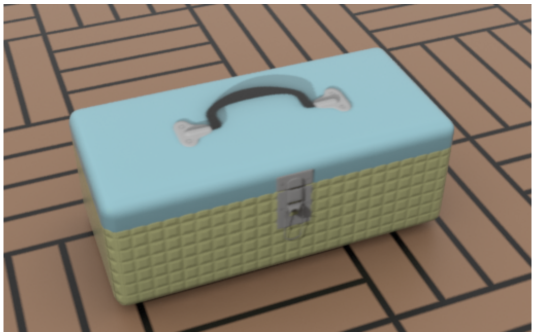
Let's use a **Render Output Bake Item** to bake the high-resolution box to a low-resolution version, ignoring the other meshes in the scene.

1. Click **New Render Output Bake Item** to add a **Bake Item** to the **Shader Tree**.
2. In the **Bake Item Properties**, select the **Render Output Bake Item** tab.
3. Select the low-resolution mesh from the **Target Meshes** dropdown.
4. Select the high-resolution mesh from the **Source Meshes** dropdown.
5. Set **Render Outputs** to **Final Color Output**.
6. Set **UV Map** to the low-resolution box UV map.
7. Check **Bake from Source**.
8. Set **Distance** to a suitable value, such as **100 mm**.
9. Click **Bake Selected** to fire the bake.

   The **Render Display** opens and the bake begins.
10. Once the bake is finished, save it by clicking **Save Image** on the **Render Display**.



The image below shows the final bake loaded into the scene and applied to the low-resolution mesh.

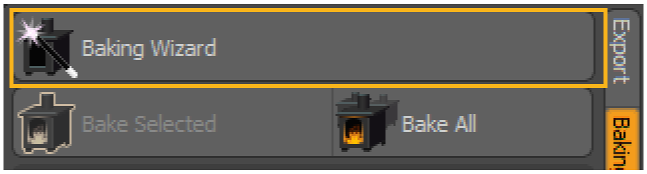


Using the Bake Wizard

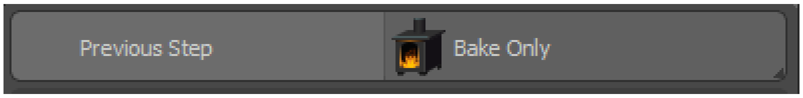
Modo provides a **Bake Wizard** to guide you through the baking process. You need to have the source meshes, target meshes, and UV maps in place (depending on the bake type) before using the wizard.

In addition to replicating the features offered by the **Bake Item**, the **Bake Wizard** lets you auto-generate outputs as required. This means you don't have to add the output to the **Shader Tree** in advance. If you use this feature, you can assign the output to a target **Material**, and Modo automatically slots the baked output into the **Material**.

Click **Baking Wizard** on the **Baking** tab in the Toolbox to open the wizard.

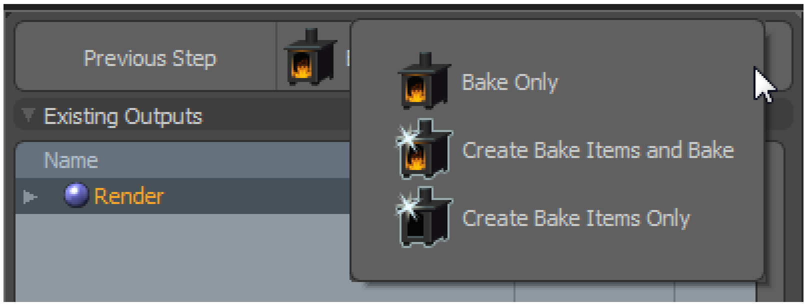


The **Bake Wizard** has four pages. Configure each page and step through using the **Next Step** button. Each page is explained in the following sections. Once configured, click **Bake Only** to fire the bake.


Previous Step
Bake Only

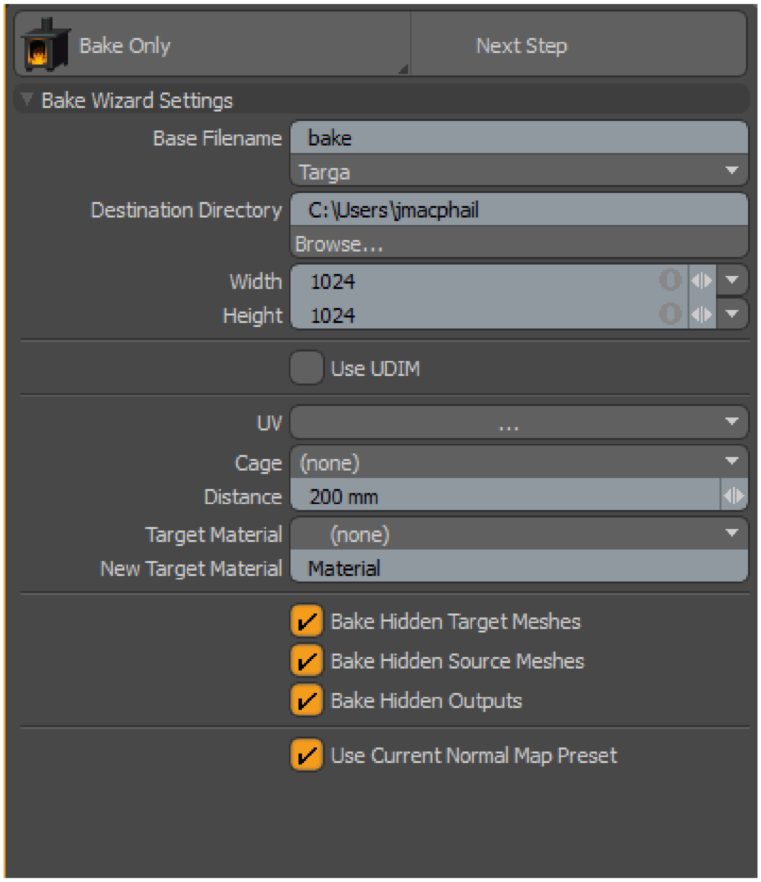
Bake Wizard - Step One - Mode of Operation and Settings

The **Bake Wizard** dialog has three modes of operation, determined by the selection in the top-center button. Right-click on Bake Only to view the option and select the mod of operation you want to use.


Previous Step
Existing Outputs
Name
Render
Bake Only
Create Bake Items and Bake
Create Bake Items Only

- **Bake Only** - Just bake, don't generate a **Bake Item**.
- **Create Bake Items and Bake** - Bake and add a configured **Bake Item** to the **Shader Tree**.
- **Create Bake Items Only** - Don't bake, but add a configured **Bake Item** to the **Shader Tree**. You can then use the **Bake Item** to perform the bake.

Set **Bake Wizard Settings** and click **Next** to move to the next step.



Base Filename: bake
Targa
Destination Directory: C:\Users\jmacphail
Width: 1024
Height: 1024
☐ Use UDIM
UV: ...
Cage: (none)
Distance: 200 mm
Target Material: (none)
New Target Material: Material
☑ Bake Hidden Target Meshes
☑ Bake Hidden Source Meshes
☑ Bake Hidden Outputs
☑ Use Current Normal Map Preset

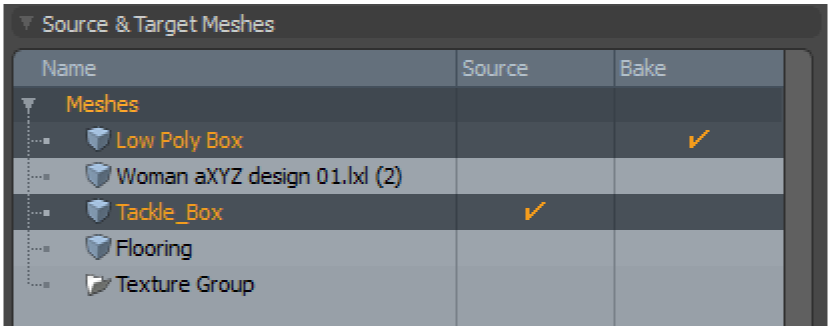
Bake Wizard Settings	
Base Filename	The filename prefix. You can also choose the preferred image format.
Destination Directory	Baked image destination directory. By default, it is set to your **$HOME** directory.
Width	Allows you to specify the width.
Height	Allows you to specify the height.
Use UDIM	Check this if you are using UDIMs. If checked, enter the **Start** and **End** UDIM tile positions. For more information, see UDIM Workflow.
UV	The UV Map that is used for the bake.
Cage	If you are using a **Morph Map** as a **Cage**, select this here.
Distance	If you are not using a **Cage**, set **Distance** to the preferred ray offset.

Target Material	The wizard lets you auto-generate a texture output or **Render Output**. This setting determines the destination material.
New Target Material	Set the name a new material group to be created for grouping any new outputs.
Bake Hidden Target Meshes	Bake to target meshes that are hidden.
Bake Hidden Source Meshes	Include all hidden source meshes in the bake.
Bake Hidden Outputs	Bake to hidden outputs.
Use Current Normal Map Preset	Texture Bake Items have an option to use the current normal map preset, which means that the selected tangent basis is automatically applied and any generated normal maps are set up according to the current normal map preset's settings before baking happens.

Bake Wizard - Step Two - Source & Target Meshes

This page lets you select source and target meshes. Click in the **Source** column to toggle the mesh as a source. Click in the **Bake** column to toggle the mesh as a target.

Once done, click **Next** to move to the next step.



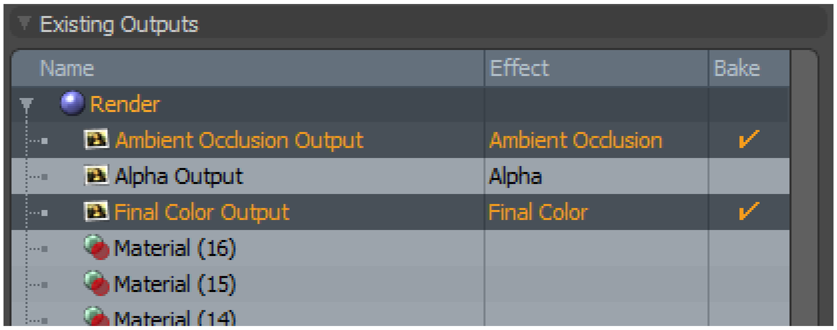
Source & Target Meshes

Name	Source	Bake
Meshes		
Low Poly Box		✓
Woman aXYZ design 01.lxl (2)		
Tackle_Box	✓	
Flooring		
Texture Group		

Note: You can select multiple sources and targets.

Bake Wizard - Step Three - Existing Outputs

This page lets you select outputs from those already in the **Shader Tree**. Click in the **Bake** column to toggle the output selection.



Existing Outputs

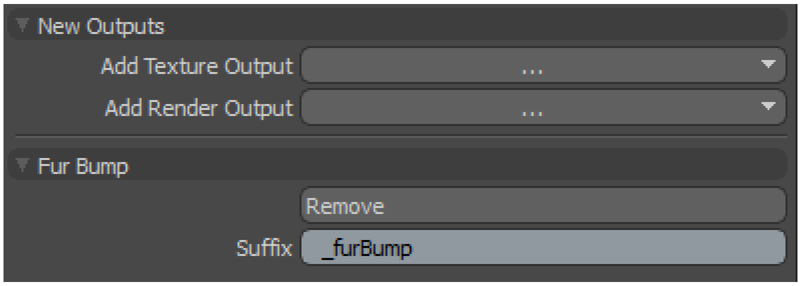
Name	Effect	Bake
Render		
Ambient Occlusion Output	Ambient Occlusion	✓
Alpha Output	Alpha	
Final Color Output	Final Color	✓
Material (16)		
Material (15)		
Material (14)		

Bake Wizard - Step Four - New Outputs

This page lets you can create new outputs to receive the bake using pre-configured output types. This provides a convenient alternative to adding items in the **Shader Tree**.

Once selected, each selection is listed in the Bake Wizard dialog.

Once added, the output appears in the page. The following is an example of the texture Fur Bump added.



New Outputs

Add Texture Output: ...

Add Render Output: ...

Fur Bump

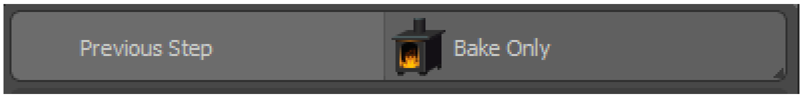
Suffix: _furBump

Each output has the following controls:

- **Remove** - delete the new output.
- **Suffix** - an identifying string to add to the new output name in the **Shader Tree**.

Each output may have additional parameters. For example, if you add an **Ambient Occlusion** preset, you can also edit the **Rays** and **Range** parameters.

Once done, click **Bake Only** to fire the bake operation.

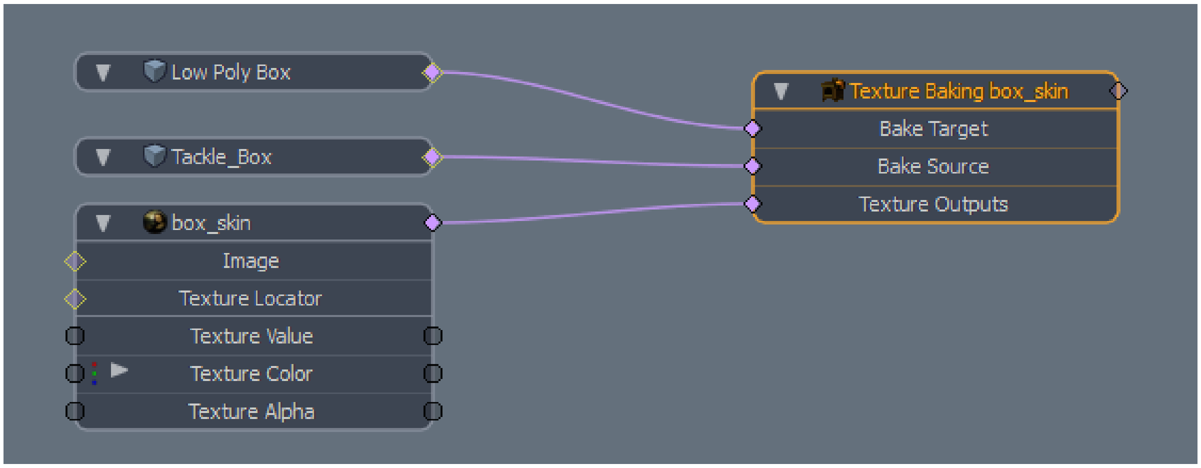


The Render Display window displays your results.

Schematic Nodes: Bake Items

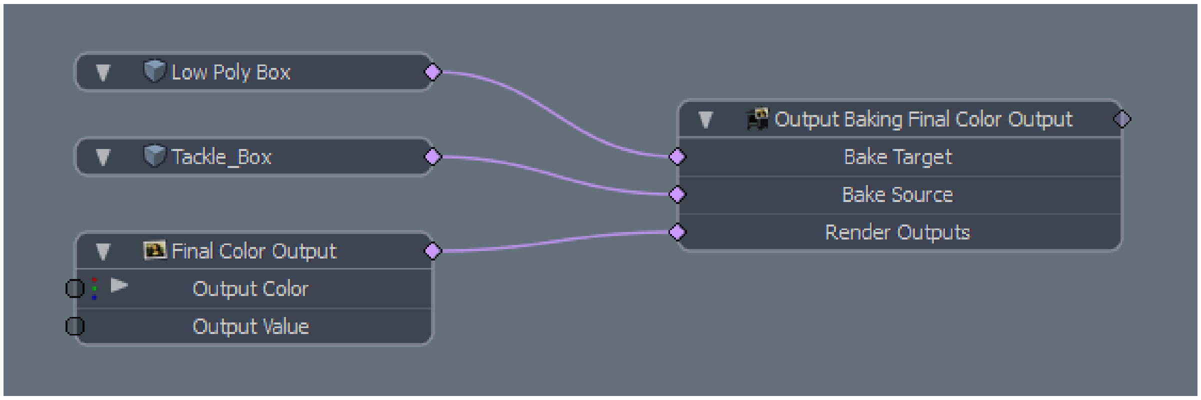
Texture Bake Items and **Render Output Bake Items** are also available in the **Schematic Viewport** under the **Bake Items** category.

Schematic Node: Texture Bake Item


Low Poly Box
Tackle_Box
box_skin
Image
Texture Locator
Texture Value
Texture Color
Texture Alpha
Texture Baking box_skin
Bake Target
Bake Source
Texture Outputs

Texture Bake Item Node	
Bake Target	Target meshes.
Bake Source	Source meshes.
Texture Outputs	Bake **Texture Outputs**.

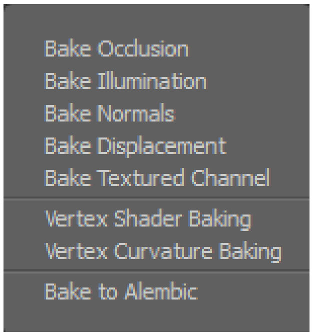
Schematic Node: Render Output Bake Item


Low Poly Box
Tackle_Box
Final Color Output
Output Color
Output Value
Output Baking Final Color Output
Bake Target
Bake Source
Render Outputs

Render Output Bake Item Node	
Bake Target	Target meshes.
Bake Source	Source meshes.
Render Outputs	Bake **Render Outputs**.

Creating Baked Items using the Mesh Operations list Bake Tools

From the contextual menu in the **Items List**, you can enable baking of certain mesh features. These bake options start a process of generating UVs, packing them, completing a bake (through a bake item), and a mesh sub-item is added to the **Shader Tree** automatically. You can then save the resulting bake to an image. Internally, the images are linked to the mesh, which makes them a sub-item of the mesh. The entire bake process happens through the use of a bake item that is added to the scene and it stores among other things the name of the folder where baked textures are saved to. The following options are available.

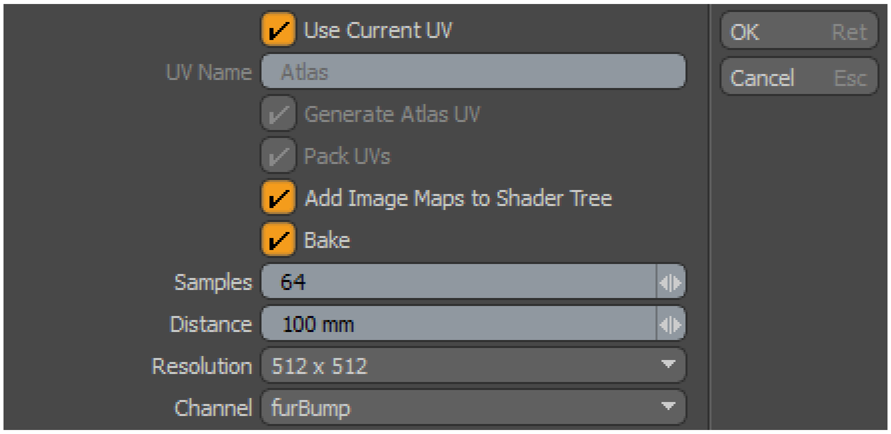

Bake Occlusion
Bake Illumination
Bake Normals
Bake Displacement
Bake Textured Channel
Vertex Shader Baking
Vertex Curvature Baking
Bake to Alembic

Bake Mesh Operation Option	**Description**
Bake Occlusion	Bakes an occlusion map. Once created, an Ambient Occlusion Output clip file is saved to the **Images** tab in the **Mesh Operations** list on the left panel. For information on the available options, see Mesh Operation list Bake Options.
Bake Illumination	Bakes an illumination map. For more information, see Vertex Shader Baking.
Bake Normal	Bakes a normal map. Modo can either bake the normal textures applied to

	the item or do object-to-object baking if the mesh item has set the high-resolution mesh item link. This link is found in the mesh item properties.
Bake Displacements	Bakes a displacement map. A Displacement Map is a grey scale image that represents the difference in position between a low-resolution foreground mesh and a high-resolution background mesh. Modo can either bake the Displacement Map applied to the item or do object-to-object baking if the mesh item has set the high-resolution mesh item link. This link is found in the mesh item properties. For more information about how to generate Displacement Maps, see Bake from Object to Render Outputs.
Bake Texture Channel	Bakes textures applied on a selected texture effect.
Vertex Shading Baking	Bakes lighting and shading data, and applies it to vertex maps. Data is stored for each vertex. For more information about vertex maps, see Working with UV Maps.
Vertex Curvature Baking	Bakes the generated vertex Weight Maps of the concavity or convexity of a surface. For more information, see Vertex Curvature Baking.
Bake to Alembic	Bakes selected items to an Alembic file. For more information, see Baking to an Alembic File.

Mesh Operation list Bake Options

After selecting the bake option you want to process from the **Mesh Operations** list, a number of options are available.



☑ Use Current UV

UV Name: Atlas

☑ Generate Atlas UV

☑ Pack UVs

☑ Add Image Maps to Shader Tree

☑ Bake

Samples: 64

Distance: 100 mm

Resolution: 512 x 512

Channel: furBump

Bake Option	Description
Use Current UV	Uses the currently selected UV Map or the top most UV Map for the mesh item.
UV Name	If **Use Current UV** is not enabled, allows you to enter the name that will be used for the UV map. By default, **Atlas** is used.
Generate Atlas UV	Generates an Atlas UV map.
Pack UVs	Packs the UVs to control whether the UV islands can be moved from their current location. (A UV island is a connected group of polygons in a UV map.) Packing the UVs automates the process of organizing UV island data to fit into the desired UV space.
Add Image Maps to Shader Tree	Adds the baked textures to the **Shader Tree**. It creates a **Bake** group in the Shader Tree and inside it an item mask for each mesh item that was baked. The baked textures are stored in those masks.
Bake	Starts the bake. If you disable that no baking will happen which can be useful if you only want to generate UV maps
Samples	Specifies the number of sample for Occlusion baking and for Illumination baking.
Distance	Bakes the distance for Occlusion, Normal and Displacement Maps.
Resolution	Specifies the resolution of the baked texture.
Includes All Surfaces	For Occlusion and Illumination baking, indicates if the bake should take into account other objects in the scene.
Channel	Bakes the texture effect channel. The following options are available: **diffColor**, **envColor**. **fullColor**, **shadeAlpha**, and **shadeColor**. **Note:** This is only available for the **Bake Textured Channel** option.

Baking to an Alembic File

You can bake a scene from an imported Alembic item. In the following example, we walk you through the process of importing an Alembic file, setting a load options, and baking the item from the **Item List**.

To bake an Alembic item:

1. Click **File** > **Open**, select your Alembic file, and click **Open**.

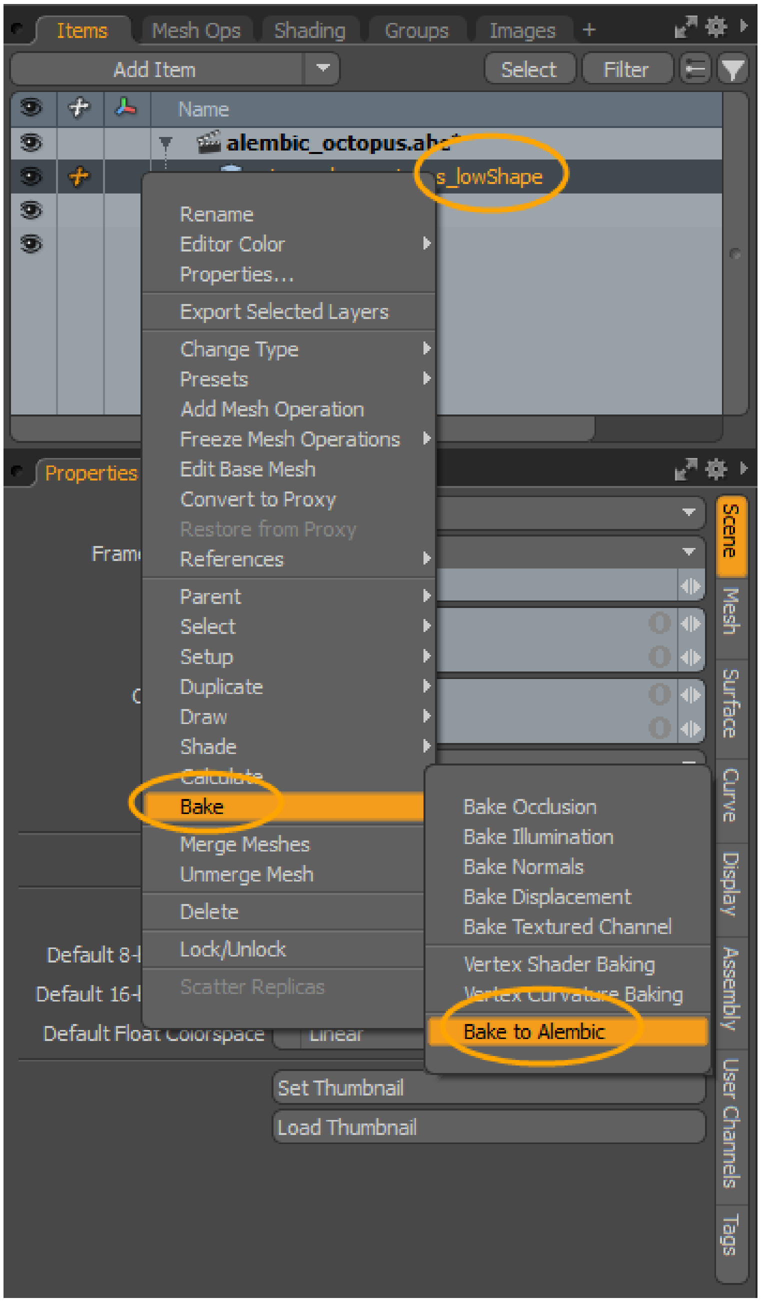
   The **Alembic Load Options** dialog displays.

2. Set **Polymesh streaming options** to **Alembic Streaming Meshes (polys only, force all)** and click **OK**.

   For more information, see **Importing Alembic Files** in the Importing and Exporting File Formats topic.

3. Make the appropriate property changes to your mesh. For more information, see Alembic Import Properties.

4. In the **Item List** on the right panel, expand the mesh item, right-click on the ***_lowShape** item, and select **Bake** > **Bake to Alembic** from the contextual menu.


Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
Name
alembic_octopus.abc
s_lowShape
Rename
Editor Color
Properties…
Export Selected Layers
Change Type
Presets
Add Mesh Operation
Freeze Mesh Operations
Edit Base Mesh
Convert to Proxy
Restore from Proxy
References
Parent
Select
Setup
Duplicate
Draw
Shade
Calculate
Bake
Merge Meshes
Unmerge Mesh
Delete
Lock/Unlock
Scatter Replicas
Bake Occlusion
Bake Illumination
Bake Normals
Bake Displacement
Bake Textured Channel
Vertex Shader Baking
Vertex Curvature Baking
Bake to Alembic
Properties
Default Float Colorspace
Linear
Set Thumbnail
Load Thumbnail
Scene
Mesh
Surface
Curve
Display
Assembly
User Channels
Tags

The **Bake to Alembic** dialog displays.

5. Enter a name for the Alembic file, navigate to the appropriate directory you want to save this file to, and click **Save**.

Network Rendering and Baking

Bake Items are designed to work with render bake network rendering, as bake parameters are fully self-contained in bake items. The items are then saved to the scene, and behave the same on all networked render workers. When baking with bake items, a specific bake can be triggered by baking a specific sequence number. This is similar to rendering a particular frame. If a scene is set up in such a way that there are 10 things to bake (let's say five assets with diffuse and normal maps), each of those bakes can be triggered by firing the baking command with a sequence number between 0 and 9. This can be handled by a render farm manager by swapping out the render commands for the relevant bake commands.

Channel Swizzling

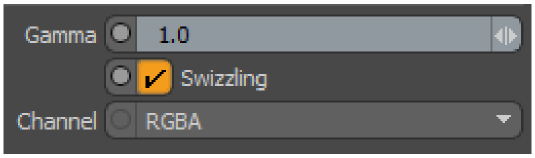
Channel swizzling allows you to assign a specific channel of RGBA to a material parameter, and set another channel to a different material parameter from the same image. In Modo, you can do this by using multiple texture layers with channel swizzling enabled.

With this feature you can set up your textures in Modo in a way that you can export them to Unity or Unreal Engine.

You can find the channel swizzling option in the **Properties** panel of a texture layer.

To enable channel swizzling:

1. In the **Shader Tree**, select the texture layer you need.
2. In the **Properties** panel, in the **Texture Layers** subtab, you can see the **Swizzling** option.



Gamma 1.0

☑ Swizzling

Channel RGBA

3. Enable the toggle, and select the channel you'd like to use in **Channel**.

   The following channel options are available:

   - **RGBA** (default) - Use all components of the color for RGBA channel
   - **RGB** - Set channel as RGB1 (alpha to 1)

- **Alpha** - Set channel as AAA1
- **Red** - Set channel as RRR1
- **Green** - Set channel as GGG1
- **Blue** - Set channel as BBB1

If **Swizzling** is disabled, the original **Alpha Channel** mode is used. The options are **Use**, **Ignore**, and **Alpha Only**.

Here's an example for channel swizzling:

You can use a texture that stores the Metallic, Ambient Occlusion, and Smoothness values in the appropriate channel:

- **R** - Metalness
- **G** - Ambient Occlusion
- **B** - Unused
- **A** - Smoothness

You can then have the same texture hooked up to different shader effects and use channel swizzling to output the correct channel. The image below shows channel swizzling from a single texture to multiple shader effects.

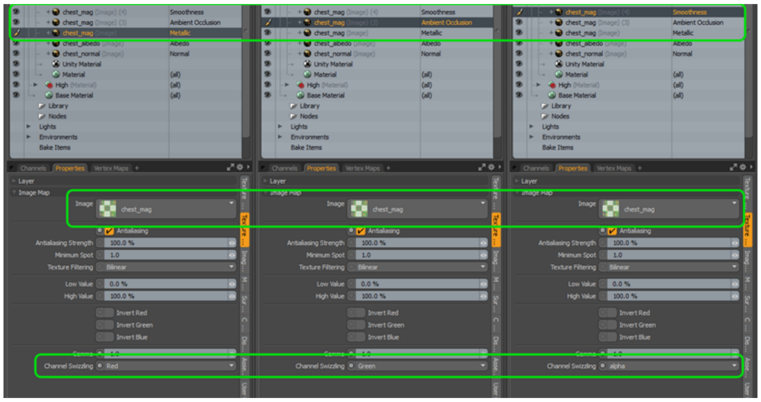

chest_mag (Image) (4) Smoothness
chest_mag (Image) (3) Ambient Occlusion
chest_mag (Image) Metallic
chest_albedo (Image) Albedo
chest_normal (Image) Normal
Unity Material
Material (all)
High (Material) (all)
Base Material (all)
Library
Nodes
Lights
Environments
Bake Items
Channels
Properties
Vertex Maps
Layer
Image Map
Texture ...
Imag...
M...
Sur...
C...
Dis...
Asse...
User

Image: chest_mag
☑ Antialiasing
Antialiasing Strength: 100.0 %
Minimum Spot: 1.0
Texture Filtering: Bilinear
Low Value: 0.0 %
High Value: 100.0 %
☐ Invert Red
☐ Invert Green
☐ Invert Blue
Gamma: 1.0
Channel Swizzling: Red

Image: chest_mag
☑ Antialiasing
Antialiasing Strength: 100.0 %
Minimum Spot: 1.0
Texture Filtering: Bilinear
Low Value: 0.0 %
High Value: 100.0 %
☐ Invert Red
☐ Invert Green
☐ Invert Blue
Gamma: 1.0
Channel Swizzling: Green

Image: chest_mag
☑ Antialiasing
Antialiasing Strength: 100.0 %
Minimum Spot: 1.0
Texture Filtering: Bilinear
Low Value: 0.0 %
High Value: 100.0 %
☐ Invert Red
☐ Invert Green
☐ Invert Blue
Gamma: 1.0
Channel Swizzling: alpha

Rendering

Rendering is the act of converting the scene information into a bitmap image by evaluating all the geometry and lighting information in a given file. Generally, you would expect the resulting rendered image to be a fully ray traced and shaded image with full shadows, reflection and the like, much like a photograph depicts an entire scene when taking a picture with a camera.

Rendering with OctaneRender

OctaneRender® is the world's first and fastest unbiased, spectrally correct GPU render engine, delivering quality and speed unrivaled by any production renderer on the market. OTOY's Octane PRIME render plugin is bundled with Modo.

OTOY® is proud to advance state of the art graphics technologies with groundbreaking machine learning optimizations, out-of-core geometry support, massive 10-100x speed gains in the scene graph, and RTX raytracing GPU hardware acceleration.

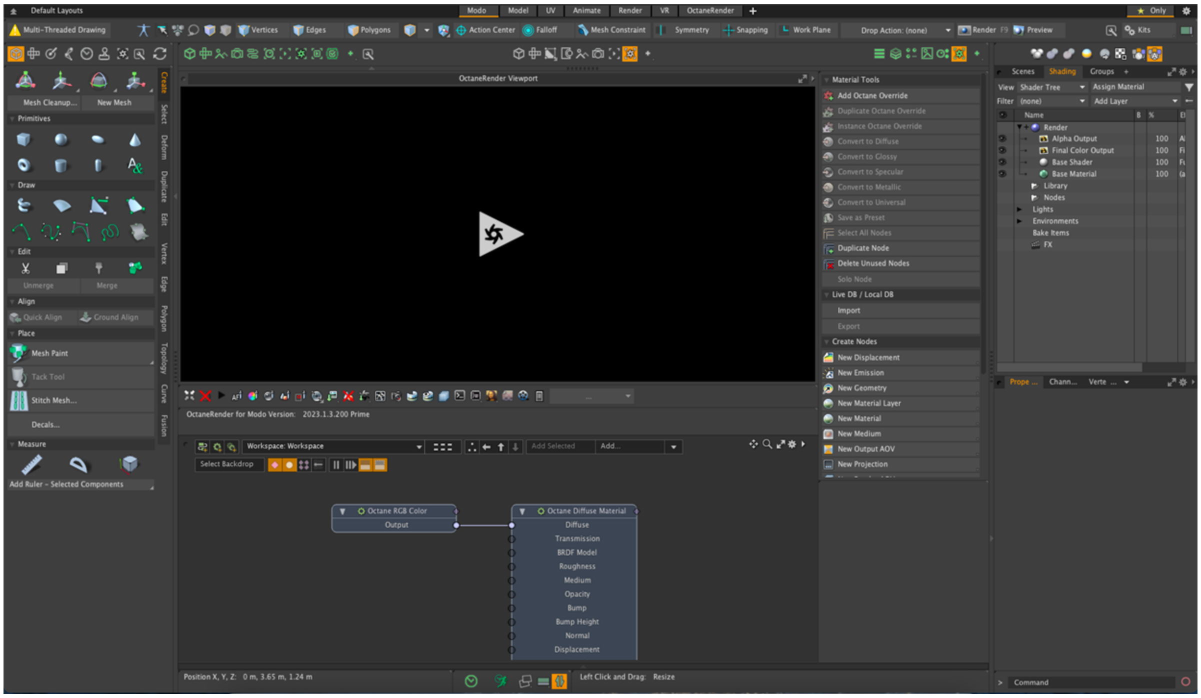
Compatibility

OctaneRender® is supported on GPU cards ranging from Kepler GPUs as the minimum requirement to RTX 4xxx GPUs on Windows and Linux, and on Apple Silicon (M1/M2/M3) hardware.

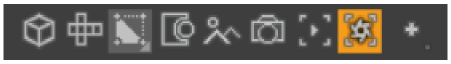
AMD and Intel GPUs are currently not supported.

The Modo Layout

The OctaneRender Prime plugin features are now available directly in the **Modo** layout, by default.

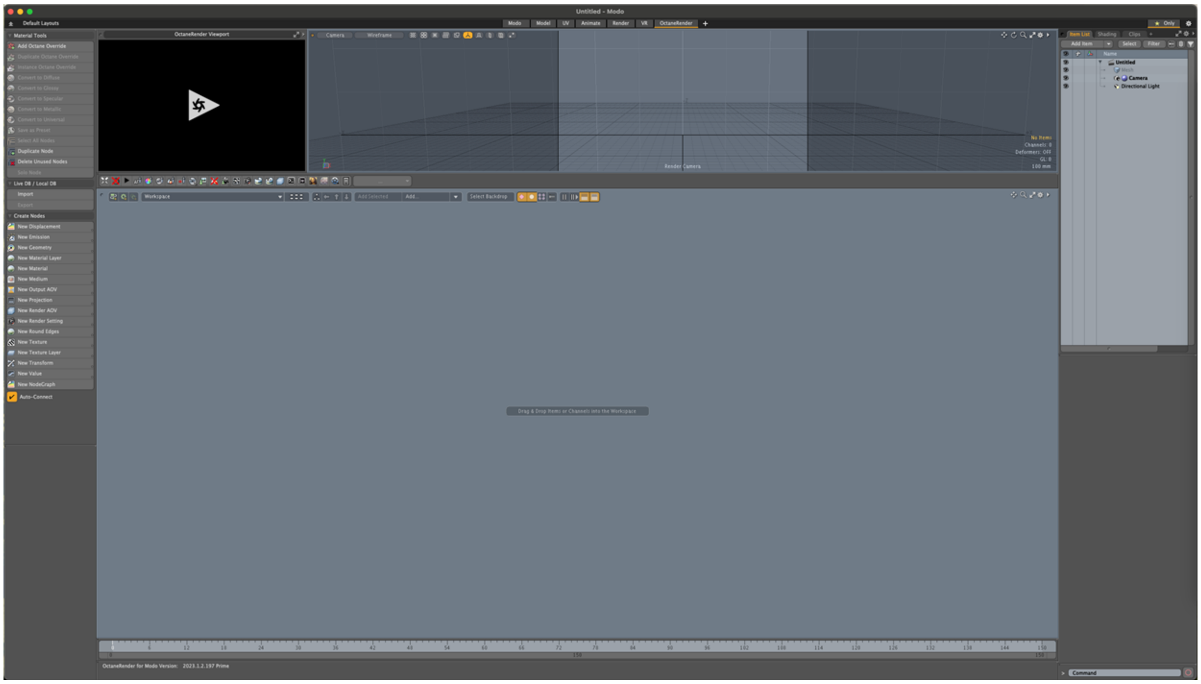


You can open it by selecting the Octane icon in the toolbar:



The OctaneRender® Layout

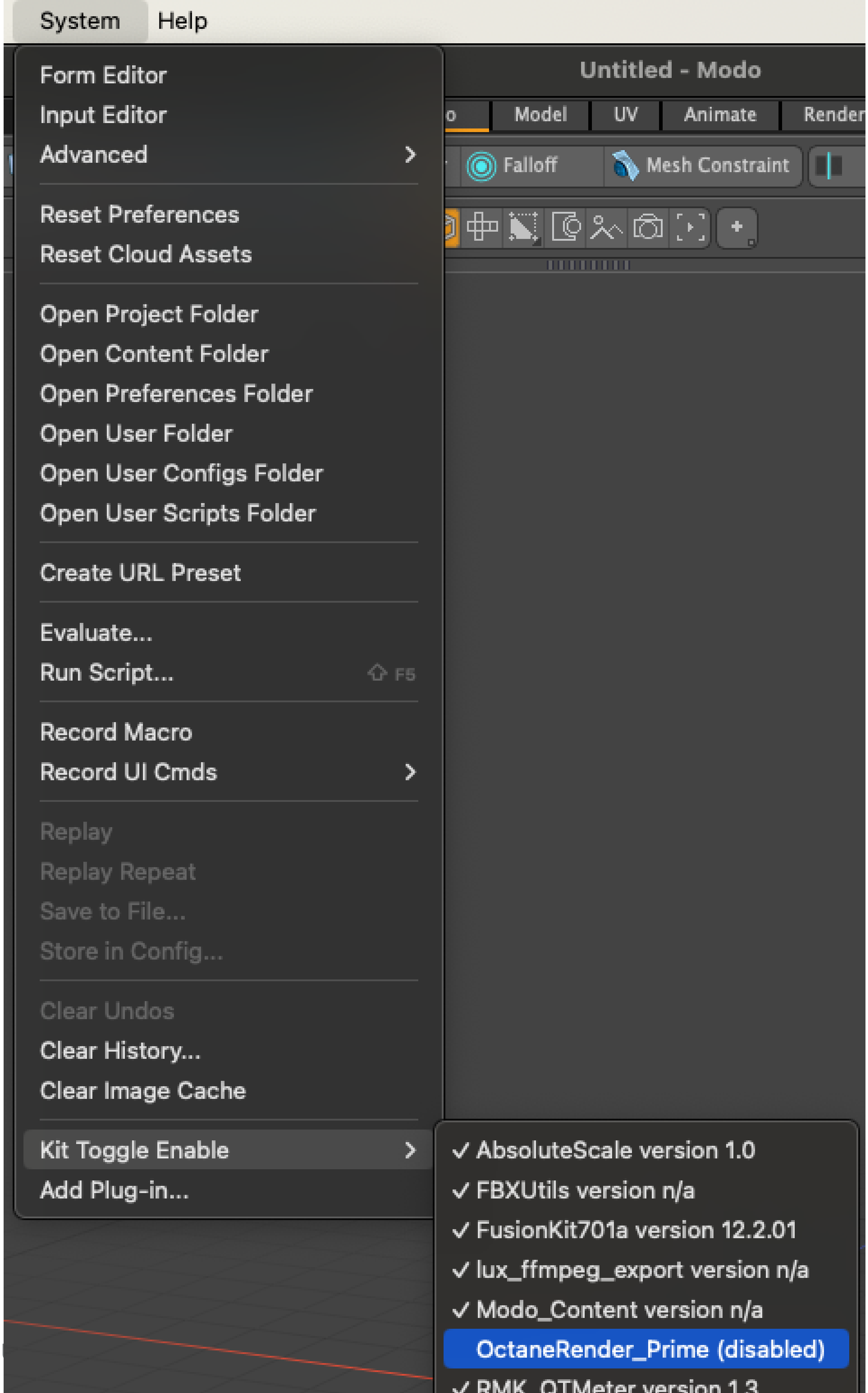
You can still access the separate **OctaneRender®** Layout if needed, by selecting the tab on the top bar.


Untitled - Modo
Default Layouts
Modo
Model
UV
Animate
Render
VR
OctaneRender
Only
Material Tools
Add Octane Override
Duplicate Octane Override
Instance Octane Override
Convert to Diffuse
Convert to Glossy
Convert to Specular
Convert to Metallic
Convert to Universal
Save as Preset
Select All Nodes
Duplicate Node
Delete Unused Nodes
Solo Node
Live DB / Local DB
Import
Export
Create Nodes
New Displacement
New Emission
New Geometry
New Material Layer
New Material
New Medium
New Output AOV
New Projection
New Render AOV
New Render Setting
New Round Edges
New Texture
New Texture Layer
New Transform
New Value
New NodeGraph
Auto-Connect
OctaneRender Viewport
Camera
Wireframe
Render Camera
No Items
Channels: 0
Deformers: OFF
GL: 0
100 mm
Item List
Shading
Clips
Add Item
Select
Filter
Name
Untitled
Mesh
Camera
Directional Light
Workspace
Select Backdrop
Drag & Drop Items or Channels into the Workspace
Command

OctaneRender for Modo Version: 2023.1.2.197 Prime

Upgrading to and Using the Octane Studio+ Plugin

If you wish to upgrade to use OTOY®'s OctaneRender Studio+ version, then you can disable the built-in PRIME kit from the **System** > **Kit Toggle Enable** menu. Enabling or disabling the kit will require a restart of Modo.


System
Help
Untitled - Modo
Model
UV
Animate
Render
Falloff
Mesh Constraint
Form Editor
Input Editor
Advanced
Reset Preferences
Reset Cloud Assets
Open Project Folder
Open Content Folder
Open Preferences Folder
Open User Folder
Open User Configs Folder
Open User Scripts Folder
Create URL Preset
Evaluate...
Run Script... ⇧ F5
Record Macro
Record UI Cmds
Replay
Replay Repeat
Save to File...
Store in Config...
Clear Undos
Clear History...
Clear Image Cache
Kit Toggle Enable
Add Plug-in...
✓ AbsoluteScale version 1.0
✓ FBXUtils version n/a
✓ FusionKit701a version 12.2.01
✓ lux_ffmpeg_export version n/a
✓ Modo_Content version n/a
OctaneRender_Prime (disabled)
✓ RMK_QTMeter version 1.3

On MacOS, since the Studio+ version is the same as the built-in PRIME version, having both enabled/installed may cause Modo to crash on launch. For this case, uninstall the external kit (remove it from the Content/Kits folder), which will allow Modo to start. You can then disable the internal "PRIME" kit and then install the external one.

Note: The OctaneRender® PRIME version is limited to a single GPU.

Article: For more information on setting up OctaneRender in Modo, please read this knowledge base article.

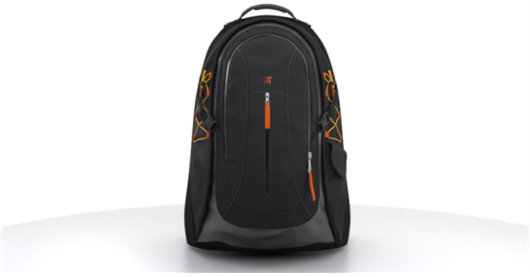
Rendering with mPath

mPath is a ground-up redesign of the Modo renderer, with dramatic reduction in controls, versatile hybrid compute capabilities, and granular refinement features. It offers many improvements over the Default renderer, including GPU acceleration for Ray Tracing with **NVIDIA OptiX** and **Metal (GPU)** to improve render performance on Apple hardware.

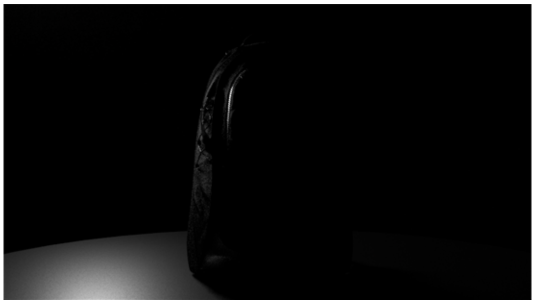
Note: Multiple GPUs are not currently supported by the mPath renderer.

If you have scenes with many lights and you want to render out separate light passes, mPath is a better option for you over the Default renderer as each light can be rendered to a separate render output as part of one single render process. This dramatically reduces render times and improves flexibility in compositing workflows.

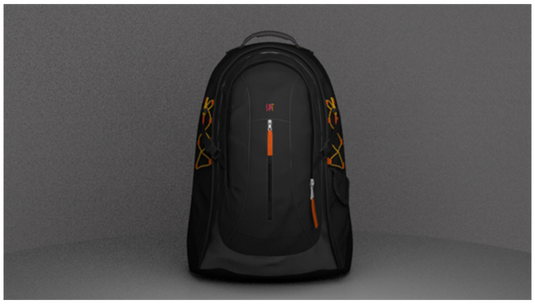
Note: See Render Outputs Per Light for more information.



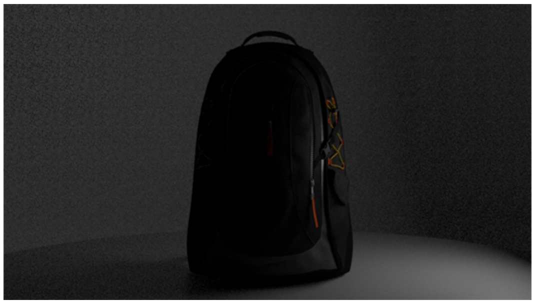
The Final Color output for this render, rendered using mPath.



The Light Path Expression output generated for the light tagged "lumi" in the light's **LPE Label**.



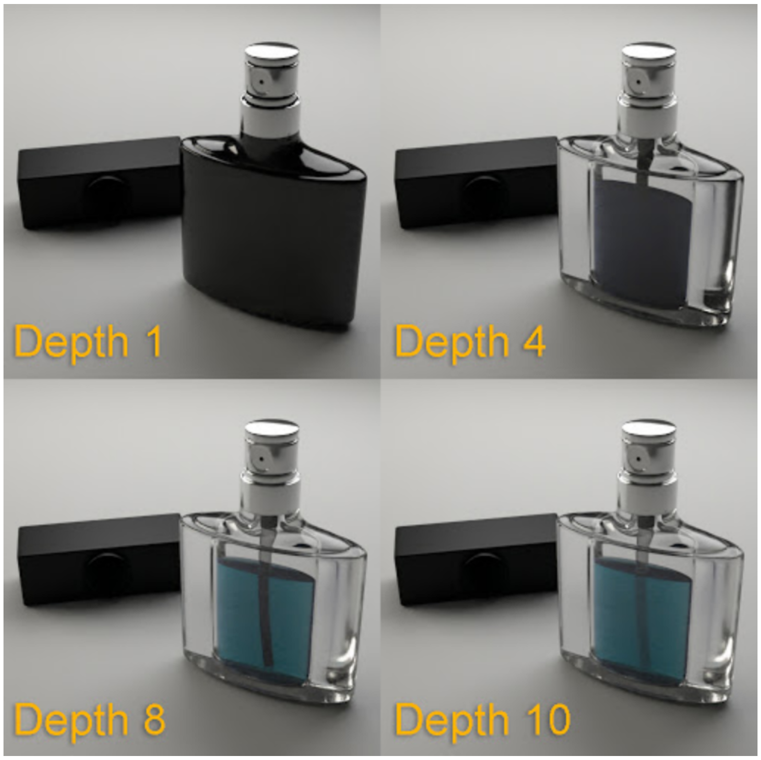
The Light Path Expression output generated for the light tagged "lumiL" in the light's **LPE Label**.



The Light Path Expression output generated for the light tagged "lumiR" in the light's **LPE Label**.

The mPath rendering workflow is also best utilized for speed improvements in Ray Tracing and Shadow Ray Tracing, and for effects like depth of field. Shading results are also more accurate with mPath, for example in Fresnel and refractive materials.

The following images demonstrate different ray depth values. Improved contact shadows, and improved reflections between close reflective components occur as Reflection Depth increases. There is increased accuracy with transparent components as Refraction Depth is increased. Accurate refractions require high depth values.



Rendering Scenes Using mPath

1. In the Shader Tree, select your **Render** item.

   This opens this render item's **Properties**.

2. In the **Frame** sub-tab, set **Renderer** to **mPath**.
3. In the **Settings** sub-tab, select your **Ray Tracing Engine**.
4. Select your **Shading Engine** from the dropdown.

   See Current Limitations in mPath for a list limitations and supported features.

> **Tip:** mPath is a hybrid renderer and can take advantage of the CPU and GPU in tandem. The default **Foundry SSE (CPU)** and **Shading Tree (CPU)** settings uses only the CPU for raycasting and shading. To use the GPU for raycasting and shading, select **NVIDIA OptiX (GPU)** and **NVIDIA CUDA (GPU)**, or to use improved performance for CPU raycasting, select **Intel Embree (CPU)**. If you're running Modo under macOS Catalina (10.15), or later, you can also select **Metal (GPU)** to improve render performance on Apple hardware.

5. Adjust the other mPath settings in the Render Item's **Settings** sub-tab.

   For more information on what the different options are, see Render Item: mPath Renderer Settings.

Note: mPath automatically uses Environment Importance Sampling if an active texture layer such as an HDR image map is affecting the environment. See Environment Importance Sampling to learn more.

6. Render your scene by pressing **F9** or along the menubar, clicking **Render** > **Render**.

   The render starts and the **Render** window opens.

Note: To decrease noise in the rendered image, we recommend using a Denoiser. See Denoising mPath Renders for more information.

Note: Press **P** during an mPath render to pause to rendering. The current render can then be saved as an image.

Interactive mPath Renders

Modo's mPath renderer has an interactive mode, which allows you to pan, zoom, and rotate the camera in the Render viewport without adjusting the camera in the scene. In interactive mode, mPath constantly renders the scene to update the output when the camera position changes.

To start an interactive mPath render, you can:

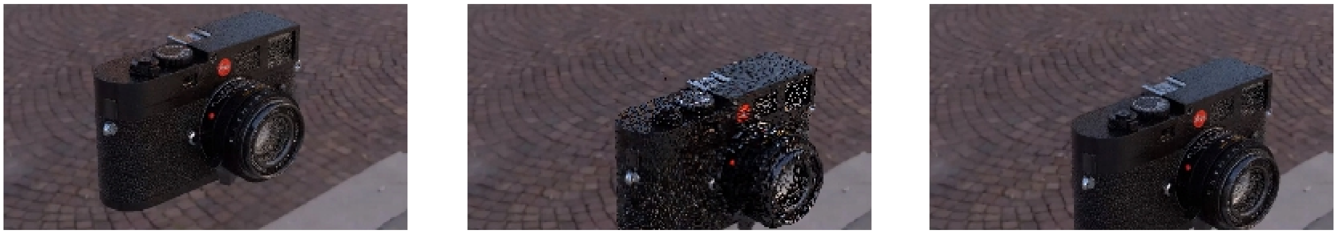
- press **Alt**+**F9**,
- navigate to the **Render** menu and select **mPath QuickCam**, or
- click and hold the **Render** button in the **Render** viewport and select **mPath QuickCam**.

Render iterations continue until you press **Abort** in the Render viewport, **Esc** on the keyboard, or **P** to pause the render.

Tip: You don't need to select **mPath** from the **Render** dropdown to start an interactive render. If you have the **Default** renderer active, Modo activates mPath for the interactive render and then switches back to **Default** automatically.

In interactive mode, you can:

- pan the camera by left-clicking and dragging in the viewport,



Panning from left to right.

- dolly in and out by scrolling the mouse wheel, and

Dollying in.

- rotate the camera around the center of interest by holding **Alt** and left-clicking and dragging in the viewport.

Tip: You can also rotate the camera directly by holding **Alt** and right-click and dragging in the viewport.

Rotating left to right.

- Adjust the Intensity of selected lights
- Adjust the Focal Length, Target Distance, Focus Distance and F-Stop of your render camera

Note: Additional settings such as light and camera settings can be accessed by pressing **Space** on your keyboard and navigating to the **QuickCam** tab.

Denoising mPath Renders

Rendering with mPath can produce noisy results due to the way path tracing resolves an image. In path tracing, the render direction of each path relies on random numbers which are different for each pixel. This results in nearby pixels having differing results from one another, creating a noisy looking image. As more samples are added to each pixel, eventually the image becomes less noisy but this can take a long time.

Using a denoiser with your mPath renders means the path tracing process can be completed before all pixels are fully converged, but you still end up with a smooth, high-quality image. This saves you a lot of time without compromising on quality. You must select a denoiser from the **Denoising** property inside your render output if you want to denoise your renders.

Tip: See Denoising for more information on the available denoisers.

To denoise your mPath renders before rendering:

1. Select your Render Output item in the **Shading** list, for example the **Final Color Output**.

   For more information, see Render Outputs and Basic Render Outputs.
2. In the Render Output **Properties**, select a denoiser from **Denoising**.

   If a denoiser is not available on your system, Modo displays a warning icon ⚠ next to it. You can still select it, allowing you to set up the scene for rendering on a different system, but a warning dialog is produced when the render is started on an unsupported system.
3. In the Render Output **Properties**, enable **Denoise Per Iteration** if you are using Render Outputs per Light.

   See Render Outputs Per Light for more information.
4. Render your scene by pressing **F9** or along the menubar, clicking **Render** > **Render**.

   The render starts and the **Render** window opens. You can verify the denoiser in the **Denoising** section in the **Statistics** panel of the Render window.

Render Outputs Per Light

With Modo's Default renderer, you have to create separate renders for each light. mPath rendering allows you to create individual light outputs within a single render by giving you the ability to add Light Path Expressions.

Tip: If you are using a **Denoiser** with your mPath render, enable **Denoise Per Iteration** in the render output settings to ensure each light pass is denoised correctly. See Denoising for more information.

Light Path Expressions

Light Path Expressions are character strings that can be used to restrict which kinds of light paths contribute radiance to a particular render output image. A path contribution is only added to a render output if its sequence of events (the vertices of the path) matches the pattern specified by the render output's Light Path Expression string.

All paths start at the camera and end at something that emits light, which can be an emissive surface, a volumetric effect, a light source, or the environment. In between the starting and ending events, zero or more scattering events can occur, in which light bounces off or passes through a surface.

To add a Light Path Expression:

Tip: To use Light Path Expressions, make sure **mPath** is selected as your **Renderer** in the **Render** item **Properties**.

1. Click **Render** > **Add Per-Light Render Outputs**.

   A **Light Path Expressions Output** is added to your **Shader Tree** for each individual light present in the scene, using the naming convention **LPE: C.*'<name of light / LPE label>'**.

Tip: Use the **LPE Label** inside the item **Properties** to specify a label to be used on that item for Light Path Expressions. LPE labels can be added to lights, materials, environments, and volumetric surfaces.

2. Select the **Light Path Expressions Output** in the **Shader Tree**.
3. Enter your Light Path Expressions in the **Light Path Expression** field in the **Properties**.

Use the following examples to specify your Light Path Expression, or use one of the presets from the **Light Path Expression** dropdown.

Starting Event Types

C signifies the camera. Since all paths begin at the camera, all valid Light Path Expressions must begin with **C**.

Scattering Event Types

Syntax	**Usage**
<RD>	Diffuse reflection event.
<RS>	Specular reflection event.
<TD>	Diffuse transmission event, which includes subsurface scattering and translucency.
<TS>	Specular transmission event, which includes refraction and dissolve.
R	A reflection event, where the path continues on the same side as the surface normal. It can be either Diffuse (**<RD>**) or Specular (**<RS>**).
T	A reflection event, where the path continues on the opposite side from the surface normal. It can be either Diffuse (**<TD>**) or Specular (**<TS>**).
D	A diffuse scattering event. It can be either Reflection or Transmission (**<RD>** or **<TD>**).
S	A specular scattering event. It can be either Reflection or Transmission (**<RD>** or **<TD>**).

Ending Event Types

Syntax	**Usage**
O	Object. An emissive object (a surface with a nonzero Luminous Amount and Luminous Color).
V	Volumetrics. Currently in Modo it is only used for light added by fog.
L	Light. A direct light source.
B	Background. In Modo this means the light from the environment item.

Special Characters

Syntax	**Usage**

. (period)	A wildcard that can match any event type.
***** (asterisk)	Zero or more instances of the preceding event type. Currently in Modo '*****' can only occur immediately after the wildcard '**.**', once per light path expression.
[ ]	An 'or' group, which can match any of the event types listed within the brackets.
' ' (single quotes)	A text label that can restrict the preceding event to one or more items that have been assigned the label appearing within the quotes. Currently, labels cannot appear within square brackets or in between the '**.**' and '*****' characters.

Examples

Syntax	**Usage**
C.*	All paths. This is equivalent to the Final Color render output.
C<RD>.*	Direct and indirect diffuse reflection.
C<RS>L	Highlights (specular reflection of direct lights).
C.*L'sun'	All shading resulting from direct lights having the label 'sun'.
C.*[OB]	All shading resulting from light from emissive objects or the background.
CDS.*	Caustics (light that has scattered from a specular to a diffuse surface).

Viewing Render Outputs Per Light

Once you have created a render using **F9** or one of the render commands in the **Render** menu, you can switch between your different light outputs in the **Render** window **Output** dropdown.

Current Limitations in mPath

There are currently a number of limitations within the mPath renderer. As these limitations are resolved, this section will be updated.

Unsupported Features in CPU Render Settings

- Volumetrics are not supported
- Shading models other than Physically Based or Principled are not supported and will give results differing form what is expected
- Custom materials such as skin, hair, and cel shading are not supported
- The Matte Clipping feature is not supported
- Direct Caustics (photon mapping) are not supported. However, indirect caustics are supported
- The Ambient Light feature is not supported
- The Vertex Irradiance Cache feature is not supported
- Negative weights in antialiasing filters are not supported, so filters that include negative weights, such as Catmull-Rom and Mitchell-Netravali, produce different results
- Shadow Density render outputs are not supported
- Multiple frame passes are not supported
- Same Surface Only subsurface scattering is not supported
- Double Sided off is not working (all surfaces are double sided in mPath)
- Keying (animating) the Renderer channel of the Render item is not supported
- Rendering a pass group with mixed renderers can cause unexpected results

Note: We recommend that you group outputs and passes for each renderer in separate pass groups and then render them with separate render commands.

mPath Limitations specific to the Intel Embree Ray Tracing Engine

- Deformation motion blur is not supported, but transformation blur is supported

Unsupported Features in GPU Render Settings

mPath Limitations specific to the NVIDIA OptiX Ray Tracing Engine

- Non-NVIDIA GPUs and older NVIDIA driver versions are not supported
- Deformation motion blur is not supported, but transformation blur works as expected

mPath Limitations specific to the Metal Ray Tracing Engine

- Motion blur is not supported
- Cylinder primitives (hair/fur) are not supported, but fur strips can be rendered with the Metal ray engine

mPath Limitations specific to the NVIDIA CUDA Shading Engine

- Motion blur is not supported
- Render booleans are not supported
- Rounded edges are not supported
- Anisotropic specular reflection is not supported
- Sheen is not supported
- Dispersion is not supported
- Image map projections other than UV are not supported
- UDIM tiles other than 1001 are not supported
- Procedural textures (other than displacement mapping) and gradients are not supported
- Volumetrics are not supported
- Light types: Photometric, Cylinder, Dome, Mesh, and Portal are not supported
- Light falloffs other than Inverse Distance Squared are not supported
- Light linking is not supported
- Fog types: Underwater and Layered are not supported
- Multiple environment items are not supported
- Environment importance sampling is not supported
- Render outputs other than Final Color, Alpha, Albedo, and Shading Normal are not supported
- Alpha types other than Opacity are not supported
- Light path expressions are not supported
- Texture Blending Modes
- Shadow Catcher
- Textured Lights
- Texture Locator Item (UV Rotation, Texture Offset Strength, Texture Falloff, and Texture Replicator)
- Scale on Material Groups

mPath Effects Only Supported by the NVIDIA CUDA Shading Engine

Basic Channels

- Diffuse Color
- Dissolve
- Roughness
- Specular Amount

- Transparent Amount
- Transparent Color

Fur

- Fur Curls
- Fur Density
- Fur Direction
- Fur Flex
- Fur Growth Jitter
- Fur Length
- Fur Root Bend
- Fur Root Frizz
- Fur Stray Hairs
- Fur Vector

Special Effects

- Stencil

Surface Shading

- Normal

Render Items

The **Render Item** always occupies the top spot in the **Shader Tree**; this is no mistake considering its importance. It can be thought of as a general container for global scene render settings. When the **Render Item** is selected, its various attributes become available in the **Properties** viewport. It is broken into three sub-tabs of related settings: **Frame**, **Settings** and **Global Illumination**:

- Render Item: Frame
- Render Item: Settings
- Render Item: Global Illumination

Render Item: Frame

The **Render** item's **Frame** tab allows you to adjust render settings specific to the rendered frame itself, such as its dimensions.

Render Options

Apply Preset	Allows you to create and save custom render settings as presets to reuse in other scenes. Apply Preset • Store Preset - Stores render item channels as a preset. Only selected channels are saved. To select channels, open the **Channels** tab and **Shift**+click on the channels you want to save. • Remove Preset - Removes the render preset. Built-in presets cannot be removed.
Renderer	Choose between Modo's **Default** renderer and **mPath**.

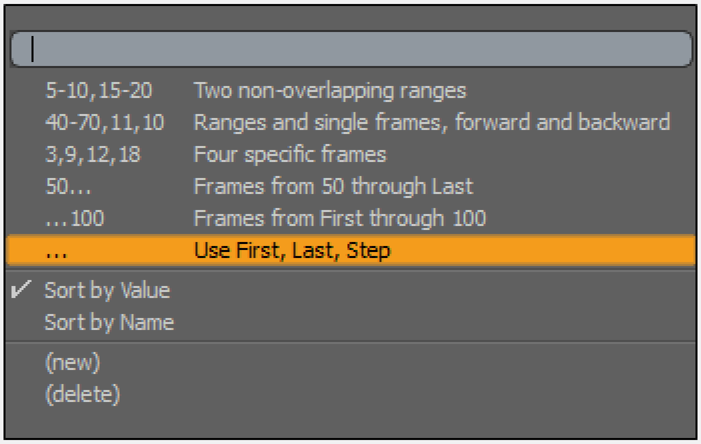
Sequence Options

First Frame/Last	In animated scenes, the frame range specifies the sequence of frames rendered when you use the **Render Animation** command. **First Frame** specifies the first frame rendered, while **Last Frame** specifies the last frame rendered.
Step	When rendering animated sequences, the frame **Step** value skips the specified number of frames. For example, when testing rendered sequences, rendering with a frame step of 2 (every other frame) cuts the overall render time in half. A frame step of 4 (every fourth frame) cuts the overall render time by three quarters and so on. This gives you a good idea of the final animated scene, albeit at a lower frame rate. This function is relative to the **Frame Range**. If you specify frame 7 as your first frame with a step of 2, frames 7, 9, 11, 13, and so on are rendered. Swapping the **Frame Range** values and setting the value to a negative number causes Modo to render frames in reverse order.

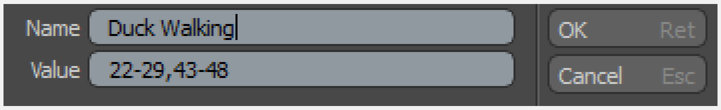
<table>
<tr>
<td>Frame Range</td>
<td>

Controls which frames are rendered during an animation rendering. You can enter in a frame range using commas or dashes to indicate specific frames or ranges of frames to render. The default ellipsis value (**...**) renders all frames between the start and end ranges.

This image shows examples of combining range numbers.


5-10,15-20 Two non-overlapping ranges
40-70,11,10 Ranges and single frames, forward and backward
3,9,12,18 Four specific frames
50... Frames from 50 through Last
...100 Frames from First through 100
... Use First, Last, Step
✓ Sort by Value
Sort by Name
(new)
(delete)

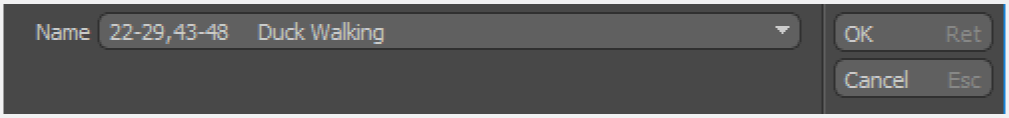
You can create a new range by clicking **(new)**, specifying the **Name** and **Value**, and click **OK**. This new preset is added to the dropdown menu.



Name: Duck Walking
Value: 22-29,43-48

OK Ret
Cancel Esc

To delete a preset range, click the down arrow, select **delete**, choose the preset, and click **OK**.



Name: 22-29,43-48 Duck Walking

OK Ret
Cancel Esc

To sort your presets, select **Sort by Value** or **Sort by Name**.

</td>
</tr>
<tr>
<td>Output Pattern</td>
<td>

These pattern options provide a means to automatically name files when saving rendered animations, as either flat or layered images. Instructions appear in the **< >** brackets, and those in the **[ ]** (square brackets) are ignored in instances that don't apply. For example:

- **<LR>** adds an **L** for Left or an **R** for Right when rendering stereo images.
- **<output>** names the file for the render output item (uses the actual name specified for the **Render Output** item in the Shader Tree).

</td>
</tr>
</table>

	• **<pass>** names the file for the Render Pass. • **<F>** is the frame number (add an extra **F** for extra leading zeros). So in practice, if you write **Robot_[<LR>]_<pass>_<FFFF>** and the resulting file name for the left render of the **BG** named Render Pass is: **Robot_L_BG_0001.exr** Where the **L** for stereo output would only be assigned when rendering in stereo. Since it is in brackets, it would be ignored for non-stereo renders. **Tip:** To disable the output pattern there is also the additional **<none>** command, which eliminates any of the file naming additions.

Frame Options

Resolution Unit	This control allows you to select how the rendered frame's image size is specified, either as pixels, inches or centimeters.
Frame Width/Height	**Frame Width** and **Height** allow you to specify the size of the frame rendered when invoking the **Render** command. Dependent upon the resolution unit, if **Pixels** is selected, then you can specify the size in pixel units, or if **Inches** is specified, you can set the printed resolution of the rendered image in inches. When set to **Inches**, the **DPI** setting is used to determine the number of pixels rendered. The typical widescreen video size of 720 pixels by 480 pixels is the default size.
DPI	Allows you to directly determine the number of **Dots Per Inch**. This is particularly useful when defining image height and width using the **Inches** resolution unit, as it is used to determine the final pixel resolution of the rendered images. It is also added to files on export, determining the eventual printed size of the image.

	Note: This option is only available in the Render Item, **Channels** viewport. You can only see this information if you have used inches or centimeters.
Pixel Aspect Ratio	Most computer screens use a **Pixel Aspect Ratio** of 1 which yields square pixels by default. However, some target media use non-square pixels to display images, such as NTSC television (0.90). Use this setting to properly render the image for its intended destination. **Note:** Modo still displays the image as if they were square pixels, so rendered images may look squashed/stretched depending on this setting.

Bucket Options

Bucket Width/Height	The **Bucket** size defines the number of pixel values calculated concurrently and is controlled by the **Height** and **Width** settings. The default value of 32 by 32 pixels means that each bucket is 32 pixels wide and 32 pixels high. Using larger buckets increases the amount of memory required but can result in speed improvement. However, if multiple threads are being used, it is important to remember that using smaller buckets can result in a better load balanced total frame. When using multiple buckets, it is unlikely that they finish rendering at exactly the same instant, therefore there is usually one bucket/thread computing when the other bucket(s)/thread(s) are finished. If the buckets are very large, the amount of time the render engine is using only one thread increases, which decreases the impact of the additional processors on final render time. You might come to the conclusion that you should set the bucket size as small as possible, but this is incorrect as well. System performance may decrease with the overhead to manage the

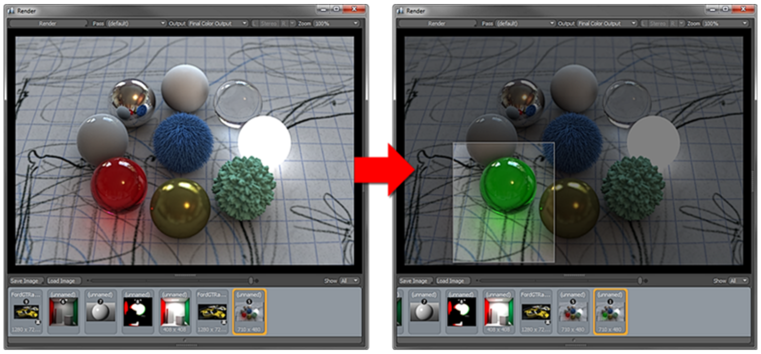
	buckets, while reducing anti-aliasing quality (as bucket borders cannot be anti-aliased to their neighboring pixels outside the bucket; decreasing the size of the bucket increases the number of border buckets). We recommend that the default bucket size (32 x 32 pixels) is typically the ideal size when rendering video resolution sized frames.
Order	**Order** determines the pattern in which the buckets are rendered. In some cases there can be a small performance benefit to certain patterns. In particular, the **Hilbert** pattern is designed to maximize bucket edge concurrency to reduce the amount of data that must be purged and loaded, whereas **Random** is most likely the least efficient in that regard. The following bucket orders are available: • **Rows** - Renders the buckets in rows from left to right, starting and the top and working down. • **Columns** - Renders the buckets in columns from top to bottom, starting on the right side and working to the left. • **Spiral** - Renders the buckets from the center spot outward, in a clockwise spiraling pattern. • **Hilbert** - Uses a specially formulated pattern that snakes across the screen making certain that there is as much concurrency from bucket to bucket across the entire frame. This pattern can result in slightly improved render performance on certain hardware architectures that are not as efficient with bus bandwidth as it reduces the amount of data being loaded and purged from bucket to bucket. • **Random** - Renders the buckets in a random order across the screen.
Reverse Order	When enabled, the order of buckets is reversed. For example, the **Columns** pattern would start at the bottom right hand side of the image and work its way bottom to top in columns towards the left of the screen.
Write Buckets to Disk	Designed to facilitate renders of enormous resolution, this option causes each completed bucket to store its frame buffer data on disk rather than in system memory.

The frame buffer of an image contains a tremendous amount of data and can become quite large, even for video resolution images. For instance a 640 x 480 pixel image requires upwards of 5 MB to hold in the frame buffer. Doubling the image resolution quadruples the number of pixels and memory requirement for the frame buffer. When rendering images for large format media, such as printed billboards or IMAX film, the frame buffer size can become unwieldy. Activating **Write Buckets to Disk** drastically reduces the memory overhead.

Note: This feature is not compatible with Network Rendering.

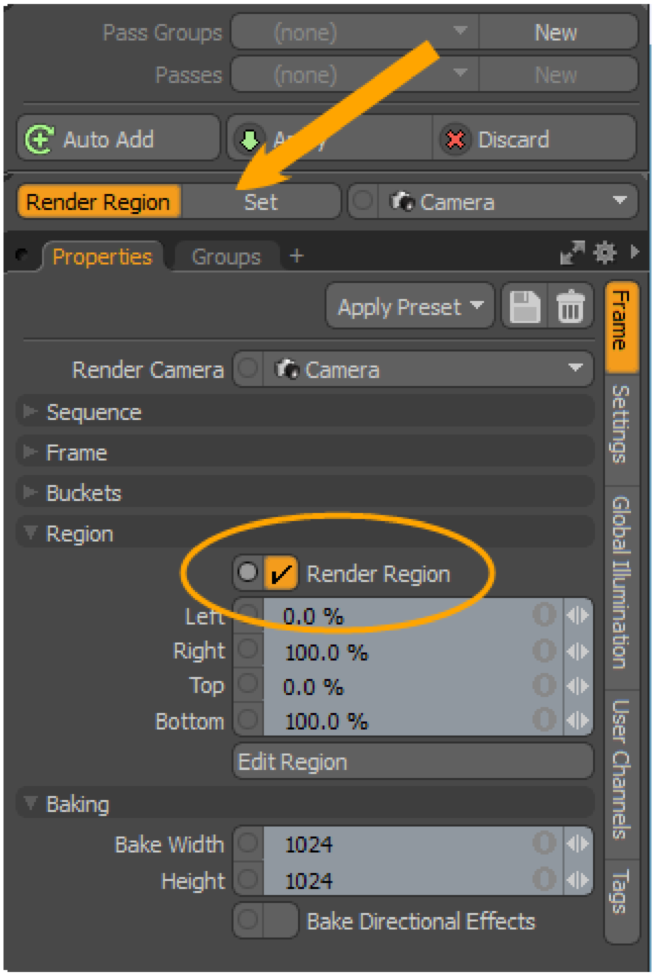
Region Options

The 3D process is one of constant iteration and refinement. Use the **Render Region** rendering functionality to refine your rendering. When enabled, the next time a render command is invoked (such as by pressing **F9**), only the area within the region specified is rendered. The function can be disabled by toggling the **Render Region** setting.



To activate and set the Render Region:

1. On the right panel, click **Render Region**.


Pass Groups
(none)
New
Passes
(none)
New
Auto Add
Apply
Discard
Render Region
Set
Camera
Properties
Groups
+
Apply Preset
Frame
Settings
Global Illumination
User Channels
Tags
Sequence
Frame
Buckets
Region
Edit Region
Baking
Bake Directional Effects

Render Camera: Camera

☑ Render Region

Left: 0.0 %

Right: 100.0 %

Top: 0.0 %

Bottom: 100.0 %

Bake Width: 1024

Height: 1024

☐ Bake Directional Effects

2. Use the **Left**/**Right**/**Top**/**Bottom** functions to specify the region manually.

Render Region	This button toggles the **Render Region** function on and off.
Left/Right/Top/Bottom	These percentage settings work in conjunction with each other to specify the total rendered region width and height. For example, you can render the left half of an image, regardless of the output resolution or aspect ratio, by setting the **Right** value to 50%. You can also set a specific area by click dragging over the rendered image in the **Render Preview** window. When you do that, the render region toggle is automatically activated. Other than disabling the function with the **Render Region** toggle, a single click anywhere on the **Render Display** restores the render to the full frame.

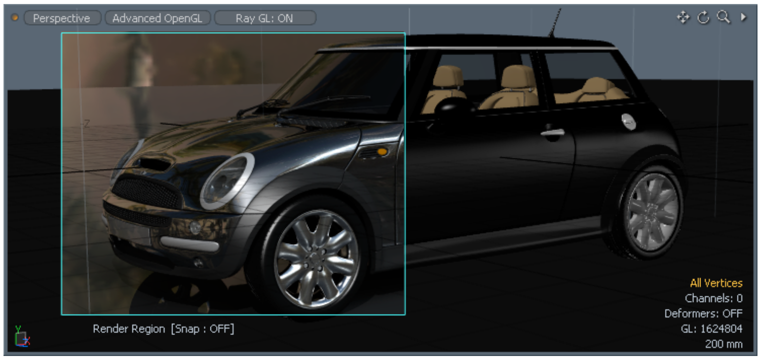
Edit Region	Activates the Render Region tool. For more details, see Using the Render Region Tool.

Baking Options

Bake Width/Height	These options control the pixel resolution of baked images, independent of the **Frame Width** and **Height** settings used for rendering. This resolution is used when invoking any of the Bake to Render Output commands.
Bake Directional Effects	When enabled, the render includes shading effects that vary with the view direction.

Using the Render Region Tool

The Render Region tool allows you to select a region within the 3D viewport to limit the RayGL rendering to a specific area of the frame. As the RayGL option is applicable to any 3D viewport, the Render Region tool further streamlines the process of iteration and refinement when creating a scene.



1. Click **Render** > **Render Region Tool**.
2. Select the command and left-click and drag within a 3D viewport to define the region.
3. Enabling the **RayGL** to limit the subsequent updates to the area inside the frame.

   Dropping the tool does not disable the limited region, so you can continue to work on a scene while limiting the region of the RayGL rendering.

To disable Render Region:

- Toggle the **Enable Option** found in the tool's **Properties** panel or disable the Render Regionoption in the **Render** item's **Frame** sub-tab.

Tip: You can adjust the quality settings of the **RayGL** option using the controls found in the Preferences panel.

Tip: The **Render Region** function is also very useful when you want to render a rather large image for print and are having difficulty doing so. Use **Render Region** to render smaller sections of the total image and assemble the pieces in an external image editing application.

Render Item: Settings

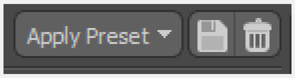
The **Render** item's **Settings** tab allows you to adjust render settings specific to rendering quality.

For settings specific to the mPath and Default renderers, see Render Item: mPath Renderer Settings and Render Item: Default Renderer Settings.

Render Item: Default Renderer Settings

The **Render** item's **Properties** tab allows you to adjust render settings specific to rendering quality.

The following Render Item properties are available on the **Settings** tab for the **Default** renderer.

Option	Description
Apply Preset	Allows you to create and save custom render settings as presets to reuse in other scenes.  • Store Preset 💾 - Stores render item channels as a preset. Only selected channels are saved. To select channels, open the **Channels** tab and **Shift**+click on the channels you want to save.

Option	Description
	• Remove Preset 🗑 - Removes the render preset. Built-in presets cannot be removed.
Renderer	Choose between Modo's **Default** renderer and **mPath**.

Legacy Renderer Options

Progressive Rendering Options

The following Render Item properties are available on the **Progressive Rendering Options** tab.

Option	Description
Convergence Target	The **Convergence Target** option is used to set the rendering percentage value used by the **Preview Viewport** for progressive final rendering. The Preview Viewport output is based on the primary render output. The default value is set to 97.5%. **Tip:** You can also set this value by clicking the **lock** icon in the **Preview Viewport**, which is set to 100% as the default. For more information about Preview locking, see Preview Viewport.
Max Render Time	Specifies the rendering time to use when saving a frame to a disk. This option is used by Preview Viewport with the default option settings enabled and with the **Render Animation** option enabled. The time is specified in minutes and it can be set to less than a minute. For example, 0.5 equals half a minute. **Tip:** You can launch the progressive rendering window by clicking **Rendering** > **Open Progressive Baking**.

Antialiasing

There are three main settings that control the quality of rendered pixels. These include **Antialiasing**, **Refinement Shading Rate**, and **Refinement Threshold**. To create the highest quality imagery, Modo does all of its computations at the sub-pixel level; taking multiple tiny samples, within the space of a single pixel, and then averaging their result. However, rather than computing the entire image this way, over and over (the "brute force" method), Modo uses elegant logic to determine how to best create the image. To balance quality with performance, Modo provides methods to determine which areas of the image should receive the greatest number of samples.

Option	Description
Antialiasing	At the top-most level, the **Antialiasing** control dictates the number of samples taken per pixel. Modo allows you to specify from 1 to 1024 samples per pixel in powers of 2 (1, 2, 4, 8, 16, and so on, up to 1024). We can think of these samples as sub-pixel fragments, imagining a pixel as a square that can be divided into smaller squares. When **Antialiasing** is set to 8, the pixel is broken down into eight pieces, and each of these pieces is evaluated and then averaged together to give us the final high-quality pixel. Since it would be expensive time-wise to evaluate every single pixel in an image repeatedly, we have some additional controls to limit the areas that receive the maximum number of evaluations.

Option	Description
	Tip: The **Antialiasing** setting's effect is most apparent on geometric edges. A setting of 8 is generally acceptable, and is a good trade-off between quality and speed, though in reproducing effects, such as **Depth of Field** or **Motion Blur**, you need to increase this value, possibly up to 1024, to produce a pleasing result. If you find your shaded edges, such as cast shadows, refractions, reflections, and procedural- and image-based textures are aliased, then reducing the shading rate value or lowering the refinement threshold helps to increase shaded edge quality. If you are finding specific areas of an image to be noisy, and it cannot be solved by **Antialiasing** and refinement options, such as with soft reflections, blurry refractions, soft shadow edges, or subsurface scattering, it may simply be a case where raising the number of samples specific to the effect eliminates the noise. These settings can be found in the associated **Properties** panels. Always keep in mind though, as more calculations are taken into account, the longer an image takes to render.
Antialiasing Filter	Allows you to choose different antialiasing filters. The **Antialiasing Filter** determines the pattern to use when evaluating a pixel. The default is **Gaussian**, which performs very well in most cases, and offers a good balance of performance and quality. **Box**, **Triangle**, **Catmull-Rom**, and **Mitchell-Netravali** are additional options. **Catmull-Rom** generally yields slightly sharper results than **Gaussian**, while **Mitchell-Netravali** offers good results when dealing with problematic moire (created by fine texture patterns).
Frame Passes	This controls how many times a single frame is rendered (not to be confused with Render Passes, which are entirely different). The results of each pass are evenly blended to make up the final frame. For example, the random numbers used to generate fur are varied for each pass, so the resulting blended fur looks softer. If motion blur is enabled, each frame pass covers just part of the exposure. If there are four frame

Option	Description
	passes, the first pass covers the first quarter of the interval from shutter open to shutter close. The random sampling for **Depth of Field** effect are also varied between passes, so additional passes help to smooth out troublesome soft focus. Increasing this value above 1 increases render times significantly, so unless you need softer fur or higher quality motion blur and depth of field, there is no added benefit to increasing this value.
Field Rendering	**Note:** The **Final Rendering** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. For scanline televisions, there were originally two half frames that made up each whole frame, which were interlaced on top of each other to make a single image. Enabling the **Field Rendering** option properly renders two discreet moments in time and interlaces the resulting half-frames into a single image (in essence, doubling the frame rate). This option is useful to compose Modo-generated elements into video that was originally captured this way. **Tip:** When using **Field Rendering**, it's recommended that you switch to **Box** antialiasing to eliminate field cross-talking. Also, post effects, such as **Bloom**, are generated on a whole-image basis, so should only be applied in a compositing package. Modern film and video capture is largely progressive, where each frame represents only a single moment of time, so in most cases **Field Rendering** is not necessary.

Shading Quality

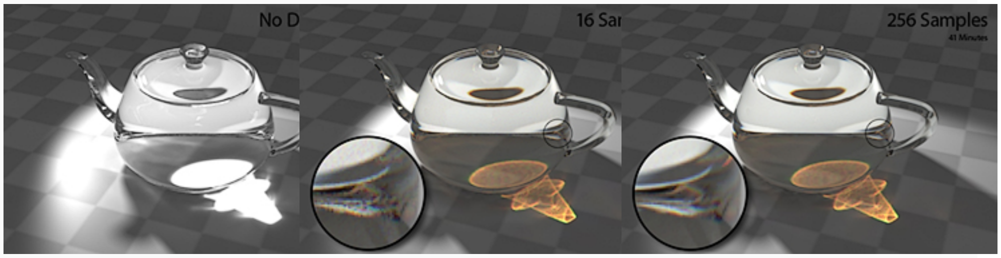
The following **Shading Quality** options are available for the **Render** item:

Option	Description
Shading Rate	Specifies the approximate spacing between shading samples, measured in pixels, used for the environment. It also acts as a cap on the rates specified in the various shader items, allowing shading detail to be increased across the entire frame. The main reason you might want to lower this value is to improve the motion blur of volumetrics. The default is 0.5. Lowering this increases shading sampling, smoothing out sharp reflections, shadows, and textures.
Refinement Shading Rate	Tells the engine how finely to evaluate any pixel that falls within the **Refinement Threshold**. When a pixel is being evaluated, first, it is rendered through the standard antialiasing application. Neighboring pixels are then tested for **Refinement Threshold** and, if a pixel is within the threshold, then the pixel is finished and the engine moves on to the next. If it's outside of the threshold, then the **Refinement Shading Rate** value kicks in, and the pixel is further evaluated. The value acts as a multiplier of the pixel's size, so a setting of 0.5 splits the pixel in half both ways, resulting in 4 additional shading samples calculated. A setting of 0.1 splits the pixel 10 times both directions, resulting in 100 additional shading samples calculated. The default value of 2.0 disables refinement shading.
Refinement Threshold	This control dictates how much contrast is acceptable between adjoining samples. Set to 100%, the **Refinement Threshold** does nothing at all, as it accepts brightness steps as high as 255 (the upper bound of a 24 bit image). At 0%, no pixels are inside the threshold, and every pixel is refined. The percent amount spans the 0 to 255 brightness steps, so a threshold of 50% accepts pixel contrasts of 128 steps, whereas a setting of 10% only tolerates 25 brightness steps between samples. Just like the refinement shading rate, we see that an increase in quality when set to lower values, but it also increases computation, requiring longer render times.
Refine Bucket Borders	Due to the "bucket" nature of Modo's render engine, adjacent pixels in neighboring render buckets can't be used to decide whether a border pixel needs further refinement, because they are either not rendered yet, or fully rendered and no longer active. By enabling **Refine Bucket Border**, Modo

Note: The Refinement Shading options are hidden by default. Click the **More** button at the bottom of the panel to display them.

Option	Description
	assumes that all border pixels in a given bucket require refinement.
Reflection Depth	Modo renders images by ray tracing, sort of like shooting tiny arrows out from the camera's lens and seeing what they hit. If a ray hits an object that is reflective, that ray bounces off the object until it hits another. Should that surface also be reflective Modo again bounces that ray until it finds a surface that is not reflective. If the ray doesn't find a surface that isn't reflective, the **Reflection Depth** threshold stops the ray from continuing on its path, keeping Modo from infinitely tracing rays between reflective surfaces. The default of 8 bounces is fine for most scenes, and is a good compromise between render accuracy (quality) and render time. In some instances, you may find a need to increase this setting to accurately portray all the reflective surfaces in a scene, or find you need to reduce it to shorten render times.
Refraction Depth	Like **Reflection Depth**, the refraction depth is calculated by ray tracing. When a ray enters a refractive surface with some transparency, the ray bends the appropriate amount and travels until it hits its next surface. If that surface is also refractive, again the ray is bent and sent along its path until it hits a non-refractive surface or is stopped by the **Refraction Depth** threshold from traveling further. The default setting of 8 is often good enough for most scenes, and is a good compromise between render accuracy (quality) and render time. In scenes with many transparent surfaces, you may find a need to increase the setting to accurately portray all your transparent surfaces, or find that you need to reduce it to shorten render times.
Ray Threshold	For some scenes, firing all those refraction and reflection rays can get expensive time-wise. **Ray Threshold** eliminates those rays that have little to no impact on the final rendered image. As rays are fired, they are given an importance value that can decrease or increase at each bounce, depending on the surface values contribution. Should the rays' importance fall below the **Ray Threshold**, Modo decides whether to kill the ray or trace it further (up until the total depth value specified). The killing of rays reduces the overall number and depth of rays traced, but has little visual impact on the final image. Increasing the **Ray Threshold** value further eliminates rays, though at the expense of image quality.

Option	Description
Maximum Radiance	Controls the maximum radiance carried by secondary rays. This can be helpful in reducing noise and preventing fireflies (single bright pixels) caused by small but very bright features of a scene, such as tight specular highlights or surfaces very close to point light sources. It acts as a multiplier for the most highly-exposed render output in the scene. Setting the **Maximum Radiance** to 10.0 (the default value) means that a ray is ten times brighter than the brightest render output. **Note:** This option doesn't clamp primary rays (anything seen directly by the camera).
Boost Importance	
Reflection Samples	A global setting used as a multiplier of the Base Material **Properties** > **Material Trans** tab, **Subsurface Scattering** > **Subsurface Amount** value and of the Base Material **Properties** > **Material Rays** tab, **Ray Tracing** > **Importance** value. Setting this parameter to 0 enables the **Reflection Rays** option for each Material item. For more information, on Reflection Rays, see Material.
Refraction	A global setting used to edit the refraction level for all materials at once. If this setting is set to 0, the material settings are used instead. **Refraction** specifies the number of rays Modo uses to calculate the material property **Refraction Roughness** setting. As the roughness amount grows, the number of rays necessary to produce a smooth effect increases, at the expense of longer render times.

Option	Description
	Note: A rough surface does not necessarily need more samples than a sharp surface. During the rendering of the **Refraction** value higher than 0, the setting is multiplied by the **Importance** value, which measures how much a particular shading calculation can affect the final pixel color. For example, if the **Reflection Samples** is set to 64 but the material being shaded is only 50% reflective, only 32 rays are fired to compute a blurry reflection. Using **Importance** in this way, allows a single **Reflection Samples** value to adapt to a variety of materials. For information about the Material **Ray Tracing** > **Importance** option, see Material.
Subsurface	A global setting that determines the contribution of diffuse shading. This option uses the **Subsurface Amount** percentage value to determine the contribution of diffuse shading made by subsurface scattering. For more information about **Subsurface Amount**, see Material.
Spectral Samples	When rendering any transparent refractive surfaces with **Dispersion**, the **Spectral Samples** value determines the number of rays that each dispersion ray is broken into. The higher the number of rays, the smoother the dispersion calculations are, at the expense of longer render times.  No Dispersion - 1 Minute 16 Spectral Samples - 3 Minutes 256 Spectral Samples - 41 Minutes
Indirect Samples	**Note:** The **Indirect Samples** option is only available when **Indirect Illumination** is enabled in the **Global Illumination** tab. See Indirect Illumination for more information. This value represents the number of samples taken for each pixel in the

Option	Description
	image when using the Monte Carlo indirect illumination model (irradiance caching disabled). Increasing this value results in higher quality at the expense of longer render times.
Environment	**Note:** The **Environment** option is only available when **Environment Importance Sampling** is enabled in the **Global Illumination** tab. See Environment Importance Sampling for more information. Determines the number of samples taken of the environment used for global illumination. More detailed environments benefit from additional samples, increasing the accuracy of the final render, but also increasing render time. On low resolution or low detail environments, you can keep this value low to reduce render time.
Irradiance Samples	**Note:** The **Irradiance Samples** option is only available when **Irradiance Caching** is enabled in the **Global Illumination** tab. See Irradiance Caching for more information. This option determines the number of Irradiance rays.

Direct Lighting Options

The following **Direct Lighting** options are available for the **Render** item:

Option	Description
Direct Light Samples	Modo uses an adaptive light sampling method that takes into account various attributes from direct light sources, such as their brightness, relative position to the shaded surface, and their intensity falloff, among other settings. These values combine to generate an overall importance amount for each light in a given scene, while still producing an unbiased result. Generally speaking, this method of working renders faster and produces higher quality shading, especially in scenes with many light sources,

Option	Description
	allowing the render engine to focus its sampling on only those lights that are most important. The main benefit being that light (and shadow) quality can be controlled by a single global setting rather than adjusting a **Samples** value per light, as done in earlier versions, simplifying the overall rendering process. The **Light Samples** value represents the total number of samples generated for all lights in a given scene. Shading noise can be decreased or eliminated by increasing this value, in exchange for longer render times. The higher the value, the smoother the shaded result. Common input values are available under the **Value Preset** pop-up. To disable adaptive light sampling, set this value to '0'. This, in turn, enables the **Samples** value per light, and works as in previous versions. In rare cases, there may be situations where individual lights don't produce the desired results. In these cases, a light's **Importance** setting can be increased or decreased depending on the desired outcome, acting as a multiplier to its evaluated importance. This setting is found under each light's **Properties** panel when selected.
MIS	**Note:** The **MIS** option is hidden by default. Click the **More** button at the bottom of the panel to display it. MIS stands for Multiple Importance Sampling and is a way of intelligently sampling direct lighting for both specular and diffuse shading, greatly reducing noise without a significant increase in render time. For most cases, the **MIS** default setting of **Both** produces the best results. In rare cases, you may wish to limit the MIS calculations to just the **Specular** shading or just to **Diffuse** shading. Setting **MIS** to **None** disables Multiple Importance Sampling, reverting to the legacy sampling method. It should be noted that **MIS** works with all shading models for diffuse shading, but MIS specular shading only works on the Energy Conserving and Physically Based models when **Match Specular** and **Blurry Reflections** are enabled for reflection shading.
Shadows	Enabled by default, the **Shadows** checkbox directs Modo to globally render all shadows cast by direct lights. Direct lights can be any of the light items,

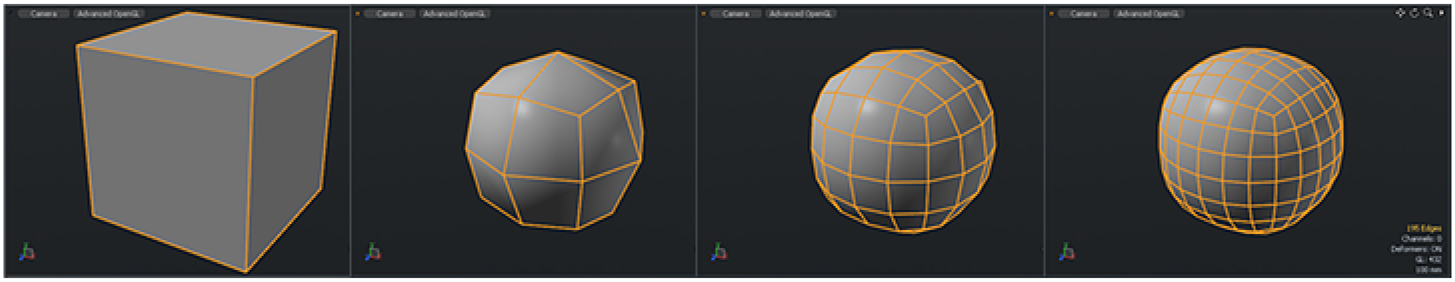
Option	Description
	such as Distant Lights, Area Lights, Point Lights, Spot Lights, Dome Lights, Cylinder Lights, and Photometric Lights. Shadows cast by luminous polygons when Global Illumination is enabled are not affected by this setting (their shading is part of the indirect illumination). Shadows can also be disabled on a per-light basis in each light's properties panel, by setting shadow type to **none**.

Geometry Options

The following **Geometry** options are available for the **Render** item:

The first two settings under the **Geometry** section are global controllers that relate to Subdivision Surface (SDS) models in a scene. **Subdivision Surfaces** is a means of producing a smooth, continuous mesh at render time, based on a low resolution proxy model (sometimes called the "cage" or **Limit Surface**). Generally, when straight polygon modeling, surfaces are made from faces - each its own tiny flat plane that, when shaded together, simulates a smooth surface - but silhouette edges can reveal the faceted nature of the model. The traditional fix for this limitation is to simply add more polygons, but controlling all those real polygons can quickly become unwieldy. For a SDS model, the polygons in the low resolution model are automatically divided and refined recursively in such a way that the resulting model is a smooth organically curving surface.

In Modo, you can toggle any object as an SDS model by simply pressing the **Tab** key. You can control the level (number of times the models is subdivided) using the **Subdivision Level** setting found in the Mesh Item properties viewport. Higher values produce smoother models, but generate more geometry that requires more processing time and memory to render. When the **Adaptive Subdivision** setting is enabled, this function overrides the Mesh Item setting at render time allowing you to adaptively control all the Subdivision Surfaces in a scene globally.

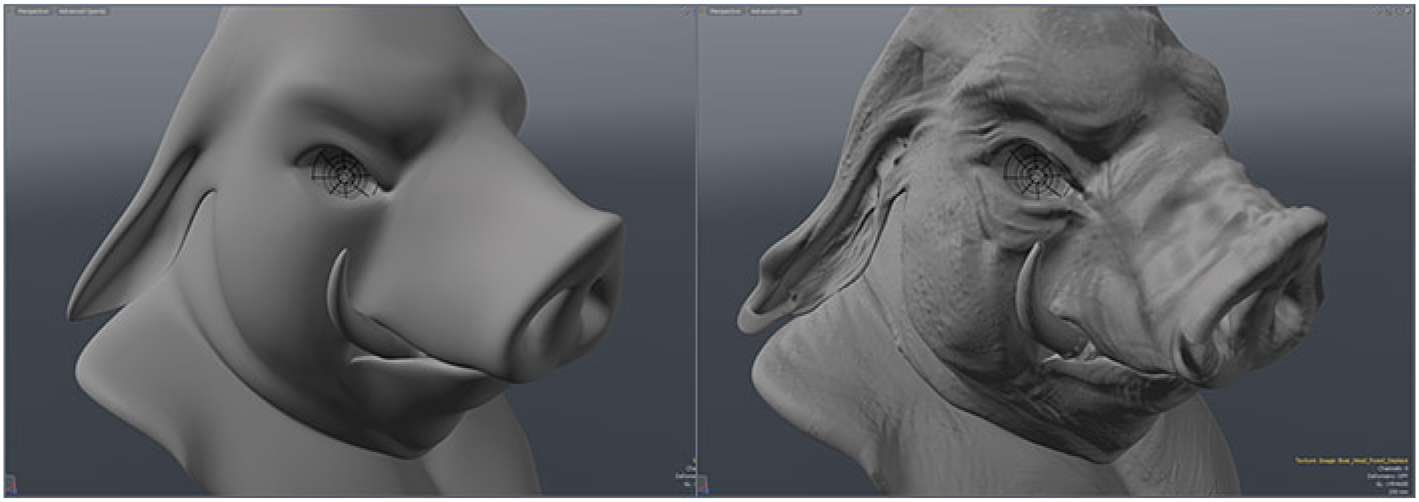


Original 'Cage' Geometry SubD level of 1 SubD level of 2 SubD level of 3

Option	Description
Adaptive Subdivision	Toggles the **Adaptive Subdivision** function in Modo. When enabled, Modo adaptively tessellates all SDS meshes at render time. Which is to say, depending on the user settings, the mesh is refined repeatedly until all polygonal edges fall within the set subdivision rate.
Subdivision Rate	When **Adaptive Subdivision** is enabled, the subdivision rate sets the level threshold for all SDS geometry in a scene. Defined as pixels, the default rate is 10, which means that Modo picks a subdivision level for that mesh, such that the length of edges for any group polygons in the largest subdivision patch appear no more than 10 pixels long (most are shorter than that). This is very useful since it means that subdivision levels automatically adapt to each mesh's distance from the camera, the zoom factor, and similar. If you really want to minimize your polygon count, you can set the subdivision rate to a huge number such as 1000. On the other hand, if you never want to see a faceted edge in a render you can reduce this number to 5 or less. Be warned that reducing the subdivision rate comes at the expense of additional system memory use and performance loss.

The next five settings control Modo's **Micropolygon Displacement** function: a means of adding fine geometric detail to surfaces by way of a texture-based controller. Similar in nature to Subdivision Surfaces, polygonal surfaces are automatically divided and refined recursively producing finer and finer micropolygons, which are then displaced (pushed in or out in the polygons normal direction) based on a grayscale value (and user settings in the Texture Locator item).

This can result in an incredible amount of detail for models that would be difficult or impossible to create otherwise. This is similar to a bump map, but where a bump map is a rendering trick that simulates detail on a surface and often looks imperfect, displacement is actual geometry. **Displacement** maps can be created in Modo, using any of the painting or sculpting tools, generated in an external application and applied to the object, or simply applied as a procedural texture layer in the **Shader Tree**. For best results, you should enable SDS for any surfaces with displacement applied. Please refer to the Effect - Texture Item topic for more information on working with displacement maps.



SDS Limit Surface

Displacement Map applie

Option	Description
Micropoly Displacement	When an object's surface has displacement applied in the **Shader Tree**, this checkbox toggles the rendering of micropolygon displacement on a global level.
Displacement Rate	When **Micropoly Displacement** is enabled, the **Displacement Rate**, sets the threshold to which the geometry is subdivided into micropolygons. Similar to the **Subdivision Rate**, micropolygon displacement is adaptive, evaluating every polygon edge separately. The distance from the camera to the center of the edge (the focal length), and the render resolution are used to convert the **Displacement Rate** (in pixels) into a distance in world space. If the edge is longer than this and also longer than the **Minimum Edge Length**, it is split in half. This process continues recursively until all edges satisfy those requirements.
Displacement Ratio	**Note:** The **Displacement Ratio** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When set to 1.0, objects are tessellated a consistent amount across their entire surface, but when displacing large areas, such as a ground plane in an environment, where a good deal of the geometry lays outside of the

Option	Description
	view of the camera, most of those polygons are going to waste, filling up your system's memory. The **Displacement Ratio** reduces the number of displaced polygons outside the camera's view. The higher the number, the greater the reduction in polygons. Keep in mind though that reflective and refractive objects still are able to see these areas outside the camera's view, so high settings may begin to introduce artifacts to your image.
Minimum Edge Length	**Note:** The **Minimum Edge Length** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. Each edge in the mesh is tessellated until polygons satisfy the **Displacement Rate** setting. Ungoverned, this can go out of control resulting in ridiculous numbers of polygons. The **Minimum Edge Length** sets a minimum length for any polygonal edge. Once an edge has reached that minimum length it can no longer be split. A simple and effective throttle for controlling the amount of polygons in your scene.
Smooth Positions	**Note:** The **Smooth Positions** checkbox is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When a surface is diced into micropolygons, the initial positions of the micropolygon vertices lie on a curved surface based on the original smooth vertex normals. So, if you start with a regular non-SDS sphere, for example, it'll be diced into a smooth sphere, and then the displacement texture is applied to that. You could even use this as an alternative to subdivision surfaces by just applying a displacement texture that is zero everywhere (although the resulting surfaces are different from SDS in that they actually pass through the original vertices). The ability to turn off position smoothing, which means that all the initial positions of the micropolygon vertices lie exactly on the flat planes of the original polygons. Doing this in the non-subdiv sphere example would cause the diced version to look faceted. The reason this feature was added was that it's necessary to turn off position smoothing when using a

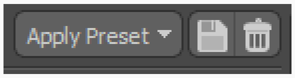
Option	Description
	displacement map created by object-to-object baking, since the distances in the map are measured based on the original polygons of the low-poly object and not the smoothed polygons.
Displacement as Bump	**Note:** The **Displacement as Bump** checkbox is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When this option is enabled, Modo also applies any displacement maps as a bump map, providing a finer detail evaluation that requires lower number of subdivisions to achieve. This can be a great way to reduce the amount of memory necessary to render complex scenes when combined with lower subdivision rates.
Indirect LOD	**Note:** The **Indirect LOD** checkbox is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When this option is enabled, Modo holds an additional lower resolution version of displaced geometry used strictly in calculating indirect lighting, reducing overhead in GI calculation. However, it may end up using slightly more memory as two distinct copies of the same geometry may be held in memory simultaneously.
Render Output Masking	

Render Item: mPath Renderer Settings

The **Render** item's **Settings** tab allows you to adjust render settings specific to rendering quality.

The following **Render** item properties are available on the **Settings** tab for the **mPath** renderer.

Tip: See Rendering with mPath to learn more about the mPath rendering workflow and denoising.

Option	Description
Apply Preset	Allows you to create and save custom render settings as presets to reuse in other scenes. You can set any of the property settings within the sub-tabs and click the Store Preset 💾 button. You can remove a render preset by clicking the Remove Preset 🗑 button.  • Store Preset 💾 - Stores render item channels as a preset. Only selected channels are saved. To select channels, open the **Channels** tab and **Shift** or **Ctrl/Cmd**+click on the channels you want to save. • Remove Preset 🗑 - Removes the render preset. Built-in presets cannot be removed.
Renderer	Choose between Modo's **Default** renderer and **mPath**.

mPath Settings

Note: mPath automatically uses Environment Importance Sampling if an active texture layer such as an HDR image map is affecting the environment. See Environment Importance Sampling to learn more.

Option	Description
Ray Tracing Engine	When **Renderer** is set to **mPath**, this allows you to choose whether to use a CPU only, or a GPU ray tracing engine. The following options are available: **Note:** A warning icon ⚠ appears next to the following options if they are not available on your current hardware. • **Foundry SSE (CPU)** - Uses only the CPU for ray tracing. • **Intel Embree (CPU)** - Offers increased performance for CPU ray tracing,

providing you have an Intel processor that supports SSE instructions. See Intel Embree for more information.

Note: Deformation Motion Blur is currently unsupported with **Intel Embree (CPU)**. An "unable to create Embree scene" warning appears if you attempt to render with this features using Embree. Use **Foundry SSE (CPU)** if you require these features.

- **NVIDIA OptiX (GPU)** - Offers increased performance, allowing for multiple buckets to take advantage of GPU raycasting simultaneously.

Note: If you see an "Unable to create context" warning while trying to render with **Intel Embree (CPU)** or **NVIDIA OptiX (GPU)**, it may imply a driver or hardware issue. See Intel Embree or NVIDIA OptiX to check for updated driver and hardware support.

Note: Modo is using the OptiX 7 ray tracing engine. Transform motion blur is supported by this engine, but deformation motion blur is not currently supported.

Tip: The **NVIDIA OptiX** option works on non-RTX cards, however true RTX hardware shows increased performance over non-RTX. Be sure to download the latest NVidia drivers, version 435 or higher.

- **Metal (GPU)** - Modo's Metal support provides GPU extension and shadow ray tracing to improve render performance on Apple hardware. You can still select this option if you don't have Apple hardware so that you can set the mPath Settings before rendering on a remote Mac.

	Note: Due to Modo's minimum supported version of macOS Catalina (10.15) the Metal ray engine does not currently support: • Motion Blur • Cylinder primitives (hair/fur). However, fur strips can be rendered with the Metal ray engine.
Shading Engine	Select the shading engine to use along side the Ray Tracing Engine: **Note:** A warning icon ⚠ appears next to the following options if they are not available on your current hardware. • **Base Material (CPU)** - shade using a default base material. This mode is useful for quick, lower quality performance comparisons to see how materials behave. • **Shading Tree (CPU)** - the default option, shade using the primary CPU. • **NVIDIA CUDA (GPU)** - shading with improved quality and speed using NVIDIA CUDA on the primary GPU, if available. You can still select this option if you don't have an NVIDIA GPU on your local machine so that you can set the mPath Settings before rendering on a remote CUDA capable machine.

Note: See Current Limitations in mPath for more information on the limitations of rendering with mPath.

Maximum Quality	Antialiasing primary quality control. Specifies the minimum and maximum number of samples per pixel. The renderer iterates through a series of passes. When **Maximum Quality** is set to a low value, you get a quick first iteration, providing a noisy preview of the scene.
Antialiasing Filter	Allows you to choose different antialiasing filters. The **Antialiasing Filter** determines the pattern to use when evaluating a pixel. The default is **Gaussian**, which performs very well in most cases, and offers a good balance of performance and quality. The other options are: • **Box** • **Triangle**

	Note: The **Antialiasing Filter** option may be hidden. Click the **More** button at the bottom of the panel to display all controls.
Noise Threshold	Specifies the degree of noise that's acceptable to determine when a pixel is skipped. If a pixel inside a render bucket achieves a satisfactory level of refinement, it is skipped for each additional iteration, speeding up the render. Higher values lead to faster but noisier renders. **Tip:** The default value of 5% is recommended, however 10-20% can be acceptable if denoising is applied afterwards.
Diffuse Depth	Specifies the maximum number of bounces for Diffuse rays before they are automatically terminated. Higher values increase render time but produce more accurate results.
Reflection Depth	Specifies the maximum number of bounces for Reflection rays before they are automatically terminated. Higher values increase render time but produce more accurate results.
Refraction Depth	Specifies the maximum number of bounces for Refraction rays before they are automatically terminated. Higher values increase render time but produce more accurate results.
Path Threshold	Eliminates rays that have little or no impact on the final rendered image. As rays are fired, they are assigned an importance value. This value can increase or decrease at each bounce, depending on the surface values contribution. If a ray's importance value falls below the **Path Threshold**, Modo decides whether to kill the ray or trace it further. Increasing this value can decrease render time at the cost of quality. **Note:** The **Path Threshold** option may be hidden. Click the **More** button at the bottom of the panel to display all controls.
Maximum Radiance	Controls the **Maximum Radiance** carried by secondary rays. This can be helpful in reducing noise and preventing fireflies (single bright pixels) caused by small but very bright features of a scene, such as tight specular highlights or surfaces very close to point light sources.

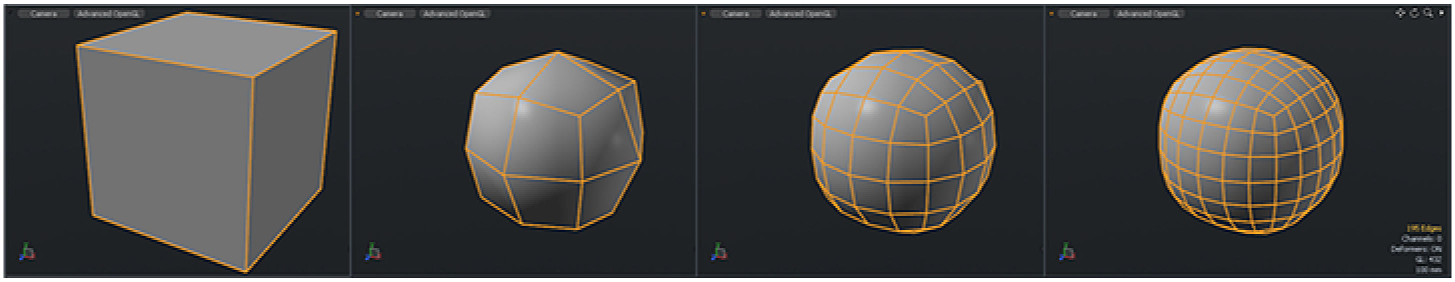
	It acts as a multiplier for the most highly-exposed render output in the scene. Setting the **Maximum Radiance** to 10.0 (the default value) means that a ray is ten times brighter than the brightest render output. It is recommended to keep this value above 1. **Note:** The **Maximum Radiance** option may be hidden. Click the **More** button at the bottom of the panel to display all controls.
Indirect Caustics	Indirect caustics are a natural rendering by-product of physically-based rendering. You have the following options to control their creation: • **None** - Disables all indirect caustics within a scene. • **Reflection Only** - Caustics are only calculated for reflective surfaces. • **Refraction Only** - Caustics are only calculated for refractive surfaces. • **Both** - Caustics are calculated for both reflective and refractive surfaces. **Note:** The **Indirect Caustics** option may be hidden. Click the **More** button at the bottom of the panel to display all controls.

Geometry

The following **Geometry** options are available for the **Render** item:

The first two settings under the **Geometry** section are global controllers that relate to Subdivision Surface (SDS) models in a scene. **Subdivision Surfaces** is a means of producing a smooth, continuous mesh at render time, based on a low resolution proxy model (sometimes called the "cage" or **Limit Surface**). Generally, when straight polygon modeling, surfaces are made from faces - each its own tiny flat plane that, when shaded together, simulates a smooth surface - but silhouette edges can reveal the faceted nature of the model. The traditional fix for this limitation is to simply add more polygons, but controlling all those real polygons can quickly become unwieldy. For a SDS model, the polygons in the low resolution model are automatically divided and refined recursively in such a way that the resulting model is a smooth organically curving surface.

In Modo, you can toggle any object as an SDS model by simply pressing the **Tab** key. You can control the level (number of times the models is subdivided) using the **Subdivision Level** setting found in the Mesh Item properties viewport. Higher values produce smoother models, but generate more geometry that requires more processing time and memory to render. When the **Adaptive Subdivision** setting is enabled, this function overrides the Mesh Item setting at render time allowing you to adaptively control all the Subdivision Surfaces in a scene globally.

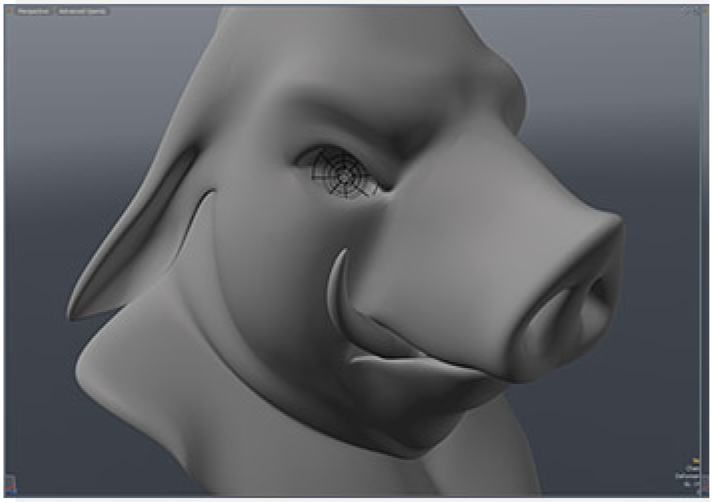


Original 'Cage' Geometry SubD level of 1 SubD level of 2 SubD level of 3

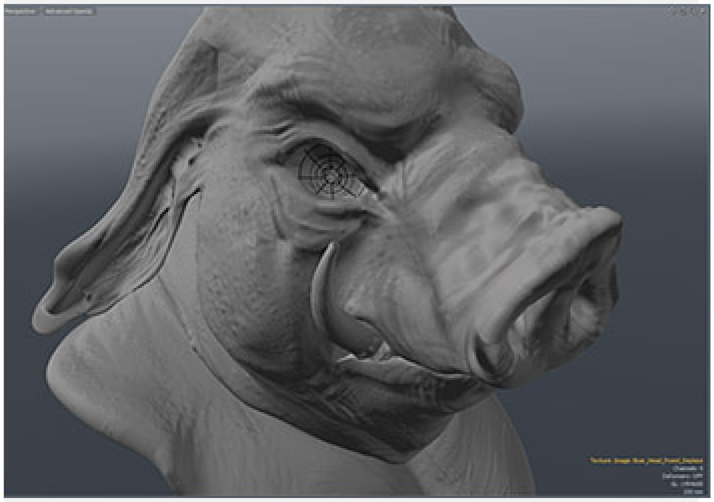
Option	Description
Adaptive Subdivision	Toggles the **Adaptive Subdivision** function in Modo. When enabled, Modo adaptively tessellates all SDS meshes at render time. Which is to say, depending on the user settings, the mesh is refined repeatedly until all polygonal edges fall within the set subdivision rate.
Subdivision Rate	Enabling **Adaptive Subdivision** sets the subdivision rate for the level threshold for all SDS geometry in a scene. Defined as pixels, the default rate is 10, which means that Modo picks a subdivision level for that mesh, such that the length of edges for any group polygons in the largest subdivision patch appear no more than 10 pixels long (most are shorter than that). This is very useful since it means that subdivision levels automatically adapt to each mesh's distance from the camera, the zoom factor, and similar. If you really want to minimize your polygon count, you can set the subdivision rate to a huge number such as 1000. On the other hand, if you never want to see a faceted edge in a render you can reduce this number to 5 or less. Be warned that reducing the subdivision rate comes at the expense of additional system memory use and performance loss.

The next five settings control Modo's **Micropolygon Displacement** function: a means of adding fine geometric detail to surfaces by way of a texture-based controller. Similar in nature to Subdivision Surfaces, polygonal surfaces are automatically divided and refined recursively producing finer and finer micropolygons, which are then displaced (pushed in or out in the polygons normal direction) based on a grayscale value (and user settings in the Texture Locator item).

This can result in an incredible amount of detail for models that would be difficult or impossible to create otherwise. This is similar to a bump map, but where a bump map is a rendering trick that simulates detail on a surface and often looks imperfect, displacement is actual geometry. **Displacement** maps can be created in Modo, using any of the painting or sculpting tools, generated in an external application and applied to the object, or simply applied as a procedural texture layer in the **Shader Tree**. For best results, you should enable SDS for any surfaces with displacement applied. Please refer to the Effect - Texture Item topic for more information on working with displacement maps.



SDS Limit Surface



Displacement Map applied

Option	Description
Micropoly Displacement	When an object's surface has displacement applied in the **Shader Tree**, this checkbox toggles the rendering of micropolygon displacement on a global level.
Displacement Rate	When **Micropoly Displacement** is enabled, the **Displacement Rate**, sets the threshold to which the geometry is subdivided into micropolygons. Similar to the **Subdivision Rate**, micropolygon displacement is adaptive, evaluating every polygon edge separately. The distance from the camera to the center of the edge (the focal length), and the render resolution are used to convert the **Displacement Rate** (in pixels) into a distance in world space. If the edge is longer than this and also longer than the **Minimum Edge Length**, it is split in half. This process continues recursively until all edges satisfy those requirements.

Option	Description
Displacement Ratio	**Note:** The **Displacement Ratio** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When set to 1.0, objects are tessellated a consistent amount across their entire surface, but when displacing large areas, such as a ground plane in an environment, where a good deal of the geometry lays outside of the view of the camera, most of those polygons are going to waste, filling up your system's memory. The **Displacement Ratio** reduces the number of displaced polygons outside the camera's view. The higher the number, the greater the reduction in polygons. Keep in mind though that reflective and refractive objects still are able to see these areas outside the camera's view, so high settings may begin to introduce artifacts to your image.
Minimum Edge Length	**Note:** The **Minimum Edge Length** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. Each edge in the mesh is tessellated until polygons satisfy the **Displacement Rate** setting. Ungoverned, this can go out of control resulting in ridiculous numbers of polygons. The **Minimum Edge Length** sets a minimum length for any polygonal edge. Once an edge has reached that minimum length it can no longer be split. A simple and effective throttle for controlling the amount of polygons in your scene.
Smooth Positions	**Note:** The **Smooth Positions** checkbox is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When a surface is diced into micropolygons, the initial positions of the micropolygon vertices lie on a curved surface based on the original smooth vertex normals. So, if you start with a regular non-SDS sphere, for example, it'll be diced into a smooth sphere, and then the displacement texture is applied to that. You could even use this as an alternative to subdivision surfaces by just applying a displacement texture that is zero everywhere

Option	Description
	(although the resulting surfaces are different from SDS in that they actually pass through the original vertices). The ability to turn off position smoothing, which means that all the initial positions of the micropolygon vertices lie exactly on the flat planes of the original polygons. Doing this in the non-subdiv sphere example would cause the diced version to look faceted. The reason this feature was added was that it's necessary to turn off position smoothing when using a displacement map created by object-to-object baking, since the distances in the map are measured based on the original polygons of the low-poly object and not the smoothed polygons.
Displacement as Bump	**Note:** The **Displacement as Bump** checkbox is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When this option is enabled, Modo also applies any displacement maps as a bump map, providing a finer detail evaluation that requires lower number of subdivisions to achieve. This can be used to reduce the amount of memory necessary to render complex scenes when combined with lower subdivision rates.
Indirect LOD	**Note:** The **Indirect LOD** checkbox is hidden by default. Click the **More** button at the bottom of the panel to display all controls. When this option is enabled, Modo holds an additional lower resolution version of displaced geometry used strictly in calculating indirect lighting, reducing overhead in global illumination calculation. However, it may end up using slightly more memory as two distinct copies of the same geometry may be held in memory simultaneously.
Render Output Masking	Supports group masks and layer masks in the **Shader Tree**.

Render Item: Global Illumination

The **Render** item's **Global illumination** tab allows you to adjust render settings specific to global illumination, a rendering technique that utilizes bounced light to produce incredibly realistic rendered images.

The following Render Item properties are available on the **Global Illumination** tab.

Note: The Global Illumination tab is only available when the **Default** renderer is selected from the **Renderer** property. This tab is not available when **mPath** is selected from the **Renderer** property.

Option	Description
Apply Preset	Allows you to create and save custom render settings as presets to reuse in other scenes. You can set any of the property settings within the sub-tabs and click the **Save Preset** button. To remove a render preset, click the **Remove Preset** button.
Render Camera	Allows you to select a specific camera from which the rendered image is created. While a basic Modo project begins with a single default camera, you can add additional cameras in the **Item List** using the **Add Item** button and selecting **Camera**. These cameras, like any item in the list, can be renamed by right-clicking and choosing **Rename**. By default, each new camera is assigned a numeric differentiator, such as **Camera(2)**, **Camera(3)**, and so on. Once additional cameras have been added to a project, the **Render Camera** dialog is used to select them. Additionally, the current camera can be animated to switch to different views during a sequence or select different cameras for individual Render Passes.

Ambient Light

The following **Ambient Light** options are available:

Option	Description
Ambient Intensity	Adds an even illumination globally to all diffuse surfaces in a scene. This is most useful for simulating additional bounces of light. Some 3D artists recommend driving this value very low or off altogether and favor adding more complex lighting scenarios or using actual global illumination bounces to achieve adequate light fill. However, using a little ambient light can be a very useful optimization to any scene; if setup properly, it can look very natural and at very little cost in render time. Especially when compared to additional lights and indirect illumination bounces. The ambient intensity control is a real world radiance value for lighting simulation, and simply acts as a multiplier for the ambient color. You can change it to show luminance units (candela per square meter) in the Input Preferences. Start with the default value of .05 W/srm and then work the value up or down until the desired lighting result is achieved.
Ambient Color	This function allows you to specify a color value for the global light added by the **Ambient Intensity** setting. Choosing a good ambient color and intensity is very important when using the **Ambient Intensity** feature. If you consider that this setting is trying to simulate the effect of the accumulation of all the scattered light rays in a project, it is easier to imagine how to choose the color and intensity. To assist the color selection, you can decrease the **Ambient Intensity** to zero and render your image. Look at all the pixels and their brightness and think about what the average overall tone or color is. That color is what you should choose for the **Ambient Color** control.

Indirect Illumination

Traditionally, 3D rendering applications have used direct light sources to illuminate a scene, various virtual lights would be defined, such as spot light or a point light, and rays could be traced from their origin. Based on their position, it was simple to calculate if a surface was illuminated or in shadow. This rendering method was fast and accurate but didn't produce the most realistic results. Light in the real world isn't static, but actually bounces around in an environment, absorbing and reflecting the colors each ray encounters. A green ball placed next to a white wall picks up some subtle color bleeding between the two surfaces when illuminated; the brighter the light, the more pronounced the bleeding effect is. Global illumination is a means to simulate this behavior of light producing far more realistic results more easily than could ever be obtained with direct light sources only.

Modo allows you to use both direct and indirect lighting sources in any scene. These two models are computed separately, and their respective products are added together to give the final shaded result. Each model has specific advantages (and disadvantages). When used together wisely, you can optimize both the speed and quality of rendered images with an incredible amount of control.

To fully understand the indirect illumination settings, it is critical to first understand the technical process of indirect illumination. While it is intuitive to imagine light coming from a 3D light or a luminous surface, travel to reach the surfaces, then bounce around a room, the actual process of rendering with indirect illumination is quite the opposite. As the surface is evaluated, rays are cast outward from the surface randomly and evaluated when they strike other surfaces in the scene. The sum of those evaluations is what contributes to the color and brightness of the original surface.

To get a more precise idea of how indirect illumination is estimated at a point on a surface, imagine the top half of a transparent globe resting on the surface, so that the point's surface normal is poking through its north pole. Rays are fired from the surface point through random positions within each cell formed by the latitude and longitude grid lines, with one ray per cell. These rays go out and hit either other surfaces or the distant environment, and the average color that they see is the indirect irradiance estimate (irradiance being the technical term for incoming light).

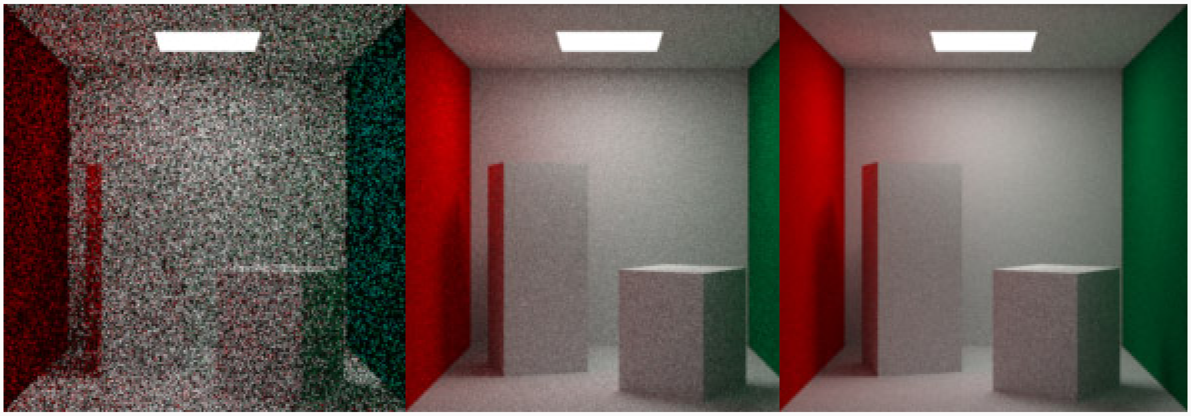
Now imagine we need to shade a surface on a moonlit night; the environment is practically all black except for one concentrated bright region, the moon. Each shading point on the surface sends rays as described above. For some points, maybe two of their rays hit the bright region, while only one ray hits other points, and the rest of the rays see black. With some points getting twice the irradiance of others, you can predict that the surface may look quite splotchy if irradiance caching is on (or grainy if it's off). However if we subdivide the transparent hemispheres more finely (that is, use more rays), the number of hits and misses become much more consistent between neighboring surface points, smoothing out the shading.

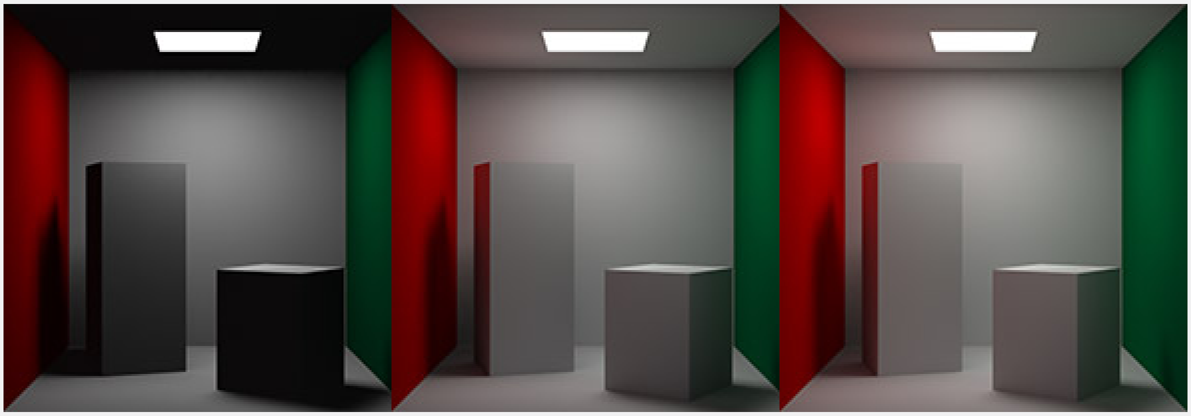
While indirect illumination in Modo is based on this hemispheric sampling technique, there are two very different approaches to the use of these samples. The default method utilizes a technique called irradiance caching (IC). The concept behind this technique is that by leveraging a smaller number of more accurate samples and blending between them, you can achieve an image of perceived quality in a shorter amount of time than sampling every pixel with lesser quality, which often results in grainy or noisy images. When irradiance caching is disabled, Modo falls back to generating a hemispherical shading sample for every pixel in the image (Monte Carlo method). As a result, you must be careful about the number of rays you use as this number is multiplied by the millions of pixels in your image. With irradiance caching active, Modo intelligently samples the scene at strategic locations and then interpolates between them for a smoother overall final frame.

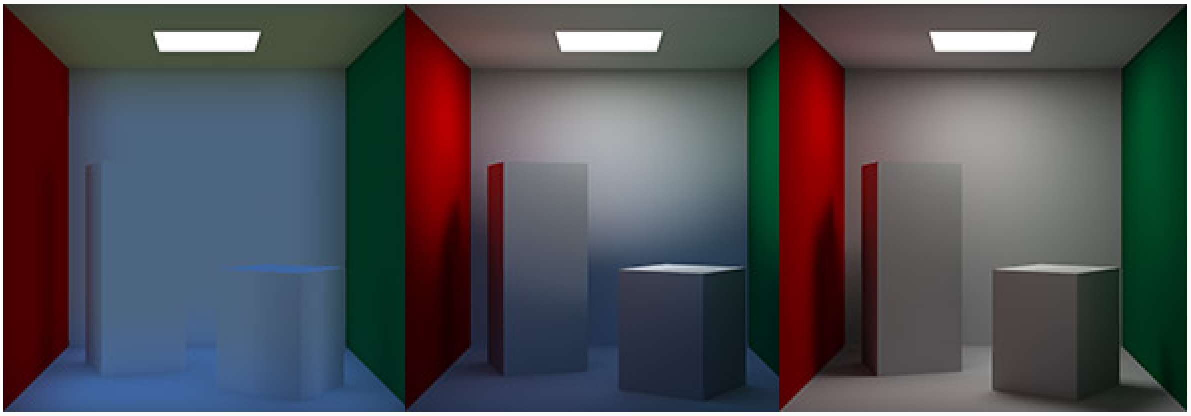
The simplest way to think about Monte Carlo and irradiance caching is that the Monte Carlo method uses a lower quality (fewer rays) sample at every single pixel location, whereas IC uses fewer, much higher quality (more rays) samples and blends them together. As a result, when the samples are not accurate enough in Monte Carlo, there is significant variance between neighboring pixels, which appears visually as grain. When using IC, the variance is spread across from sample to sample, which yields splotches visually. With Monte

Carlo, you can remedy this by increasing the number of rays per pixel. This can cause render times to increase dramatically. IC provides several approaches to reducing artifacts, including increasing the number of rays, adding super sampling, and increasing the number of samples required to create a blend, called Interpolation Values.

Now that you have an overview of the basic concept, the following definitions have more context and enable you to more effectively balance your scene in performance/quality when working with indirect illumination.

Option	Description
Enable	Activating this toggle enables Modo to calculate global illumination.
Indirect Samples	This value represents the number of samples taken for each pixel in the image when using the Monte Carlo indirect illumination model (irradiance caching disabled). Increasing this value results in higher quality at the expense of longer render times. Global Illumination - Monte Carlo (IC disabled), 4 bounces, Ray Threshold 0%  (default) 64 Indirect Rays -12s 4096 Indirect Rays -13m 22s 16384 Indirect Rays -51m 40s **Note:** When **Indirect Illumination** is enabled, the **Indirect Samples** option is also available in the **Settings** tab.
Bounces	The default value of **1** means that a single bounce is used to calculate how a surface's environment affects it. This is less accurate than a multi-bounce solution since in the real world, photons bounce all over when illuminating an environment. However, the sacrifice in quality using fewer bounces is rewarded by increased performance. By increasing the number of indirect bounces, the indirect rays are fired from the initial surface and bounce off the first surface they hit, then continue traveling and hitting surfaces until the maximum number of bounces is met.

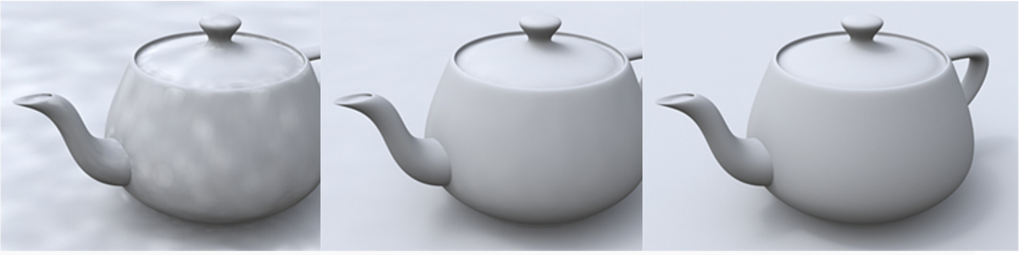
Option	Description
	Note: As you add bounces, the number of calculations and render time increases and while the result is technically more accurate, there are naturally occurring diminishing returns with each additional bounce. We recommend that you adjust your scene with a single bounce and then add additional bounces to see how great an impact they have on the end result. If a scene is mostly illuminated by a single small area of light, adding bounces can dramatically improve the overall brightness and look of the render. Indirect Bounces (IC enabled), Ray Threshold 0%  (default) 1 Bounce -30s 2 Bounces -57s 4 Bounces -1m 14s **Tip:** To reduce the number of bounces required, you can increase the **Ambient Intensity** value. Since the effect of multiple bounced illumination rays is to boost the overall lighting by some averaged color, you can reduce the number of bounces and set the **Ambient Color** and **Intensity** to mimic those final bounces. This saves time and yields similarly realistic results.
Range	This value determines how far an indirect ray can travel before it is terminated. In the case of a ray being terminated due to the indirect range value, it is assumed that the ray eventually hits the environment background, so this value is returned to the shading engine. This is a very useful way to optimize render times. Reducing the indirect range generally improves render speed.

Option	Description
	Note: Setting this value too low creates unnatural lighting effects, as many rays that otherwise would have ultimately hit a geometric surface resulting in a shadow may get cut off early, illuminating the surface with the backdrop color, rather than shading it with the surrounding geometry. A setting of 0 disables indirect range, utilizing the full scale of your scene. The following images show Indirect Range (IC enabled inside a 1m cube w/ default Environment), Ray Threshold 0%:  200mm Range -29s 800mm Range -48s 1.6m Range -57s
Indirect SSS	When rendering surfaces that employ subsurface scattering (SSS), you can choose how the indirect lighting in the scene affects those surfaces. When selecting **Direct Only**, Modo doesn't use any indirect (bounced) lighting, but only calculate SSS based on the direct lighting within the scene, that is, distant lights, spotlights, point lights, and so on. This mode is useful in many cases as global illumination's effect on SSS is usually fairly subtle. However, in instances where there are no direct lights in your scene, such as those lit entirely by image-based lighting (HDRI), you need to choose either **Indirect Affects SSS** or **Both** to see the effects of SSS for any given surface. • **Indirect Affects SSS** - When selecting this option, Modo uses any global illumination in your scene, as well as the direct lighting to calculate the subsurface scattering effects on your object's surface. • **SSS Affects Indirect** - When selecting this option, Modo calculates bounced indirect light, taking into consideration the subsurface scattering effects of any given surface. • **Both** - This setting allows global illumination to affect subsurface scattering, as well as for the subsurface scattering to affect the indirect illumination calculations. This setting is the

Option	Description
	most render-intensive.
Indirect Volumetrics	Enabling this checkbox directs Modo to consider volumetric lighting effects when calculating global illumination.

Environment Importance Sampling

The following **Environment Importance Sampling** options are available:

Option	Description
Enable	Importance sampling is a feature that takes into account the brightness of HDR images used for Image Based Lighting. Brighter areas of the image have a greater effect on the lighting result, so when enabled, samples concentrate around these brighter areas, producing a more accurate, smoother result with fewer IC rays (which means faster rendering). This feature essentially makes the process of applying a blurred low-resolution copy of the Environment HDRI unnecessary. **Note:** Importance sampling slightly changes the look of the final rendered image, most obviously producing more accurate shadows, which may or may not be a desirable result.
Environment Samples	Determines the number of samples taken of the environment used for global illumination. More detailed environments benefit from additional samples, increasing the accuracy of the final render, but also increasing render time. On low resolution or low detail environments, you can keep this value low to reduce render time.  The image on the left has importance sampling disabled and uses an HDR

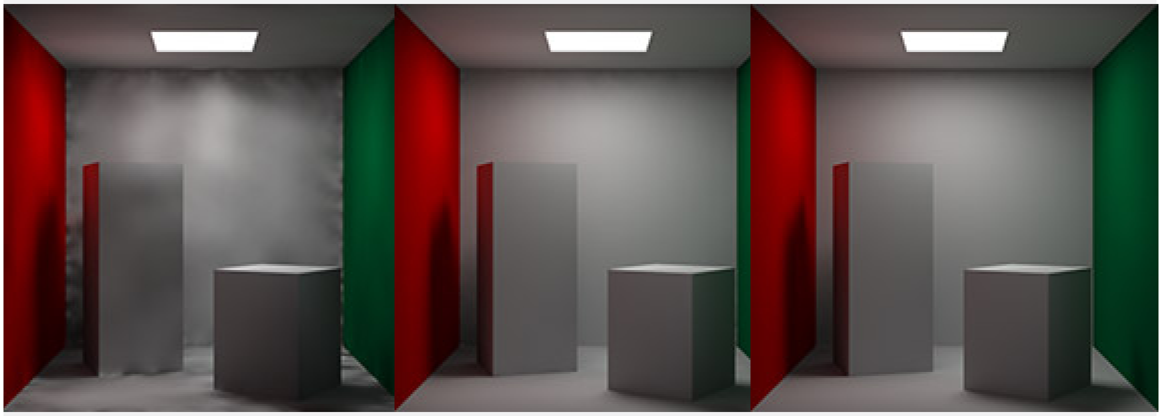
Option	Description
	image for environment lighting with a bright spot (sun). With 128 IC rays, undesirable splotches are visible that are eliminated in the middle image by increasing the IC rays 8 times to 1024 rays. The third image enables **Importance Sampling** at default values and resets the IC rays back to 128 rays. Note the crispness of the shadow and the complete lack of any splotches, producing superior result with only a slightly longer render than the left image. **Note:** When **Environment Importance Sampling** is enabled, the same option is available in the **Settings** tab, called **Environment**.

Irradiance Caching

The following **Irradiance Caching** options are available:

Option	Description
Enable	When rendering indirect illumination in Modo, irradiance caching is the default method. Disabling this toggle directs Modo to use the more intensive Monte Carlo method to calculate global illumination.
Irradiance Usage	Modo offers two ways to calculate global illumination. By default, there is the very fast irradiance caching option that saves values and refers to them when needed. The second way is to disable IC, Modo then uses Monte Carlo path tracing. There is no caching of Monte Carlo values, so results can be slow and noisy, however, with enough rays and time, it can produce very clean and accurate results. The way ray evaluation works in Modo can be thought of as layers. When lighting with multiple bounces (having more than a single bounce defined in the **Indirect Bounces** control), each bounce is its own layer. The **Irradiance Usage** option allows you to control what layer of rays use irradiance caching during a render and which use Monte Carlo. The reasons for these options are to allow the strongest influencing rays (the first bounce) to use the more accurate Monte Carlo Method and blend that with the faster and smoother results of the subsequent IC evaluation. To mimic the results of Modo 701 and earlier, use the **First and Second Bounce** option.

Option	Description
	• **First Bounce Only** - Only the first outward bounce from the camera uses IC, any subsequent bounces use Monte Carlo. • **Second Bounce Only** - The first outward bounce from the camera uses Monte Carlo and then IC for the second bounce and back to Monte Carlo for any subsequent bounces. • **First and Second Bounce** - Both the first and second bounces use IC and subsequent bounces use Monte Carlo. Irradiance Cache 1m 02s Monte Carlo 4m 24s Hybrid 1m 16s
Irradiance Samples	Irradiance rays are technically the same as indirect rays in that they are the rays fired out from the surface in order to sample indirect illumination. For convenience, Modo has both irradiance and indirect rays settings since the two forms of indirect illumination typically require vastly different numbers of samples. Irradiance caching relies on much higher quality samples distributed sparsely across the project, whereas indirect illumination without irradiance caching uses lower quality samples at every pixel. Having two values allows you to easily switch between irradiance caching and traditional indirect illumination without constantly adjusting the number of rays for each sample. The following images show Irradiance Rays, Ray Threshold 0%:

Option	Description
	 64 Rays -3.1s 1024 Rays -6.7s 4096 Rays -12.4s **Note:** When **Irradiance Caching** is enabled, the same option is available in the **Settings** tab, called **Irradiance**.
Supersampling Toggle	**Note:** The **Supersampling Toggle** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. After all rays have been fired for a particular irradiance evaluation, this feature looks at the resulting ray color of each hemisphere cell and then sends additional rays through those cells that differ a lot from their neighbors, getting a more detailed look at high contrast areas of the environment and therefore a more accurate estimate. Currently about 25% more rays are fired, so for a setting of a 100 rays, you really get 125, but the results are better than if you had just used a setting of 125 without supersampling. This is because the rays are going in the more important directions. It is recommended to leave this option on all the time. It generally improves your render quality with minimal performance impact.
Outlier Rejection	**Note:** The **Outlier Rejection** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. This option specifies whether to disregard rays that are much brighter or darker than the average values in a particular direction when irradiance cache values are being computed and **Supersampling** is enabled. It can be useful for smoothing indirect

Option	Description
	illumination in some cases and help to eliminate noise produced by very small, very bright illumination in a given scene.
Irradiance Rate	As irradiance caching smoothly blends between neighboring IC values, a little breathing room is necessary between samples to provide space for that blend to occur. **Irradiance Rate** provides that space as it sets a minimum distance between IC samples, calculated by pixels. So a setting of 2.5 means the minimum distance between two samples is 2.5 pixels away. Increasing this value causes IC values further away from each other to blend together, providing a smoother result, however increasing the value too high may result in reduced accuracy in areas of high detail.
Ratio	**Note:** The **Ratio** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. While **Irradiance Rate** sets the minimum distance between IC values, **Irradiance Ratio** sets the maximum distance. The ratio is a multiplier of the rate, so a setting of 6 multiplied by a rate of 2.5 returns a value of 15, meaning that IC values are no closer than 2.5 pixels and no further apart than 15 pixels. The ratio ultimately reduces the number of samples necessary across smooth surfaces where shading changes are minimal.
Interpolation Values	**Note:** The **Interpolation Values** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. Specifies a minimum for how many nearby values to interpolate. Let's say it's set to 3. If we're shading a point and can only find two nearby previously computed values in the cache, then it forces the computation of a new value at the current point, and the final irradiance at that point is a blend of all three values. So it tends to smooth out the shading. Increasing this value generally improves render quality at the expense of render time.
Walkthrough Mode	When rendering an animation where only the camera is moving in the scene, such as for architectural walk-throughs, enabling the **Walkthrough Mode** option tells Modo to never delete any irradiance cache values. Each frame continues to calculate pre-passes, generating new values when necessary and reusing existing values when available. Since Modo never needs to generate a new (possibly different) value, this can help to

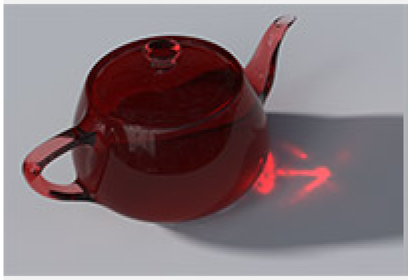
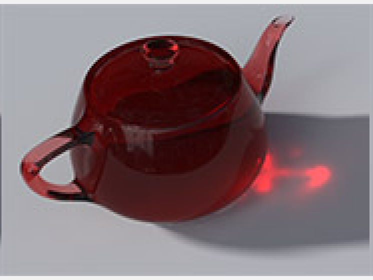
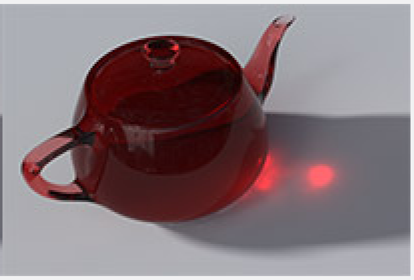
Option	Description
	eliminate jittery shading sometimes evident on pristine architectural surfaces. However, for longer sequences it can create a large number of IC values, which may slow down rendering of later frames.
Store Direct Light	**Note:** The **Store Direct Light** checkbox is hidden by default. Click the **More** button at the bottom of the panel to display all controls. This option includes illumination from direct light sources in addition to the usual indirect illumination in the secondary irradiance cache values, saving time whenever a first bounce indirect ray hits a surface that already has cached secondary values. This is useful for applications like architectural interior rendering. **Note:** This option is only applicable if there are at least two GI bounces and **Irradiance Usage** is set to **Second Bounce Only** or **First and Second Bounce**.
Retrace Threshold	**Note:** The **Retrace Threshold** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. If an indirect ray travels less than this distance, it ignores the second level IC and reverts to path tracing. This helps prevent light leaks (especially when using **Store Direct Light**), and can speed up rendering highly-detailed geometry, since irradiance caching doesn't work so well in those cases. **Note:** This option is only applicable if there are at least two GI bounces and **Irradiance Usage** is set to **Second Bounce Only** or **First and Second Bounce**.
Save Irradiance/Load Irradiance	**Note:** The **Save Irradiance** and **Load Irradiance** options are hidden by default. Click the **More** button at the bottom of the panel to display all controls. Using the **Save Irradiance** and **Load Irradiance** functions allow you to save an

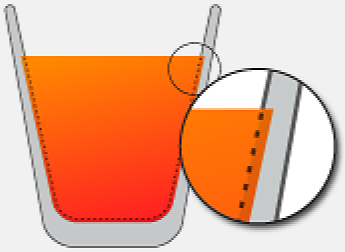
Option	Description
	irradiance cache solution to disc and then load it at a later time for reuse. The typical workflow is to enable the **Save Irradiance** option, render a frame to generate the values and then disable the option. Next, enable the **Load Irradiance** option to load and reuse the generated values for subsequent renders. There are several reasons for doing this, but the main one is to reduce the overhead of calculating the IC values at each render, for scenes where lighting and position are not changing. To use it, enable the **Save Irradiance** checkbox and click the **Save Irradiance File** button to define a filename and location. Click **Save** to continue. The next time a full render is generated the irradiance cache saves to disc upon completion of the render. When the render is complete, disable the **Save Irradiance** option and enable the **Load Irradiance** option. With the **Save Irradiance File** button, locate the saved irradiance cache **.lxi** file defined earlier and press **OK**. Subsequent renders then reuse the values in the saved file. Saving and Loading of irradiance cache files when combined with **Walkthrough Mode** rendering of animated sequences may produce a very large file which has the potential to slow down rendering due to the amount of time required to locate and read the target IC values.

Caustics

When light refracts through or reflects off of a surface, these bent rays are focused together, creating a bright pattern referred to as a caustic. Think of the light dancing on the swimming pool's floor, or the bright flash you see from a car's windshield on a sunny day and you'll know what it is. Simulating this effect in Modo is quite easy with global illumination as caustics are a natural rendering byproduct of image-based lighting. Caustics from direct light sources, such as area lights, distant lights, spot lights and such, require the use of photons map tracing to calculate the effect, enabled with the following **Caustics** settings.

Option	Description
Direct Caustics	This checkbox tells Modo to calculate caustics for all reflective and transparent surfaces.
Total Photons	The total photons control sets a ceiling on the total number of photons fired in the scene. The photon count is divided amongst all active 3D lights in a scene with a bias based on the power of the light, ensuring energy conservation.

Option	Description
Local Photons	Local photons indicate the number of photons required for each pixel sampled. When the pixel is rendered a search along the surface locates the nearest photons up to the local photon count. The default setting of 32 indicates that 32 photons are used for each pixel rendered. Increasing this value creates a smoother overall caustic effect at the expense of detail, while decreasing the value results in a sharper caustic effect with increasing graininess.  (default) 32 Local Photons  128 Local Photons  256 Local Photons
Walkthrough Mode	When rendering animations containing caustics, where the camera is the only item moving in the scene, it is helpful to enable the **Walthrough Mode** option. This tells Modo to cache the caustic calculations, frame to frame, reusing the initial calculations, thus ensuring each subsequent frame matches precisely, producing a smoother rendered result.
Indirect Caustics	**Note:** The **Indirect Caustics** option is hidden by default. Click the **More** button at the bottom of the panel to display all controls. Indirect caustics are a natural rendering by-product of image based lighting. In instances where you wish to control their creation, you have several choices to do so. • **None** - Disables all indirect caustics within a scene. • **Reflection Only** - Caustics are only calculated off of reflective surfaces. • **Refraction only** - Indirect caustics are only calculated though refractive surfaces. • **Both** - Indirect caustics are calculated for both reflective and refractive surfaces.

Option	Description
	 **Tip:** When modeling overlapping transparent surfaces, such as glasses or bottles containing liquids, it is recommended that the liquid layer overlap the glass layer by a small amount. In no case should there be a gap between the two surfaces. This image illustrates a cutaway view of a shot glass, the black outline, filled with a liquid, the red shading. Note how the red object extends beyond the interior surface of the glass. This is necessary to obtain the desired results.

Camera Items

The **Camera Item** is a virtual lens into the 3D world inside Modo. Everything rendered is seen from the camera's point of view, and the camera's settings determine how the 3D world is captured. The camera works in conjunction with the **Render Item**, generating a virtual snapshot in the Render Display.

The image that is displayed depends on your render output. This acts as a virtual developer by determining what actually gets rendered when any render command is invoked.

Tip: You can also invoke the renderer by pressing **F9**.

The **Camera Item** contains the properties that define the projection, frame aspect ratio, depth-of-field, and other attributes. However, the actual frame resolution is set within the **Render Item** under the **Frame** properties.

Adding a Camera Item

Each new Modo scene has a single camera item by default. A new camera can be added to the scene by the following methods:

- Select **Item** > **Create Camera** from the menu.

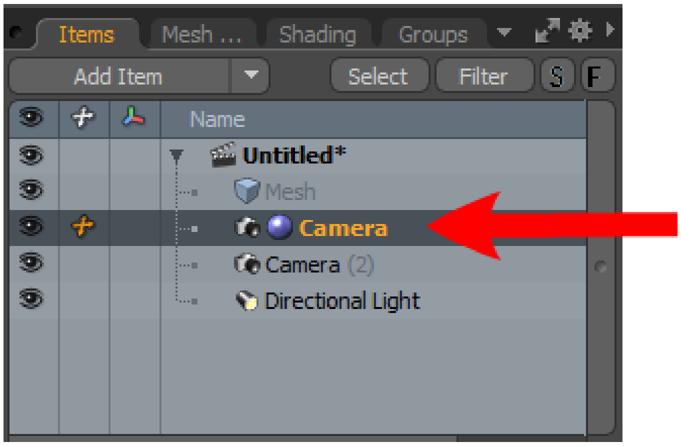
OR

- In the **Item List**, click **Add Item** and select **Locators** > **Camera**.

Choosing a Render Camera

Additional cameras are helpful in providing alternate views of a scene, easing navigation, or for providing close-up views of a particular area, especially in situations where the hero (principle) camera is animated and locked, avoiding accidental changes. Only a single camera may be rendered at a time, the particular camera used is specified in the Render Item: Frame sub-tab under the **Render Camera** option. Additionally, alternate cameras may be selected within the **Render Preview** viewport using the select camera option available, without affecting the **Render Camera** used for final rendering.

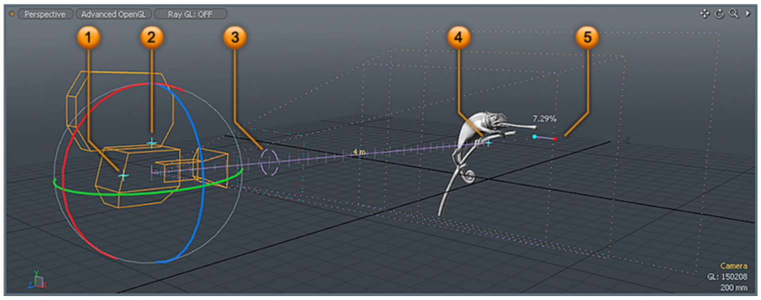
To quickly identify which camera is used as the render camera, in the **Items lists**, bold text and a render icon is displayed. In the image below, an additional camera was added to our scene to demonstrate the difference in the display of camera items. To set a camera in your Items list to a render camera, right-click on the camera item and select **Set as Render Camera**.



Working with the Camera

Camera attributes, such as **Position** and **Rotation**, can be modified in the **Properties** panel whenever the Camera Item is selected in either the Item List or the Shader Tree. By default, the camera icon is only visible in the Animate and Render 3D Viewports, but by changing the visibility in the viewport options, they can be made viewable in any viewport.

Hover the mouse pointer over the 3D viewport you wish to modify, and press **O** on the keyboard to open the **3D viewport Properties** panel.



Camera Items can be modified interactively in the 3D viewport while in **Items** mode (Camera Items have no editable components). Use the standard transform tools to move (**W**) and rotate (**E**) the camera. When making fine adjustments, you may find it easier to see how the camera is viewing the scene. This can be done by changing the viewport view button in the upper left to **Camera**. When set as such, black bars may be drawn over the viewport if the aspect ratio of the **Frame Size** is different from the viewport's aspect ratio. The area inside the bars represents the area of the scene that renders when a render command is invoked.

To position the camera while in this view mode, various keyboard/mouse options are available.

Keyboard/Mouse Options	**Camera Actions**
Shift+**Alt**+Left-click and drag	Move camera item
Alt+Left-click and drag	Rotate around Target Distance
Alt+Right-click and drag	Rotate around Camera center
Ctrl+Middle-click and drag	Rotate around the Z axis
Ctrl+**Alt**+Left-click and drag	Move Forward/Backward only
Ctrl+**Alt**+Right-click and drag	Adjust Focal Length
Roll Mouse Wheel	Move Forward/Backward only

When tool handles are set to **Advanced** (within the Preferences or Tool Pipe), additional handles become visible, allowing you to interactively adjust camera setting in the viewport. The handles are:

- Adjust Focal Length/Field of View
- Adjust F-Stop for Depth of Field
- Adjust Convergence Distance
- Adjust Focus Distance

- Parallax Offset Difference value (distance between items for left and right eye in final rendered frame) expressed as percentage of frame width

  You can drag the handle around the scene to get an idea how far apart elements appear on the screen when rendered as a stereo image.

  Generally speaking, for comfortable viewing you should keep the sum of the values for the closest elements in frame to the farthest elements around 5% and under. Adjustments can be made to **Interoccular Distance** and the **Convergence Distance** options to help in obtaining this result.

Tip: You may find it more intuitive to adjust the camera when the Action Center is set to **Local**, keeping the transform handles aligned to the Camera Item itself.

On occasion, you may wish to modify the size of the camera icon within the 3D viewport. This can be accomplished with the **Size** setting in the Display viewport, which offers additional settings for modifying the camera's icon representation, including the ability to label the item, enable the **Safe Area** display, and toggle the display of the **Stereo Volume**. You can also adjust the camera's **Scale** option in the **Properties** panel.

Note: Changing the size of the camera icon in no way affects the rendering outcome or the overall scale of the scene.

Markup

The Markup tool allows you to draw directly onto selected camera views so you can add annotations, notes, and comments for your animations or stills. Assign Markup to the current camera item by selecting the camera and clicking **Markup** in the **Images** properties. You must make the current camera active in the viewport before you can create the Markup image. Once assigned, you can use a selection of Markup tools and controls to draw on your viewport. See Markup for more information.

Tip: The Markup tool can be used in conjunction with animated scenes by moving to a frame and drawing. The Markup images for each frame are displayed when scrubbing through frames or during normal animation playback.

Foreground and Background Images

Each camera can be assigned a foreground and background image to supplement a scene. The foreground image is only visible in the render view so as not to obscure any geometry, whereas the background image is always visible from the camera. To preserve the background image aspect ratio, you can override the render resolution with the background image resolution. For detailed instructions, see Camera Images.

Camera Matching

The camera background image acts as a backdrop to geometry when rendering a scene. To make the geometry look like it's in the backdrop you can use Camera Matching. Camera Matching adjusts the camera position so that the perspective in the camera background image matches the perspective of the Modo world. This allows you to place geometry as if it were part of the background image. For full instructions, see Camera Matching.

Camera Image Planes

In addition to foreground and background images, Modo cameras support multiple image planes. Using image planes, you can position a material so that it faces the camera at any distance from the camera viewpoint. This lets you layer images, allowing you to insert geometry between the layers. For full instructions, see Camera Images.

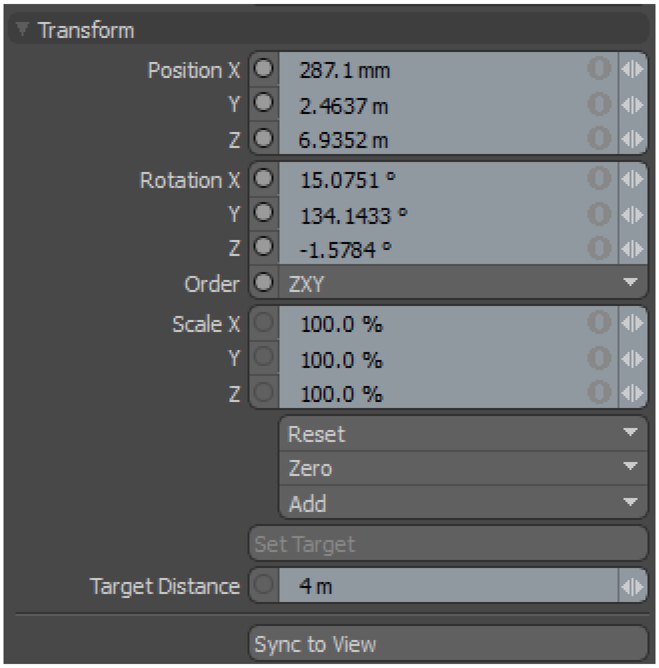
Camera View Properties

The first property is the camera's **Name**, which can be edited to help identify the camera in the **Items** list.

Other **Camera View** properties:

- Transform Properties
- Projection Properties
- Resolution Override Properties
- Images Properties
- Background Image
- Film Back Properties

Transform Properties



Position X 287.1 mm
Y 2.4637 m
Z 6.9352 m
Rotation X 15.0751 °
Y 134.1433 °
Z -1.5784 °
Order ZXY
Scale X 100.0 %
Y 100.0 %
Z 100.0 %
Target Distance 4 m

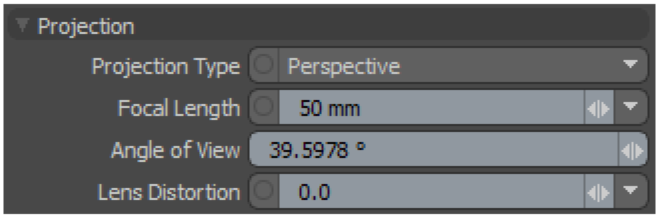
These properties define the camera position and orientation.

Transform	
Position	An item transform that allows you to numerically position the Camera Item in XYZ space. By default, **Position** transforms originate from the item's center position.
Rotation	An item transform that allows you to numerically set the rotation of the Camera Item. By default, **Rotation** transforms originate from the item's center position.
Order	Allows you to set the order in which rotations are applied to the Camera Item. Changing the order that rotations are applied can sometimes help to reduce, or eliminate gimbal lock.
Scale	An item transform that determines the size of the icon in the 3D viewport. Scale otherwise has no effect on the final rendered image.
Reset	Resets the selected transform values to (0,0,0), returning the items to their default state.
Zero	Returns the Camera Item's **Center** position to the world space center (0,0,0), without changing the position of the Camera Item itself.
Add	**Transform Items** are the channel groups associated to an item that store its transform

	values, controlling its position, rotation, and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add **Transform Items**. One is by simply transforming the target item with one of the various transform tools (or by editing the values input fields). This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the **Channels** list while keeping the default 0,0,0 values (a necessary step for **Referencing**; in order to override channels, they must first exist).
Set Target	This option allows you to target specific items in a scene, automating the rotation of an item, so that it continuously points toward the targeted item. To use **Set Target**: 1. Make sure you are in **Items mode**, and select the **Camera Item** (and only the Camera Item). 2. Hold down the **Ctrl** key and select a single additional item in the **Items** list that you wish to target (such as a locator), in that order; as the selection order is important. 3. Click **Set Target** under the **Camera View** options sub-tab to enable the function. Remember though, that the camera looks directly at the item's center position, regardless of the location of geometry within the target layer. Once activated, additional options appear: • **Remove Target** - Removes the target link between the two items. • **Enable** - Toggling this option off temporarily disables the targeting function while retaining the link between the items. • **Set Focus Distance** - This option sets the focal distance for DOF when targeting camera items to other item. • **Roll** - Provides the ability to offset the item's rotation angle away from the target. • **Time Offset** - Specify the time in seconds for the camera following to be ahead or behind the item movement. The camera movement is still dictated by the item movement, but when the **Time Offset** is a positive value, the camera tracks ahead by the specified number of seconds, effectively predicting the item's movement. If the **Time Offset** is a negative value, the camera will lag behind by the specified number of seconds.
Target Distance	This value defines the point around which the camera orbits. When you start a fresh project, the camera is back 4M on Z and the **Target Distance** is 4M. That is why it orbits around 0.

	This value is reset automatically when you zoom in the 3D viewport. Modo fires a ray out from the camera and automatically sets the target distance based on the first surface the ray intersects.
Sync to View	This option changes the position and orientation of the currently selected **Camera Item** to match the currently selected 3D viewport as closely as possible. There are some caveats in that the camera's view scale depends on the focal length and image resolution (for aspect) and therefore cannot be matched precisely.

Projection Properties

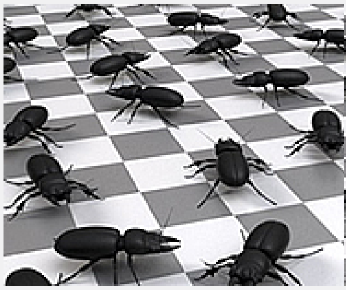


Projection Type: Perspective
Focal Length: 50 mm
Angle of View: 39.5978 °
Lens Distortion: 0.0

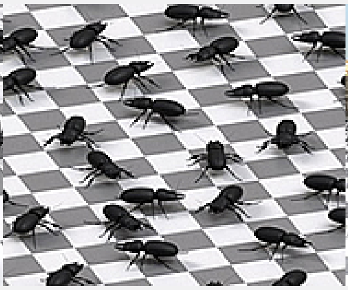
These properties define how the world is seen through the camera based on the projection and lens properties.

Projection	
Projection Type	Modo can render traditional perspective cameras as well as orthographic cameras and spherical cameras. This control allows you to choose the desired camera type: • **Perspective** cameras show perspective distortion where the scale of an element appears to diminish in the distance. This traditional perspective view is what most artists would expect when rendering; similar to what a real world camera might capture. **Perspective** cameras are found most often in visual FX, product and architectural visualization, and anywhere artists are trying to mimic reality. • **Orthographic** cameras do not exhibit any perspective distortion as all rays are fired from the camera in parallel lines. **Orthographic** rendering is very useful in news/info graphics and architectural plan rendering. • **Spherical** cameras capture the entire 360 degree view of a scene.

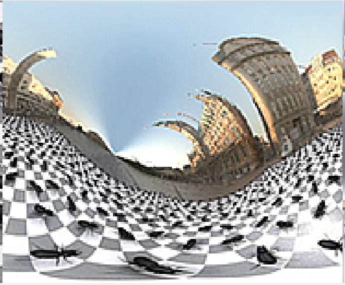
	Spherical cameras are useful in capturing images that can be later applied as image based lighting and reflection maps, among other things. • **Cylindrical** projection captures the entire 360 view from the perspective of a tube, removing the polar areas of the projection. • **Cylindrical** and **Spherical VR** projection types are intended for applications like virtual reality. They differ from the previous types in that they are not flipped, but centered on the camera view direction.



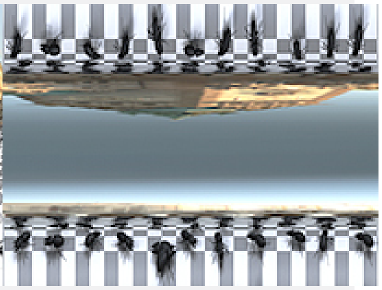
Perspective Camera



Orthographic Camera



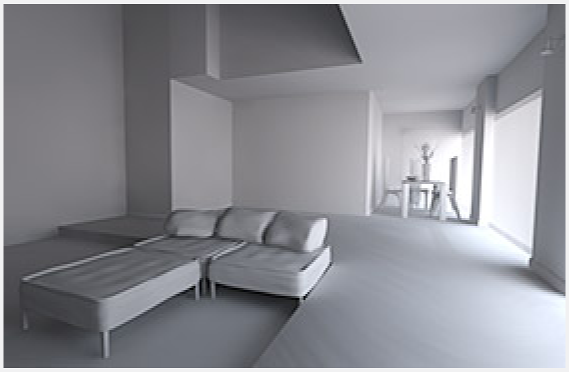
Spherical Camera



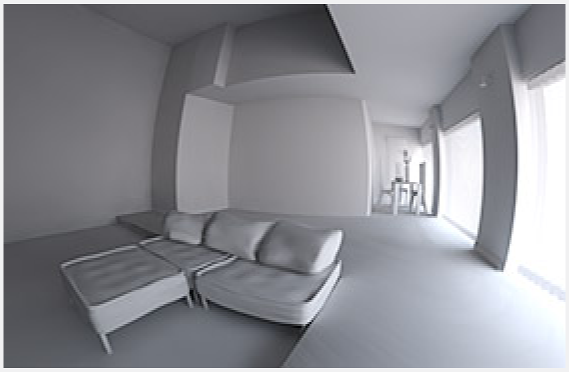
Cylindrical Projection

Focal Length	This setting controls the amount of zoom, just as it would with a real world camera. The actual definition of **Focal Length** is the distance between the lens and the point where the rays converge after having been refracted by the lens. Wide angle lenses (those with short focal lengths) capture more of a given scene than their telephoto counterpart. Lower focal lengths exhibit more perspective distortion, conversely, longer focal lengths minimize the effect of perspective. To zoom out you should decrease the focal length and to zoom in you increase the value. This control is set in real world metric units to match physical cameras. **Focal Length** can be adjusted on the **Camera Properties** panel or interactively in the 3D viewport. If you are looking through the Camera (and 3D viewport set to **Camera**) you can use **Ctrl**+**Alt**+right-click and drag to adjust the **Focal Length** interactively (**Ctrl**+**Alt**+left-click and drag moves the camera itself forwards and backwards).
Angle of View	In some instances, it may be easier to specify an angle of view instead of a **Focal Length**. This setting allows you to specify the **Angle of View** as a number of degrees horizontally, and once set, the **Focal Length** is updated to reflect the new value.
Lens Distortion	In a real world camera, light rays are bent through a spherical lens element

	resulting in a slightly distorted image. 3D rendered images don't pass through a lens and likewise don't exhibit the distortions we're accustomed to in photography. The **Lens Distortion** setting can be used to introduce these familiar distortions in a realistic way. Positive values create an increasing **Barrel** distortion, and negative values create increasing **Pincushion** distortion effects.

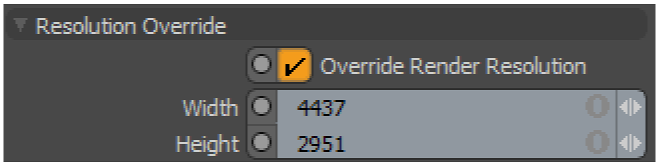


Standard CG Wide Angle View



Same view w/ Lens Distortion applied

Resolution Override Properties



Resolution Override

☑ Override Render Resolution

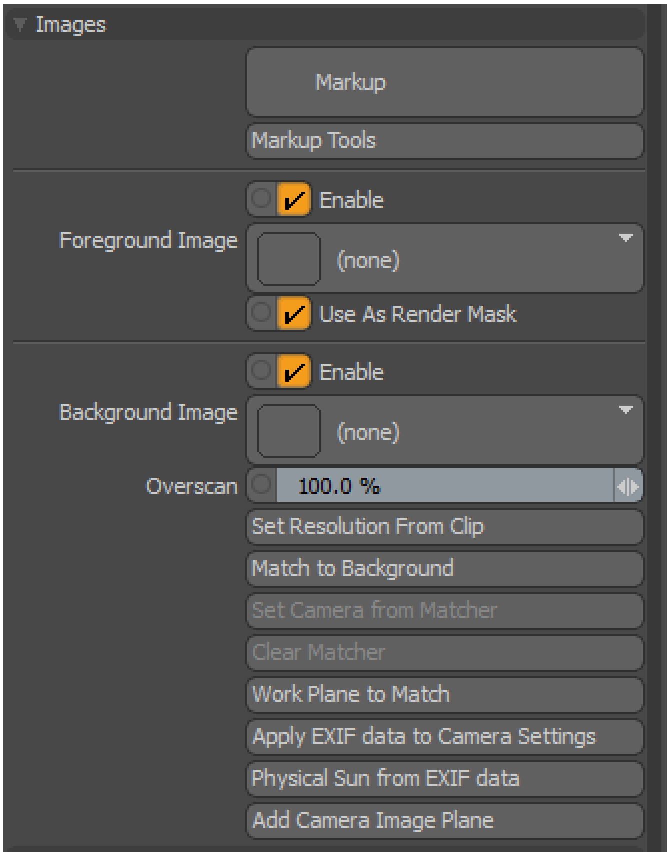
Width 4437

Height 2951

These properties allow you to set a resolution to override the Render Item resolution.

Resolution Override	
Override Render Resolution	Check this to specify a resolution to use for rendering through this camera.
Width	The width in pixels.
Height	The height in pixels.

Images Properties


Images
Markup
Markup Tools
Foreground Image
Enable
(none)
Use As Render Mask
Background Image
Enable
(none)
Overscan
100.0 %
Set Resolution From Clip
Match to Background
Set Camera from Matcher
Clear Matcher
Work Plane to Match
Apply EXIF data to Camera Settings
Physical Sun from EXIF data
Add Camera Image Plane

The camera item **Images** properties panel.

Markup

The Markup tool allows you to draw directly onto selected camera views so you can add annotations, notes, and comments for your animations or stills. See Markup for more information.

Markup	Pressing the Markup button assigns a Markup drawing to the currently selected camera. **Note:** You must have the selected camera active in the viewport for the Markup tool to become available.
Markup Tools	Opens the Markup tool popover. See Markup.

Foreground Image

These properties allow you to add a foreground image, which may be also used as to mask the render output. For more information, see Foreground and Background Images.

Foreground Image	
Enable	Enables the Camera image to use the **Foreground Image**. When disabled, the Foreground Image is not displayed in your rendering but it is not removed. This option is **Enabled** by default.
Foreground Image	Add a foreground image to the near viewing plane. This is only visible when rendered.
Use As Render Mask	Check this to use the foreground image as a mask for the render output. White areas in the image are fully transparent, and black areas are fully opaque and omitted from the render. The darker the shade, the more opaque the mask.

Background Image

These properties allow you to work with background images and image planes. For more information, see Camera Images.

Background Image	
Enable	Toggles to enable or disable Camera images that are used for **Background Image** without removing it. This option is **Enabled** by default.
Background Image	Add a background image to the far viewing plane. This can be used for Camera Matching to match the perspective of the camera to the background image.
Overscan	Scale the background image without changing the aspect ratio.
Set Resolution from Clip	Set the render resolution to match a selected clip. The **Background Image** is used if one has been added.

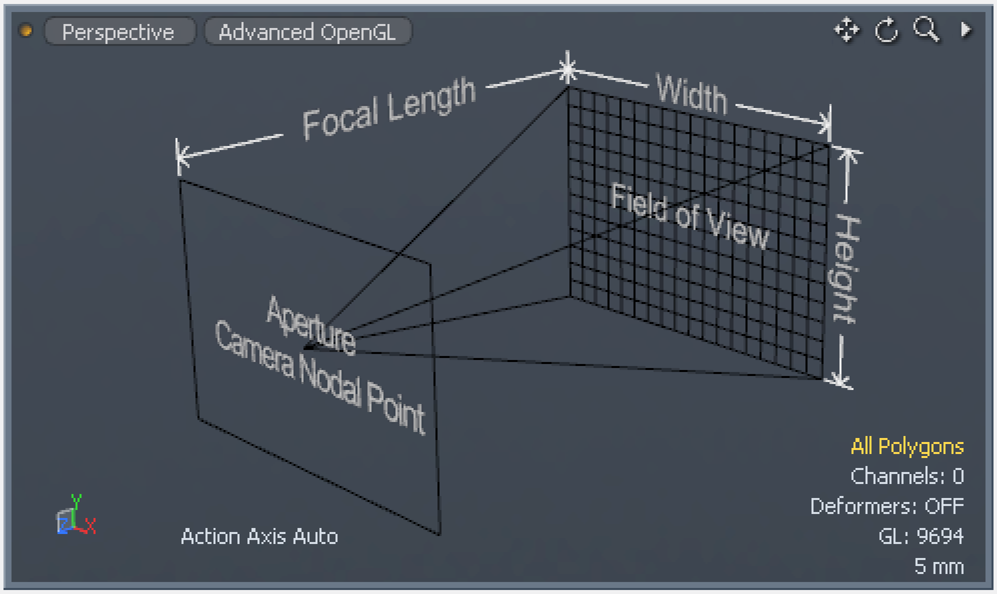
Match to Background	Click this to start the Camera Matcher. For full instructions see Camera Matching.
Set Camera from Matcher	Match the camera perspective to the image perspective according to the perspective lines that have been drawn when using the Camera Matcher. In short, match the camera.
Clear Matcher	Reset the Camera Matcher.
Workplane to Match	Match the Work Plane to the axes and position as drawn in the Camera Matcher.
Apply EXIF data to Camera Settings	Use EXIF data to define the camera's **Focal Length**. See EXIF Data.
Physical Sun from EXIF data	Use EXIF data to create a **Physical Sun** directional light. See EXIF Data.
Add Camera Image Plane	Add a quad polygon on an image plane in the camera frustum. This opens the **Camera Plane** properties tab. Adjust the **Depth** to position the plane at a suitable distance from the camera viewpoint. The quad is scaled automatically to fill the camera frame. See Image Planes.

Film Back Properties

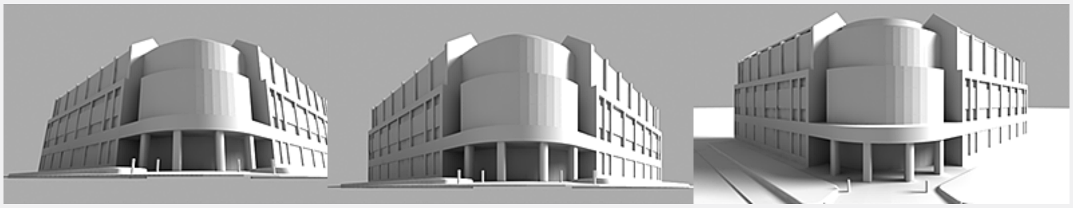
In the real world, every camera has certain attributes of fixed physical size. In the analog days, this was based on the film size (such as 35 mm, 70 mm, and so on). With digital cameras, it is the sensor size (such as 1/3, 1/2, APS, and so on). The size of this fixed capture area works in conjunction with the focal length to control the field of view that is captured. Modo offers the ability to modify the virtual capture area to any conceivable size.

For example, should you wish to compose 3D rendered elements together with images from a camera, for the most convincing results, matching the real-life sizes and aspect ratios of the original camera is highly recommended. Many popular preset sizes are provided for convenience, but you can also define custom values with the **Width** and **Height** settings. The **Film Fit** setting allow you to control Modo's behavior when the frame size specified in the Render Item, is a different aspect ratio than the film back settings.

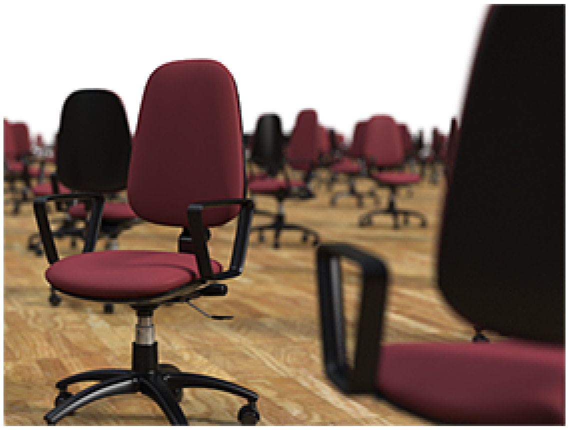
Film Back	
Film Back Preset	This control allows you to choose from a number of available preset real world cameras. Choosing a preset automatically sets the **Film Width** and **Height**, and the corresponding **Frame Size** (in the **Render Items** > **Frame Tab**). You can also save your own presets by selecting **(new preset)** from the bottom of the list. When selecting this option, a dialog displays asking you to name your preset. Your new preset is saved in Modo's config file and is available immediately for selection.



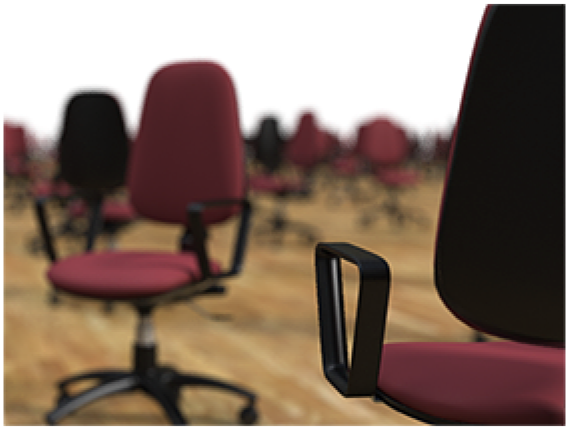
Film Width/Height	**Film Width** and **Film Height** correspond to the exposed region of the film negative in a real camera set in real world measurements. In some applications, this is known as the **Film Gate**. One way to visualize this is to imagine the camera frustum as a pyramid whose base dimensions are the **Film Width** and **Height** and terminating at the apex. The distance from the base to the apex is the **Focal Length** (the apex being the position of Modo's virtual aperture and the position where the camera rotates, which incidentally, is also the nodal point). The base of the pyramid can be shifted around within its plane (shearing the pyramid) by using the **Film Offset** controls.
Film Offset X/Y	This function allows you to offset the position of the **Film Back** without changing the position of the Camera Item itself, providing unique perspective views not otherwise possible. The function is especially useful when rendering architectural type images that prefer two point perspective as opposed to three. By rotating the camera on the X axis to 0° (eliminating the vertical perspective), the offset can be adjusted to modify the camera's view (framing the subject properly) while retaining

	the perspective of the camera's original position. The three images below illustrate the differences between film offset and simply moving the camera. **A.)** The camera standard perspective view with the subject centered in the frame. **B.)** Rotates the camera X axis to 0° and uses a **Film Offset** to frame the subject eliminating the vertical perspective. **C.)** Is the default perspective camera again with no offset, but the camera was rotated to 0° on the X axis again and the camera's position was modified upward to frame the subject producing a dramatically different perspective than that offered by **B**.  A.) Normal Perspective Camera B.) 0° X-axis rotation w/ Film Offset C.) 0° X-axis rotation w/ camera offset
Film Roll	Allows for 2D camera rotation. The results are displayed in the 3D Viewport when the animation is played. The rotation is relative to the Film Offset X/Y values.
Film Fit	**Film Fit** only matters when the Camera Item's film gate (defined by the **Film Width** and **Height** above) has a different aspect ratio than the Render Item's resolution gate (defined by the actual frame's **Width**, **Height**, and **Pixel Aspect Ratio**). In that case, the two possibilities are that the horizontal borders coincide, or the vertical borders coincide. The **Fill** and **Overscan** options automatically select whether to match the horizontal or vertical borders in order to make the resolution gate fit within the film gate or vice versa. You can think of the film gate (**Film Width** and **Height**) as the size of the film in the camera in real world units, with the film being analog (not divided into pixels). Then you take the exposed film frame and lay it on a digital scanner (the resolution gate), which is divided into pixels, and which might have a different shape (wider or taller). It's the resolution gate that gets saved to the output image file.

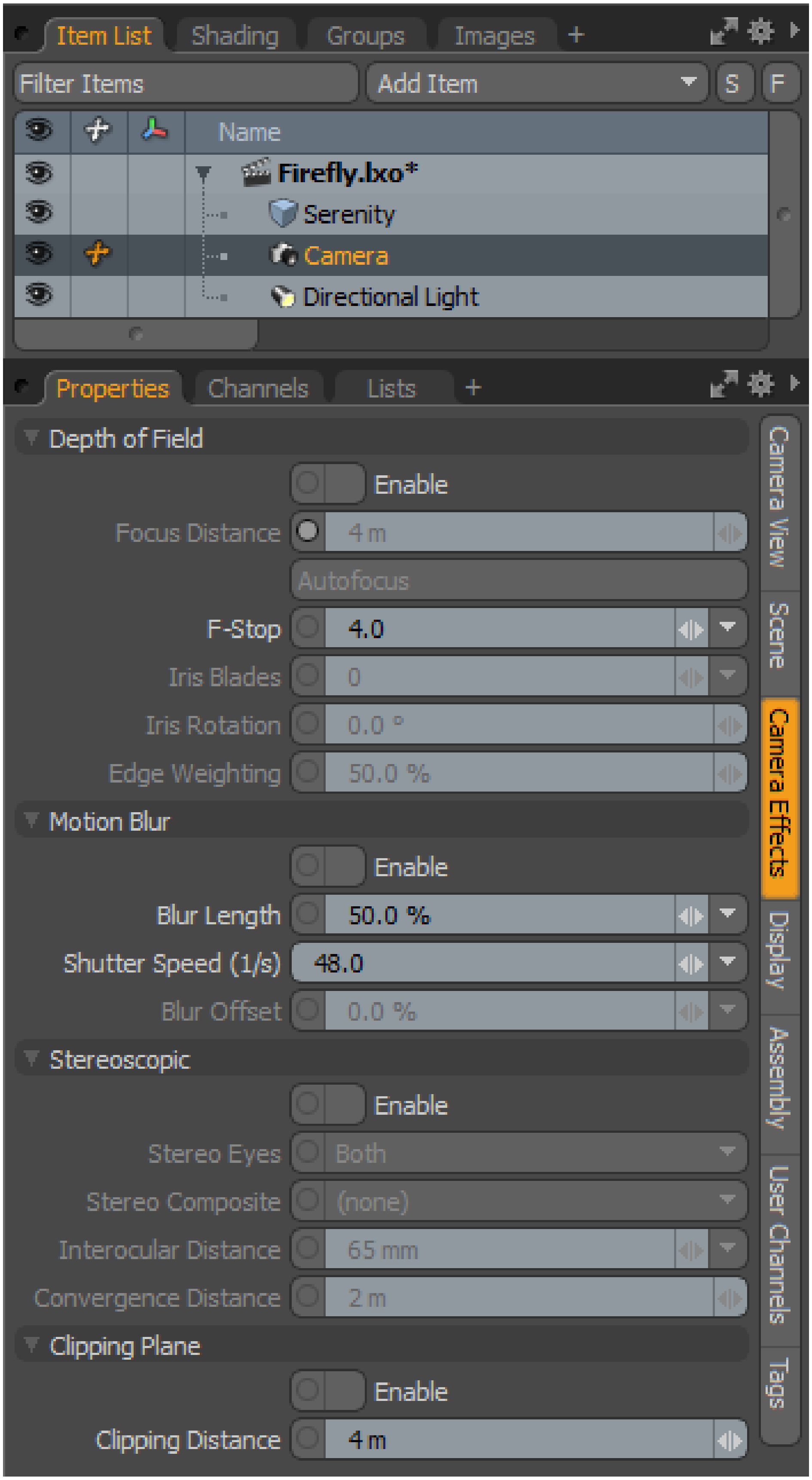
Camera Effects Properties



Distant Chair in Focus



Near Chair in Focus



Item List
Shading
Groups
Images
Filter Items
Add Item
S
F
Name
Firefly.lxo*
Serenity
Camera
Directional Light
Properties
Channels
Lists
Camera View
Scene
Camera Effects
Display
Assembly
User Channels
Tags

Depth of Field

○ Enable

Focus Distance 4 m

Autofocus

F-Stop 4.0

Iris Blades 0

Iris Rotation 0.0 °

Edge Weighting 50.0 %

Motion Blur

○ Enable

Blur Length 50.0 %

Shutter Speed (1/s) 48.0

Blur Offset 0.0 %

Stereoscopic

○ Enable

Stereo Eyes Both

Stereo Composite (none)

Interocular Distance 65 mm

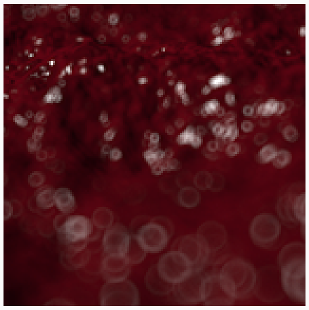
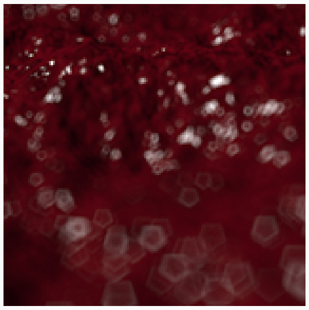
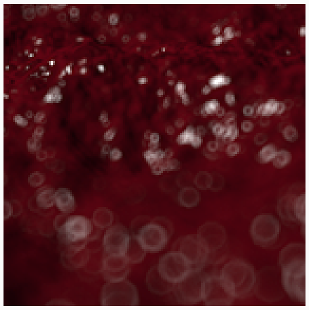
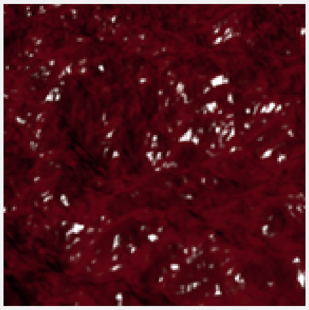
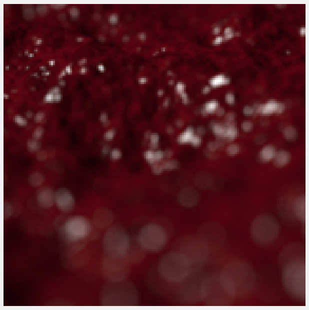
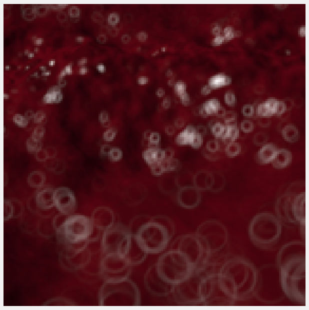
Convergence Distance 2 m

Clipping Plane

○ Enable

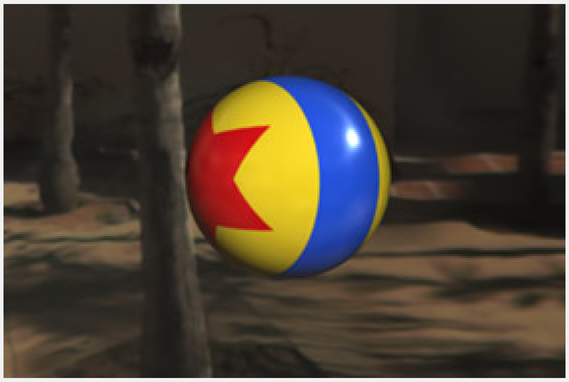
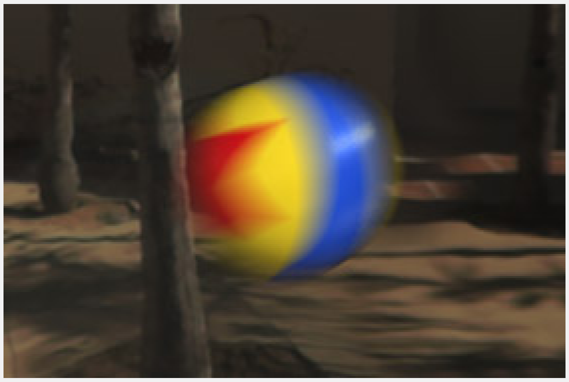
Clipping Distance 4 m

Option	Description
Depth of Field	
Depth of Field (DOF), as it relates to photography, is the area of acceptable focus in an image; any area outside gets increasingly blurry as it recedes from the focal plane. The softening of edges as they recede from the plane is highly dependent on the **F-Stop** setting. Smaller values represent larger apertures producing softer blurs, conversely, higher values represent smaller apertures producing less blurriness.	
Enable	This option toggles the **Depth of Field** effect on and off when rendering.
Focal Distance	When rendering with **Depth of Field** enabled, this value sets the distance out from the camera center at which the image is perfectly in focus. To quickly set the focus distance for the camera, you can use the **Autofocus** button or press **Ctrl**+**A** with the mouse over the **Camera** viewport. You can also set this value interactively with the **Item Transform** tool (**Y**) when the **Camera** item is selected.
Autofocus	This command sets the **Focal Distance** to whatever geometry is at the direct center of the current camera view. Setting the **Focal Distance** is only useful when rendering with **Depth of Field** enabled. You can also use the keyboard shortcut **Ctrl**+**F** when the mouse is over any **Render Preview** window.
F-Stop	This control on the Modo camera is only effective when **Depth of Field** is enabled. In this case, lowering the **F-Stop** decreases the depth of field (apparent area of focus) just as it would in real-world cameras. **Note:** **F-Stop** does not affect exposure of light since the Modo camera captures real world radiance values as they are striking the virtual CCD. This provides linear light values to the final image which provides an opportunity for post processing the exposure for maximum flexibility.
Iris Blades	This controls the way the blurriness looks in images with shallow depth of field, generally referred to as **Bokeh**. Bright areas, when out of focus tend to take on the shape of the camera's iris, so a smaller number of blades make more obvious geometric patterns in the bokeh shapes, while larger values look more circular. A setting of **0** makes a perfectly circular shape.

Option	Description
 Iris Blades 0 (default)  Iris Blades 5  Iris Blades 9	
Iris Rotation	This option changes the rotation angle of the virtual iris when rendering shallow depth of field in images, which in turn rotates the bokeh shapes in the final rendered output.
Edge Weighting	When Modo calculates **Depth of Field**, rays are fired in a pattern that simulates the iris blades producing the bokeh effect shapes. The edge weighting option concentrates those rays toward the center or edges of the iris, depending on the value. A value of 0% concentrates all the values at the center of the iris, while a setting of 100% concentrates the rays toward the outer edge. A value of 50% evenly distributes the rays. As light tends to be stronger in the outer edge of the iris in the real world, setting around 60% to 70% tend to produce the most realistic results.
 Edge Weight 0%  Edge Weight 50%  Edge Weight 100%	

Motion Blur

Real cameras don't freeze time, but capture brief moments, causing moving objects to blur in the frame. This natural motion blur that our eyes are accustomed to can be simulated in Modo. Once **Motion Blur** is enabled, Modo takes time samples forward and backward from the rendered frame blending them

Option	Description
	together to create the blur effect. Depending on the strength of the effect or the speed of motion in your frame, you may find it necessary to increase the number of anti-aliasing samples. Adjustments to **Blur Length** (like adjusting the exposure time for more or less blur) and **Blur Offset** can be found in the Camera Item's properties.  Animated Ball, Motion Blur Disabled  Animated Ball, Motion Blur Enabled
Enable	This option toggles the **Motion Blur** effect on and off when rendering.
Blur Length	This control affects the length of the motion blur effect when rendering animations. The default value of 50% is similar to a real-world camera set with a 180 degree shutter. In Modo, the 50% value results in the shutter being open for half of the exposure time of that frame. The blur effect is calculated by taking time samples both forwards and backwards from the current frame time and blending them together for the final blurred result. As such, it may be necessary to increase the level of anti-aliasing to produce a smooth result.
Shutter Speed (1/s)	Sets the shutter speed of the camera. The slower you set the shutter speed, the more pronounced the motion blur effect gets. You can enter a value manually or use one of the value presets by clicking on the small arrow on the right of the field. Here you can also save new presets by clicking **(new)** or remove an existing preset by clicking **(delete)**.
Blur Offset	This lets you control the center position in time where Modo samples its motion blur from. A setting of zero centers the motion blur on the current rendered frame, while a setting of -100% moves the position backwards one entire frame and a setting of 100% moves the position forward one entire frame.

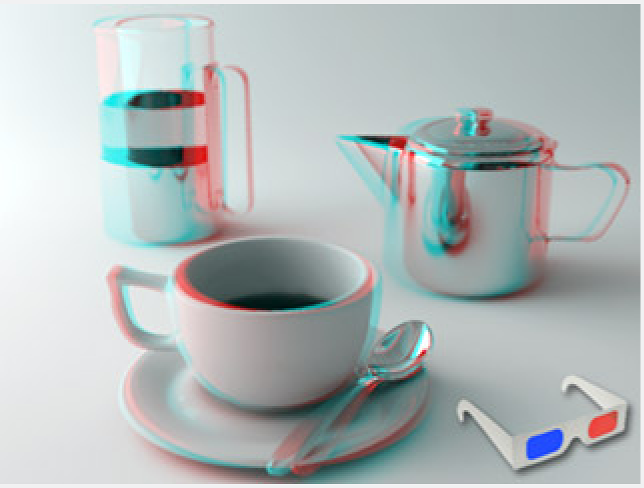
Option	Description
	Tip: The **Antialiasing** setting's effect is most apparent on geometric edges. A setting of 8 is generally acceptable and is a good trade off between quality and speed, though in reproducing effects such as **Depth of Field** or **Motion Blur** you may need to increase this value, possibly all the way up to 1024 to produce a pleasing result. If you find your shaded edges, such as cast shadows, refractions, reflections, procedural and image based textures are aliased (not smooth), then reducing the shading rate value or lowering the refinement threshold can help to increase shaded edge quality. If you are finding specific areas of an image to be noisy, that cannot be solved by Antialiasing and Refinement, such as soft reflections, blurry refractions, soft shadow edges, subsurface scattering, it may simply be a case where raising the number of samples specific to the effect eliminates the noise. These setting can be found in the associated **Properties** panels. Always keep in mind though, as more calculations are taken into account, the longer an image takes to render.

Stereoscopic

We see with two eyes, each slightly offset from the other, providing our brains with enough information to create a sense of depth to the world around us. Modo scenes are likewise 3D with depth, but the monitors we view scenes with are two-dimensional, as are the renders Modo creates. Activating the stereoscopic function in Modo brings back that missing depth information by rendering two offset images, that when viewed in tandem, reveal the depth present in your original scene.

The **Stereoscopic** function, when enabled, renders two separate frames, one appended **L** for left and **R** for right, noting which eye the frame was rendered for.

Note: When enabled, the Stereoscopic function is rendering two completely separate frames, so increased render times are to be expected. If applicable, only a single Irradiance Cache is calculated for stereoscopic renders.

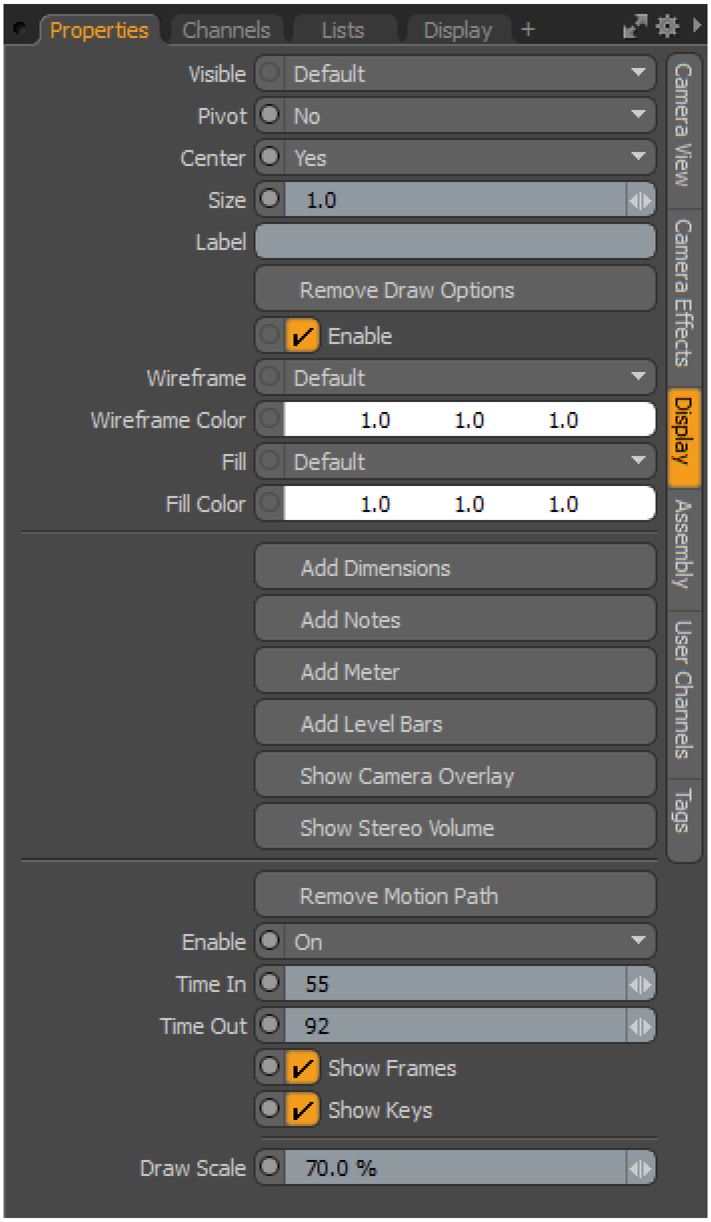
Option	Description
	
Stereoscopic Enable Toggle	This setting is the global toggle to enable or disable the Stereoscopic function when rendering. When enabled, Modo renders two frames slightly offset from each other (like our eyes). The resulting images can then be combined in a variety of ways that, when viewed appropriately, reveal the depth present in your original scene.
Stereo Eyes	When Stereoscopic rendering in enabled, you can use this option to choose which eye is rendered. Choose to render only the Left eye, only the Right eye, or both simultaneously. When rendering both, you can additionally choose a **Stereo Composite** option to determine how Modo combines the resulting stereo images for final output.
Stereo Composite	Modo offers a variety of options for composing stereo images into a single image for viewing. Which option you choose is largely based on how the resulting image is viewed: • **Anaglyph Red-Cyan** - This method replaces the red channel or the right image with the red channel from the left image. • **Anaglyph Grayscale** - This method makes both frames into grayscale (luminance) images, then puts the left one in the Red output channel and the right one in the Green and Blue channels. • **Anaglyph Red-Cyan Half Color** - This method basically converts the left image to grayscale and puts it into the Red channel, and uses the right frame's Green and Blue channels for the output Green and Blue channels. This technique reduces 'retinal rivalry', which is a disturbing phenomenon caused by differing brightness of what should be the same objects seen by either eye.

Option	Description
	• **Anaglyph Red-Cyan Optimized** - This method further combats retinal rivalry by discarding the red channel from both images, keeping the right frame's cyan component, and mixing the left frame's Green and Blue channels at 70%/30% ratio to use in the output red channel. • **Anaglyph Red-Blue Least Squares** - This algorithm was invented by Eric Dubois. He used the optical properties of standard red and blue filters used for anaglyph glasses and derived magic numbers for blending stereo pairs into color anaglyphs by minimizing the error using a "least squares" method. • **Side-by-Side** - Half-width frames are rendered and composed side by side into full width regular frames, with Left on Left and Right on Right. • **Side-by-Side Full Width** - Full sized frames are rendered separately and composed side by side in double width frames, with Left on Left and Right on Right. • **Side-by-Side Cross Eyed** - Half-width frames are rendered and composed side by side into full width regular frames, with Left on Right and Right on Left. • **Side-by-Side Cross Eyed Full Width** - Full sized frames are rendered separately and composed side by side in double width frames, with Left on Right and Right on Left.
Interocular Distance	The **Interocular Distance** technically means the distance between the center point of a person's two pupils, but here it is the measured offset between the two rendered images in Modo. The default setting of 65 mm is the average distance between the eyes of an adult. For scenes modeled to real world values, this is an appropriate setting that produces good results. Scenes of varying scale or exceptionally close/far objects may require this distance to change.
Convergence Distance	The convergence distance is the point within 3D space where the two offset images converge. In front of this point, objects appear to come forward, and behind this point, object appear to recede. It is easily understood by imagining it as the focal point of your scene, much like with DOF. Where the focal distance is the position in space where all objects are in focus, objects in front or behind this position gradually blur, the **Convergence Distance** is the point in the scene you wish to direct your viewers eyes, as it is the most comfortable to view. This is exactly how your eyes work as well. If you were sitting in a restaurant and your eyes were focused on your glass of water on the table, this would effectively be the convergence point of your two eyes, your dinner plate would be in front of the glass coming forward, and your guest would be further away on the other

Option	Description
	side of the table.
Clipping Plane	
Clipping refers to removing part of a scene from a rendered image while leaving the actual scene geometry intact. You can define a distance where all (or selected) surfaces are clipped, by enabling the **Clipping Plane** option and then defining a **Clipping Distance**. You can disable clipping on a per-material basis by disabling the **Enable Surface Clipping** option in the Material item. **Tip:** A similar effect can be obtained through the use of Render Booleans that are not camera view dependent.	
Enable	When enabled, any scene items within the bounds of the camera position and the clipping plane are clipped from the scene, removing them from the rendered image while retaining the actual scene geometry.
Clipping Distance	Defines the distance away from the camera where items are clipped.

Camera Display Properties

The camera display options can be accessed from the **Display** tab in the camera's **Properties** viewport. These options determine how the camera is drawn in the 3D viewport.


Properties
Channels
Lists
Display
+
Camera View
Camera Effects
Display
Assembly
User Channels
Tags
Remove Draw Options
Add Dimensions
Add Notes
Add Meter
Add Level Bars
Show Camera Overlay
Show Stereo Volume
Remove Motion Path

Visible: Default
Pivot: No
Center: Yes
Size: 1.0
Label:
☑ Enable
Wireframe: Default
Wireframe Color: 1.0 1.0 1.0
Fill: Default
Fill Color: 1.0 1.0 1.0
Enable: On
Time In: 55
Time Out: 92
☑ Show Frames
☑ Show Keys
Draw Scale: 70.0 %

Camera: Display	
Visible	Sets visibility for the item itself in the 3D viewport. Options are: • **Default** - respects viewport settings. • **Yes** - sets the item as always visible. • **No** - disables visibility entirely. • **No (Hide Children)** - disables visibility of all related child items.
Pivot	Sets visibility for the 3D item center. Options are:

	• **Default** - respect viewport settings. • **Yes** set the center as always visible. • **No** - disable center visibility entirely.
Center	Sets visibility for the 3D item center. Options are: • **Default** - respect viewport settings. • **Yes** - set the center as always visible. • **No** - disable center visibility entirely.
Size	A multiplier that adjusts the display size of the locator.
Label	Displays next to the locator in the viewport for easy identification. It adds two additional properties to control the display of the label in the viewport: **Show Label** and **Label Offset**.
Show Label	Temporarily enables/disables the display of the label in the viewport without losing settings. This option is only available when a label is specified.
Label Offset X, Y	Specifies the X and Y distance away from the locator where the label appears. This option is only available when a label is specified.
Add/Remove Draw Options	The **Draw Options** control the look of the locator in 3D Viewports, they are mostly used to define a custom wireframe color for the camera item.
Enable	Toggles the camera draw options.
Wireframe	Determines the camera wireframe draw style. Options are: • **Default** - The standard white wireframe. • **User** - The color can be customized using the **Wireframe Color** setting. • **Background Light** - A color to retain camera wireframe visibility against a light background. • **Background Dark** - A color to retain camera wireframe visibility against a dark background.
Wireframe Color	Determines the camera wireframe color when the **Wireframe** mode is set to **User**.
Fill	If the viewport style is set to **Solid** (see 3D Viewport Styles), this setting

	determines the fill color of 3D items. • **Default** - The standard gray fill. • **User** - The color can be customized using the **Fill Color** setting. • **Background Light** - A color to retain item visibility against a light background. • **Background Dark** - A color to retain item visibility against a dark background.
Fill Color	Determines the item fill color when the **Fill** mode is set to **User** and the viewport style is set to **Solid**.
Add/Remove Dimensions	When enabled, dimension values for the overall bounding box size appear in the viewport around the selected mesh item. An additional **Dimensions** sub-tab shows in the **Properties** viewport, allowing further display customizations.
Add/Remove Notes	When enabled, you can define multiple lines of text (up to nine) that show up when the item is selected. An additional **Multi-Line Notes** sub-tab shows in the **Properties** viewport, allowing further customization.
Add/Remove Meter	When enabled, displays an on-screen analog-style meter graph (this looks like a speedometer) that can be rigged to show individual values in the 3D viewport. An additional **Dial Meter** sub-tab shows in the **Properties** viewport, allowing further customization.
Add/Remove Level Bars	When enabled, displays an on-screen equalizer-type graph for visualizing multiple on-screen values. An additional **Equalizer Bars** sub-tab shows in the **Properties** viewport, allowing further customization.
Show/Hide Camera Overlay	When enabled, displays the camera overlay graphics. An addition **Camera Overlay** sub-tab shows in the **Properties** viewport. For a full description of camera overlays see Camera Overlay Features.
Show Stereo Volume	Enables the display of the **Stereo Volume** guides in the 3D viewport to give you an understanding of the amount of parallax in a given stereo scene. When enabled, a series of options to control the size and opacity of the guides can be found in the **Camera** item's properties panel only when the **Show Camera Overlay** option is enabled.
Add/Remove Motion Path	Displays the animation motion path (if any) for the camera. Clicking **Add Motion Path** reveals further customization parameters.
Enable	Determines if the path draws. Options are: • **Off** - don't draw the motion path.

	• **On** - draw the motion path irrespective of selection status. • **Selected** - draw the motion path if selected.
Time In	The number of frames before the current time in the timeline that are displayed by the curve.
Time Out	The number of frames after the current time in the timeline that are displayed by the curve.
Show Frames	Toggles the dots along the curve that mark each frame along the motion path. The dots can be useful to see how the element moves in time, but can also cause clutter for some types of motion and so can be disabled.
Show Keys	Toggles the white dots that mark the keyframes along the motion path.
Draw Scale	This extrudes a ribbon shape outward from the motion path that represents the orientation angle of the animated element. The larger the scale value, the wider the ribbon representation in the viewport.

Camera Overlay Features

Modo's camera view can be augmented with a number of overlays to help you compose a scene. You need to enable these options by clicking **Show Camera Overlay** in the **Display** tab of the camera item **Properties**. This adds the **Camera Overlay** tab to the **Properties** panel.

> **Tip:** The **Camera Overlay** guides only appear when a viewport is set to the **Camera** view type. Also, the camera must be set to **Visible** within the viewport. Ensure **Show Camera** is enabled in the Visibility options and the camera is visible in the Items (Scenes) List.

Camera Overlay

The **Camera Overlay** section allows you to toggle and resize the safe area guides. It also contains controls for adjusting the appearance of any overlays.

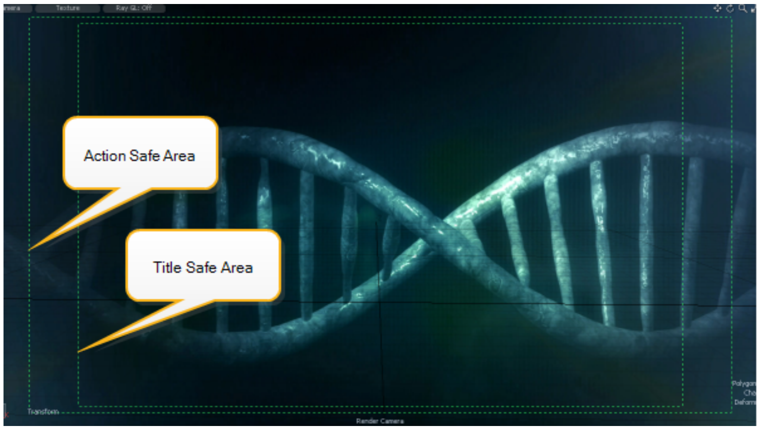
Safe areas help you take overscan into account, when composing the camera view. Overscan occurs in broadcasting, and refers to the situation in which the image is blown-up larger than the visible screen size. This means that not all of the image is seen on screen as the image is magnified and cropped.

To manage overscan, broadcasters work within safe areas. A safe area is a region of the image that should not be affected by overscan. Two universal safe areas have been adopted:

- Title Safe - An area where text is guaranteed not to be cropped. Text such as titles or subtitles should be placed in this area.
- Action Safe - A larger area that marks the edge of the picture that is likely to be seen on screen. Unlike text, it is acceptable for the action to reach the edges of the Action Safe area.

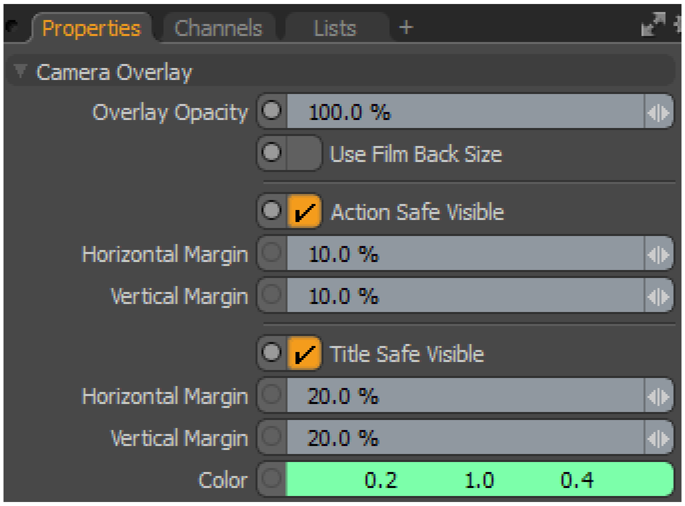
Note: It is not advisable to have text up against the edge of the screen, and so the Title Safe area is always smaller than the Action Safe area.

The image below shows typical Title Safe and Action Safe area boundaries as shown in Modo.



Note: The safe areas used in the image above correspond to those used by the BBC for 16:9 images. The title safe area is 80% horizontal by 90% vertical, and the action safe area is 93% horizontal by 93% vertical.

Camera Overlay Properties



Camera Overlay

Overlay Opacity: 100.0 %

○ Use Film Back Size

☑ Action Safe Visible

Horizontal Margin: 10.0 %

Vertical Margin: 10.0 %

☑ Title Safe Visible

Horizontal Margin: 20.0 %

Vertical Margin: 20.0 %

Color: 0.2 1.0 0.4

Camera Overlay	
Overlay Opacity	Adjusts the transparency of the overlay. This applies to the safe areas, the Composition Guides, and the Lines.
Use Film Back Size	Check this to calculate overlay size based on the Film Back Properties size, rather than the render image size. **Note:** Leave this unchecked unless you have a specific reason to do so. Older versions of Modo (pre-902) used the film back size to generate overlays, but this was found to be problematic. The option has been left in to maintain compatibility with previous versions. That said, it may be useful if you wish to use the film back size as a guide when rendering the scene to different image resolutions.
Action Safe Visible	This option toggles the visibility of the Action Safe area boundary. The Action Safe area is the region in which the important action of a scene should be clearly visible. The bounds of this area are likely to match the edges of the viewing screen.
Horizontal Margin	This determines the Action Safe boundary distance away from the left and right edges of the camera viewport. The value applies to each side of the boundary. To show a horizontal Action Safe

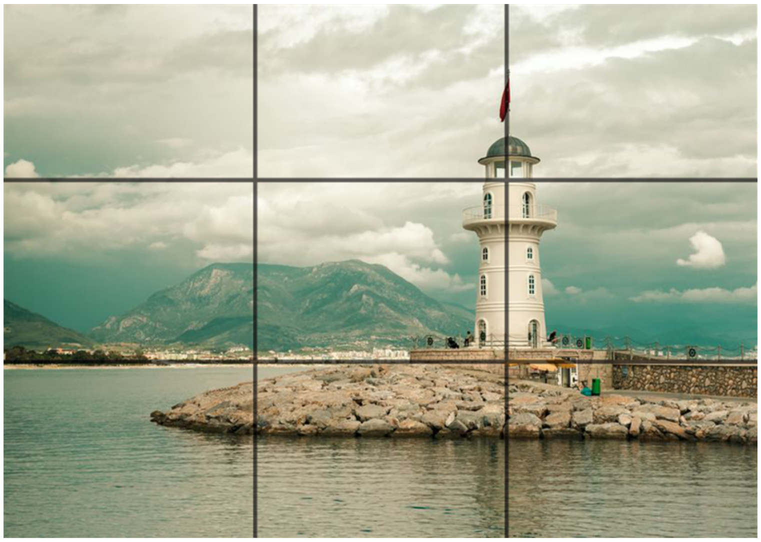
	area of 93%, you would set the **Horizontal Margin** to 3.5% (7% divided by 2).
Vertical Margin	This determines the Action Safe boundary distance away from the top and bottom edges of the camera viewport. The value applies to each side of the boundary. To show a vertical Action Safe area of 93%, you would set the **Vertical Margin** to 3.5% (7% divided by 2).
Title Safe Visible	This option toggles the visibility of the Title Safe area boundary. The Title Safe area is the region in which on-screen text and graphics are guaranteed to be visible and clearly legible.
Horizontal Margin	This determines the Title Safe boundary distance away from the left and right edges of the camera viewport. The value applies to each side of the boundary. To show a horizontal Title Safe area of 80%, set the **Horizontal Margin** to 10% (20% divided by 2).
Vertical Margin	This determines the Title Safe boundary distance away from the top and bottom edges of the camera viewport. The value applies to each side of the boundary. To show a vertical Title Safe area of 90%, set the **Vertical Margin** to 5% (10% divided by 2).
Color	The color of the safe area boundary lines.

Composition Guides

To assist you in organizing elements within a scene, Modo provides a number of camera overlay guides based on common compositional techniques. The guides are divided into two main types:

- **Grid** - Symmetrical grid-lines spaced at regular intervals, and a grid based on the golden ratio.
- **Golden Rectangle** - Division of space based on the golden ratio. This can be drawn as square and rectangular sections, a logarithmic spiral, or both.

Most of the **Grid** types consist of equally spaced lines. For example, the **Rule of Thirds** splits the view into three equal rows and columns. This is a popular compositional guideline in painting and photography, which recommends that important compositional elements (such as the horizon) should be placed along the lines, or at intersections.



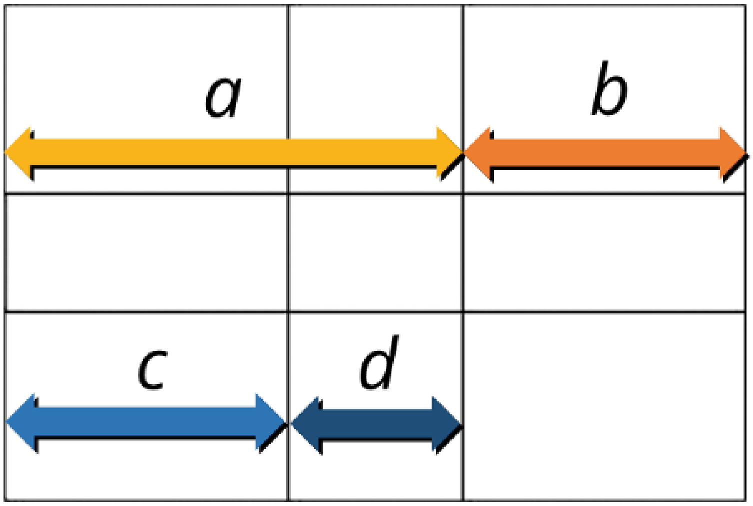
The image above illustrates the rule of thirds in practice. A thirds grid has been overlaid to demonstrate the compositional technique. Notice how the coastline and the lighthouse are closely aligned to the grid lines.

The Golden Ratio Overlays

Grid lines that are spaced equally provide a simple and popular means of composition. However, some argue that the ratio of proportions provided by the golden ratio provides a more natural and pleasing relationship. The golden ratio has its origins in ancient mathematics and is a ratio of proportions that is believed to create visual harmony. The actual value of the golden ratio is a little over 1.618 and has been assigned the greek letter φ (phi) . You can use the guides in Modo to help you apply the ratio with a high degree of accuracy.

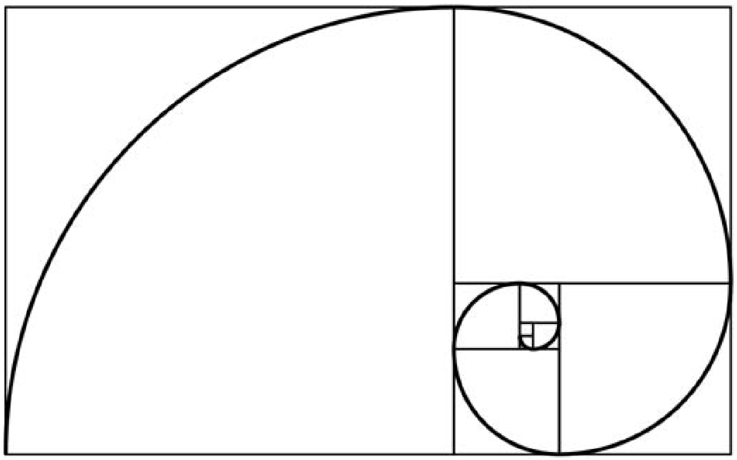
Note: If you have two lengths *a* and *b*, the ratio of their lengths is *a*:*b*. For the lengths to conform to the golden ratio, the ratio of *a*:*b* must be equivalent to the ratio *a*+*b*:*a*. In other words, *a*/*b* = (*a*+*b*)/*a*.

Modo provides a **Golden Ratio** grid in the **Grid** list (also known as a Phi Grid). Each line is intersected so that the lengths at either side of the intersection conform to the golden ratio.



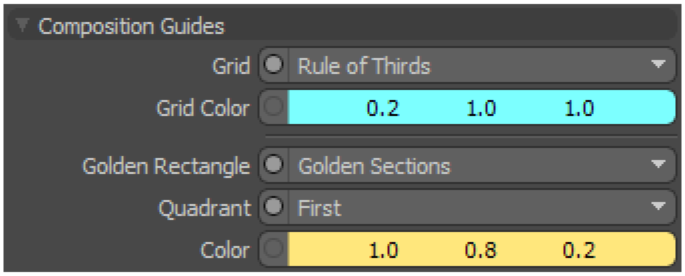
The image above shows a golden ratio grid; the ratio *a*:*b* conforms to the golden ratio, as does *c*:*d*. This also applies vertically. In practice, you'd simply use the grid lines to position important compositional elements, just as you would with the other grids.

Modo provides an overlay based on repeated division by the golden ratio, in the **Golden Rectangle** option. A golden rectangle has dimensions that conform to the golden ratio. This gives rise to an interesting property of the rectangle: If you divide the rectangle into a square and a smaller rectangle, the smaller rectangle is also a golden rectangle. By repeated subdivision, the **Golden Rectangle** grid is formed. By drawing a quarter circle through each square, a spiral is formed. This is known as a **Logarithmic Spiral** as the radius increases exponentially from the smallest square. The image below shows a sub-divided golden rectangle and the logarithmic spiral.



Why would you use this layout in Modo? Again, it presents an alternate set of lines to which you can match compositional elements. When used properly, the spiral can help guide a viewer's eye through a scene and ultimately to the main focus point. The small part of the spiral can be placed over the main focus point, with the rest of the image roughly distributed according to the proportions of the squares.

Composition Guides Properties



Composition Guides

Grid: Rule of Thirds

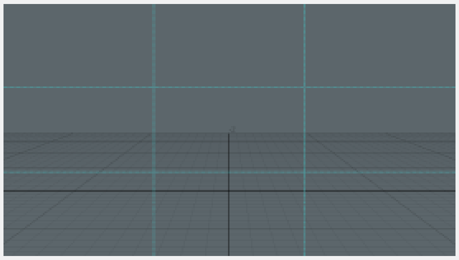
Grid Color: 0.2 1.0 1.0

Golden Rectangle: Golden Sections

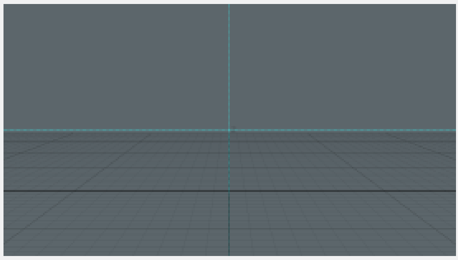
Quadrant: First

Color: 1.0 0.8 0.2

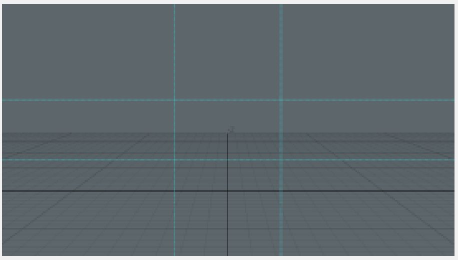
Composition Guides	
Grid	The type of grid. • **Rule of Thirds** - A 3x3 grid made of lines spaced equally along the horizontal and vertical. • **Center Lines** - Bisects the viewport horizontally and vertically. • **Golden Ratio** - A basic Phi Grid. Each line divides the line it intersects according to the golden ratio. • **Tenths** - A 10x10 grid made from line spaced equally along the horizontal and vertical. • **Square** - A grid consisting of squares. The height of each square is equal to one tenth of the viewport height. • **Off** - Hide the grid.



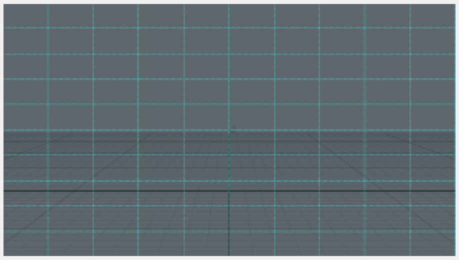
Rule of Thirds



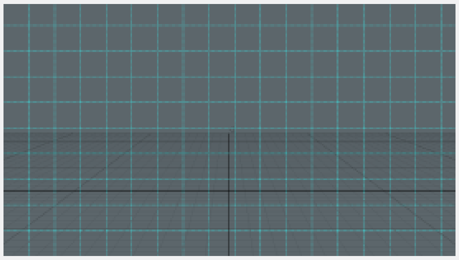
Center Lines



Golden Ratio

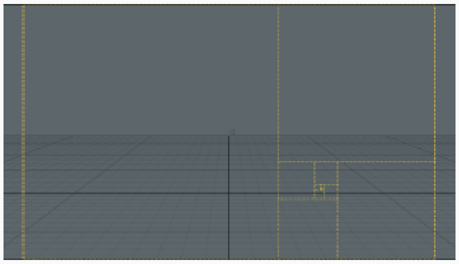


Tenths

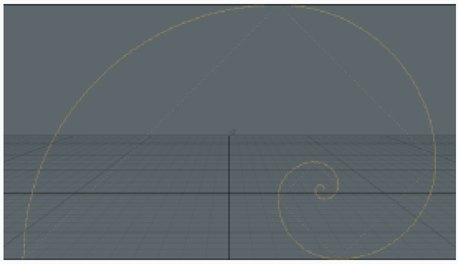


Squares

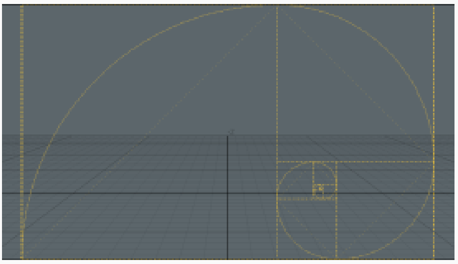
Grid Color	Use this to set the color of the grid.
Golden Rectangle	Divide the grid into regions by using the golden ratio. • **Golden Sections** - A grid formed by repeated division into squares and golden rectangles. • **Logarithmic Spiral** - A logarithmic spiral whose growth factor is equal to the golden ratio. • **Sections and Spiral** - Both of the above. • **Off** - Hide the lines.



Golden Sections



Logarithmic Spiral (first quadrant)

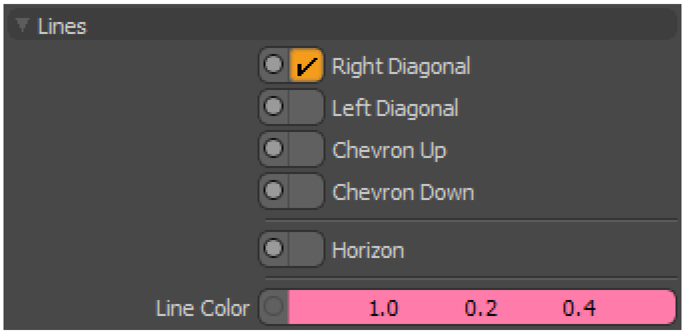


Sections and Spiral

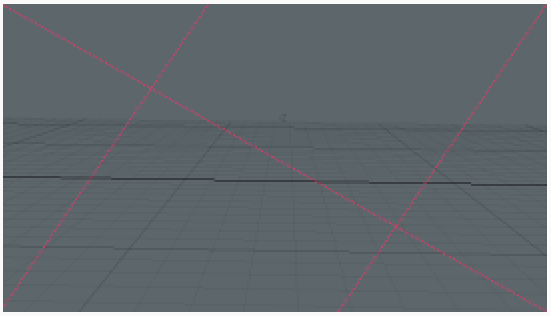
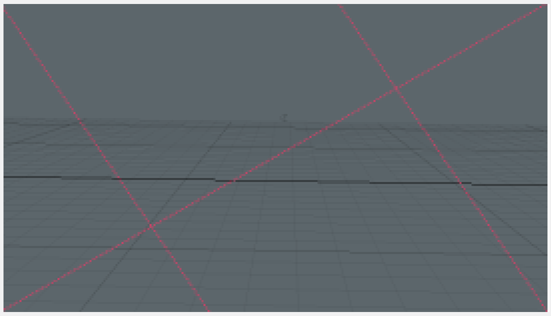
Quadrant	Choose the viewport quadrant in which to place the center of the spiral.
Color	The color of the **Golden Rectangle** lines.

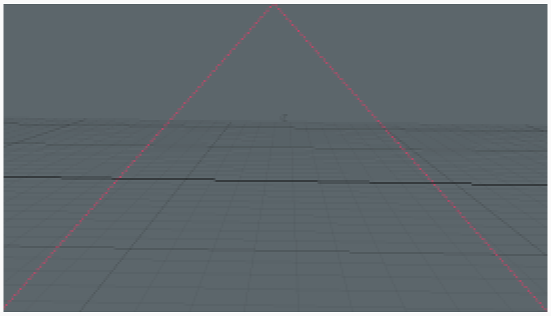
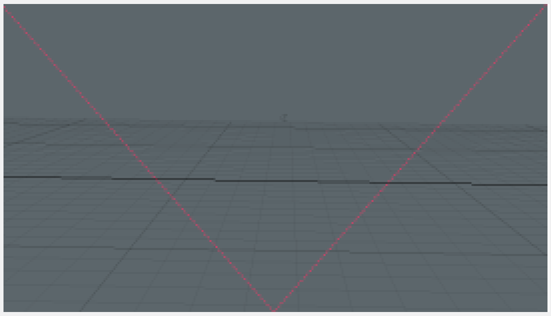
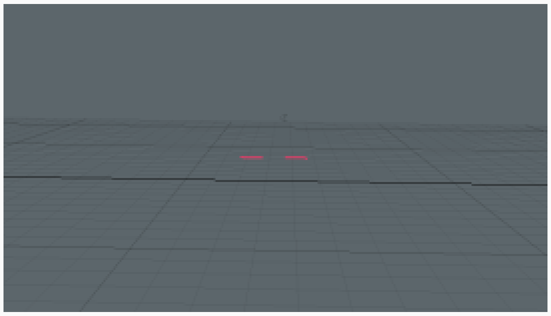
Lines

In addition to grids, Modo offers a set of diagonal lines and a horizon indicator to assist with composition. The lines help you place objects along the diagonal using common reference points, such as the corners, or center of the image. The horizon is a useful reference to show camera orientation with respect to the world. You can combine different types of lines as required.



☑ Right Diagonal
☐ Left Diagonal
☐ Chevron Up
☐ Chevron Down
☐ Horizon
Line Color 1.0 0.2 0.4

Lines	
Right Diagonal	Draws a diagonal line from the top-left to the bottom-right of the viewport, with two perpendicular lines that connect to the other corners. 
Left Diagonal	The reverse of the right diagonal. 
Chevron Up	Draws a chevron with its apex at the top of the viewport.

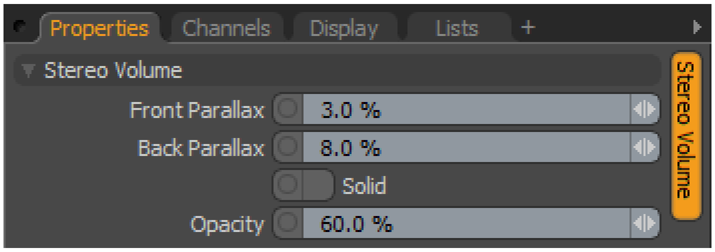
	
Chevron Down	Draws a chevron with its apex at the bottom of the viewport. 
Horizon	This indicates the horizon in relation to the current camera roll position. The horizon is always drawn at the center of the viewport and is updated when the camera is moved. 
Line Color	Allows you to choose the color of the lines.

Stereo Camera Volume

The separation of matching pixels in left and right stereo images is referred to as parallax. On the camera's convergence plane, which corresponds to the TV/Movie screen as far as the audience is concerned, there is no parallax. Objects at that distance appear to be in the same plane as the screen. Objects further than the convergence plane appear to be pushed back into the screen. The farther back they are, the greater the separation (parallax) between those pixels. Things in front of the convergence plane appear to be in front of the screen, and also have increasing parallax, though the images are shifted in the opposite direction.

If a scene exceeds a certain parallax amount, viewers may experience discomfort, or the intended effect could be ruined entirely. Conversely, if the scene doesn't have enough volume, then the stereo effect is barely noticeable. The amount of parallax into and out of the screen is critical to monitor when setting up a shot. Directors are even starting to plot depth scores so they can keep the parallax in a safe range, using it as a storytelling element as well as to catch sharp jumps which can be disturbing to viewers.

The **Stereo Volume** package on the camera enables you to define stereo volume by setting the front and back parallax amounts. You can then either use it to make sure the objects stay in range, or to measure shots by setting keyframes so it always matches the nearest and farthest elements in the scene. To get the **Depth Score**, just screen capture the Graph Editor.



Front Parallax 3.0 %
Back Parallax 8.0 %
Solid
Opacity 60.0 %

Option	Description
Stereo Volume	
This refers to the amount of depth of your scene between the front-most and back-most elements. It is an important quantity for designing a stereo image. It is referred to as a volume because it includes the entire space width, height, and depth of the frame. It is also related to the stereo budget, which is yet another way of saying the amount of depth used by the scene. You can activate the display of this important visual guide in the 3D viewport by enabling the Show Stereo Volume option in the **Display** tab. Once enabled, the following options appear in the **Properties** panel when the **Camera Item** is selected.	
Front/Back Parallax	This parameter sets the desired front and back parallax amounts. It is given as the separation distance's percentage of the screen width. When you set this, Modo computes the distance from the convergence plane where objects have that parallax amount, and draws the front or rear plane there. There are also two circles drawn with centers that are separated by the parallax amount.
Solid	When enabled, draws volume as filled quads, instead of outlines, making its

Option	Description
	apparent area more obvious.
Opacity	Determines the opacity of the drawn volume guides.

Camera Matching

Camera Matching adjusts a camera viewpoint so that the perspective in a background image matches the perspective of the Modo 3D world. This allows you to place 3D objects as if they were part of the background image.

The images below demonstrate Camera Matching in action. This is a rendered output of a series of Modo cubes overlaid onto the image of a parking lot. The perspective of the cubes and the parking lot were matched using the Camera Matcher.

ons
Effect: Shadin...
Rendering...
LUT: sRGB
(Render Camera)
Shading: Full

Note: The background image is loaded onto the camera. This is different to a **Backdrop Item**, which creates a new background plane in the **Item List**, and can be seen from any camera. To view a camera background image, you have to switch to the camera's view.

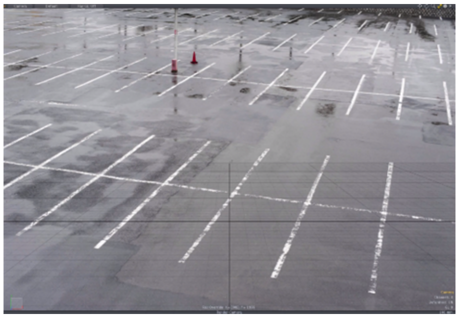
To match the camera background image to the 3D world, Modo needs to know the following:

1. Where are the lines of perspective in the image?
2. Which axis is each line parallel to?

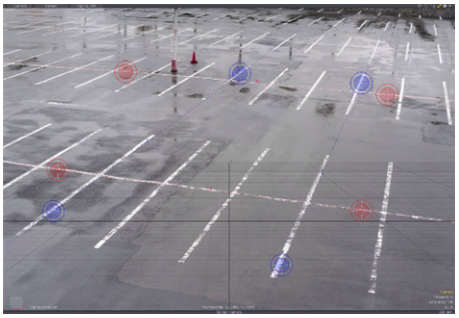
To provide Modo with this information, you must identify two pairs of perspective lines in the image, where each pair is parallel to one of the world axes. You then draw over these lines using the Camera Matcher. For example, in the image above, the markings that run from left to right provide good lines to match to the X-axis. The lines perpendicular to these, whose vanishing point is in the far distance, provide a good match to the Z-axis.

Note: One pair of lines must be orthogonal to the other pair.

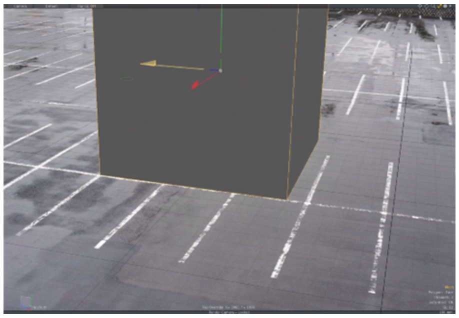
The image below shows the basic steps for camera matching.



Load the background image onto the camera.



Draw the perspective lines, and match the camera to the perspective.



Add models as required. Notice how the perspective of the model and the image now match.

The Camera Matcher positions the camera at the World Origin with an orientation to match the perspective in the background image. For additional accuracy, you can draw a fifth line that describes the length of an object in the image. This adjusts the camera position so that the scale of the image matches the scale of the Modo world.

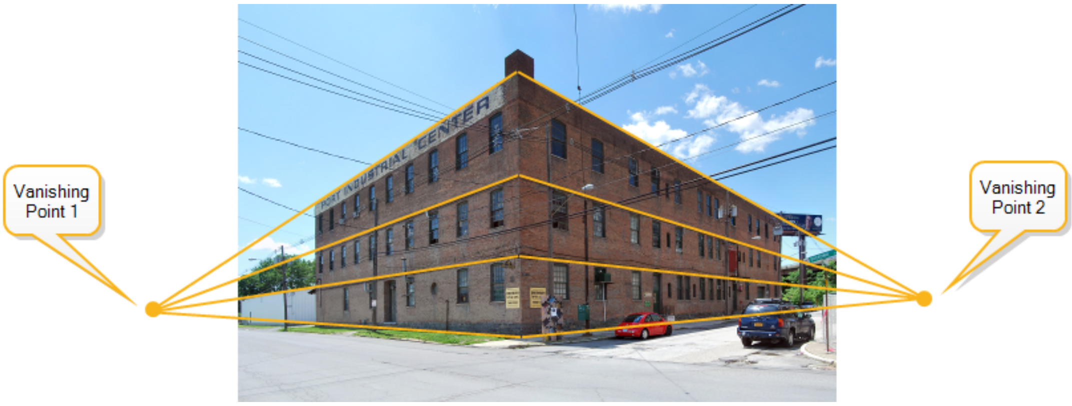
Once matched, you can view existing geometry against the background by either moving the geometry in-front of the camera, or translating the camera. Also, to help you draw new objects in front of the camera, the Camera Matcher lets you reposition the Work Plane relative to the camera.

Camera matching is performed in **Setup** mode using the **Background Image** properties tab in the camera's **Item Properties** panel.

Note: Each camera can have a single background image and camera matching configuration.

Two-Point Perspective

The Camera Matcher relies on two-point perspective, where orthogonal lines in the image converge at two vanishing points. The image below illustrates two-point perspective in a photograph of a building.


Vanishing Point 1
Vanishing Point 2

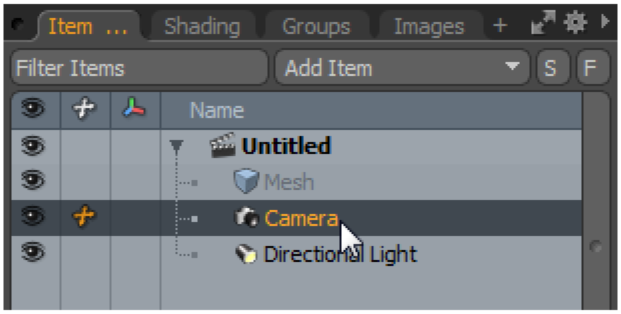
In this image, there are two orthogonal sets of lines; the lines on the left of the building corner are orthogonal to the lines on the right. Each set converges at a different vanishing point. Two-point perspective allows Modo to calculate a camera position with a high degree of accuracy.

Note: The lines used for a single axis must not run exactly parallel as they do not converge at a vanishing point.

Selecting and Looking Through the Camera

To select the camera and look through it.

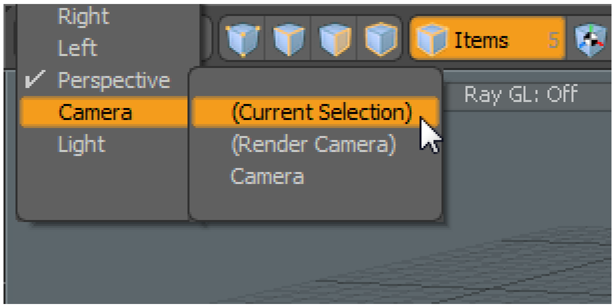
1. Click on the camera in the **Item List**.


Item ...
Shading
Groups
Images
Filter Items
Add Item
S
F
Name
Untitled
Mesh
Camera
Directional Light

2. Select the **Setup** layout.

Tip: In the **Setup** layout in **Items** mode, you can select a camera by clicking on it.

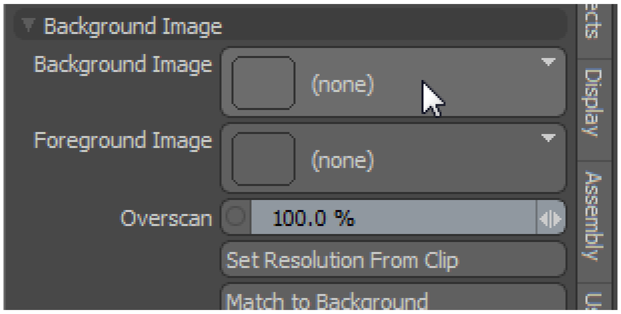
3. Switch to the camera viewpoint by selecting **Camera** > **(Current Selection)** in the view selector in the top-left of the viewport.


Right
Left
Perspective
Camera
Light
(Current Selection)
(Render Camera)
Camera
Items
5
Ray GL: Off

Adding a Background Image

Camera matching requires a background image to match against. You can load the image into the camera using the **Background Image** properties.

1. In the **Background Image** section of the camera's **Item Properties**, click on the **Background Image** control to open the image thumbnails.

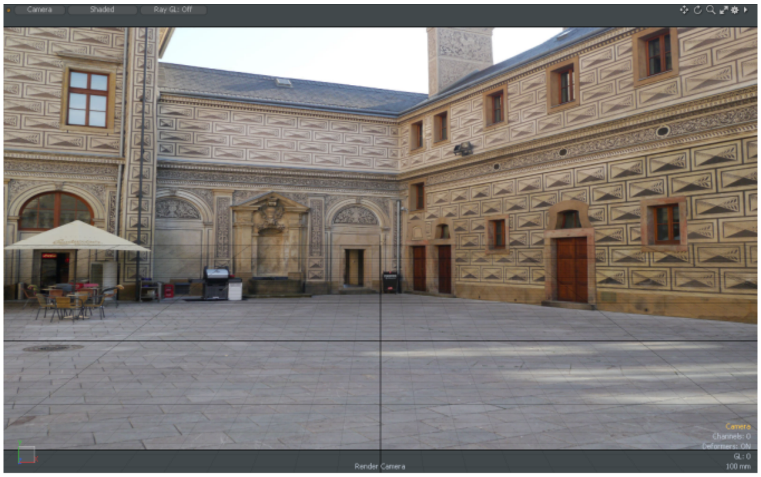

Background Image
Background Image
(none)
Foreground Image
(none)
Overscan
100.0 %
Set Resolution From Clip
Match to Background
ects
Display
Assembly
Us

2. Click the **Add Clip** dropdown and select **(load image)**.

   This opens the image file browser.

3. Browse to and select the background image. Click **OK**.

   The image appears in the background.



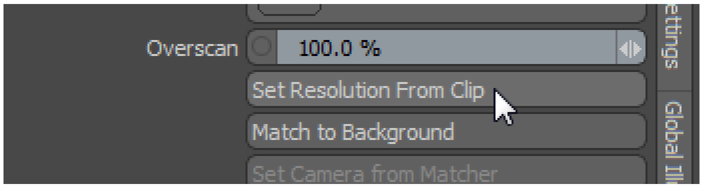
Adjusting Image Properties

If the background image has a different aspect ratio to the render resolution, the image aspect ratio is automatically adjusted to fit the render resolution. Usually, this is not wanted as the camera background image adopts a different resolution to the original source image. To override this, you can adjust the render resolution to fit the background image, avoiding any distortion of the background image. This is done using the **Resolution Override** properties.

Auto-Adjusting the Render Resolution

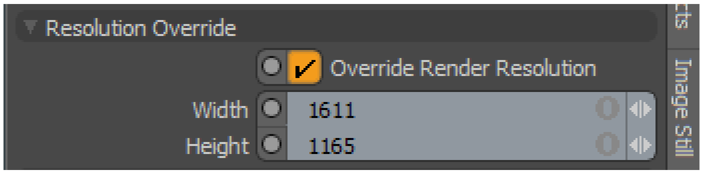
To ensure the render viewport is sized to the image, you can use the **Set Resolution from Clip** button.

1. Select the **Camera View** tab in the camera **Item Properties**.
2. In the **Background Image** properties, click **Set Resolution from Clip** to set the render resolution to be the same as the background image.



Overscan 100.0 %

This activates **Override Render Resolution** in the **Resolution Override** properties. The **Width** and **Height** values have been changed to match the image resolution.



Resolution Override

☑ Override Render Resolution

Width 1611

Height 1165

Note: **Set Resolution from Clip** also adjusts the aspect ratio of the film back for the camera, to ensure proper camera matching.

Manually Adjusting the Resolution

If you prefer not to use the source image resolution, you can resize the background image manually and use the new settings for the render resolution.

1. Check **Override Render Resolution** in the **Resolution Override** properties.
2. Edit the **Width** and **Height** values to resize the image.

Note: The **Width** and **Height** are adjusted independently unless you gang edit.

Adjusting the Color Space

You can switch the image to a different color space if required.

1. In the camera **Item Properties**, select the **Image Still** tab.
2. In the **Color** section, select the **Colorspace**.

Matching the Perspective

You match the camera to the image perspective by drawing four lines. These lines must meet the following criteria:

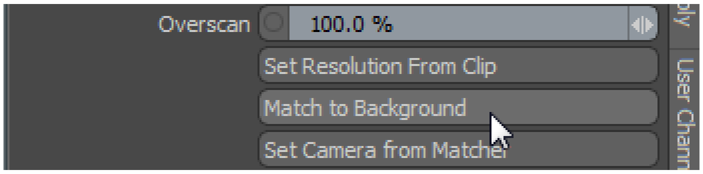
- Each line must run parallel to one of the world axes.
- Each line must be clear enough to draw over.
- Two lines should run in parallel to one axis, and two lines should run parallel to an orthogonal axis.

In the building image (see Adding a Background Image), you can see lines along the walls, at the roof, and along the floor. The back wall is orthogonal to the side wall, and they are a good match for the X and Z axes respectively. Each wall has a choice of lines that we can use for the matcher.

Note: You only need to match against two axes for the Camera Matcher to work. Choose which axes to use in the **Camera Matcher** properties.

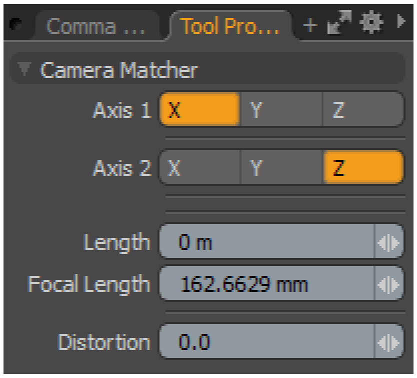
1. Ensure you're in the **Setup** layout with the camera selected, and the viewport looking through the camera.

2. Select the **Camera View** tab on the camera's **Item Properties** panel.
3. In the **Background Image** properties, click the **Match to Background** button.


Overscan
100.0 %
Set Resolution From Clip
Match to Background
Set Camera from Matcher
User Chann

4. Click on the **Tool Properties** tab on the left of the Modo window.

   This opens the **Camera Matcher** properties.

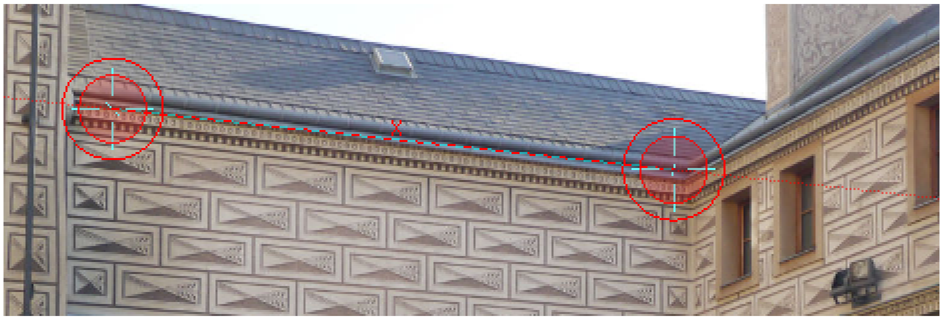

Comma ...
Tool Pro...
Camera Matcher
Axis 1
X
Y
Z
Axis 2
X
Y
Z
Length
0 m
Focal Length
162.6629 mm
Distortion
0.0

5. Choose the two axes to match against by selecting values for **Axis 1** and **Axis 2**. For example, in the image of the building above, the X and Z axes are a good choice as they both have clear matching lines.

   The first two lines you draw are matched against **Axis 1**.

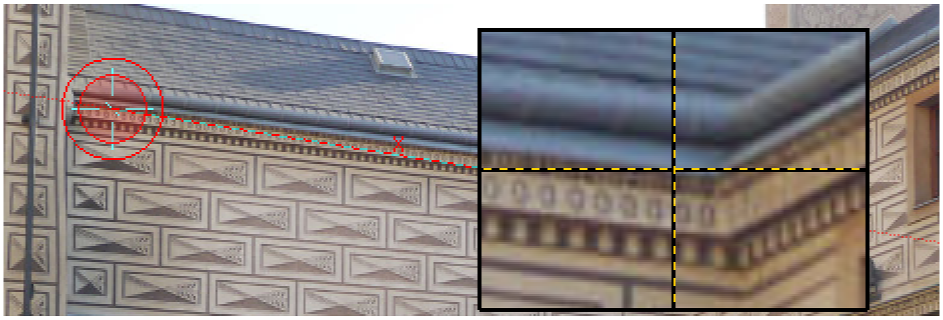
Note: The choice of axes is guided by the content of the image. If you have clear perspective lines such as building edges or road markings, use these and match them to the world axes. Features that cross the image from left to right, match the X-axis. Road markings that extend to the far horizon are a good match for the Z-axis, as would building edges that are perpendicular to those being used for the X-axis. The sides of buildings, poles, or towers that extend upwards would match the Y-axis.

6. Click and drag on the image to draw the first line. Once drawn, you can click and drag the end-points to adjust the line.



7. To make fine adjustments to the line end-point, right-click and drag the end-point.

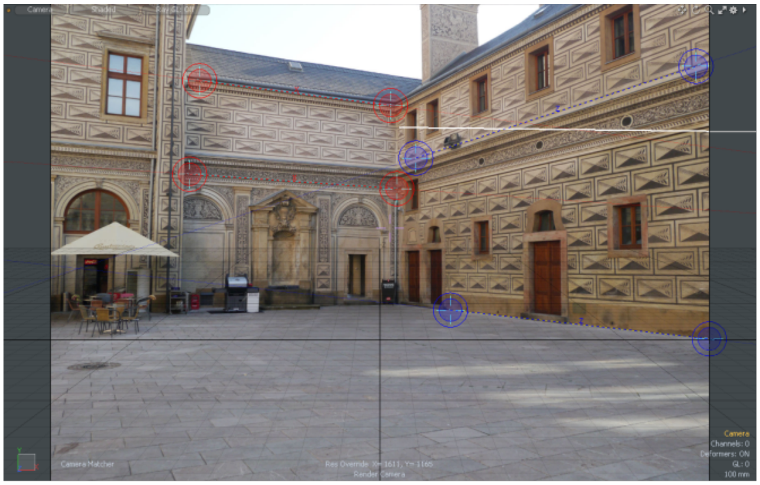
   Right-clicking on the end-point magnifies a portion of the image around the end-point.



   The cross in the center of the magnified region marks the center of the end-point.

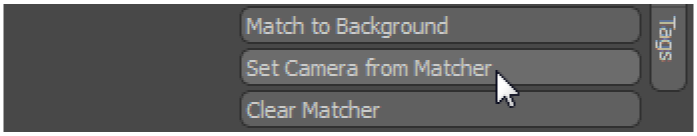
Note: The line is color coded to match the axis color.

8. Once the first line is correctly aligned, click and drag to draw the second line for **Axis 1**. You can then make adjustments as needed.
9. Click and drag to draw the first line for **Axis 2**. Notice that the color of the line differs.
10. Adjust the line to match the image.
11. Click and drag to draw the second line for **Axis 2**. Adjust as required.
12. With four lines drawn, you can make adjustments to each one until you have a good alignment.

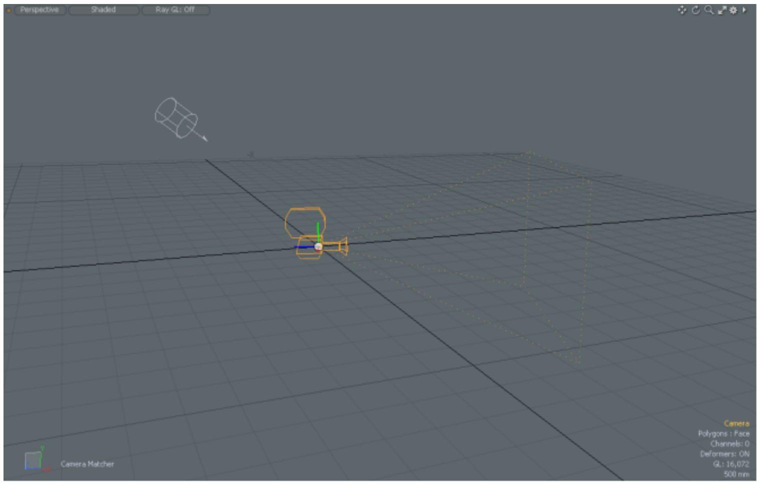


Note: At this stage, for a more accurate camera position, you can also add the length line. This is explained in Matching the Scale

13. Click **Set Camera from Matcher** to match the camera to the Modo world.



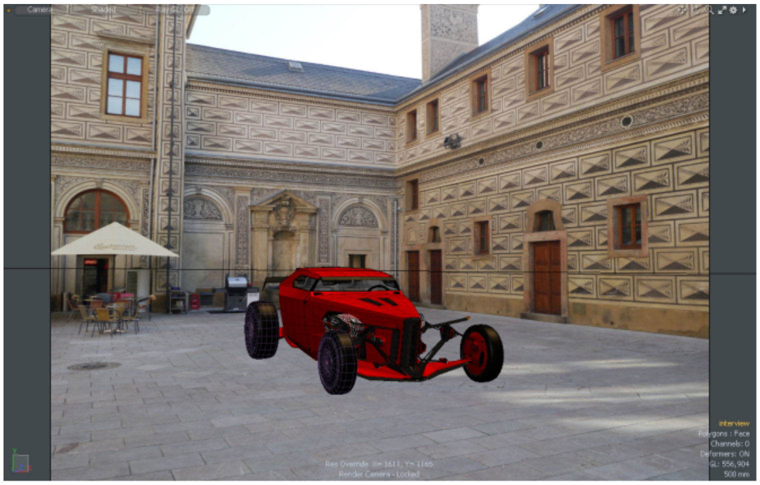
The perspective lines disappear and the camera is re-oriented at the origin. The image below shows the new camera position as seen in the **Perspective** view.



Modo objects visible from the camera now match the perspective of the background image. You can place objects as required, for more information see Viewing Objects from the Camera.

Tip: If there are objects in the world but they are not visible from the camera, switch to **Perspective** view and place the objects in front of the camera. Once visible from the camera, the objects can be manipulated in the **Camera** view.

You can also translate the camera without adjusting its orientation to face any objects in the scene. To translate the camera it must be unlocked.



Note: The camera matcher stays live, which means that any lines drawn can be edited when you click **Match to Background**, or when you click in the viewport when the tool is active.

Matching the Scale

The camera matcher lets you draw a fifth line to adjust the camera location. The new camera location enables the scale in the background image to match the scale of the 3D world. This allows you to size and place objects more accurately to match the background image.

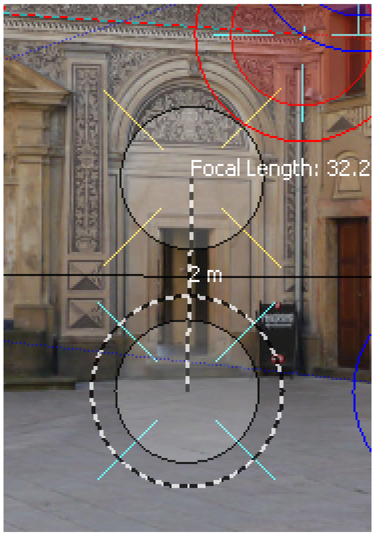
In addition to drawing the line, you need to enter the length that the line measures in the background image. For example, if there's a 2 m doorway in the image, you would draw a line along the full height of the doorway and enter 2 m for the line's length.

Note: To draw a length line, you must have drawn the four perspective lines.

1. If you've just drawn the four perspective lines and camera matcher is active, skip to step 3.
2. With the viewport on the camera, click **Match to Background** in the **Background Image** properties.
3. Click and drag to draw the length line.

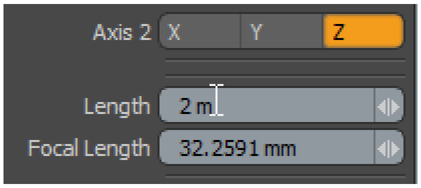
The length line is drawn dotted white and black.

Note: The end of the line with the dotted circle indicates the World Origin point. The matcher finds a position for the camera relative to this point.



Note: As a reminder, the current focal length of the camera is shown at the end of the line.

4. Adjust the line as you would with the perspective lines. Right-click+drag an end-point to magnify.
5. Open the **Tool Properties** panel on the left and enter the **Length** in meters.



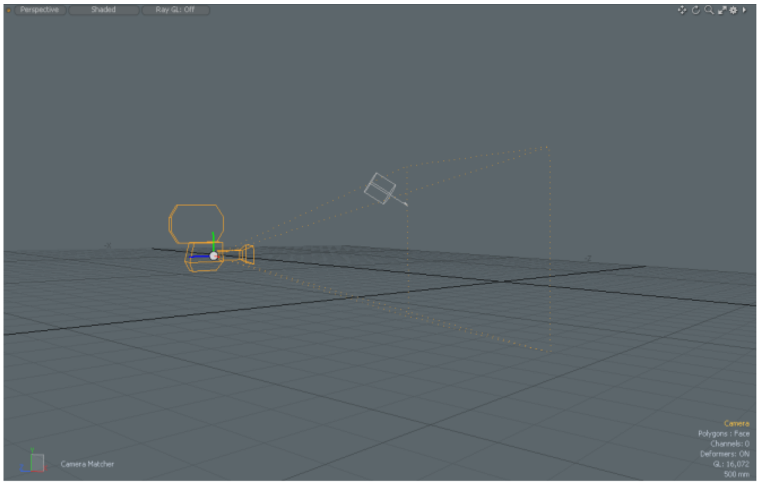
Axis 2: X Y Z

Length: 2 m

Focal Length: 32.2591 mm

6. Click **Set Camera from Matcher** to match the scale.

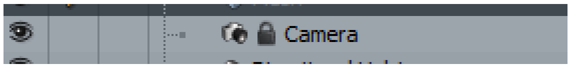
   The lines disappear and the camera is repositioned. You can swap into **Perspective** view to check the camera position.



Note: Moving the camera may mean that objects are no longer in view. If this happens, swap into **Perspective** view to move the objects back in-front of the camera.

Viewing Objects from the Camera

To place existing geometry in front of the camera, you can either move the geometry or translate the camera. Make sure that you don't rotate the camera as this will lose the perspective match. The Camera Matcher automatically locks the camera, so you need to unlock it before moving it. A locked camera is indicated by a small padlock icon as shown below.



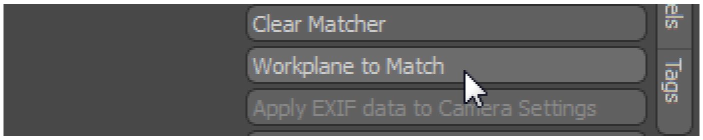
To unlock the camera, right-click on the camera item and select **Lock/Unlock**.

New geometry will be drawn on the Work Plane as normal. To draw geometry in the camera view you can set the Work Plane relative to the camera (see Matching the Work Plane.

Matching the Work Plane

To help you place objects against the background image, you can set the axes of the Work Plane to match the axes that are drawn when camera matching. This adjusts the Work Plane orientation relative to the current camera. If you have added a length line, the Work Plane is also moved to the location of the length line.

Match the Work Plane by clicking **Set Camera from Matcher** in the **Background Image** properties of the camera's **Item Properties**.

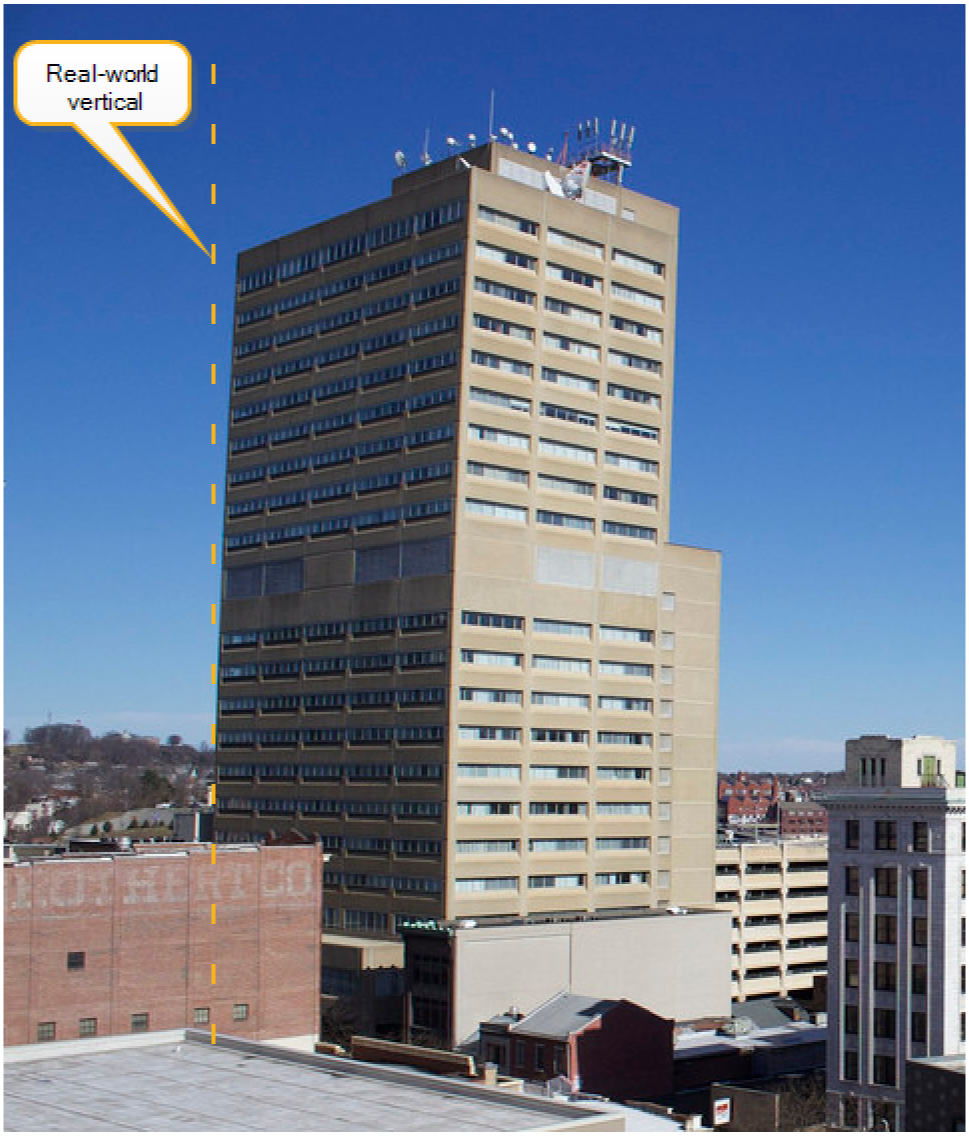

Clear Matcher
Workplane to Match
Apply EXIF data to Camera Settings
els
Tags

Note: Without a length line, the Work Plane is positioned at the World Origin.

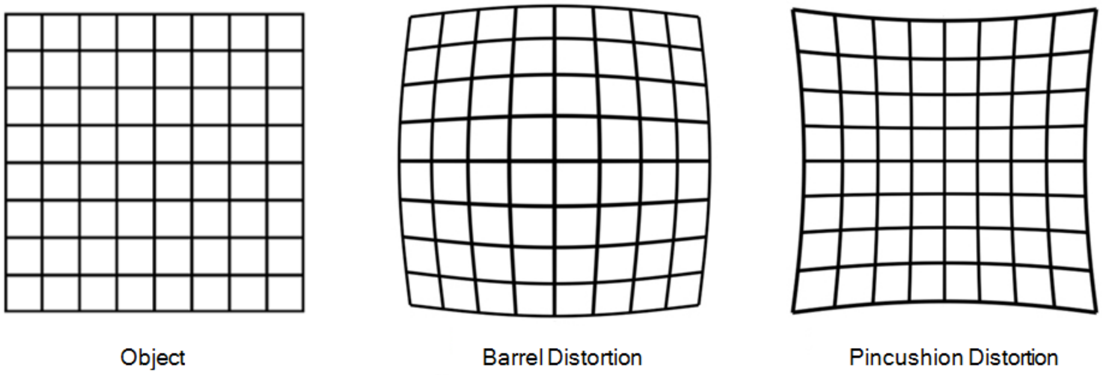
Lens Distortion

Modo's Camera Matcher includes a Lens Distortion Estimator to compensate for real-world lens distortion. Lens distortion is a common phenomena in optics, where a straight line in the real-world may appear slightly curved in a photo due to lens design. Lens distortion is most apparent in zoomed or wide-angle shots, and is easily seen at extreme focal lengths such as those in fisheye lenses.

The image below shows subtle lens distortion along the vertical edge of a building. Notice how the edge of the building curves away from the vertical axis.



Two common types of lens distortion seen in photography are barrel distortion and pincushion distortion. Barrel distortion typically occurs at wide angles, and the lines curve outwards from the center of the image. Pincushion distortion occurs in zoomed images, and the lines curve inwards. The image below illustrates the two effects.



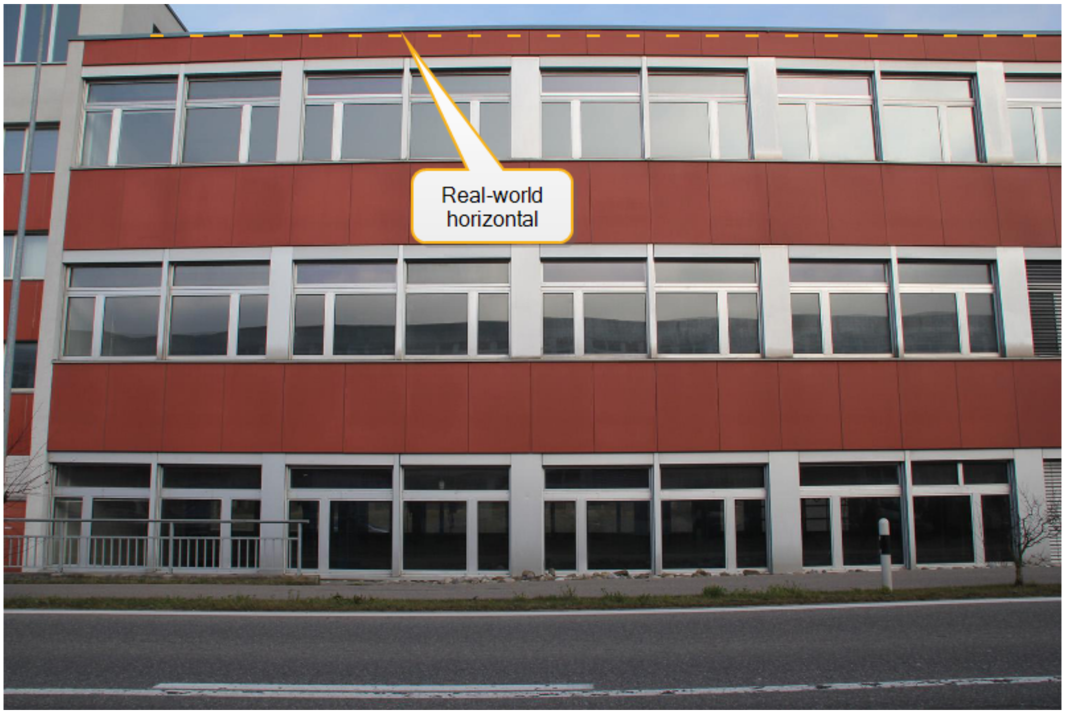
You can see in the image of the building above, the distorted edge of the building is convex, and therefore due to barrel distortion.

Adjusting for Lens Distortion

Modo's Lens Distortion Estimator ensures that the edges of objects in the 3D world are curved to match the edges in the image, where the curvature is due to lens distortion. The Estimator is able to solve for subtle lens distortion such as the type shown in the image above. It is less likely to work with extreme lens distortion as seen in fisheye lenses.

Note: You only see the effects of the Lens Distortion Estimator in the rendered output.

You can see lens distortion in the image below. Notice how the horizontal building lines are subject to barrel distortion. This is particularly visible across the roof-edge of the building.



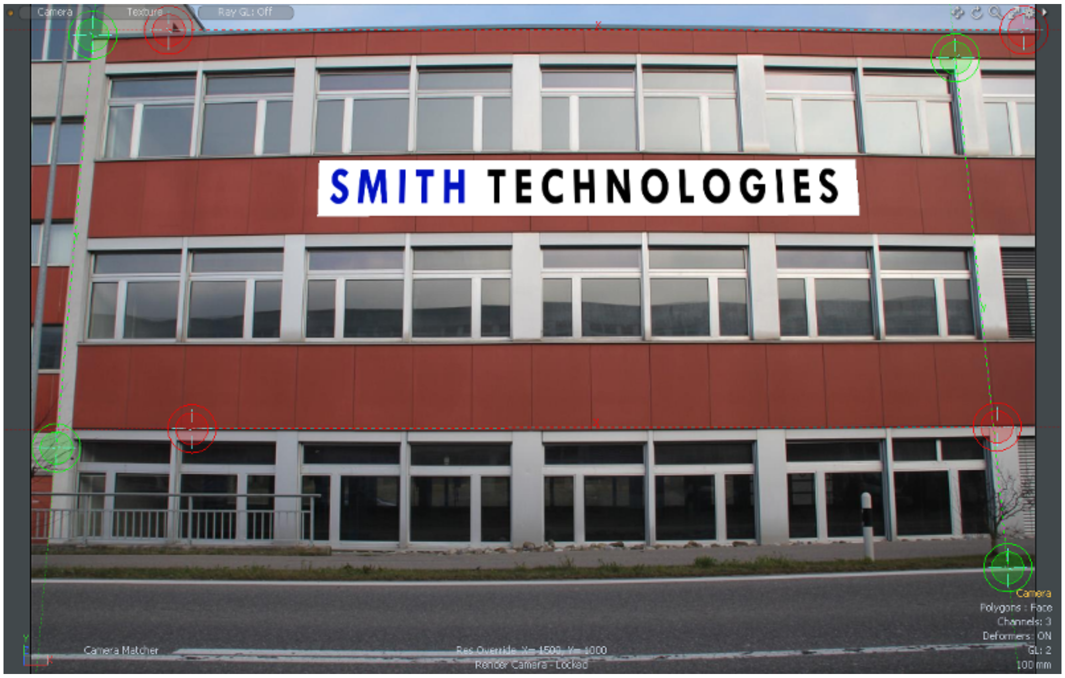
Lens distortion is applied to rendered objects by adjusting the **Distortion** parameter on the **Tool Properties** panel. **Distortion** can take positive or negative values:

- Positive values are used to adjust for barrel distortion.
- Negative values compensate for pincushion distortion.

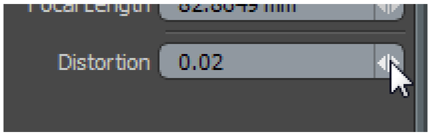
Note: Add lens distortion after you have drawn the perspective lines.

1. Position objects against the background as required. The image below shows a textured polygon that has been added to a building facade using Modo.

   Observe how the building in the image is subject to lens distortion but the Modo geometry is not.


Camera
Texture
Ray GL: Off
SMITH TECHNOLOGIES
Camera Matcher
Res Override X= 1500, Y= 1000
Render Camera - Locked
Camera
Polygons : Face
Channels: 3
Deformers: ON
GL: 2
100 mm

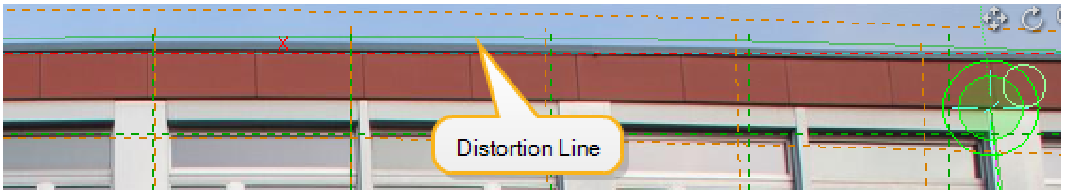
2. If the Camera Matcher is active, skip to step 4.
3. In **Setup** mode, with the camera selected and the viewport looking from the camera, click **Match to Background**.
4. In the **Tool Properties** panel, increase or decrease the **Distortion** value for barrel or pincushion distortion respectively.



Distortion 0.02

A non-zero **Distortion** value reveals the lens distortion reference grids in the camera viewport. The green grid shows undistorted lines as a base reference. The orange grid shows the degree of lens distortion according to the **Distortion** value.

5. As you adjust the value, look closely at the perspective lines. You can see that a green line is drawn between the end-points to show the effect of barrel distortion. The image below shows a green distortion line originating from the perspective line that was drawn along the roof-edge of the building.


Distortion Line

6. Match the green line to the line in the image and click **Set Camera from Matcher** to apply the distortion.

Tip: You can also use the orange grid to estimate the level of distortion required.

7. When you're happy with the **Distortion** value, click **Set Camera to Matcher**.
8. Press **F9** to render the output and check the results.

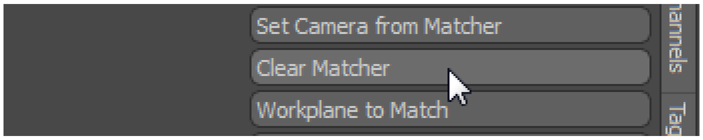
Note: To render from the camera's point-of-view, the camera must be the **Render Camera**. This is set in the Render Item's properties.

SMITH TECHNOLOGIES

9. If you need to make further adjustments, return to step 2.

Resetting the Matcher

If you want to reset the Camera Matcher and start again, click **Clear Matcher** in the **Background Image** properties of the camera's **Item Properties**.


Set Camera from Matcher
Clear Matcher
Workplane to Match
hannels
Tag

Camera Images

The **Camera View** properties include a number of features for using images with a camera. These are:

- Background image: This fills the camera frame and is rendered behind any geometry that is visible from the camera.
- Foreground Image: This fills the frame and is rendered in front of all geometry. You can also use this as a render mask.
- Overscan: Resize the foreground and background image without changing the aspect ratio.
- Camera Image Plane: You can place any number of image planes between the foreground and background positions at an adjustable distance from the camera viewpoint. An image can then be added to each plane as required.
- Apply EXIF data to Camera Settings: If an image file contains the necessary EXIF data (image metadata), you can set the camera **Focal Length** to equal the image focal length.
- Physical Sun from EXIF data: If EXIF data is available, you can auto-generate a Physical Sun directional light in the scene to match the sun in the image.
- Resolution Override: The render resolution can be defined in the camera.

For more details on the **Camera View** Property options, see Camera Items.

Foreground and Background Images

In addition to image planes, Modo lets you add a foreground and background image to a camera. A foreground image appears in front of geometry and is only visible when you render.

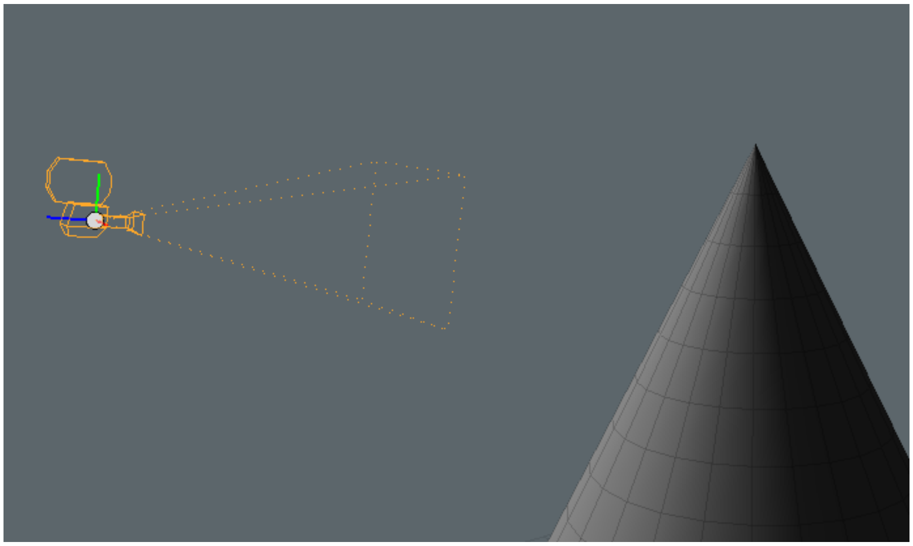
Note: The foreground image is hidden when editing so as not to obscure other items.

Background images are drawn behind other items and can be seen from the camera view when editing. To add a background image, see Adding a Background Image. Loading a background image enables Camera Matching and the use of EXIF data (if present).

Adding a Foreground Image

You can add a foreground image to any camera using the **Foreground Image** selector in the **Foreground Image** section of the **Camera View** properties.

1. Select the camera. In this example we are using a camera that is facing some geometry to illustrate the effect of the foreground image.

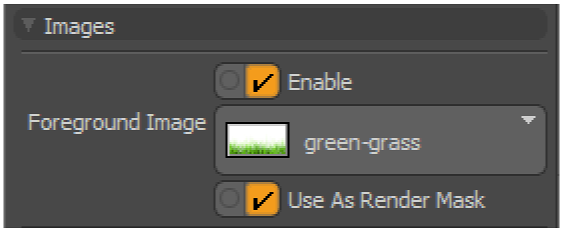


2. Click the **Camera View** tab of the camera item's **Properties**.
3. In the **Foreground Image** section, click on the **Foreground Image** selector.
4. Click the **Add Clip** dropdown and select **(load image)**.

   This opens the image file browser.
5. Browse to and select the background image. Click **OK**.

   The image is loaded into **Foreground Image**.

   In this example, we are using a transparent **.png** image with grass along the bottom. By default, the **Foreground Image** is enabled. If the **Enable** option is toggled off, the image is not displayed in your rendering or removed from your scene.



Images

Foreground Image: ☑ Enable; green-grass; ☑ Use As Render Mask

6. To see the image in place, press **F9** to open the **Render Display**.

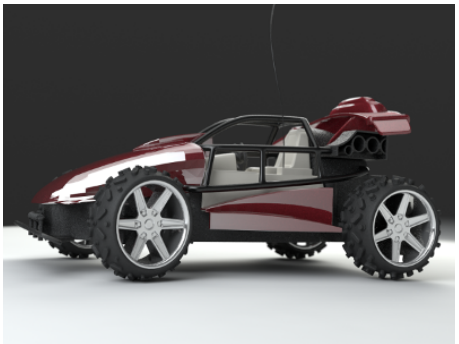
You can see the foreground image with the world geometry rendered behind it.

Note: To render from the camera, it must be set as the Render Item: Frame.

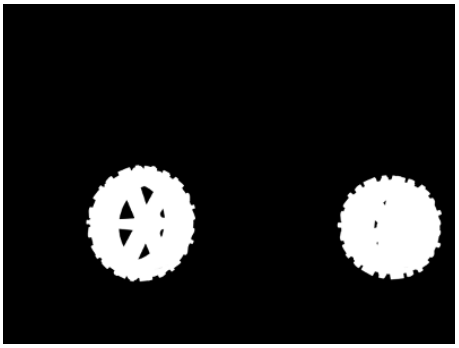
Using a Foreground Render Mask

Check **Use As Render Mask** to use the **Foreground Image** to mask the render output. This is useful if you want to quickly re-render part of the image, whilst hiding parts that don't need to be updated. Modo uses the luminosity of each pixel in the image to determine the pixel's opacity. A white pixel is rendered as transparent, black is fully opaque, and all shades in-between are semi-transparent.

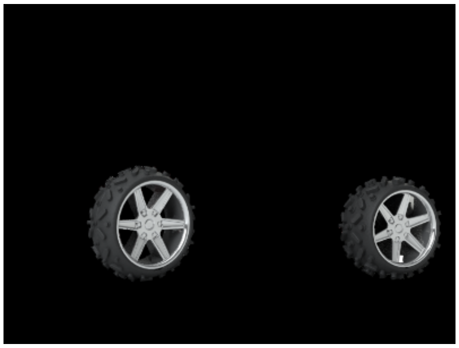
The images below demonstrate the use of a mask.



Original model, fully rendered.



Camera foreground image used as



Masked render output.

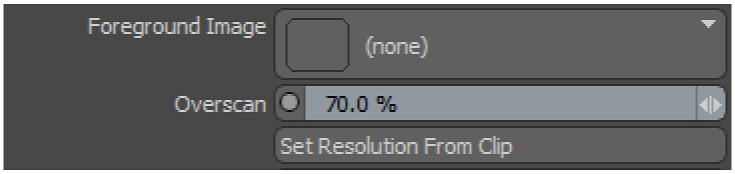
mask.

You can paint a mask manually or use an external application to create one.

Note: In the example above, the mask was created by hiding unwanted geometry and rendering the wheels to an Alpha layer.

Overscan

If you want to scale an image to fill the camera's frame without changing the image aspect ratio, you can use **Overscan**. This works with both foreground and background images. Edit the **Overscan** value as a percentage of the original image size.

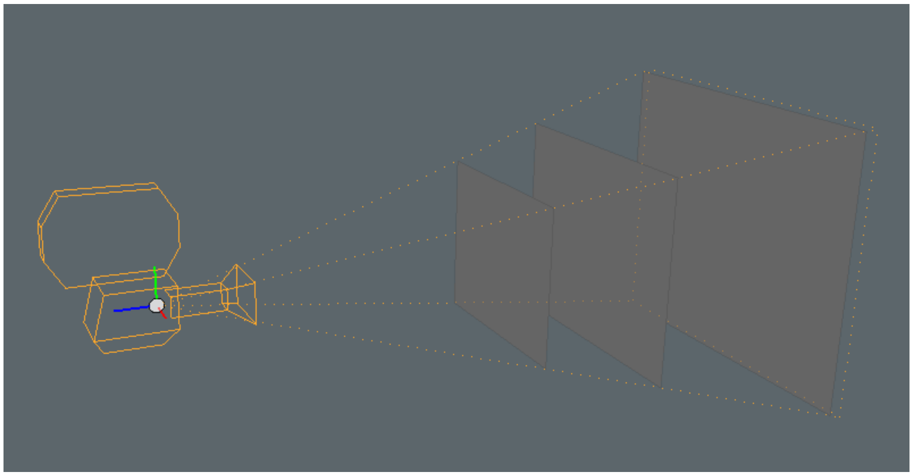


Foreground Image (none)

Overscan 70.0 %

Image Planes

Modo can generate a rectangular polygon with size, position, and rotation to match the image plane, projected from the camera at a given depth. This is useful if you want to layer images in front of the camera and insert geometry between the layers. The image below shows an example of three image planes that have been added to a camera. Notice how the polygons are scaled to fit the camera frustum.



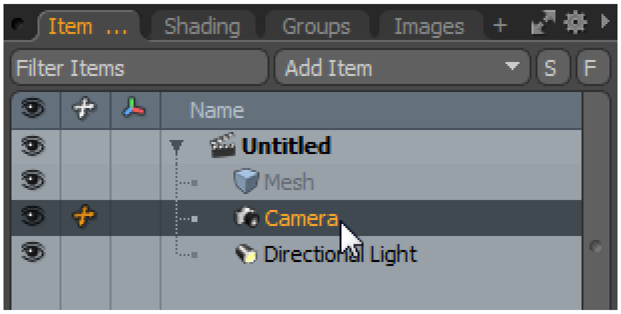
Note: To layer images, you must make use of transparency, similar to the foreground image shown above. An image without transparency fills the frame and hides items behind it.

Image Planes are added using the **Add Camera Image Plane** button on the camera's **Camera View** properties.

Adding an Image Plane

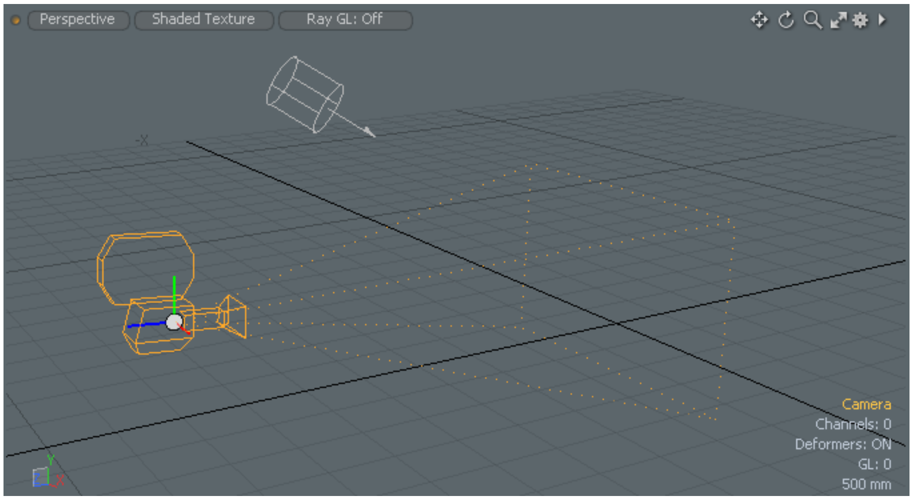
The **Add Camera Image Plane** button adds a quad (four-sided) polygon at an adjustable depth from the camera viewpoint.

1. In **Setup** mode, select the camera to use in the **Item List**.

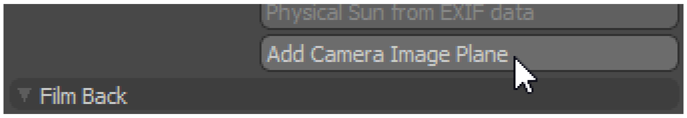

Item ...
Shading
Groups
Images
+
Filter Items
Add Item
S
F
Name
Untitled
Mesh
Camera
Directional Light

Tip: You can also select a camera by clicking on it, when in the **Setup** layout in **Items** mode.

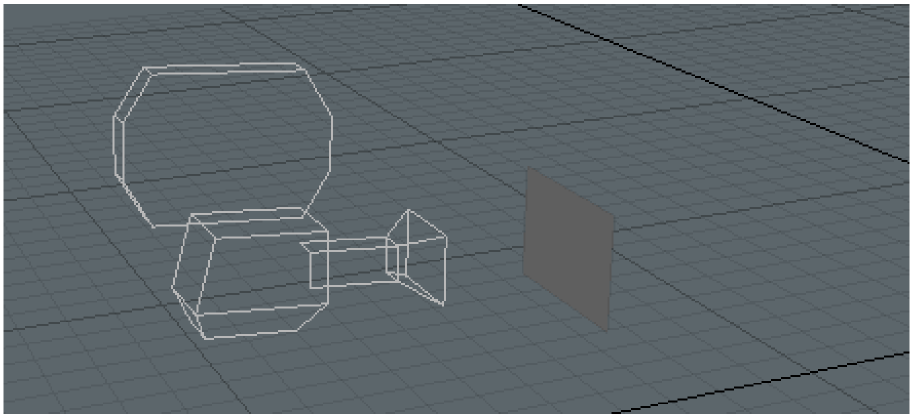
2. Ensure the 3D viewport is in **Perspective** mode with a clear view of the camera.


Perspective
Shaded Texture
Ray GL: Off
Camera
Channels: 0
Deformers: ON
GL: 0
500 mm

3. Click the **Camera View** tab in the camera item's **Properties**.
4. In the **Background Image** section, click **Add Camera Image Plane**.

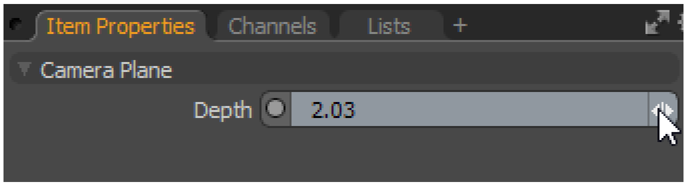


You can see the new image plane in front of the camera.



A corresponding image plane item called **ImagePlane**, is added to the **Item List**.

5. Adjust the **Depth** value to move the image plane to the required distance from the camera.



Depth 2.03

The quad will automatically scale to fit the camera frustum.

6. Add more image planes as required by repeating steps 2-5.
7. You can add a material with an image to each image plane as needed.

> **Note:** To layer images and geometry, you must use images with transparent regions that reveal background detail.

EXIF Data

An image file produced by most modern digital cameras now includes metadata to store the camera settings at the time the image was taken. The additional data is known as interchange information. A common standard for encoding images with interchange information is the Exchangeable Image File (EXIF) format. EXIF data includes settings such as ISO speed, shutter speed, aperture, white balance, camera model and make, date and time, lens type, and focal length.

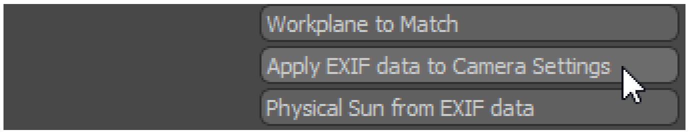
Note: At the time of writing, EXIF is limited to the **.jpg** and **.tif** image formats.

In Modo, if you are using a background image that contains EXIF data, you can use properties from the image to set the focal length or generate a Physical Sun directional light in the scene.

Note: For non-EXIF images, the options are grayed-out.

Using EXIF Data to Set Camera Properties

You can set the focal length of the Modo camera to match the focal length of the image. Open the **Camera View** tab in the **Item Properties**, and click **Apply EXIF Data to Camera Settings** in the **Background Image** section.

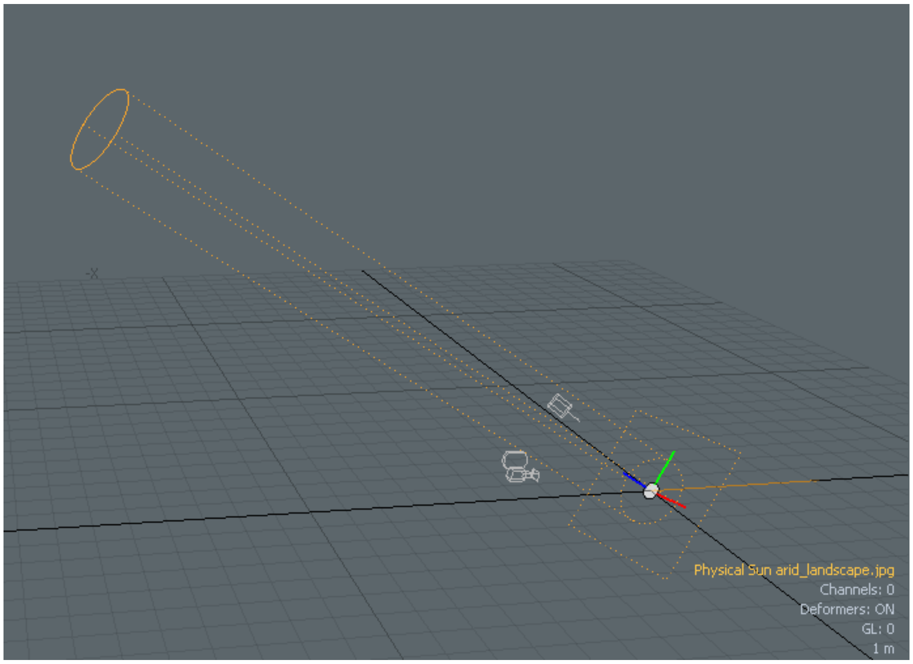

Workplane to Match
Apply EXIF data to Camera Settings
Physical Sun from EXIF data

Using EXIF Data to Create a Physical Sun

Matching the lighting in a background image can be challenging. Modo can help you by auto-generating a **Physical Sun** directional light to mimic the sunlight in the image. This is calculated using the time, date, and location information that is stored in the EXIF file.

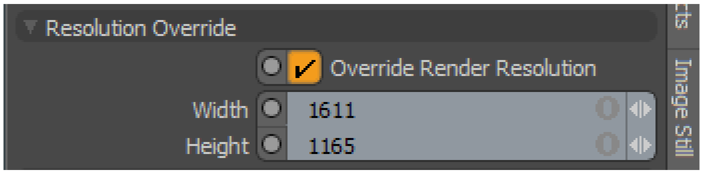
1. Open the **Camera View** tab in the **Item Properties**.
2. In the **Background Image** section, click **Physical Sun from EXIF** data.

   A **Physical Sun** directional light is added to the scene.



Resolution Override

The **Camera View** properties include a **Resolution Override** section to customize the render resolution when the camera is used as the render camera. Check **Override Render Resolution** and adjust the **Width** and **Height** values as required.


Resolution Override

☑ Override Render Resolution

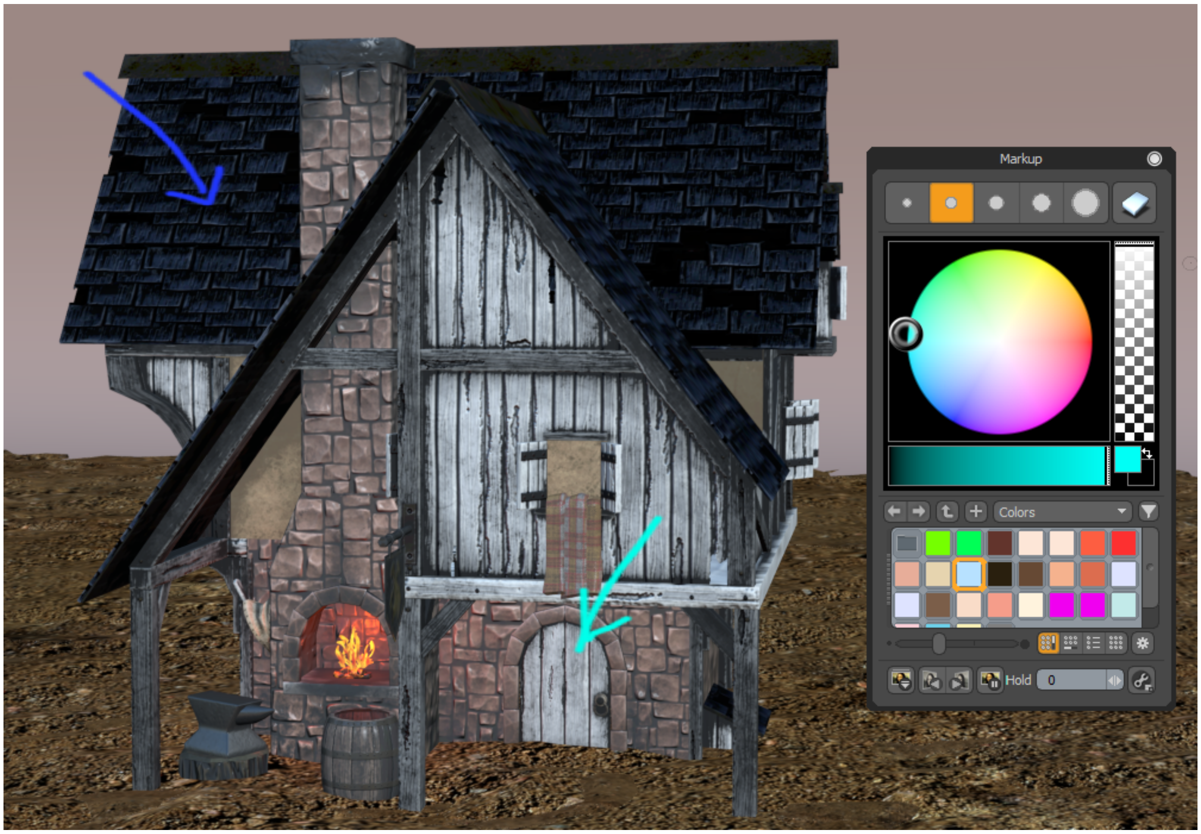
Width 1611

Height 1165

The **Set Resolution from Clip** button turns on **Resolution Override** and uses the dimensions of a selected image. If you have loaded a **Background Image**, its resolution is used. This is useful when performing Camera Matching. Without a **Background Image**, the resolution of the selected image in the Images/Clips List is used.

Markup

The Markup tool allows you to draw directly onto selected camera views so you can add annotations, notes, and comments to your animations or stills. Assign the Markup image to the current camera item by selecting the camera and clicking **Markup** in the **Images** properties. Once assigned, you can use a selection of Markup tools and controls to draw on your camera view in the viewport.



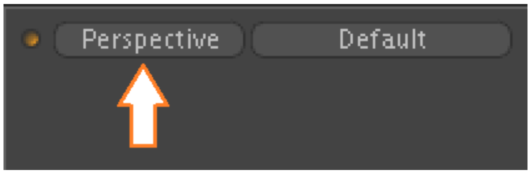
The Markup tool being used to annotate areas of the scene that require lighting and animation.

Assigning Markup to a Camera

1. Select or create a camera in the Item List.

Tip: See Camera Items for more information on adding cameras to your scene.

2. Use the camera view option in the upper left of the 3D viewport to set the selected camera to the active camera view.



3. In the camera **Properties**, under **Images** click **Markup**.

   An **Assign Markup** dialog box appears, allowing you to select a Resolution for your Markup image.

4. Select the **Resolution** you require for your image from the dropdown, and click **OK**.

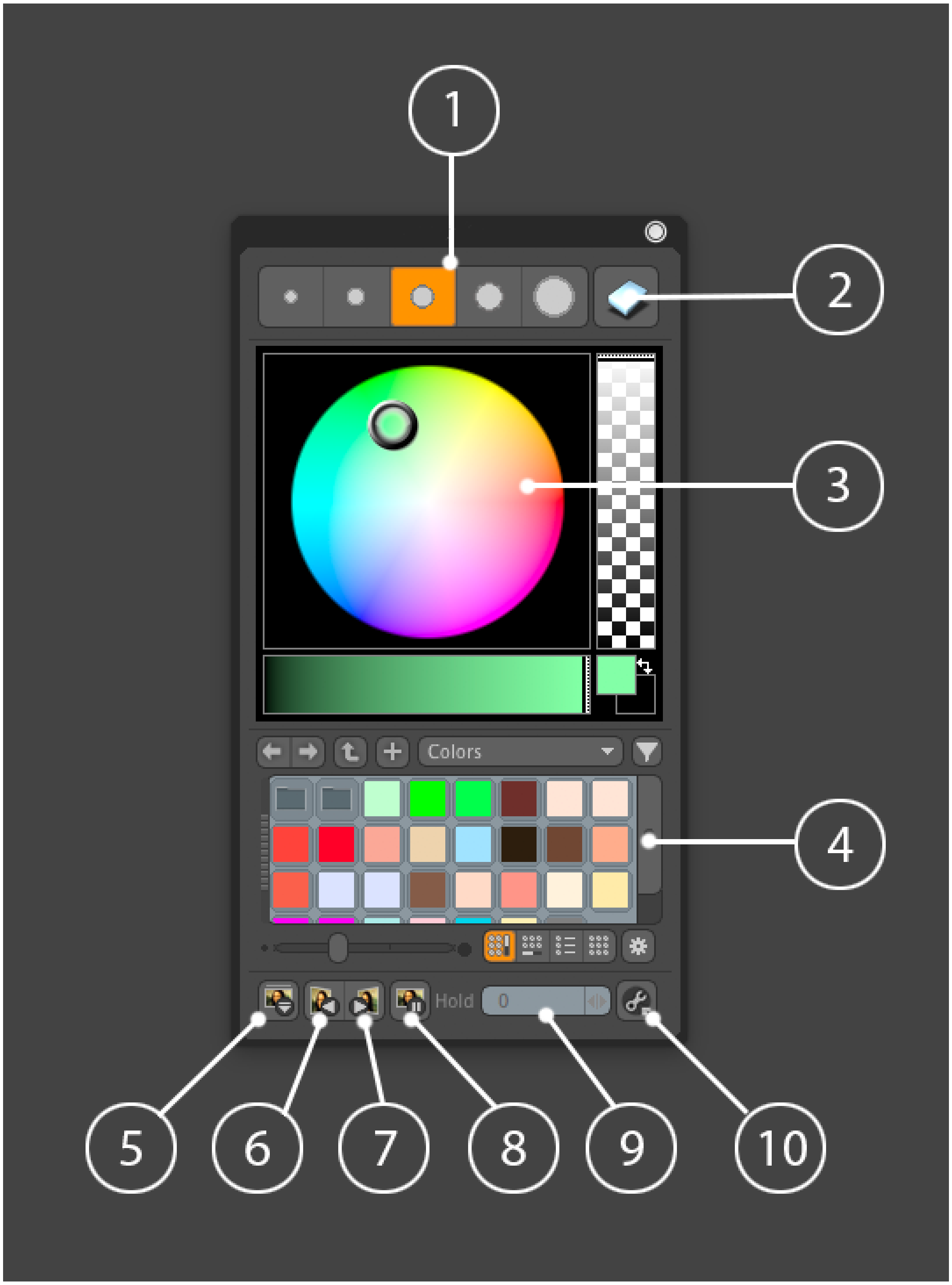
   A save dialog appears, allowing you to select a location, format, and name for your Markup image.

5. Provide a name, save location, and image format for your Markup image, and click **Save**.

   The Markup tool popover appears, an Image Sequence is created, and the **Image Sequence** tab becomes active. A Canvas item is also added to the Item List. Now you can draw in the viewport using the Markup tools.

Note: To remove Markup from a camera, delete the associated **Canvas** item found in the Item List. For example, if you want to remove Markup from "Camera (05)", deleted "Canvas : Camera (05)" from the Item List.

Markup Tool Popover

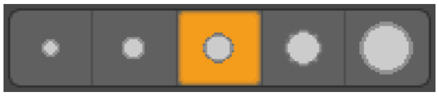

1
2
3
4
5
6
7
8
9
10
Colors
Hold
0

The Markup tool popover.

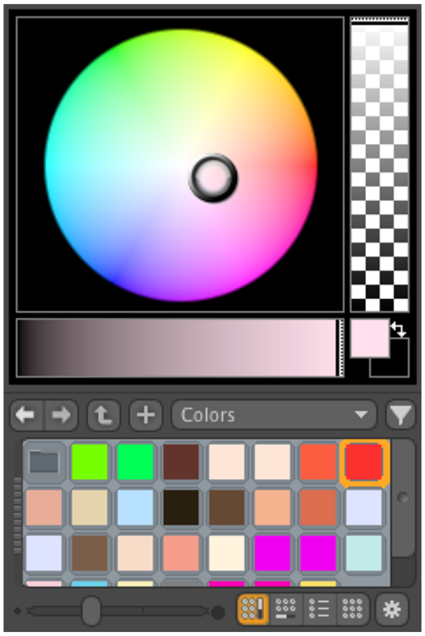
1. Preset Brush Sizes	A selection of brushes, increasing in size from left to right.
2. Eraser	Erases any drawing you have made in the viewport. **Tip:** Holding **Shift** with this tool active changes this tool to Erase Image, that allows you to clear an entire drawing in the frame.
3. Color Picker	Allows you to select a color to draw with when a brush is active.
4. Color Presets	A selection of preset colors to draw with.
5. Save	Saves any changes to the Markup image(s) to disk. If there are multiple Markup images across multiple frames, an image is saved to disk per frame affixed with the frame number.
6. Previous Image	Navigates to the previous image on the previous frame.
7. Next Image	Navigates to the next image on the next frame.
8. Toggle Frame Hold	By default, images are only displayed for the frame they were created on. Turn **Frame Hold** on to display the previous image for all frames until a new image is drawn.
9. Hold	If Toggle **Frame Hold** is off, use **Hold** to specify a number of frames to hold an image for.
10. Paint Tool Properties	Opens Modo's full paint tool properties. See Paintbrush.

Drawing in the Viewport with Markup

1. Select a preset brush size from the Markup tools popover.



2. Select a color to draw with from the Color Picker or Color Presets.



3. Hover over the 3D viewport, and click and drag to draw.

Tip: If at any point you lose your Markup tool popover, open the Camera **Properties** and click **Markup Tools** to reopen the tool popover.

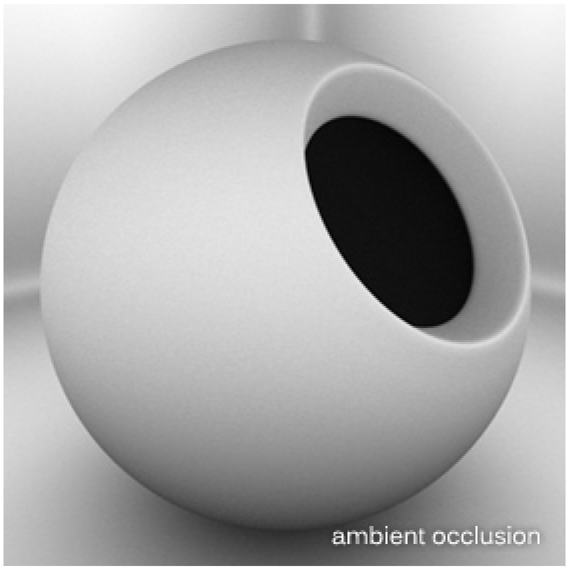
Tip: You can open the Timeline to select specific keyframes to draw on by clicking the **Time** button at the bottom of the 3D viewport.

4. Activate the **Eraser** to erase any drawings you have made in the viewport.
5. Click **Save** to store your drawing to disk.

Render Outputs

Render outputs are simply what their name describes: they are the output of a **Render** operation, the evaluated results of all visible layers of the **Shader Tree**. Invoking the **Render** command or pressing **F9** when no render outputs are present and visible produces an error message. The most common output is **Final Color**.

The **Effect** of the render output layer can be changed to a variety of options for different purposes. Samples demonstrating each render output, including detailed descriptions of each are available in the specific render outputs.



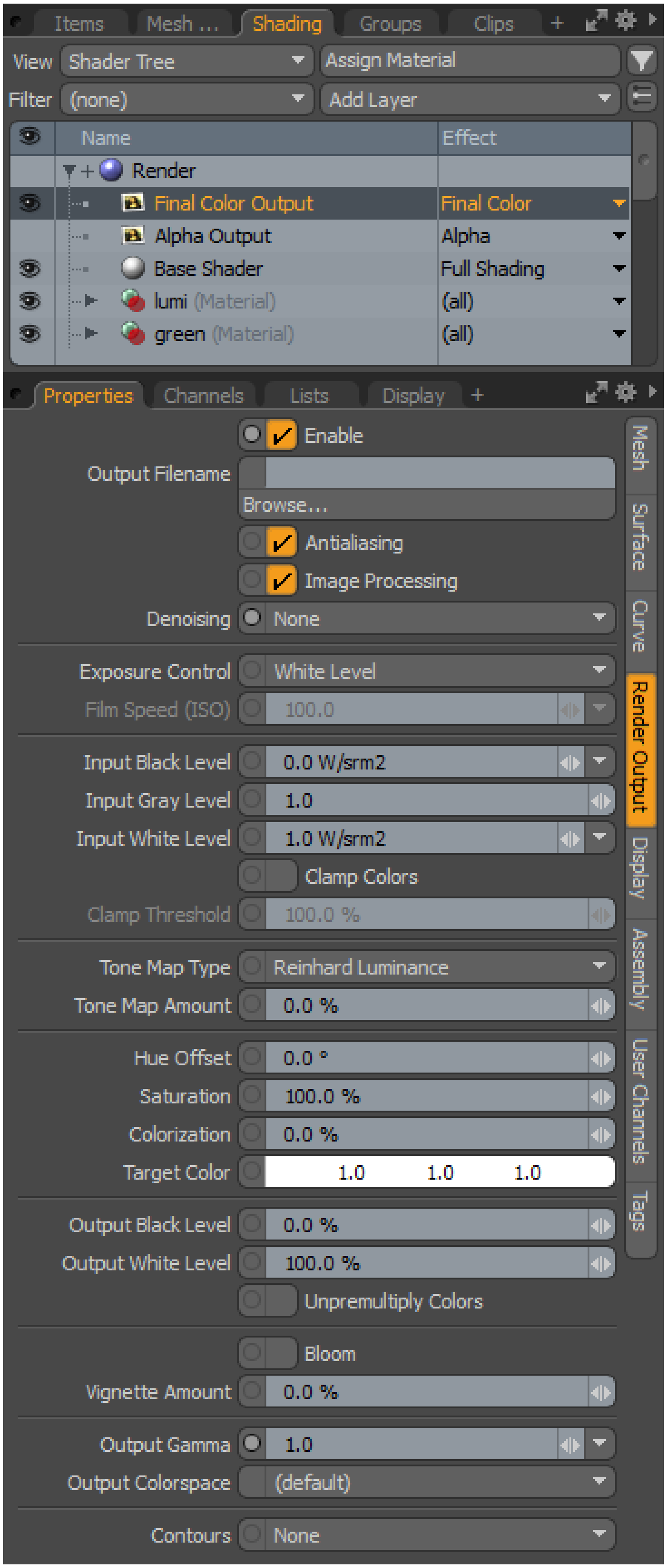
Modo offers many different render output types useful in a variety of situations, ranging from purely diagnostic to having specific compositing utility in an external application. A default scene automatically includes a **Final Color** and **Alpha** output, which works fine for most situations. You can add additional layers in the **Shader Tree** by clicking the **Add Layer** option and selecting **Render Outputs** > **Render Output** from the dropdown. You can then right-click in the **Effect** column of the resulting **Shader Tree** layer for a context menu that allows you to select alternate output types.

Additionally, shortcuts have been added that automatically specify an **Effect** type under the **Render Outputs** menu section by simply selecting the desired **Output** type. All the various output types are listed under the **Add Layer** button (**Add Layer** > **Render Outputs**), working as shortcuts to add the render output layer and setting its **Effect** at the same time.

Selecting any of the render output options adds an additional layer to the top of the tree. From this position, the output sees the results of the entire tree. Clicking and dragging a layer, and placing it within a material **Group**, limits the render output evaluation to just those layers. Individually, render outputs can be enabled either from the layer's **Enable** checkbox within the **Properties** panel, or by clicking the 👁 (eye) icon in the visibility column of the **Shader Tree**. All enabled outputs are rendered simultaneously each time the **Render** or **Render Animation** commands are invoked. Once rendered, the individual outputs appear in the Render Display window and are accessible from the layer menu therein.

Render Output Properties

Selecting any of the render outputs within the **Shader Tree** makes their editable attributes available in the **Properties** panel. Selected layers are indicated by the dark bar highlighting the layer name.


Items
Mesh ...
Shading
Groups
Clips
Properties
Channels
Lists
Display
Mesh
Surface
Curve
Render Output
Display
Assembly
User Channels
Tags

View: Shader Tree

Assign Material

Filter: (none)

Add Layer

Name	Effect
Render	
Final Color Output	Final Color
Alpha Output	Alpha
Base Shader	Full Shading
lumi (Material)	(all)
green (Material)	(all)

☑ Enable

Output Filename:

Browse...

☑ Antialiasing

☑ Image Processing

Denoising: None

Exposure Control: White Level

Film Speed (ISO): 100.0

Input Black Level: 0.0 W/srm2

Input Gray Level: 1.0

Input White Level: 1.0 W/srm2

☐ Clamp Colors

Clamp Threshold: 100.0 %

Tone Map Type: Reinhard Luminance

Tone Map Amount: 0.0 %

Hue Offset: 0.0 °

Saturation: 100.0 %

Colorization: 0.0 %

Target Color: 1.0 1.0 1.0

Output Black Level: 0.0 %

Output White Level: 100.0 %

☐ Unpremultiply Colors

☐ Bloom

Vignette Amount: 0.0 %

Output Gamma: 1.0

Output Colorspace: (default)

Contours: None

All render outputs share the same set of following attributes, with **Ambient Occlusion**, **Depth**, **Final Color**, and **Motion Vector** having additional options covered further below.

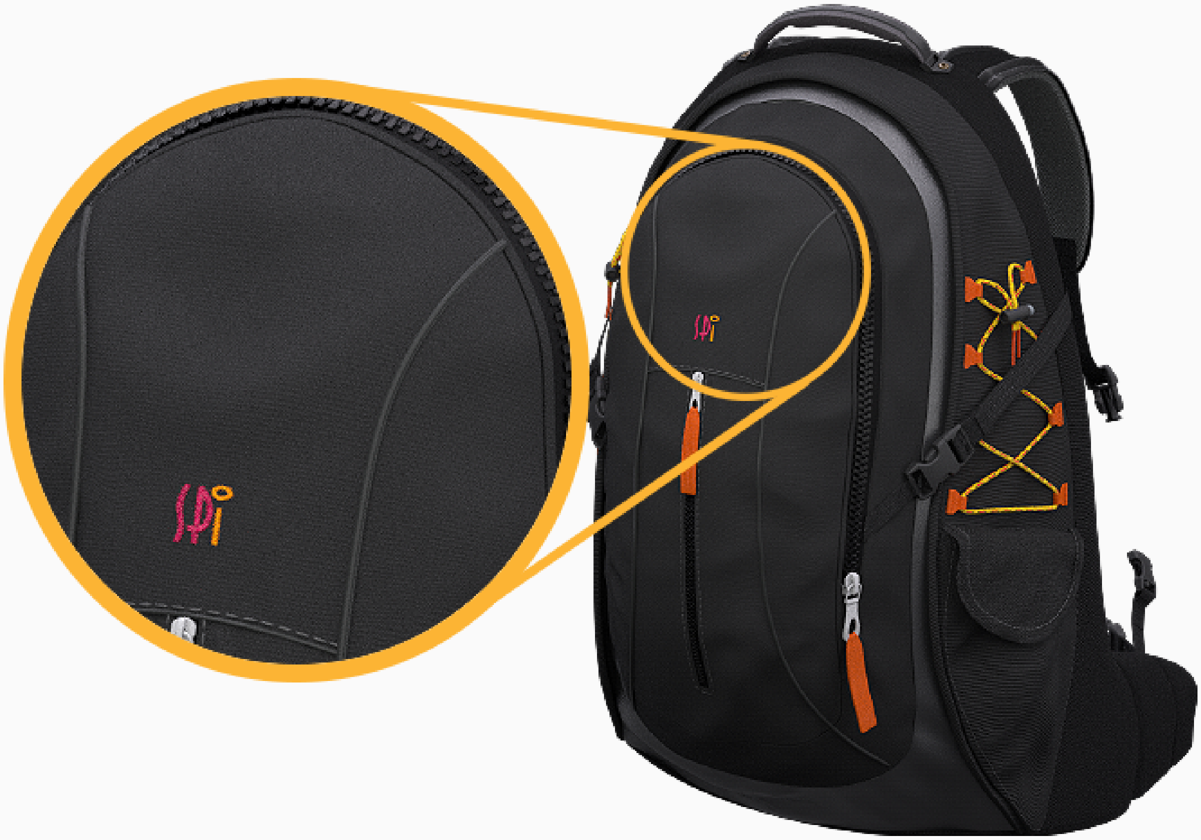
Option	Description
Enable	Allows you to temporarily enable or disable a layer without losing stored values. Disabled layers are saved with the scene file.
Output Filename	Allows you to specify a filename and location saving a rendered animation. Click the **Browse** button to open the **File Save** dialog. Specify a name and location for the rendered output. Images are saved the next time you invoke the **Render Animation** command from the menu bar. Additional automatic output naming options can be specified in the **Frame** settings of the Render item as well. **Note:** Rendering scenes using the **Render** command or **F9** does not save the image automatically to disc, but merely displays the rendered result in the **Render Display** window. For saving the image to disc, use the **Save As** option of the **Render Display** window.
Antialiasing	Used to enable or disable the rendering Antialiasing functionality on a per-output basis, allowing you to render all outputs simultaneously when only certain passes do not require antialiasing (such as **Depth**). When enabled, antialiasing is calculated for the pass; when disabled, antialiasing is not calculated.
Image Processing	Enable this option to control the post-processing applied to the rendered image. When enabled, post processing functions apply to the rendered output; when disabled, post processing settings are disregarded. Post processing can be a powerful way to adjust the final rendered image. This option makes it easy for you to quickly see to what degree the image processing is contributing to the final render without disabling all the settings individually.
Denoising	Allows you to apply post-process denoising to the render output. You can choose from either NVIDIA or AMD denoising options. NVIDIA OptiX is available for Windows and Linux systems running NVIDIA GPUs, and the AMD denoiser options are supported on all platforms. The AMD denoisers are OpenCL-based, so they run on GPUs from AMD, NVIDIA, and Intel.

Note: If a denoiser is not available on your system, Modo displays a warning icon ⚠ next to it. You can still select it, allowing you to set up the scene for rendering on a different system, but will produce a warning dialog when the render is started on an unsupported system.

Tip: Denoising is useful when you are rendering with the mPath renderer. To learn more, see Denoising mPath Renders.

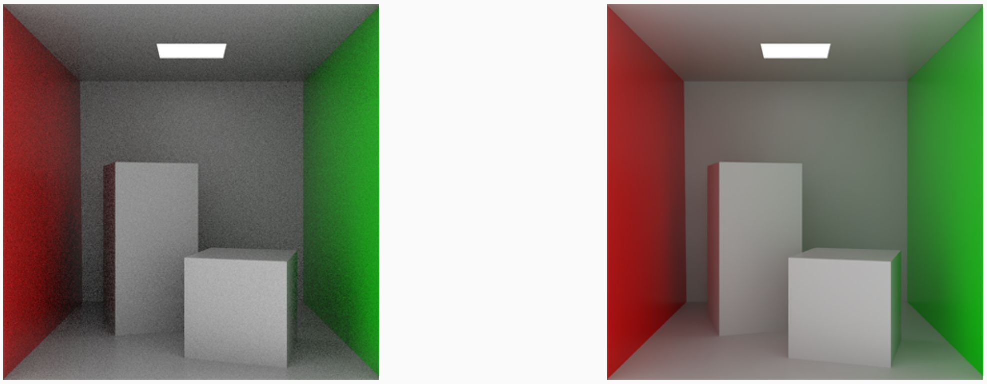
You have the following options:

- **None** - The render operation uses no denoiser on the rendered image.
- **Intel Open Image Denoise** - A high quality denoiser that uses artificial intelligence to improve image quality of the rendered image on the host CPU.



No Denoising.

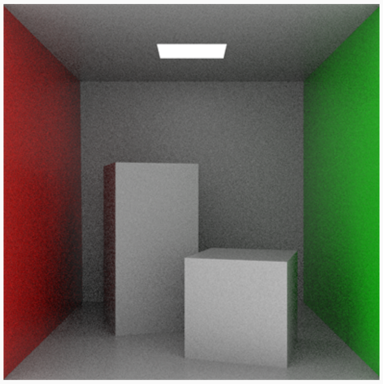
Intel Open Image Denoise with **Denoise Per Iteration** enabled.



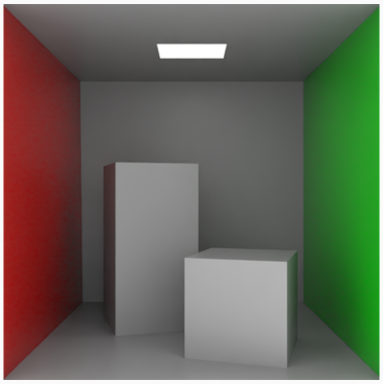
No Denoising.

Intel Open Image Denoising.

- **NVIDIA OptiX** - NVIDIA OptiX denoising works best with Monte Carlo path tracing. The following table displays the render output of a scene using the NVIDIA OptiX denoiser.



No Denoising.



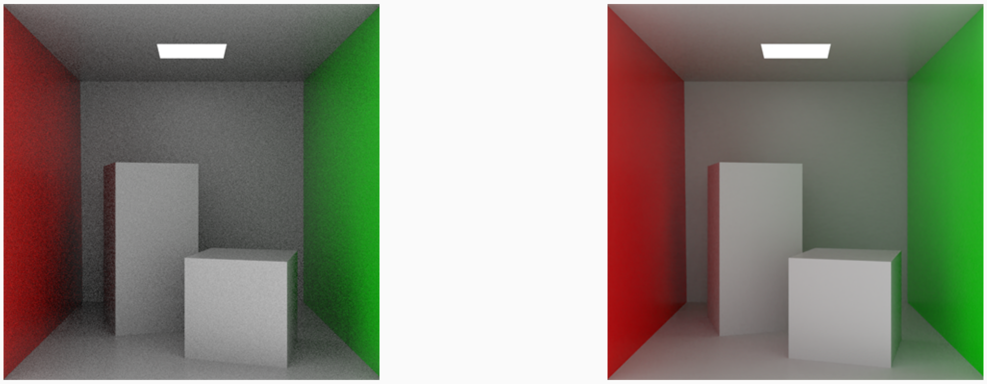
NVIDIA OptiX Denoising.

- **AMD AI** - Uses artificial intelligence to produce a high quality image on the host GPU.

Note: If you have less than 4 GB RAM on your GPU, we recommend using an alternative denoiser to **AMD AI**.

No Denoising.

AMD AI Denoising with **Denoise Per Iteration** enabled.



No Denoising.

AMD AI Denoising.

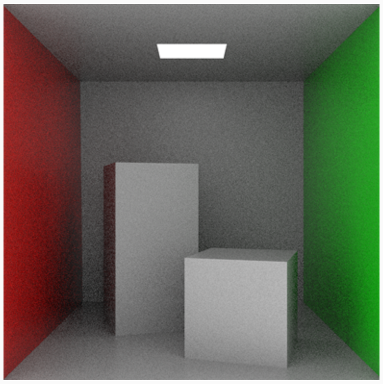
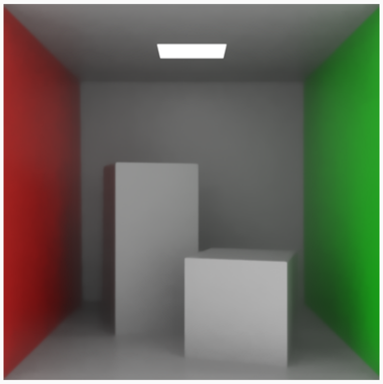
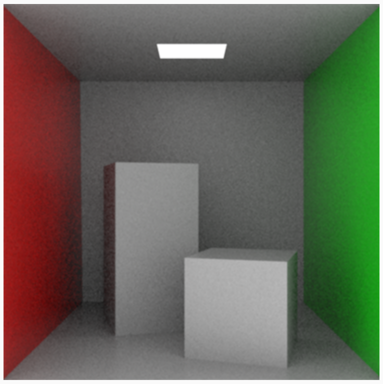
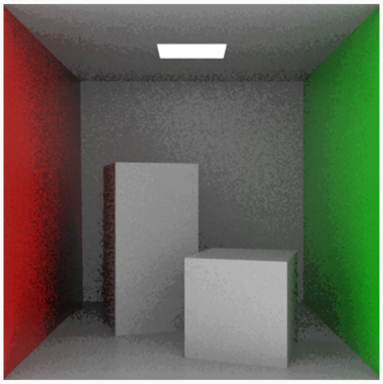
- **AMD Wavelet** - Edge-avoided À-Trous wavelet denoiser. This denoiser filter uses a technique called 'wavelets' to analyze and detect edges in images to blur noise across a surface without blurring across edges. It can use multiple image outputs to refine the analysis. In addition to the original color image, you can also use the surface normals, the depth (distance from eye), and surface IDs. Outputs with a weight set to zero are ignored. You can edit the following parameters:

Color Weight	The threshold for border detection using only the colors of the source image. Lower values preserve the borders more, but increase the noise level.
Normal Weight	The threshold for border detection using only the normals of the source image. Lower values preserve the borders more, but increase the noise level.
Depth Weight	The threshold for border detection using only the depth of the source image. Lower values preserve the borders more, but increase the noise level.
Surface Weight	The threshold for border detection using only the Surface ID. Lower values preserve the borders more, but increase the noise level.

AMD Bilateral - a non-linear, edge-preserving, and noise-reducing smoothing filter. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. You can edit the following parameters:

Blur Radius	The radius of the region that is used for output color computation. Accepts values between 1 and 50.
Image Blur Factor	The threshold for border detection in the image output. Lower values preserve borders more but increase the noise level. Blur Factors are applied to each of the four inputs.
Albedo Blur Factor	The threshold for border detection using only the surface colors without shading applied. Lower values preserve borders more but increase the noise level. Blur Factors are applied to each of the four inputs.
Normal Blur Factor	The threshold for border detection using only the surface normals for each pixel. Lower values preserve borders more but increase the noise level. Blur Factors are applied to each of the four inputs.
Depth Blur Factor	The threshold for border detection using only the Z depth of each pixel. Lower values preserve borders more but increase the noise level. Blur Factors are applied to each of the four inputs.

AMD Median - a non-linear digital filtering technique. Median filtering is widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. It operates by running through the signal entry by entry, replacing each entry with the median of neighboring entries. You can edit the **Radius** (in pixels) of the region that is used for

	denoising. The following table displays the render outputs of a scene using the different AMD denoisers. Click on an image to expand it.  No Denoising  AMD Wavelet Denoising  AMD Bilateral Denoising  AMD Median Denoising
Denoise Per Iteration	When **mPath** is set as the **Renderer** in the **Render** item, this applies denoising to each individual mPath pass using the selected **Denoising** engine.
Exposure Control	There are two possible ways to control the overall brightness of the final rendered image for any given render output. There is either **White Level** or **Photographic**. The choice can be made by selecting one of the options from the pop-up menu. **White Level** defines the amount of luminance (illumination brightness) that corresponds to pure white in the rendered image, adjusted with the **Input White Level** value. Higher values tend to darken the image, requiring a much brighter area to represent the whitest white, but can also be used to bring out detail in overblown areas. **Photographic** produces similar results but uses an **ISO** setting, more familiar to photographers. Much like how higher ISO makes a photographic image sensor more sensitive to light, in Modo a higher ISO produces brighter images (but without any added

	noise). **Tip:** For best results, when using the **Exposure Control**, **Clamp Colors** needs to be disabled.
Film Speed (ISO)	When the **Exposure Control** value is set to **Photographic** the ISO value is used to define the white level in the rendered image. Higher values produce brighter images, lower values darker.
Input Black Level	Specifies the radiance or luminance level that corresponds to pure black in the final rendered image (a pixel color value of 0.0). To use the input level controls effectively, **Clamp Colors** must be disabled.
Input Gray Level	A non-linear luminance adjustment that applies a curve-like function to the pixels, modifying the mid-range the most while attenuating the adjustment amount for values moving toward the defined **Black** and **White** points. Values above 1.0 lighten the mid-range while values below 1.0 darken the mid-range.
Input White Level	Specifies the radiance or luminance level that corresponds to pure white in the final rendered image (a pixel color value of 1.0). To use the input level controls effectively, **Clamp Colors** must be disabled.
Clamp Colors	Suppresses all color values in a rendered image to truncated 24-bit values prior to anti-aliasing, producing cleaner results in areas of extreme contrast. However, when this setting is enabled, you lose any extended dynamic range present in the image (making proper tone mapping impossible). If you intend to save your image in an high dynamic range format, leave the setting disabled. Additionally, instances where **Tone Mapping** may not be appropriate, any apparent blockiness can be minimized by adding a small amount of **Bloom**, available as an option of the **Final Color** output. When enabled, a small amount of dithering is added to reduce or eliminate banding in gradations.
Clamp Threshold	Determines the upper threshold within the image's rendered dynamic range where clamping takes place. At 100% the standard 24-bit low dynamic range values define the total brightness range; at 200%, twice the brightest pixel would be the upper limit. This setting allows you to clamp extremely bright pixels in an image for better antialiasing results, but still retain extended dynamic range in the image.
Tone	Dynamic range is measured as the difference between the darkest shadow to the brightest

Mapping	highlight in an image. Most image formats' dynamic range pales in comparison to that of the world around us. Much in the same way that a photographer struggles to capture all of the range our eyes can see using a camera, you may find your renders have shadow areas that are too dark, and highlight areas that are too bright. Luckily, Modo renders in full floating point accuracy, providing dynamic range well beyond what your monitor is capable of displaying. This allows you to modify the overall tonal balance of the rendered image in several ways, including **Tone Mapping**, a technique that is used to compress the rendered dynamic range into something viewable; shadows open revealing previously obscured details as highlight areas. The main tone mapping features are the **Tone Map Type** and **Tone Map Amount** controls. But you often find it takes a combination of settings for **Exposure Control**, **Tone Mapping**, and **Gamma** to achieve the best results. **Tip:** For more information, please refer to the Tone Mapping topic.
Tone Map Type	There are two types of tone mapping: **Reinhart Luminance**, which reduces luminance in bright areas while preserving color saturation, or **Reinhart RGB**, a method that lowers the luminosity per RGB channel, producing slightly different saturation results. Which is the better choice is highly subjective and dependent on the contents of the final render, so it's best to try both to see which is preferred.
Tone Map Amount	Controls the strength of dynamic range compression as a percentage, where, with a value of '0%,' no change is applied up to 100%, where the maximum amount of tone mapping is applied to the rendered image.
Hue Offset	Adjusts the color values of the rendered image independent of the luminosity or brightness values, shifting them across the entire spectrum in a sequential fashion. For example, when shifting a red color, moving the values in a positive fashion adjusts reds toward an orange hue, then yellow, and so on. The hue values should be thought of like a wheel, where a rotation of 180° inverts all the color values and a rotation of 360° brings them back to their initial state.
Saturation	Controls the concentration or amount of color in an image, independent of the luminosity of brightness. At 100%, colors are fully saturated, as defined in the item's material settings. Reducing the value reduces the overall color saturation down to a value of 0%, producing an image only with gray shades. Since Modo renders in full 32-bit floating point accuracy, values above 100% increase saturation without introducing the color banding and artifacts generally associated with over-saturating images.
Colorization	Use the **Colorization** options to introduce an overall color tint into an image. This can be

	purely for artistic reasons, but it can also be helpful in reversing color casts introduced by Image Based Lighting among other things. You need only to define a color with which to tint by setting a **Target Color** and then adjusting the Colorization amount, determining the strength of the color into the image. A value of 0% produces no colorization attenuating toward 100% where the image is replaced fully by the target color.
Target Color	Determines the color with which to tint the rendered image when the **Colorization** option is above 0%.
Output Black Level	Determines the output value of what is considered fully black in the image. Adjusting the value upwards changes the **Black** point to a shade of gray. As values are adjusted, intermediate values between black and white are adjusted as well.
Output White Level	Determines the output value of what is considered fully white in the image. Adjusting the value downwards changes the **White** point to a shade of gray. As values are adjusted, intermediate values between white and black are adjusted as well.
Unpremultiply Colors	When generating pixel values in rendered images, Modo**multiplies** the rendered result with the background color to calculate the final result. This "multiplied" final result is fine for most cases, however, when using alpha channels to composite multiple images together, partially transparent pixels often exhibit halos, an artifact of the previous background color. **Unpremultiply** basically reverses this effect. Activating this option won't reveal any difference in Modo's render display. You need to save the image and open it in another application to see the result. This function requires the object/scene to be rendered over a solid black background with a corresponding alpha channel. This can be easily achieved by disabling the **Visible to Camera** option in the Environment Item (and the **Visible to Refraction Rays** option, in cases where there are transparent surfaces, noting that this affects how the render looks).

Tip: Since a render output is just another layer, it can reside anywhere in the **Shader Tree**. Typically, they are at the top, directly under **Render** item, as this is where they see the result of all layers in the tree, but if a render output is placed inside of a Material Group, the render output only renders what it sees within that group. This makes it possible to render alpha channels for specific items, based on a polygon tag, instead of making some item-specific shader or hiding items in the item list. All active render outputs render each time the render command is invoked, and appears within the layer menu of the **Render Display** viewport.

Ambient Occlusion

The **Ambient Occlusion** output generates an occlusion map where occluded areas - crevices and other inaccessible areas - of a scene shade darker, while open areas render lighter colors, approximating an overall global illumination type of shading. Grainy renders can be smoothed out by increasing the **Occlusion Rays** value. Controlling the density of the occluded areas is done through the **Occlusion Range**.

- **Occlusion Rays** - specifies the number of random rays Modo shoots out from a surface when shading a pixel for **Ambient Occlusion**. Generally, the higher the number of rays, the more accurate (and smoother) the AO image is; low numbers introduce noise as variations in neighboring pixels and it becomes more evident the lower the value.
- **Occlusion Range** - specifies the length of the rays that are fired from a surface. If you have a complex object with overlapping elements, limiting the ray range is a good way to control what recessed areas are shaded darker. It also helps to push the shading further into recessed areas. A value of 0m disables the range, defaulting the value to the maximum length of the scenes overall bounding box size.

Depth

The **Depth** output generates a Z-Buffer that represents the distance from the camera to the geometric surface in shades of gray. The shade is determined by the actual distance value in Meters and requires a floating point format for saving. If the **Remap Pixel Values** option is enabled, you can define a maximum depth value and then save the resulting image to a low dynamic range format. These values are calculated the camera ray starting out white at the ray origin (camera position) and attenuating darker as they get longer becoming fully black at the specified **Maximum Depth** value. Useful for calculating Depth of Field in an external application or for composing a fog effect into a rendered scene.

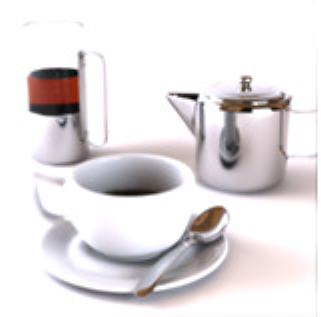
- **Remap Pixel Value** - when this option is enabled, you can specify a **Maximum Depth** value.
- **Maximum Depth** - sets a "clipping plane" for the black point so that Modo knows at what distance to shade the Z-buffer pure black.

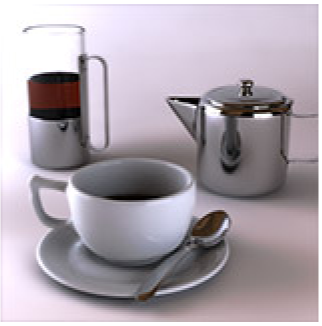
Final Color

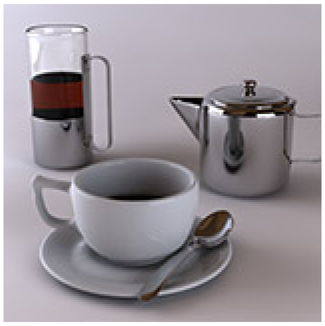
The **Final Color** output represents all the settings of the **Shader Tree**, leveraging all visible material and texture layers as well as shaders and the render item settings.

- **Bloom** - in photography, when a very bright part of an image neighbors a very dark part, the bright part appears to glow, this phenomenon is known as "bloom". Modo's **Bloom** simulates this effect by removing excess energy from a pixel and distributes it around to its neighbors. This checkbox enables the effect; once checked, two additional options become available:

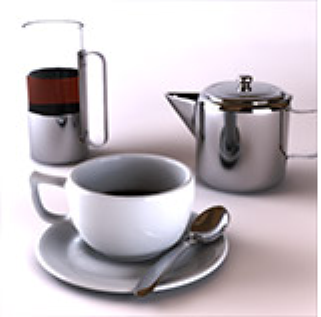
  - **Bloom Threshold** - sets the lower threshold to which pixels are effected by **Bloom**. When set to 100%, only the very hottest specular highlights bloom, where at 0% every pixel in the scene receive the bloom effect.

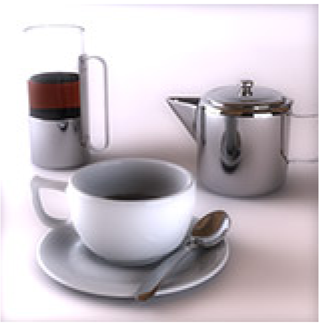

Bloom Threshold 0%

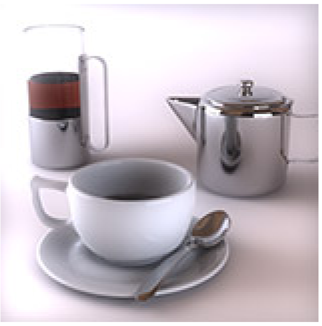

Bloom Threshold 50%


Bloom Threshold 100%

  - **Bloom Radius** - sets the distance that energy is distributed, in effect controlling the size of the glow. 2% is the default value. Larger radii impart an ethereal look to the rendered image.


Bloom Radius 1%

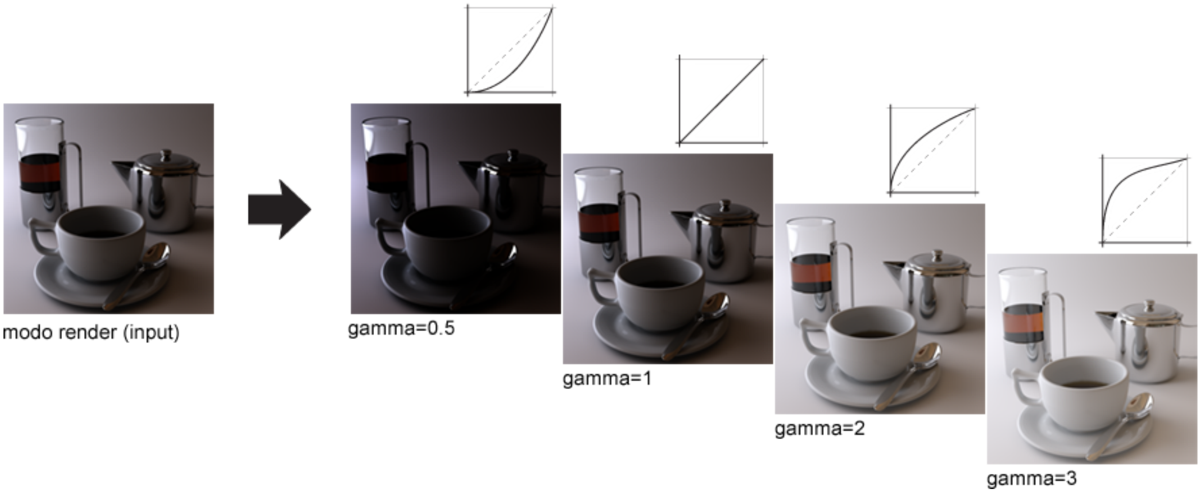

Bloom Radius 8%


Bloom Radius 16%

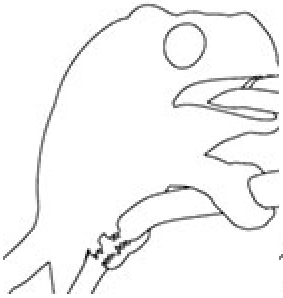
- **Vignette Amount** - in photography, vignetting is the natural reduction of an image's brightness toward the outer edges of the frame. The **Vignette Amount** in Modo, when set at 100%, is a physically-accurate vignette simulation, based on the camera's **Focal Length** and **Film Size**. Values above 100% darken the resulting vignette, and values below 100% reduce its visibility.
- **Output Gamma** - gamma is the measure of total contrast in an image and represents the relationship of an image input to an image output. In this case, the input is Modo's rendered image (what Modo is holding in its memory) and the output is either the image being displayed in the **Render Display** viewport (what you see) or the image that is saved. A gamma value of 1 is said to be **linear**, what Modo rendered, is identical to what is displayed. Modifying the **Gamma** value changes how the image is represented. Values higher than 1 lighten the image, while values lower than 1 darken it. Adjustments to the gamma value are not linear, that is, they don't change all the pixels in an image the same amount, explaining why it is often referred to as a gamma curve.

The default **Gamma** value for any new render output is derived from the global preference found in the **Preference Editor** under **Rendering**, called **Default Output Gamma**, which is set to 1.0. With the introduction of Color Management in Modo, it is no longer necessary to "degamma" texture images. The

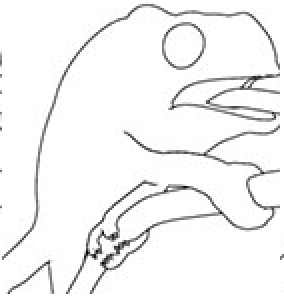
Output Gamma controls remain simply as an additional image adjustment control. Adjusting the **Output Gamma** value in the render output overrides the global preference on a per-item layer basis.



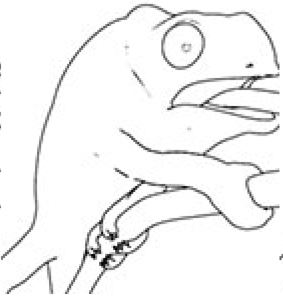
- **Output Colorspace** - defines the colorspace of the rendered frame. When set to **default**, the colorspace definition is derived from the Scene Item settings. Changing the value results in an override that can be set per output. Experienced users with specific needs may want to adjust from **default** to a specific colorspace, but for the most part, the default settings produce the correct results. For more information, please refer to the Color Management topic.
- **Contours** - Modo can render a contour outline around specified boundaries. The options available are:
  - **None** - the default value; renders no contouring outline.
  - **Surface Boundaries** - contour line renders around all surface boundary definitions (Modo Material tag boundaries).
  - **Segment Boundaries** - contour line renders around all segment boundary definitions (Modo Segment ID boundaries, which are not user-definable).
- **Contour Color** - defines the color of the contour outline using the standard Modo color controls for specifying an R, G, and B value.
- **Contour Width** - defines the thickness of the contour line applied to the render output, defined as pixels, so a value of 1 would render a line 1 pixel wide over all the boundaries.
- **Fade with Distance** - you can fade the application of the contour outline when enabling this option. Once enabled, an additional option **Fade Start Distance** determines where the fading effect start.
- **Crease Threshold** - determines the angle at which Modo draws a contour line. Lower angles generate more lines.



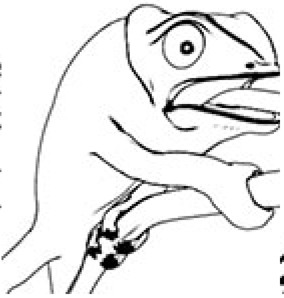
Crease Threshold 180°



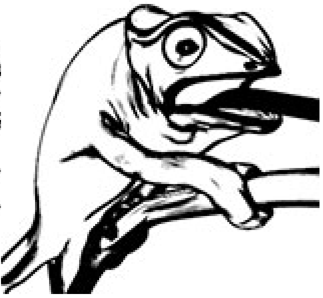
Crease Threshold 90°



Crease Threshold 45°



Crease Threshold 15°



Crease Threshold 5°

Motion Vector

The **Motion Vector** output represents the speed and direction at which an item is moving within the scene, calculated per-pixel. The main use of this render output is to apply a post motion-blur effect to the frame using an external compositing application. As rendering actual motion blur in Modo requires a fair number of antialiasing passes for a smooth result, rendering out a frame normally and applying the effect separately achieves similar results in a fraction of the time. For normal Motion Vector outputs, the red, green, and blue pixel values represent the X, Y, and Z vector components, which range from -1 to 1. For motion vector outputs, the red and green pixel values represent horizontal and vertical movement measured in pixels and can be negative requiring an extended dynamic range format to save (unless the **Remap Pixel Values** option is enabled).

- **Remap Pixel Values** - adjusts the resulting motion vector values to values appropriate to low dynamic range images, scaling values appropriately based on the **Maximum Motion** value. Enabling this option means the values are remapped (Reel Smart Motion Blur), and disabling it means the values are saved at their actual calculated result (Nuke).
- **Maximum Motion** - represents the maximum distance a blurred pixel would travel. This setting also defines the amount of scaling applied to the resulting values when **Remap Pixel Values** is enabled. When working with the Re:vision ReelSmart Motion Blur tool, you may want to make sure that **Remap Pixel Values** is enabled, and make sure the **Maximum Motion** values match in both locations for the proper result.

Render Output Masking

Render outputs support group masks and layer masks in the **Shader Tree**. This works both in Preview and the Render Display.

To enable render output masking:

1. Select the Render item in the Shader Tree.

2. In its **Channels** list, found in the lower-right panel, set the **outputMasking** channel to **True**. It is disabled by default.

   When you apply a group mask to a group containing a render output, that output is masked accordingly. The mask value is applied to the render output to mask certain parts of it. This way, you can use image maps, procedural textures, or gradients on render outputs.

Note: Depending on the scene, render time may increase slightly when render output masking is enabled, as Modo needs to store the values while evaluating the shader.

Note: For more information on group and layer masks, see Advancing the Shader Tree.

Render Output Items

Rendering is the act of converting the scene information into a bitmap image by evaluating all the geometry and lighting information in a given file. Generally, the resulting rendered image is a fully raytraced and shaded image with full shadows, reflection and the like, much like a photograph depicting an entire scene when taking a picture with a camera. In Modo, these images are the result of what are called render outputs, the visible representations of the full scene evaluations. Each render output is a separate layer in the **Shader Tree** and produces its own image layer as part of the render calculations. Every time one of the Render Commands is invoked, all of the render output layers are simultaneously calculated and represented in the Render Display viewport. The different layers can be viewed by choosing one of the specific **Output** options within the display window.

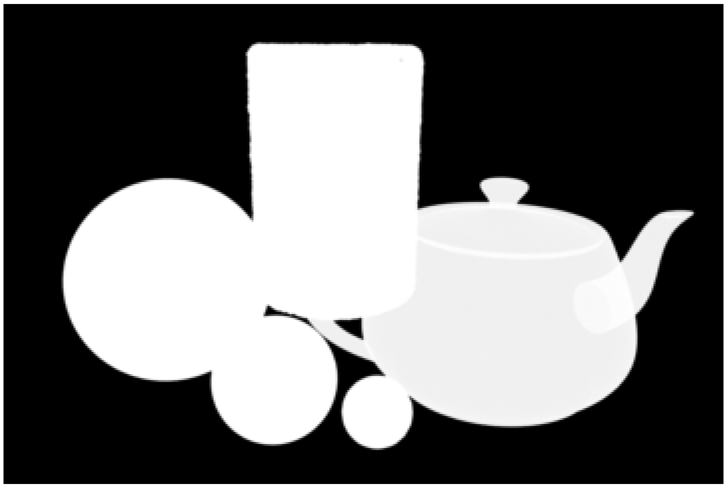
Occasionally, you only need a subset of the full evaluations, such as the diffuse color layers without any shadows. Modo offers many different **Render Output** types useful in a variety of situations ranging from purely diagnostic, to specific compositing utility in an external application. A default scene automatically includes a **Final Color** and **Alpha** output which is fine for most situations. Additional layers can be added from within the **Shader Tree**, by clicking the **Add Layer** option of the full viewport window and selecting **Special** > **Render Output** from the pop-up menu. This adds an additional layer to the top of the tree. You can then right-click in the **Effect** column of the render output for a pop-up context menu that allows you to select from the alternate output types. For information on the editable attributes of the individual outputs, see the primary Render Outputs topic.

Basic Render Outputs

These are the different **Basic** render outputs.

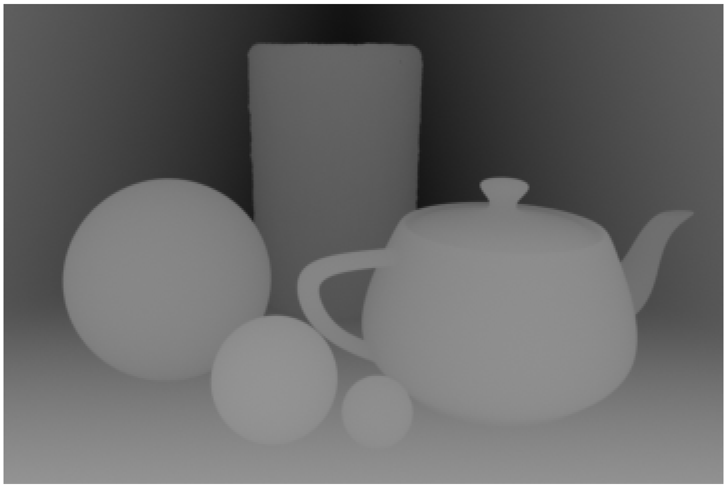
Tip: To learn more about render output properties, see Render Outputs.

Alpha



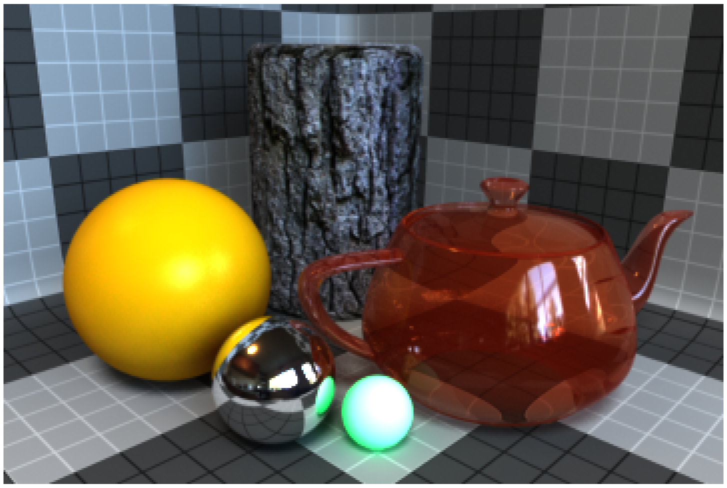
The **Alpha** output calculates the opacity of the objects based on their material settings to determine an alpha (transparency) value. With this setting, an environment does not contribute to the alpha channel, so when the final image is viewed with an application that supports alpha channels, only the rendered geometry is visible. If an object or surface has a level of transparency, it is also being considered in the alpha channel. This way, it is possible to render images with semi-transparent elements that can easily be composited over other backdrops in an application such as Nuke, After Effects or Photoshop. For best results, when compositing multiple images, the **Unpremultiply Colors** option in the corresponding **Final Color** output should be enabled. For more information on this option, see Render Outputs. There are also additional options for uniquely controlling how rendering evaluates transparency for images in the Alpha Type item.

Depth



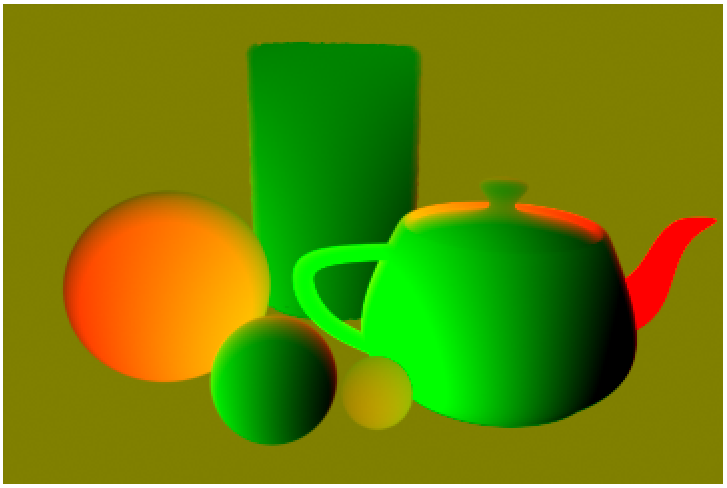
The **Depth** output encodes the distance from the camera item to the first surface intersection (transparency is ignored) and generates a grayscale Z-Buffer image that represents the measured distance in meters. When **Remap Pixel Values** is enabled, you can alternately define a **Maximum Depth** distance, remapping the values to fit within the 0-1 value range with white at the camera ramping toward full black at the defined **Maximum Depth** value. The **Depth** output is useful for calculating depth-of-field in an external application or for composing a fog effects into a rendered scene.

Final Color



The **Final Color** output represents the full evaluation of all the settings of the Shader Tree, leveraging all visible material and texture layers as well as the shader and the render item settings. The **Final Color** render output produces a full floating point accurate high dynamic range image (unless the **Clamp Colors** option is enabled, see Render Outputs).

Motion Vector

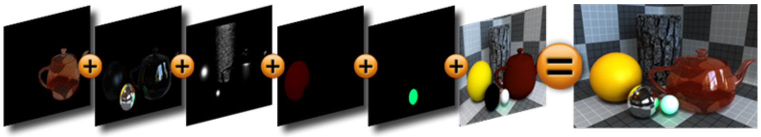


The **Motion Vector** output encodes the direction and speed of each pixel into an RGB value, where red and green represent horizontal and vertical movement respectively. This allows the resulting **Motion Vector** image to be used for further processing in an external application, using them as an input to generate motion blur for a rendered frame as a post production effect. This can greatly enhance the smoothness of motion blur and save considerable rendering time when processed this way, however, since each pixel can only have a single color value (and therefore a single motion vector), incorrect result is produced when motion blurring transparent or translucent objects moving in a scene.

Compositing Outputs

Render outputs are incredibly useful when you wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aid in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is which render outputs combine to equal the final color output. The answer is Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color, where each successive layer is added to the layer below in the composition (also called linear dodge). Rendering each layer at a gamma of 1.0 and then combining them and adding the gamma back to the fully composed layers produce the best results.

You may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to **Effect** > **Alpha** (or simply adding them to existing Shader Tree masks). Then each time a render command is invoked, all active render outputs get generated simultaneously.



Cryptomatte Render Output

The **Cryptomatte** render output is used to generate renders in the OpenEXR file format for use in compositing applications such as Nuke.

Cryptomatte identifies surfaces in Modo renders and separates them into different ID mattes. These mattes can be used to mask out specific elements when making adjustments in compositing.

In Modo, the different matte surfaces are identified in a render by adding an **LPE Label** to materials in the Shader Tree. Materials that share the same **LPE Label** are defined as the same matte surface in the render.

Note: Cryptomatte outputs are only supported with the mPath renderer. See Rendering with mPath for more information.

Note: See the Cryptomatte website to learn more.

Adding a Cryptomatte Render Output to your Scene

If you need to generate a Cryptomatte render, you must add a Cryptomatte render output to your Shader Tree. To add a Cryptomatte render output to your scene:

1. Click the **Shading** tab in the **Modo**, **Model**, or **Render** layout.
2. Click **Add Layer** > **Render Outputs** > **Cryptomatte** > **Cryptomatte**.

   The render output is added to the top of your Shader Tree.

   Note: While **Cryptomatte ID**, and **Cryptomatte Coverage** are other output options, they do not function as render outputs in the Shader Tree.

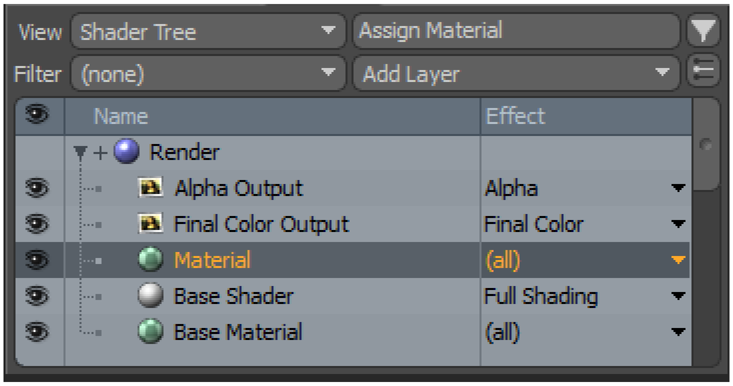
The Cryptomatte render output has several **Properties** available. These properties are shared with other render outputs. See Render Outputs for more information.

Identifying Cryptomatte Surfaces

In order for the Cryptomatte render output to correctly identify the different matte surfaces, you need to assign an **LPE Label** to materials that you wish to identify as a surface. Materials which share the same **LPE Label** are identified as matching matte surfaces in the Cryptomatte render. This gives you the flexibility to combine multiple surfaces into one matte.

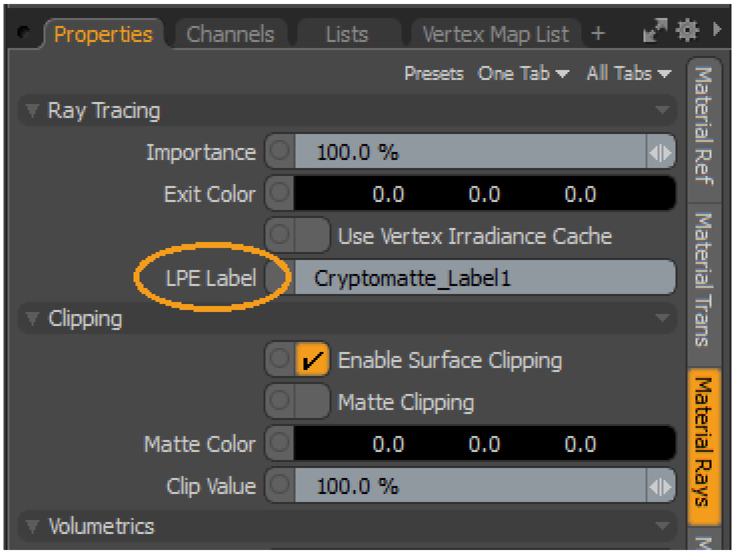
To assign a Cryptomatte ID to a material in Modo:

1. Click the **Shading** tab in the **Modo**, **Model**, or **Render** layout.
2. Select a material in the Shader Tree.


View
Shader Tree
Assign Material
Filter
(none)
Add Layer

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Material	(all)
Base Shader	Full Shading
Base Material	(all)

3. Click the **Material Rays** tab in the **Properties**.
4. In **LPE Label**, type an ID of your choice for the material.


Properties
Channels
Lists
Vertex Map List
Presets
One Tab
All Tabs
Material Ref
Material Trans
Material Rays
Ray Tracing
Clipping
Volumetrics

Importance 100.0 %
Exit Color 0.0 0.0 0.0
☐ Use Vertex Irradiance Cache
LPE Label Cryptomatte_Label1
☑ Enable Surface Clipping
☐ Matte Clipping
Matte Color 0.0 0.0 0.0
Clip Value 100.0 %

5. To assign a Cryptomatte ID to another material, select a different material in the Shader Tree.

6. Click the **Material Rays** tab in the **Properties** and type an ID of your choice in **LPE Label**.

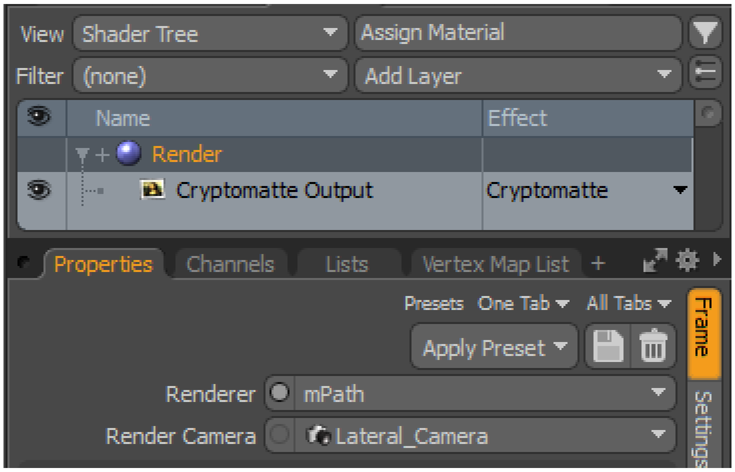
   This ID can match the ID from step 4, or it can be a new ID. If this ID matches the ID from step 4, the Cryptomatte render identifies them as the same matte surface once rendering begins.

Creating a Cryptomatte Render

A Cryptomatte render output creates an OpenEXR (**.exr**) file that can be imported to Nuke. Ensure you have a Cryptomatte render output in your Shader Tree, and materials tagged using the LPE Label. See Adding a Cryptomatte Render Output to your Scene and Identifying Cryptomatte Surfaces to learn how.

To generate a Cryptomatte render output:

1. Click the **Render** item in the Shader Tree and ensure **mPath** is selected as the **Renderer**.



View: Shader Tree

Filter: (none)

Render

Cryptomatte Output: Cryptomatte

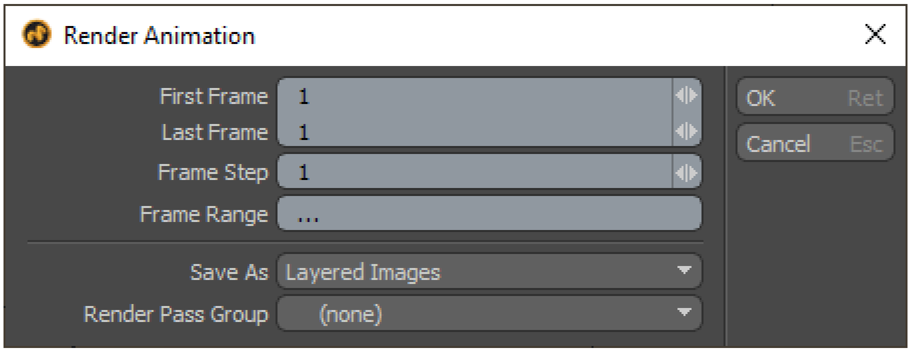
Renderer: mPath

Render Camera: Lateral_Camera

Note: Cryptomatte is only supported by mPath rendering. See Rendering with mPath for more information.

2. Click **Render** > **Render Animation**.
3. If you only need to render a single frame, set the **First Frame** and the **Last Frame** in the **Render Animation** dialog to **1**.
4. Set **Save As** to **Layered Images**.

   This produces a single **.exr** file which contains the Final Color output and Cryptomatte output which can be imported easily to applications supporting Cryptomatte.
5. Click **OK**.



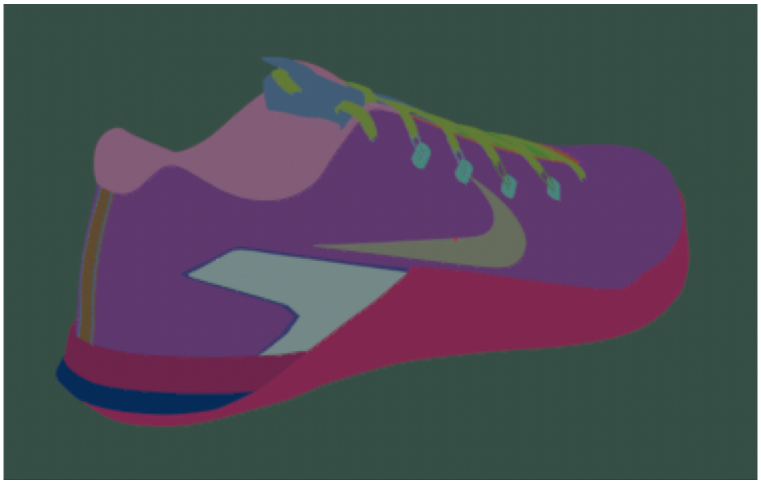
First Frame 1
Last Frame 1
Frame Step 1
Frame Range ...
Save As Layered Images
Render Pass Group (none)

If you have specified an **Output Filename** in the Final Color output, Modo asks if you want to use this filename. Click **No** and a **Save As** dialog opens for you to create a different output file.

6. Choose a save location for the render output, change **Save as type** to **Layered OpenEXR Float 32-bit (*.exr)** and click **Save**.

   A **Render** window opens and the render begins.

   At render time, the Cryptomatte output is expanded into separate buffers required for the **.exr** file. You can view each of these buffers using the **Output** dropdown. Once the render is complete, the Cryptomatte 32-bit **.exr** file is exported to the location you specified in step 2.

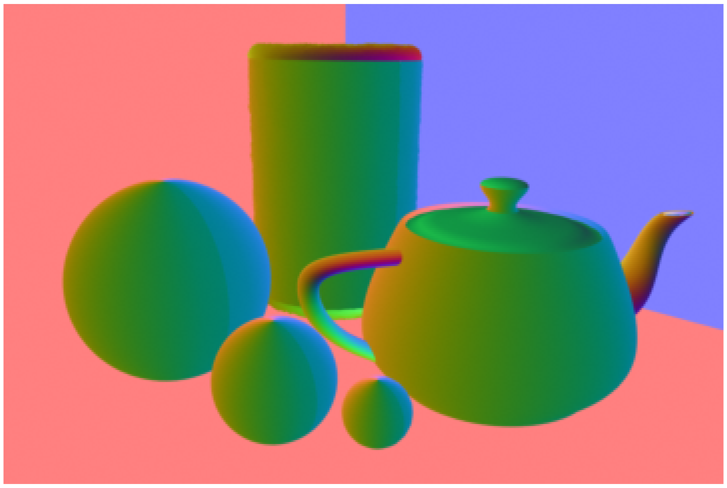


An **.exr** Cryptomatte file rendered in Modo and imported to Nuke using the Nuke Cryptomatte plug-in. Each matte surface identified in the render is assigned a different color.

Geometry Render Outputs

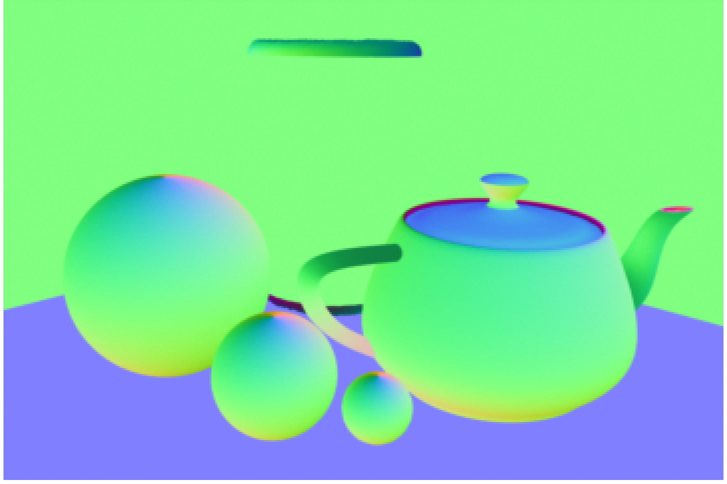
Learn about the different **Geometry** render outputs.

dPdu Vector



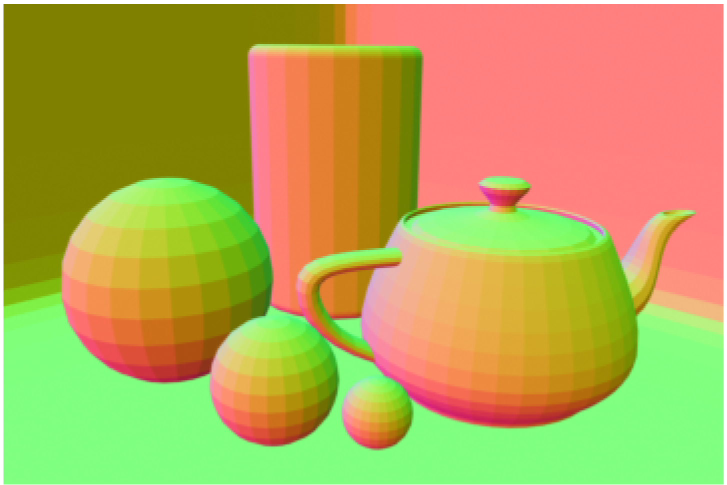
The **dPdu** and dPdv are world space vectors that point in the direction of increasing U and increasing V at each point on a surface that has a UV map. dPdu and dPdv have a big effect on the anti-aliasing of UV image maps and also define the reference directions for anisotropic highlights and reflections, so the outputs are sometimes useful for the purposes of debugging.

Dpdv Vector



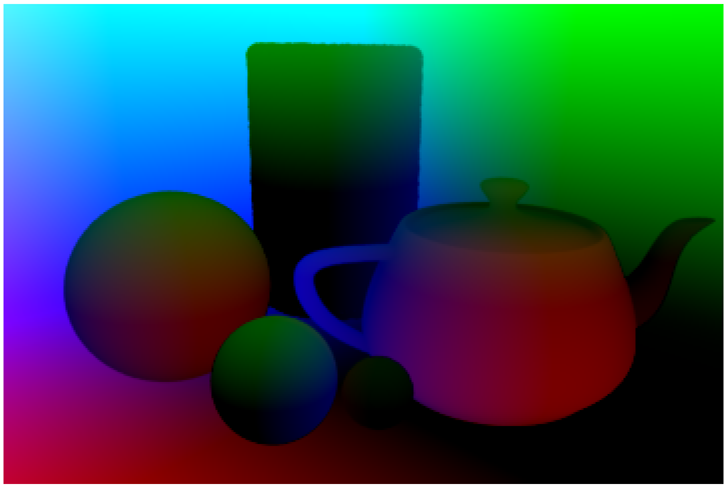
The dPdu and dPdv are world space vectors that point in the direction of increasing U and increasing V at each point on a surface that has a UV map. dPdu and dPdv have a big effect on the anti-aliasing of UV image maps and also define the reference directions for anisotropic highlights and reflections, so the outputs are sometimes useful for the purposes of debugging.

Geometric Normal



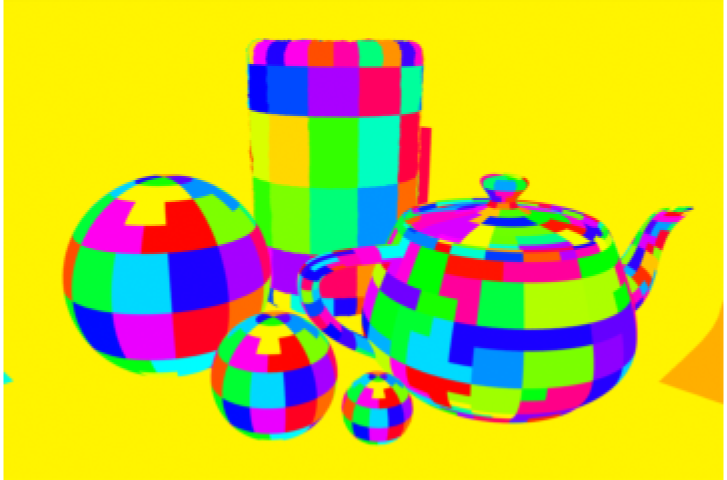
The **Geometric Normal** output simply renders all meshes with a color ramp based on the angle of their polygonal normal, where the RGB image components represent the corresponding XYZ vectors, so values are negative unless the **Normalize** option is enabled. It is useful for evaluating the continuity of a mesh's surface. In this example the Material item's **Smoothing** option has been disabled to emphasize the effect.

Object Coordinates



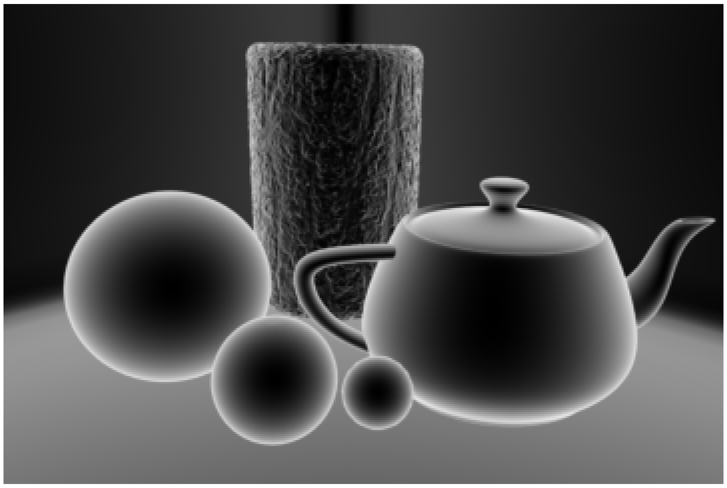
The **Object Coordinates** output generates red, green, and blue components that contain the undeformed object space X, Y, and Z coordinates of visible surfaces. The pixel values are in meters and can be negative, so a floating point image format should be used when saving this output. This theoretically allows post-processing applications to add solid textures that remain attached to surfaces during an animation.

Segment ID



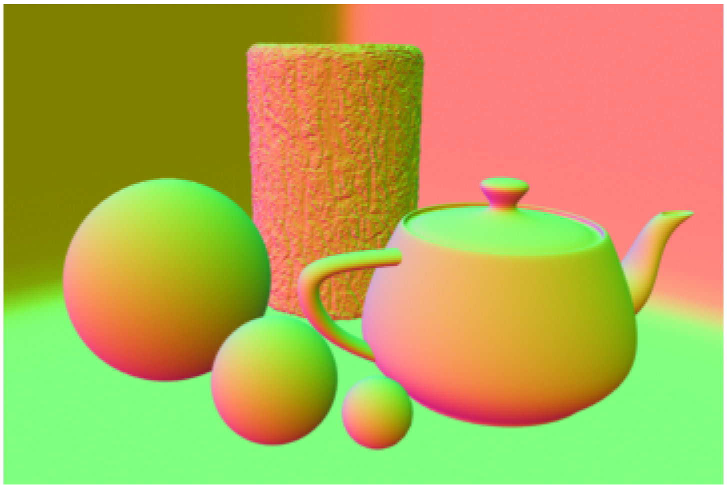
When Modo renders a project, all meshes are broken down into segments for optimization and general scene management. Using the Segment ID shading option renders each generated segment with a random color. This shading method is purely a diagnostic option.

Shading Incidence



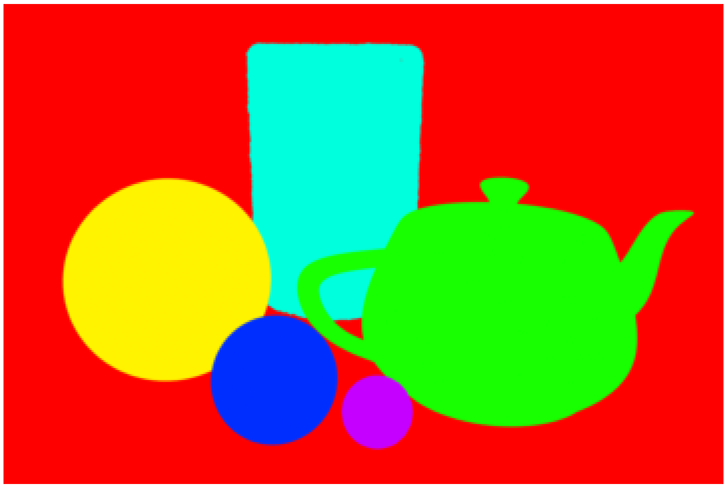
The **Shading Incidence** output renders a grayscale image the represents surfaces angle relative to the camera. Shaded surfaces that face the camera directly are rendered as black, attenuating toward white on shading that is perpendicular to the camera.

Shading Normal



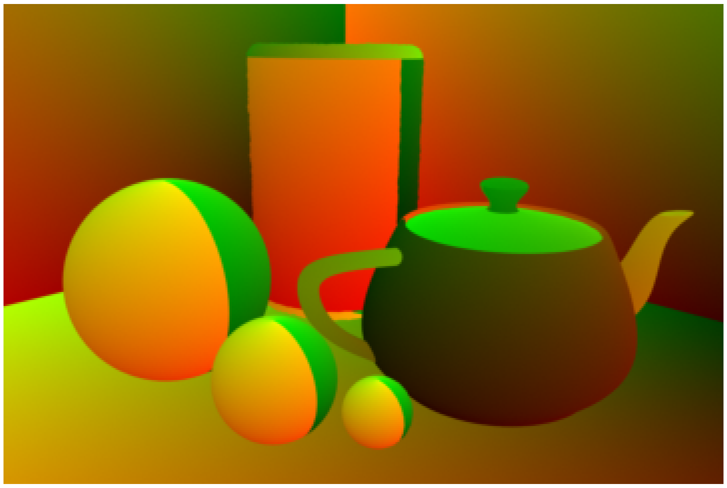
The **Shading Normal** output renders all meshes with a color ramp based on the angle of their surface's shaded normal, where the RGB image components represent the corresponding XYZ vectors, so values are negative unless the **Normalize** option is enabled. Where the Geometric Normal is calculated exclusively by the polygonal geometry, the **Shading Normal** also takes into account surfacing attribute effect on normals, such as bump maps.

Surface ID



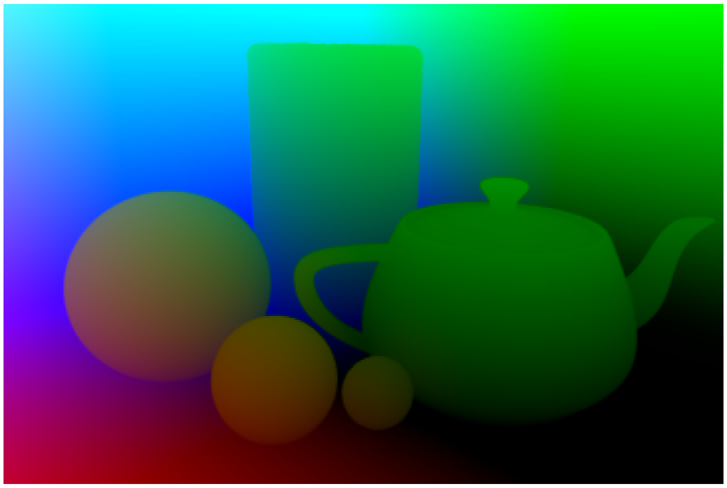
The **Surface ID** output renders each individual object with a random color. This is purely a diagnostic shading option, as it may also be helpful in spotting surface relationships in complex scenes.

UV Coordinates



The **UV Coordinates** output generates a channel that converts UV positions to a color value. Some software is then capable of utilizing this image to reapply a texture map to the object after it has been rendered.

World Coordinates

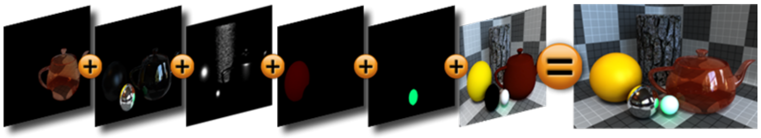


The **World Coordinates** output generates red, green, and blue components producing the final world space X, Y, and Z coordinates of visible surfaces. The pixel values are in meters, representing an actual 3D coordinate in space and can be negative, so a floating point image format should be used when saving this output.

Compositing Outputs

Render outputs are incredibly useful when you wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aid in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is which render outputs combine to equal the final color output. The answer is Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color, where each successive layer is added to the layer below in the composition (also called linear dodge). Rendering each layer at a gamma of 1.0 and then combining them and adding the gamma back to the fully composed layers produce the best results.

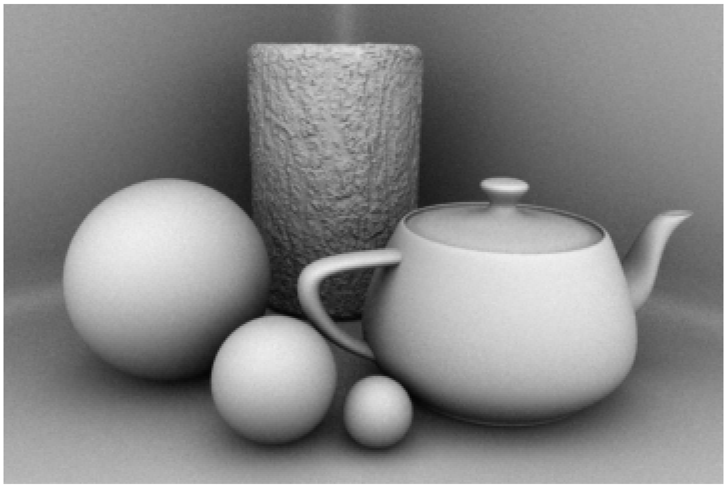
You may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to **Effect** > **Alpha** (or simply adding them to existing Shader Tree masks). Then each time a render command is invoked, all active render outputs get generated simultaneously.



Lighting Render Outputs

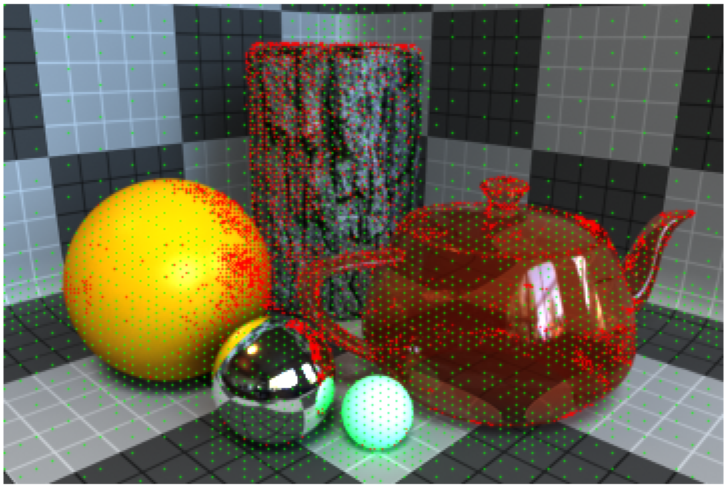
This topic describes the different **Lighting** render outputs.

Ambient Occlusion



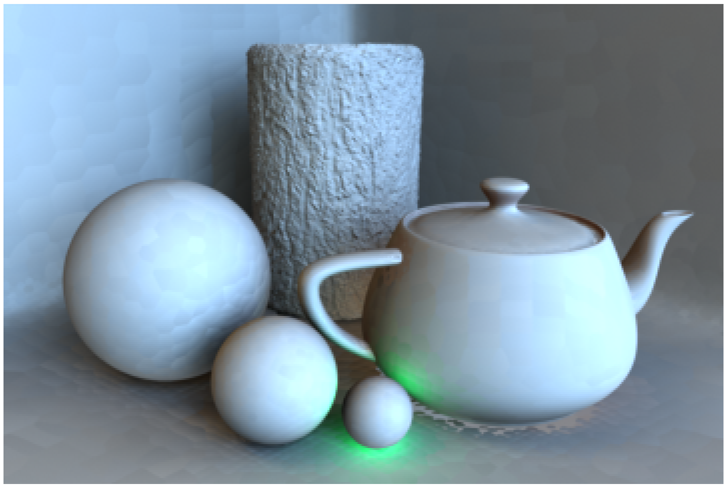
The **Ambient Occlusion** output generates an occlusion image where occluded areas, such as crevices and other inaccessible areas of a scene shade darker, while open areas render lighter colors, approximating an overall global illumination type of shading. Grainy renders can be smoothed out by increasing the **Occlusion Rays** value. You can control the density of the occluded areas with the **Occlusion Range** option. For more information on **Ambient Occlusion**, see the Render Outputs topic.

IC Positions



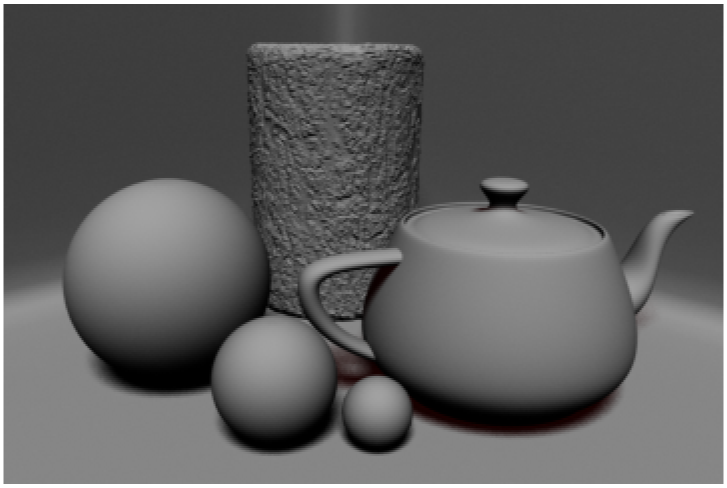
The settings in the **Render Item** > **Properties** > **Global Illumination** tab allow you to specify values that the irradiance cache uses to shade a rendered scene. The **IC Positions** output displays the pixel positions used in generating the shading based on the irradiance cache values. Green points are positions sampled during the initial pass, red points signify additional sample generated by the **Interpolation Values** setting. Useful in analyzing various IC settings.

IC Values



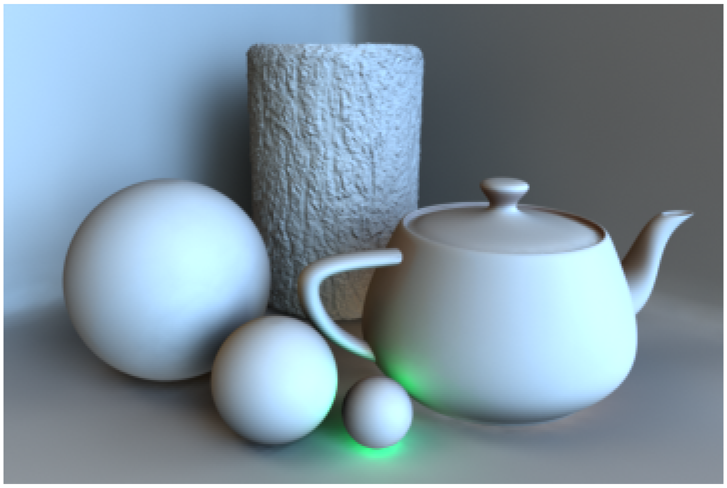
The settings in the **Render Item** > **Properties** > **Global Illumination** tab allow you to specify values that the irradiance cache uses to shade a rendered scene. The **IC Values** output displays the diffuse values calculated during the initial irradiance cache Render Pass. The image appears splotchy as each irradiance sample is not blended with nearby samples; giving one a much better understanding of the area affected by individual IC positions. Useful in analyzing various IC settings.

Illumination (Direct)



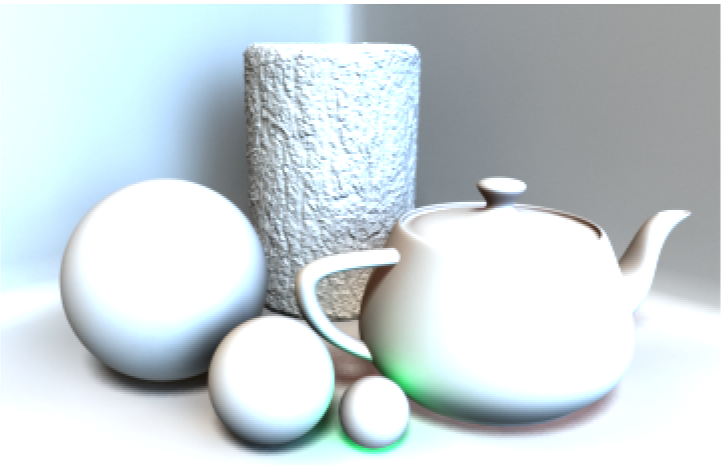
The **Illumination (Direct)** output generates a channel exclusive to the illumination in a scene from all direct light items such as distant lights, area lights, point lights and spot lights, independent of all other illumination types and surfacing attributes. For information on lights, see Lights

Illumination (Indirect)



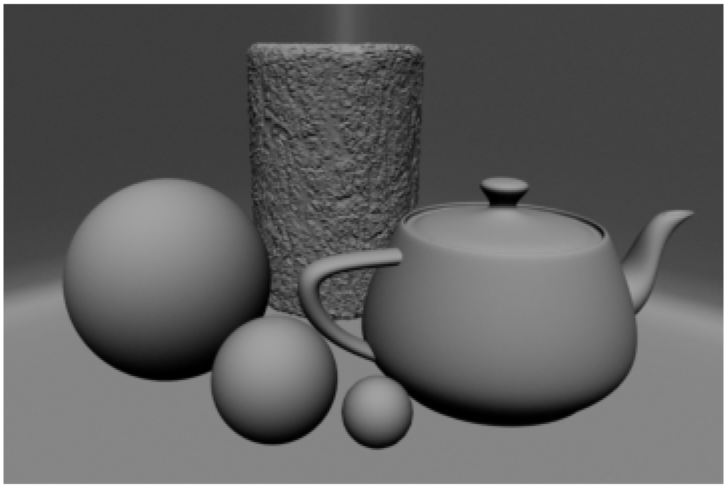
The **Illumination (Indirect)** output generates an image exclusive to the illumination in a scene from all indirect sources, such as image-based lighting and luminous polygons, independent of all other illumination types and surfacing attributes.

Illumination (Total)



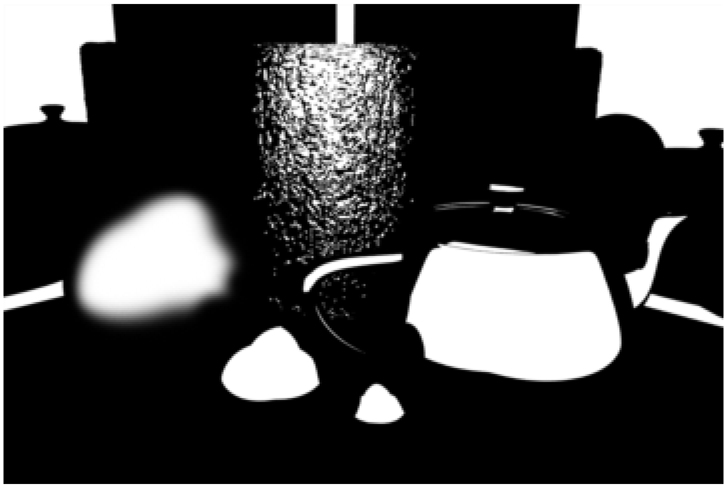
The **Illumination (Total)** output generates an image of the full illumination evaluation in a scene, including shadows cast by direct sources, but independent of all other surfacing attributes.

Illumination (Unshadowed)



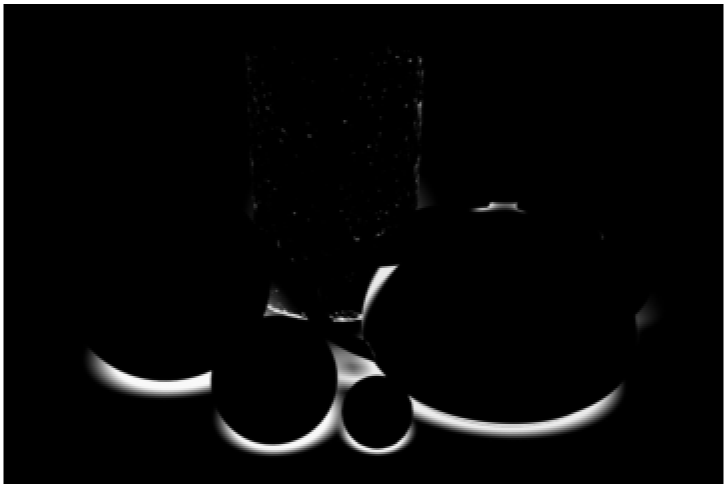
The **Illumination (Unshadowed)** output generates a channel of the full illumination in a scene excluding shadows cast by direct light sources, independent of all other surfacing attributes.

Reflection Occlusion



The **Reflection Occlusion** output renders an image that is strictly black and white, determining if a reflection ray off of an object has hit any other geometry in the scene. White if it has and black if it has not. The areas that appear as white here would be shaded with the background environment in a **Final Color** render.

Shadow Density

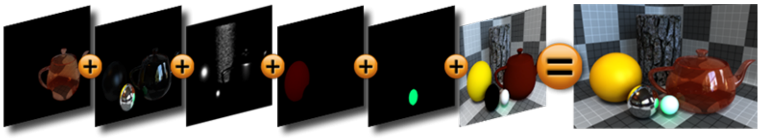


The **Shadow Density** rendering output renders a channel generated from the density of all direct light shadows in Modo, excluding color shading or texture. The darkest areas of shadow render as white, ramping toward black for the areas that are fully illuminated. When rendering in layers, it is very useful to have shadows rendered separately, giving you control of color and transparency. You can easily invert this channel in an image-editing application and layer it over the unshadowed render set to **Multiply**, giving control over how dark the shadows are based on the layer's transparency.

Compositing Outputs

Render outputs are incredibly useful when you wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aid in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is which render outputs combine to equal the final color output. The answer is Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color, where each successive layer is added to the layer below in the composition (also called linear dodge). Rendering each layer at a gamma of 1.0 and then combining them and adding the gamma back to the fully composed layers produce the best results.

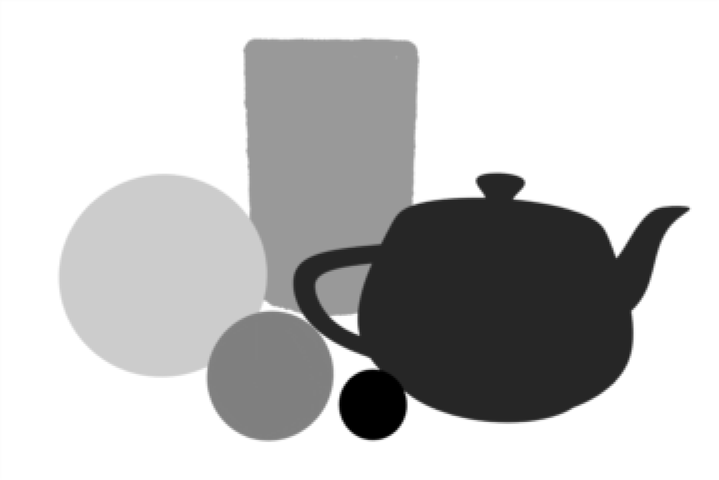
You may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to **Effect** > **Alpha** (or simply adding them to existing Shader Tree masks). Then each time a render command is invoked, all active render outputs get generated simultaneously.



Material Render Outputs

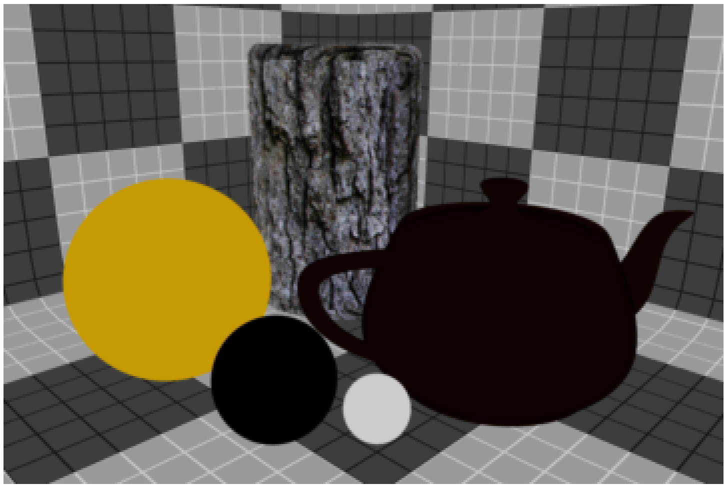
This topic describes the different **Material** render outputs.

Diffuse Amount



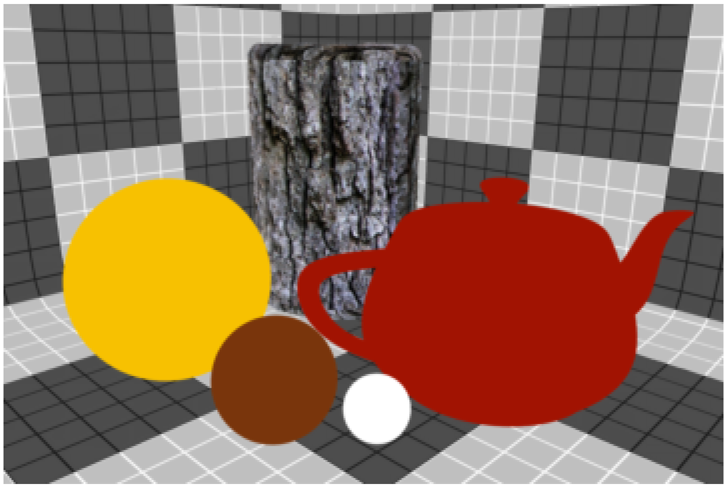
The **Diffuse Amount** output renders a scene as grayscale image that represents the diffuse amount values per surface, independent of illumination or shading.

Diffuse Coefficient



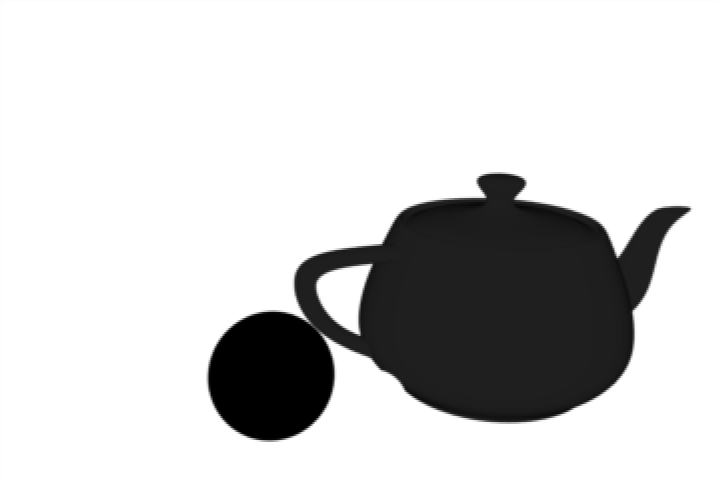
The **Diffuse Coefficient** output renders a scene as a full-color image that represents the diffuse colors of a scene taking into account **Fresnel** values and energy conservation; independent of illumination or shading.

Diffuse Color



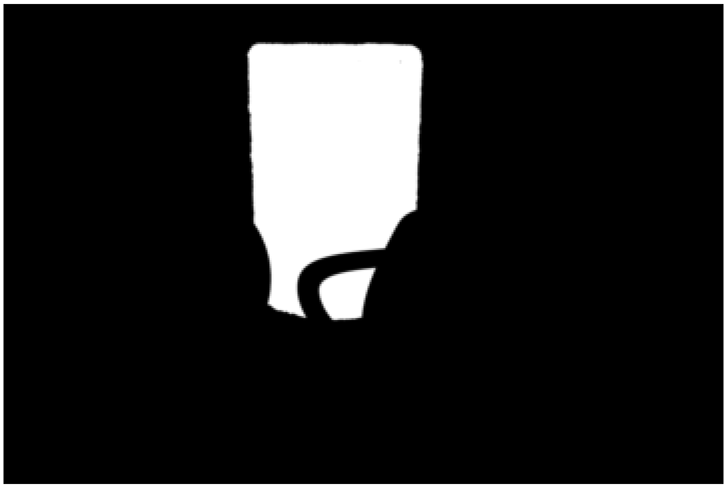
The **Diffuse Color** output renders a scene as a full-color image that represents the diffuse colors of a scene independent of illumination or shading.

Diffuse Energy Conservation



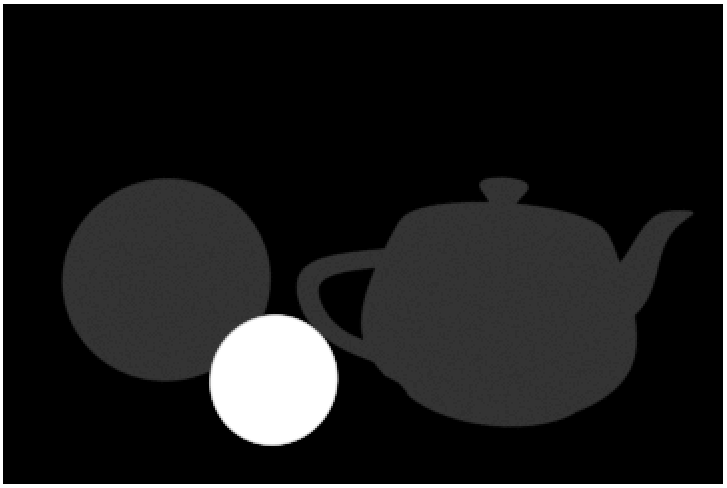
The **Diffuse Energy Conservation** output renders a scene as a grayscale image that represents the **Energy Conservation** values for surfaces based on the **Fresnel** and **Energy Conservation** option settings.

Diffuse Roughness



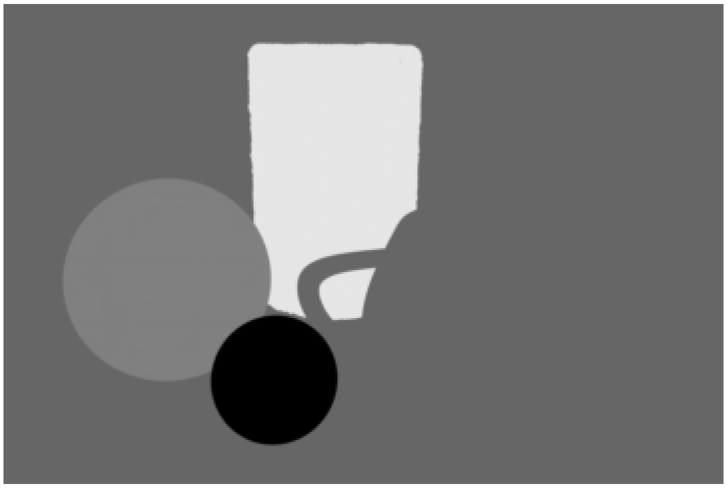
The **Diffuse Roughness** output renders a scene as a grayscale image that represents the **Diffuse Roughness** values of a Material in a scene, independent of any illumination or shading.

Reflection Coefficient



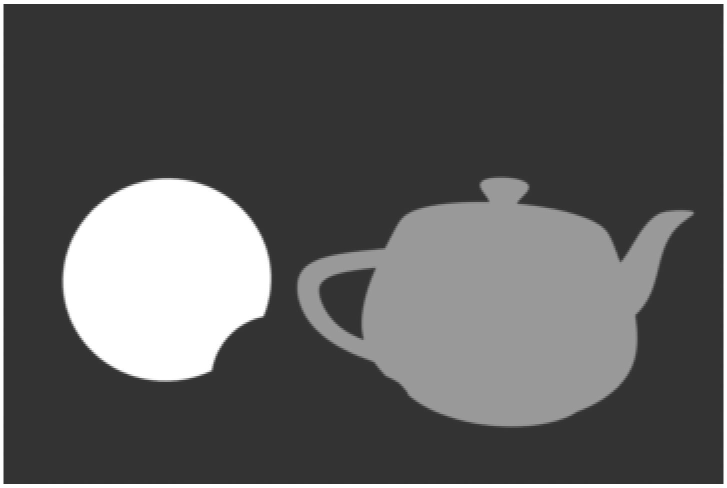
The **Reflection Coefficient** output renders a scene as a grayscale image that represents the reflection amount per surface. Black for 0% reflectivity, ramping toward white at 100%.

Roughness



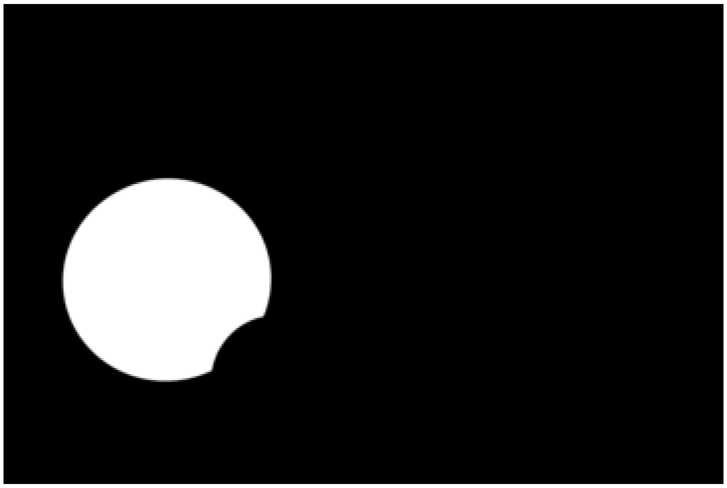
The **Roughness** output renders a scene as a grayscale image that represents the roughness amount per surface. Black for 0% roughness, ramping toward white at 100%.

Specular Coefficient



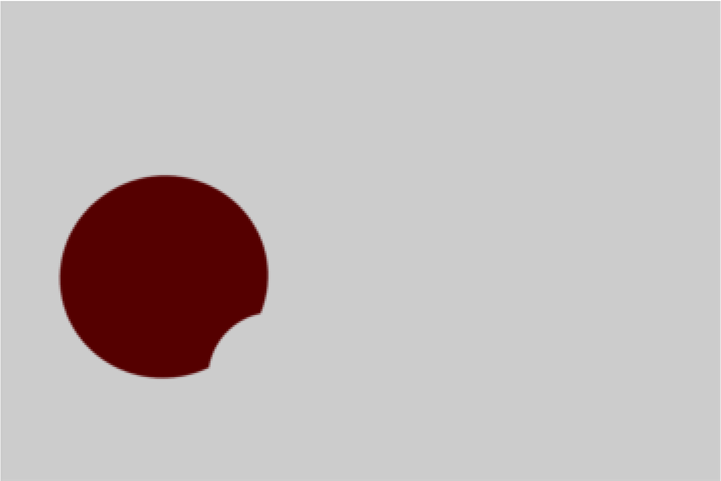
The **Specular Coefficient** output renders a grayscale image that represents the amount of specular on surfaces within a scene. Black for 0% specularity, ramping toward white at 100% specular.

Subsurface Amount



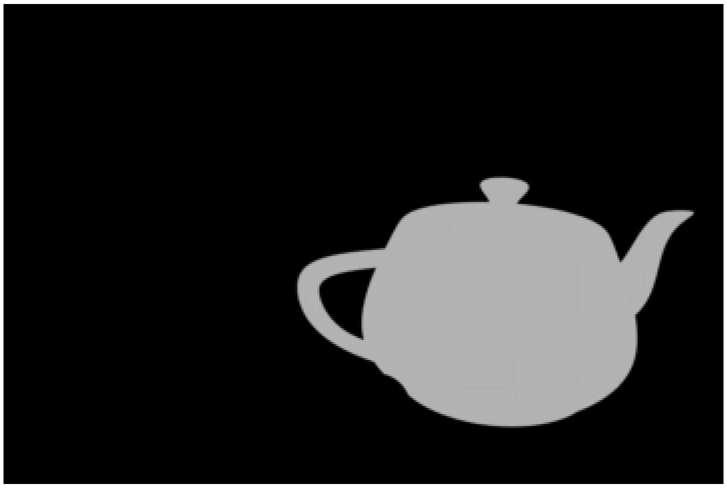
The **Subsurface Amount** output renders a grayscale image that represents the amount of subsurface scattering on surfaces within a scene. Black for 0% subsurface scattering, ramping toward white at 100%.

Subsurface Color



The **Subsurface Color** output renders a scene as full color image that represents the Subsurface coloring of a scene independent of illumination or shading.

Transparent Amount



The 'Transparent Amount' output renders a grayscale image that represents the amount of transparency on surfaces within a scene. Black for 0% Transparency ramping toward white at 100%.

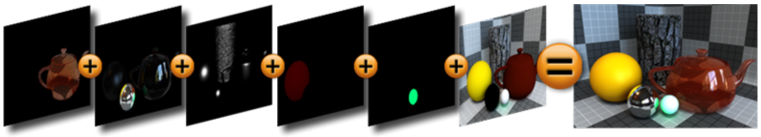
Transparent Color

The **Transparent Color** output renders a scene as a full-color image that represents the transparent coloring of a scene independent of illumination or shading.

Compositing Outputs

Render outputs are incredibly useful when you wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aid in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is which render outputs combine to equal the final color output. The answer is Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color, where each successive layer is added to the layer below in the composition (also called linear dodge). Rendering each layer at a gamma of 1.0 and then combining them and adding the gamma back to the fully composed layers produce the best results.

You may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to **Effect** > **Alpha** (or simply adding them to existing Shader Tree masks). Then each time a render command is invoked, all active render outputs get generated simultaneously.



Particle Render Outputs

This topic describes the different **Particle** render outputs.

Particle Age

The **Particle Age** output assigns a gray-shaded value to each particle based on its age (in seconds), starting at black and moving toward white. Values are stored as floating point numbers, so an extended dynamic range format is necessary to save this output.

Particle ID

The **Particle ID** output assigns a random floating point ID number to each particle, which is represented by a gray shade from 0 to 1, black to white. Values are stored as floating point numbers, so an extended dynamic range format is necessary to save this output.

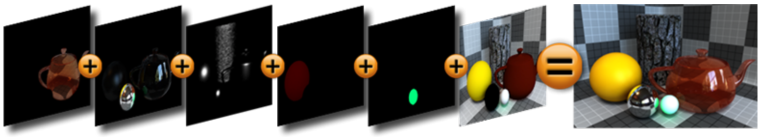
Particle Velocity

The **Particle Velocity** encodes the velocity value to a floating point number, representative of the speed the particle is traveling at the current frame. The floating point values are represented as a color in the image that extends beyond the common range of color values, so an extended dynamic range format is necessary to save this output.

Compositing Outputs

Render outputs are incredibly useful when you wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aid in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is which render outputs combine to equal the final color output. The answer is Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color, where each successive layer is added to the layer below in the composition (also called linear dodge). Rendering each layer at a gamma of 1.0 and then combining them and adding the gamma back to the fully composed layers produce the best results.

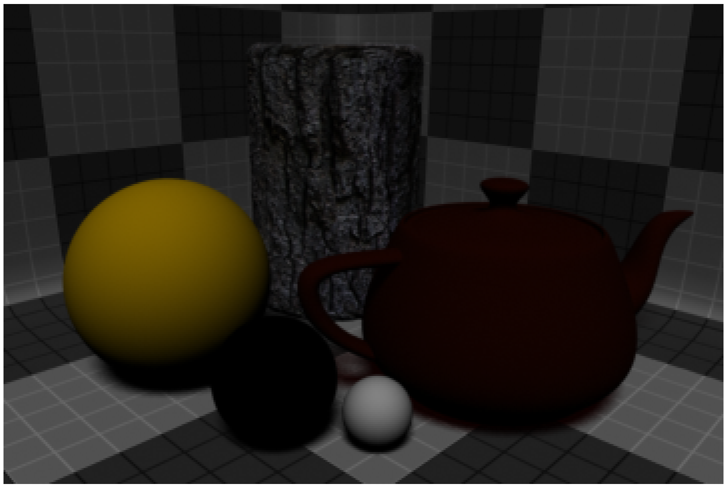
You may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to **Effect** > **Alpha** (or simply adding them to existing Shader Tree masks). Then each time a render command is invoked, all active render outputs get generated simultaneously.



Shading Render Outputs

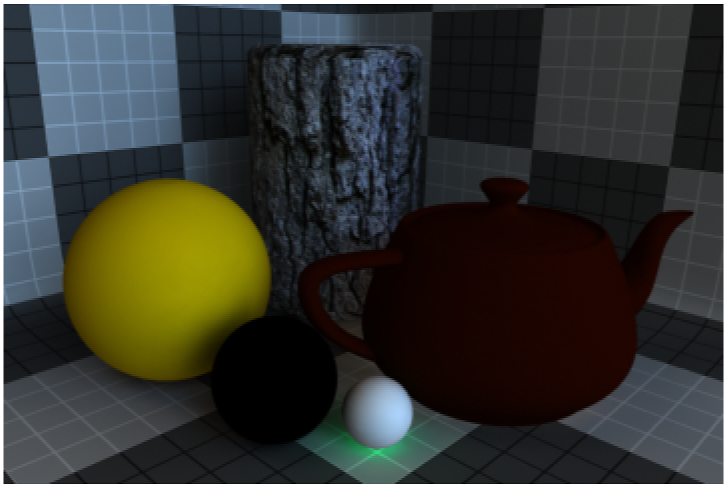
This topic describes the different **Shading** render outputs.

Diffuse Shading (Direct)



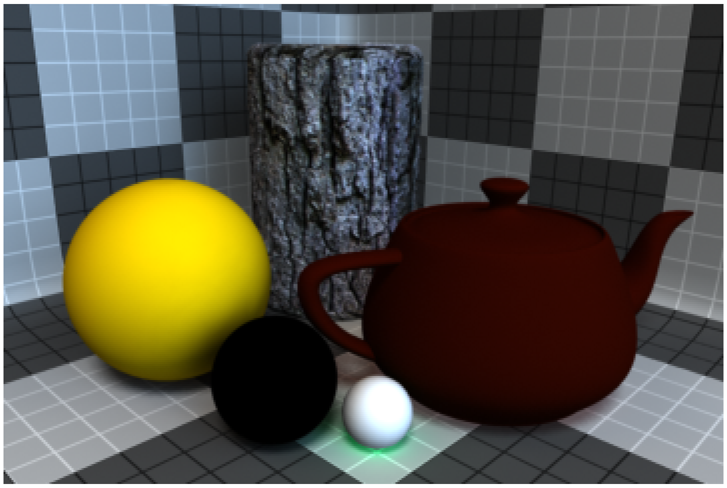
The **Diffuse Shading (Direct)** output renders a channel of the diffuse color and shading of all objects in a scene by all direct light sources, independent of other surfacing attributes. Basically, it is a combination of diffuse coefficient and direct illumination.

Diffuse Shading (Indirect)



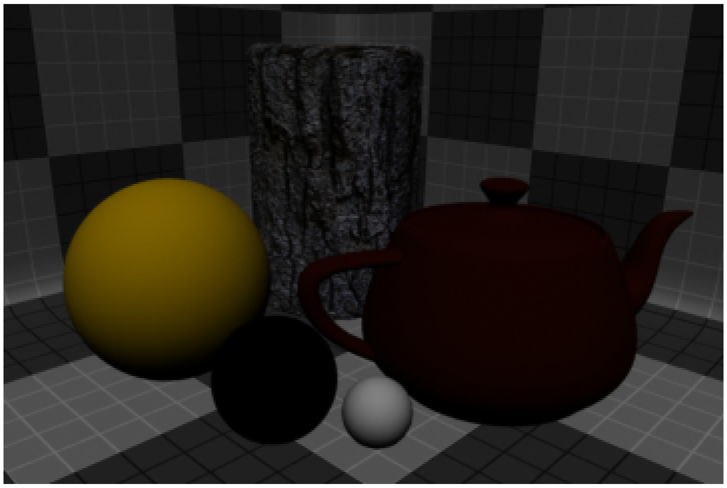
The **Diffuse Shading (Indirect)** output renders a channel of the diffuse shading of all objects in a scene by all indirect light sources, independent of other surfacing attributes. Basically, it is a combination of diffuse coefficient and indirect illumination.

Diffuse Shading (Total)



The **Diffuse Shading (Total)** output renders a channel of the diffuse color and shading of all objects in a scene by all light sources, independent of other surfacing attributes.

Diffuse Shading (Unshadowed)

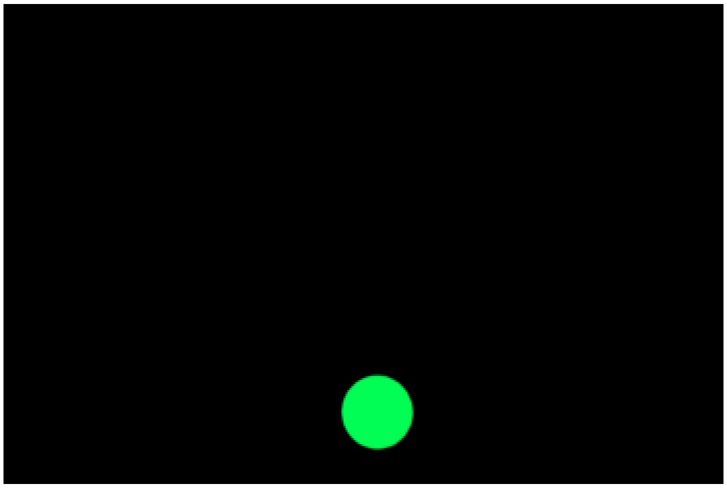


The **Diffuse Shading (Unshadowed)** output renders a channel of the diffuse color and shading of all objects in a scene by all light sources, independent of other surfacing attributes, while excluding shadows cast by direct and indirect light sources.

Light Path Expression

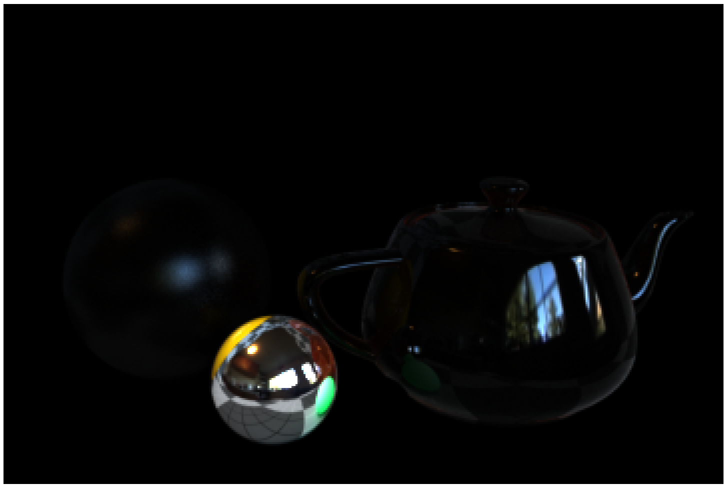
The Light Path Expression output populates a series of channels containing information on each light in your scene. At render time, each individual light path expression generated by this output shows up as individual outputs in the Render window. See Rendering with mPath.

Luminous Shading



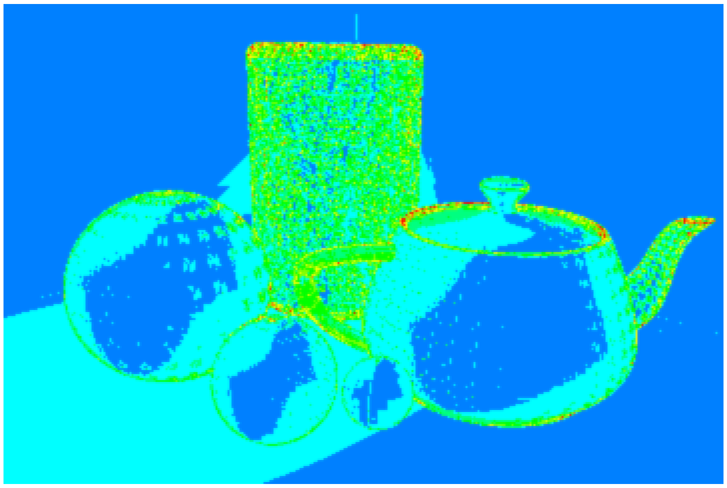
The **Luminous Shading** output generates a channel of all the luminous shading within a scene, independent of all other attributes.

Reflection Shading



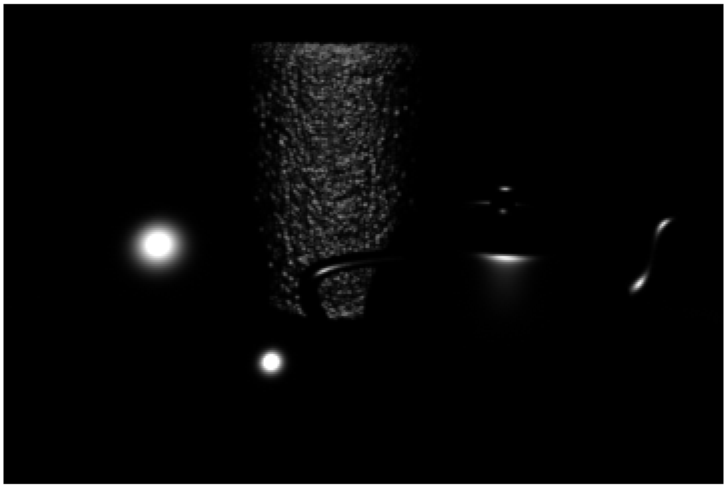
The **Reflection Shading** output generates a channel of all the reflection calculated within a scene, independent of all other attributes.

Shaded AA Samples



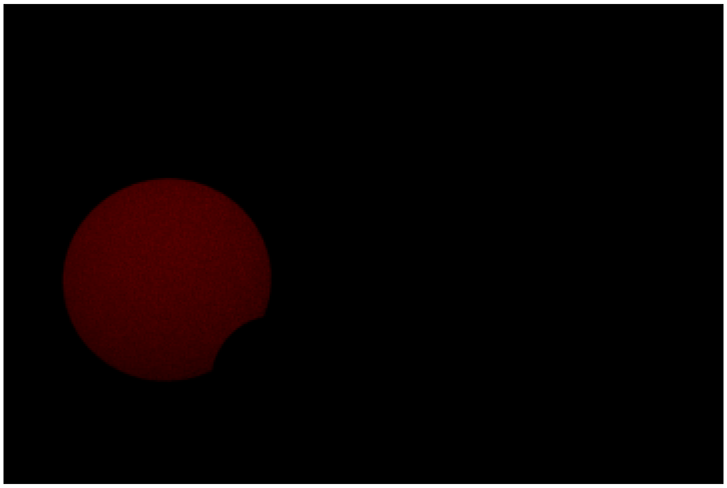
The **Shaded AA Samples** output shows the fraction of anti-aliasing samples in each pixel that was actually shaded, using a color scale ranging from blue (one sample shaded) to red (all samples shaded). This approximates the shading cost per pixel and can be used to diagnose issues related to shading rates and the refinement threshold.

Specular Shading



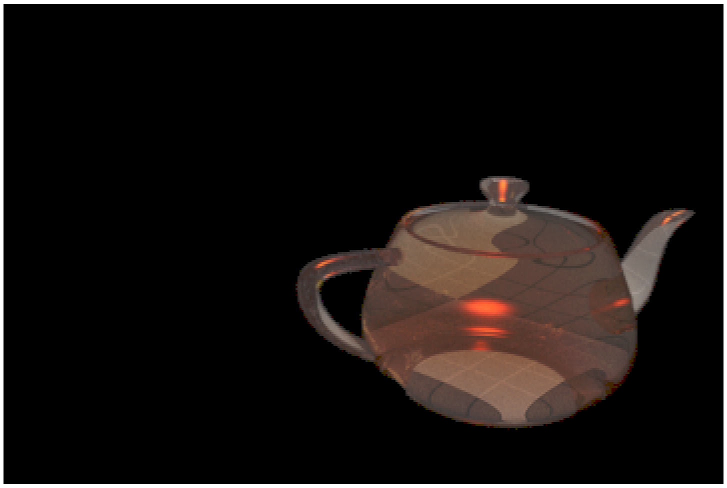
The **Specular Shading** output generates a channel of all the specular shading within a scene, independent of all other attributes.

Subsurface Shading



The **Subsurface Shading** output generates a channel of all the subsurface scattering shading within a scene, independent of all other attributes.

Transparent Shading



The **Transparent Shading** output generates a channel of all the transparent shading within a scene, independent of all other attributes.

Variance

A Variance render output of a solid sphere sat inside a sphere with 90% **Transparent Amount** and 0.9 **Refractive Index**.

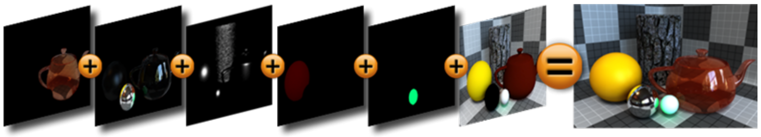
A render output that visually displays Variance in the render.

Note: The **Variance** render output only functions when the **mPath** renderer is selected.

Compositing Outputs

Render outputs are incredibly useful when you wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aid in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is which render outputs combine to equal the final color output. The answer is Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color, where each successive layer is added to the layer below in the composition (also called linear dodge). Rendering each layer at a gamma of 1.0 and then combining them and adding the gamma back to the fully composed layers produce the best results.

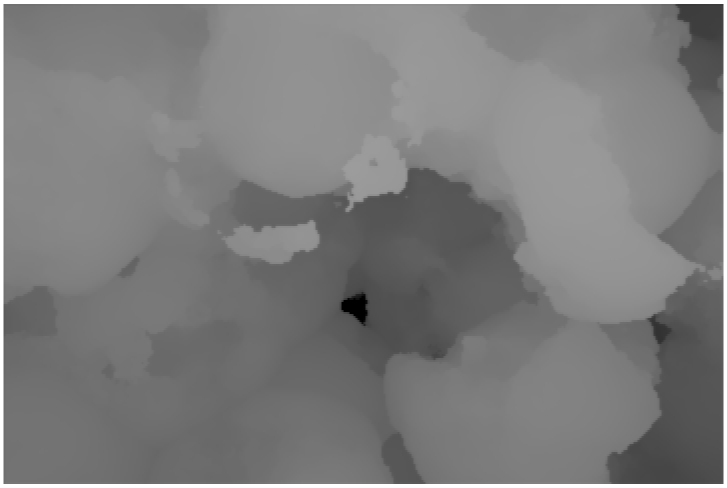
You may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to **Effect** > **Alpha** (or simply adding them to existing Shader Tree masks). Then each time a render command is invoked, all active render outputs get generated simultaneously.



Volume Render Outputs

Learn about the different **Volume** render outputs.

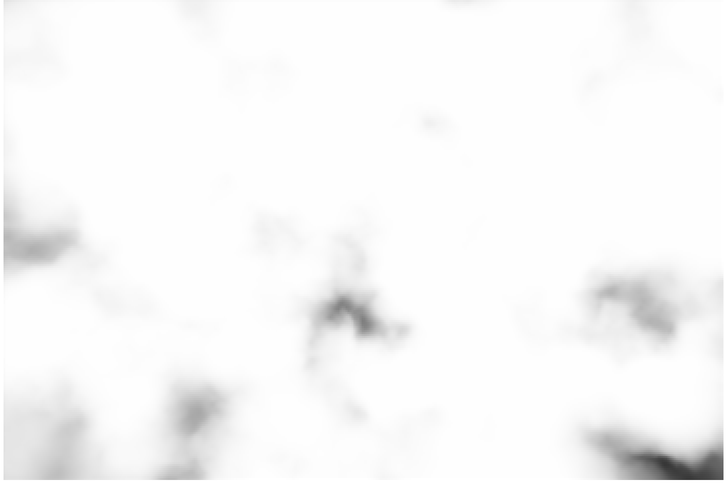
Volumetric Depth



Volumetric items are not evaluated for all render outputs items, such as **Depth**, so the **Volumetric Depth** option allows you to calculate depth values for volumetric items in the scene.

Note: Each pixel in the volume can only have a single depth value, so opacity is not considered for this type of output.

Volumetric Opacity



The **Volumetric Opacity** output evaluates the transparency of the volumetric elements in the scene, allowing you to separate the opacity of the volume independent of the rest of the scene's elements for compositing.

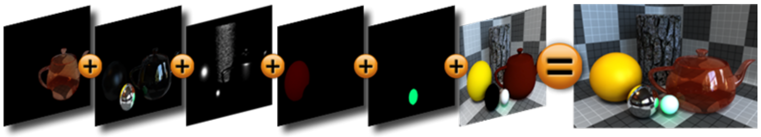
Volumetric Scattering

The **Volumetric Scattering** render output renders only the volumetric elements of a scene, including those generated by sprites, volumes and volumetric lights.

Compositing Outputs

Render outputs are incredibly useful when you wish to compose the generated layers individually in an external application, such as Photoshop, After Effects, Nuke or Fusion. This can aid in tweaking color, among other settings as a post process without requiring additional render time. The question most often asked is which render outputs combine to equal the final color output. The answer is Diffuse Shading (Total) + Transparent Shading + Reflection Shading + Specular Shading + Subsurface Shading + Luminous Shading = Final Color, where each successive layer is added to the layer below in the composition (also called linear dodge). Rendering each layer at a gamma of 1.0 and then combining them and adding the gamma back to the fully composed layers produce the best results.

You may also render out simultaneous alpha channels for each item in a scene by creating item masks for each and placing a render output set to **Effect** > **Alpha** (or simply adding them to existing Shader Tree masks). Then each time a render command is invoked, all active render outputs get generated simultaneously.



Render Commands

Rendering is the process of converting the polygons and lights in a scene, combined with the settings of the Shader Tree to produce a two dimensional bitmap image. Generated from the perspective of the render Camera Items, it is akin to taking a photograph, converting the three-dimensional scene before the lens into the still image on the camera's sensor.

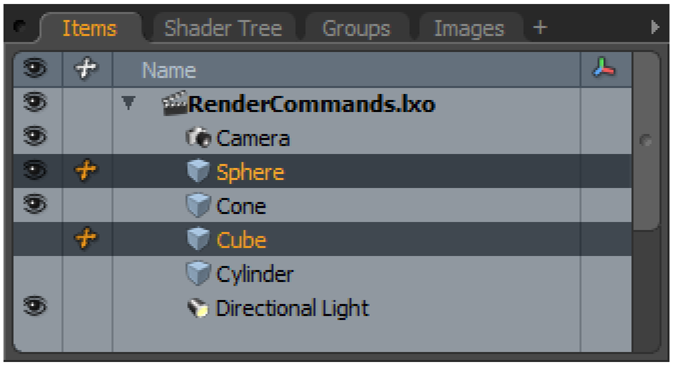
Rendering the scene is like pressing the shutter button where the settings of the current frame are evaluated and displayed as an image in the Render Display where it can be further manipulated or saved.

While the 3D GL viewports offer a decent likeness of how a scene looks, including animation and deformers, it is still only an approximation and many advanced or render intensive options are excluded completely.

Rendering shouldn't be thought of as the final step in a linear workflow of modeling, texturing and rendering either. Throughout the process, you can render and preview your scene iteratively, ensuring the proper final outcome. However, completely rendering the entire scene continuously would slow down the creative process, therefore Modo has a variety of commands to facilitate various workflows, found in the menu bar under the **Render** menu. Each is intended to give you greater control over the rendering process, ultimately providing a faster, more streamlined workflow. The various commands are especially useful in expediting the preview of large scenes, when combined with the **Auto Visibility** settings of the Items (Scenes) List. The result of each command is displayed in the Render Display window, which automatically opens each time a rendering command is invoked (if not currently open).

Note: Images in the **Render Display** are cached upon render completion, but are not saved to the system. Please reference the Saving Images page of the documentation for more on how to save rendered images, see Saving Images.

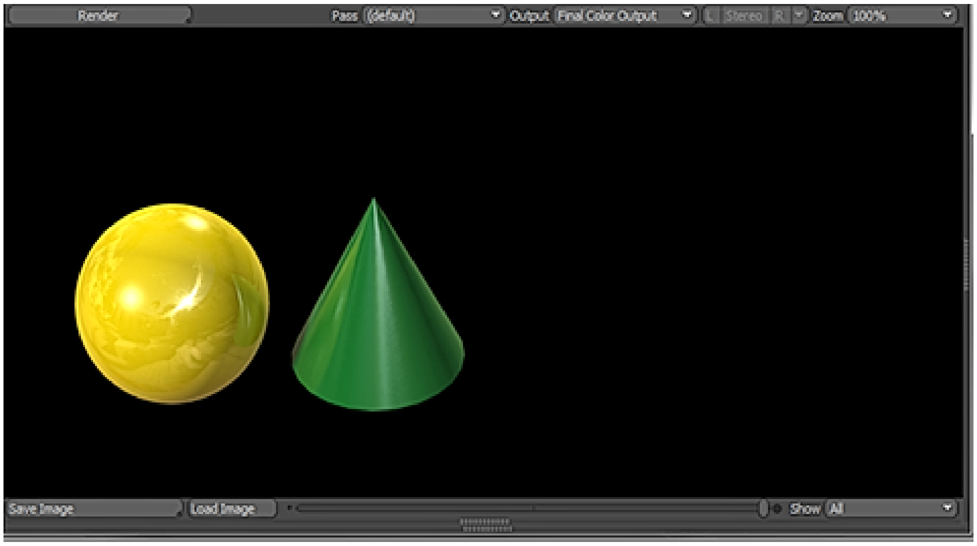
The various commands play off the visibility of items in the Items (Scenes) List (the eye '👁' column) or as defined in the Display Viewport, which can be set to override **Item List** settings. For instance, in the example below, there are four simple items, a sphere, a cone, a cube and a cylinder. The sphere is visible and selected, the cone is visible, but not selected and the cube is not visible but selected, and the cylinder is neither visible nor selected; one of each possible states.


Items
Shader Tree
Groups
Images
+
Name

RenderCommands.lxo
Camera
Sphere
Cone
Cube
Cylinder
Directional Light

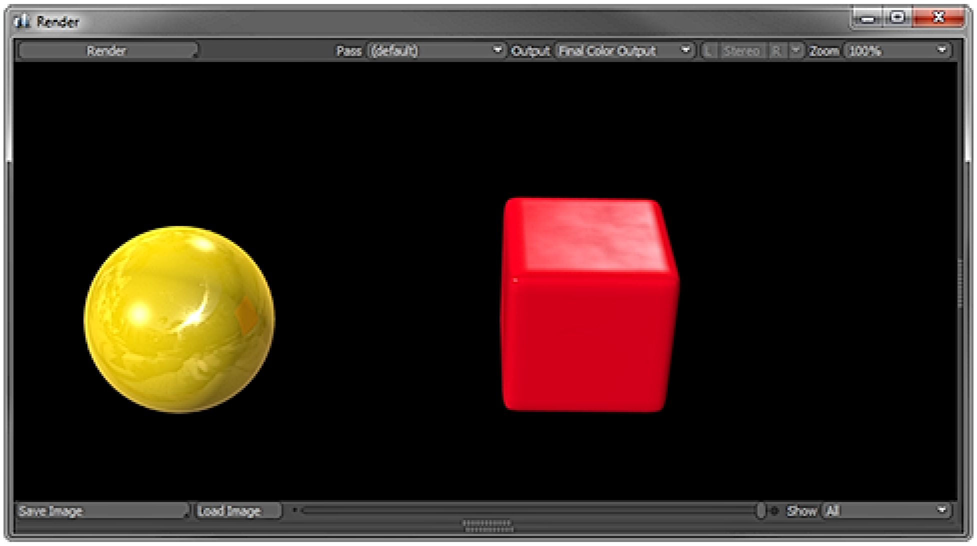
Render

The **Render** command is the most common of the rendering commands, it can be invoked when pressing **F9** on the keyboard. This option respects the visibility column of the **Item List**. Visible Items are rendered regardless of selection, items not visible are not rendered. From the above **Item List**, only the sphere and cone are visible, therefore the only items in the **Render Display** window.


Render
Pass
(default)
Output
Final Color Output
L
Stereo
R
Zoom
100%
Save Image
Load Image
Show
All

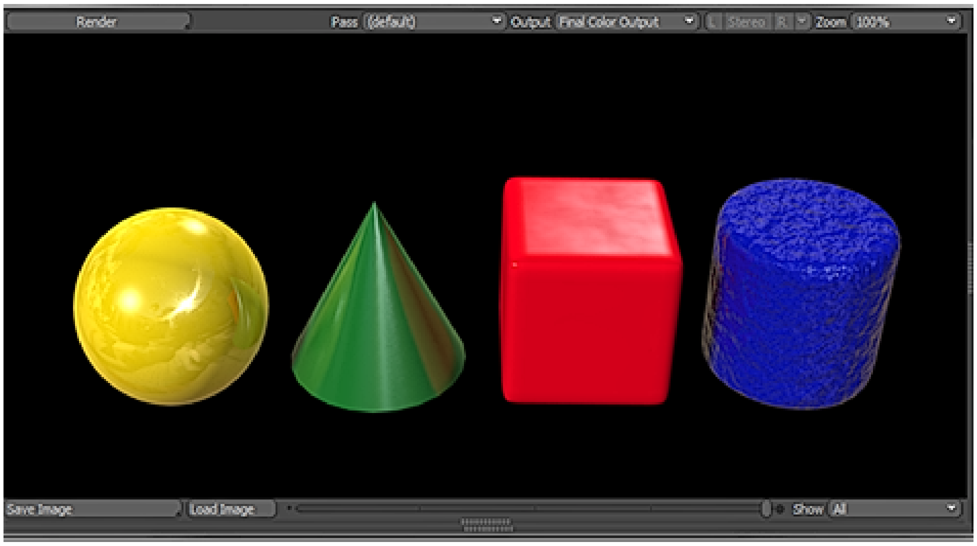
Render Selected

The **Render Selected** command only renders selected items that is, items that are highlighted in the **Items** list, regardless of their visibility. Therefore, the sphere and cube are the only items in the **Render Display**, according to the **Item List** above.


Render
Render
Pass
(default)
Output
Final Color Output
Stereo
R
Zoom
100%
Save Image
Load Image
Show
All

Render All

The **Render All** command renders all items in the **Item List** regardless of visibility, therefore all four items in the list are rendered when using this command.

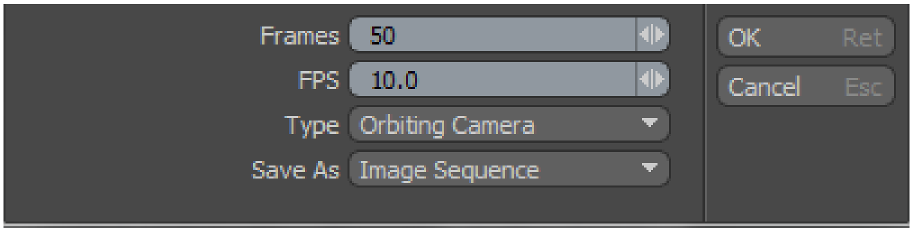

Render
Pass
(default)
Output
Final Color Output
Stereo
R
Zoom
100%
Save Image
Load Image
Show
All

Render Current View

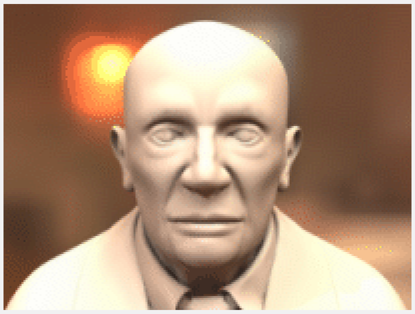
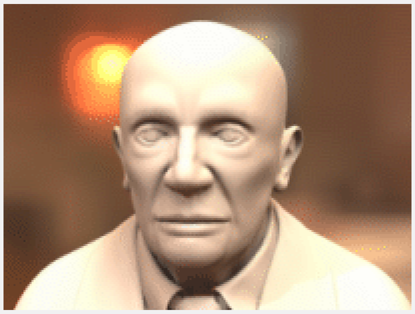
The 3D OpenGL viewports also respect visibility of the **Item List**, although the display style of background (unselected) items is controlled differently. If you need to render from the unique perspective of a 3D OpenGL viewport (such as any perspective or orthographic views), you can utilize the **Render Current View** command, which temporarily moves the current **Render Camera** to match the position of the viewport specific camera, rendering the scene, however, the focal length and aspect ratio cannot be matched precisely, producing a rendered image that is an approximation of the viewports view. Although not exact, this function gives you the ability to move freely through a scene, previewing items without moving the camera item itself. If you wish to move the camera item to match the view of a viewport, use the **Sync To View** command. You can also enable the **RayGL** option per viewport to get a preview rendered representation directly in the viewport.

Render Turntable

The **Render Turntable** command provides you a simple means to generate 360° turntable animations without the need to keyframe any animations. When you choose the command, the following dialog opens.



Frames 50
FPS 10.0
Type Orbiting Camera
Save As Image Sequence

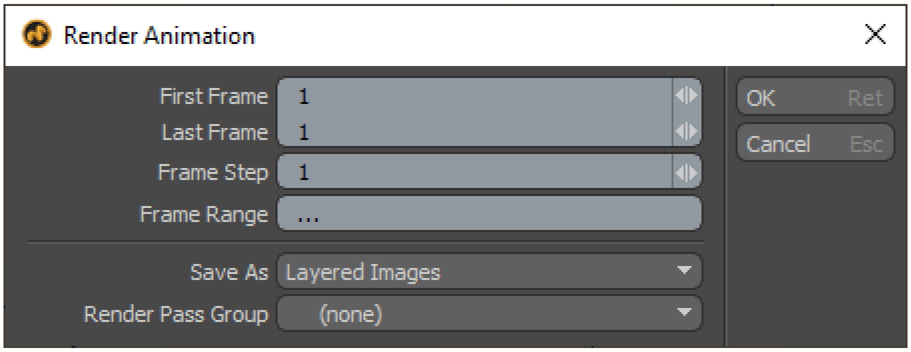
Option	Description
Frames	Sets the total length of the rendered animation in frames.
FPS	Frames Per Second defines how fast or slow the animation plays back if you choose to render the turntable as a video file.
Type	There are two options for the turntable type; **Orbiting Camera** rotates the camera item around the origin and **Spinning Geometry** spins the geometry itself around the origin, lights and other elements remain still. The main different between the two is in how light and shadow play over the surface as it rotates.  Orbiting Camera  Spinning Geometry

Option	Description
Save As	Determines how the rendered turntable is saved to disc. There are three options: **Image Sequence**, **Movie** and **Layered Images**. **Image Sequence** saves a series of numbered frames, **Movie** renders out a platform appropriate video file. The **Layered Images** option renders out a sequence of frames, saving all the different render outputs of the Shader Tree within a single layered file. **Note:** File formats supported by Modo are fully covered on the Saving Images page of the documentation.

Note: Before generating a turntable, you need to position the camera, framing the image as you wish. This is the base position of the camera. Make sure the geometry is centered at the origin, as this is the center of rotation for both the **Orbiting Camera** and **Spinning Geometry** types of turntables.

Render Animation

The **Render Animation** command instructs Modo to render out a sequence of frames, by default, determined by the **Scene Range** values. You can override this range by adjusting the values directly in the **Render Animation** dialog. You can then choose how the sequence of frames is rendered and saved to disc. The **Render Animation** command defaults to saving files to the name and location defined in the render output, if specified, asking if you would like to override or accept the designations. When no filename is specified, Modo opens an OS-specific dialog, prompting you to choose the filename and save location.


Render Animation

First Frame 1
Last Frame 1
Frame Step 1
Frame Range ...
Save As Layered Images
Render Pass Group (none)

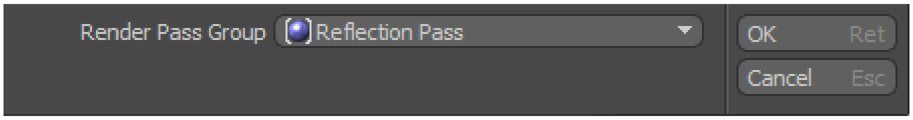
Option	Description
First/Last Frame	Determines the sequence of frames rendered.
Frame Step	Determines the progression of rendered frames, useful in skipping frames to speed up a render. A **Frame Step** of 1 renders every frame, a Frame Step of 2 renders every other frame and so on. To play back a sequence at the correct speed, divide the real FPS (frames per second) by the **Frame Step** value to get the new FPS.
Save As	Modo provides several options for saving the image sequences. • **Image Sequence** - Each rendered frame is saved individually, with the frame number appended to the filename as well as **L** or **R** for stereo pairs, if applicable. • **Movie** - Frames are rendered as a single animation file. Format option are dependent on platform. This option is also necessary to save still images as stereo pairs when rendering stereoscopic sequences. • **Layered Images** - This option acts exactly like the **Image Sequence** option, however, each active render output is combined into a layered file, which can be very useful in reducing clutter when compositing images in an external application, reducing the number of files necessary to import. **Note:** Check to make sure your target application can accept layered files before proceeding with this option.
Render Pass Group	Determines the pass group to render. Pass groups are separate iterations of a single scene that can be rendered sequentially. You can define as many passes as you like to a pass group. When that group is specified, each iteration is rendered and saved according to the settings of the Render Output Items, with naming that conforms to the settings of the Frame **Output Pattern** option. For more information on creating and working with Render Passes, see Render Passes.

Render Passes

Passes are collections of setting that when rendered produce multiple variations of an scene from the same camera. The **Render Passes** command facilitates the rendering of a pass group when rendering still images; as the standard **Render** commands only render the currently selected pass.

Note: For more information on Render Passes, see Render Passes.

Selecting the **Render Passes** command opens the **Render Passes** dialog where you can select the target pass group and all the passes within the selected group render consecutively and display in the **Render Display** viewport.


Render Pass Group: Reflection Pass

Open Preview

This command opens a new window that contains a Preview Viewport, which uses the Modo render engine in a progressive refinement mode, allowing you to see all changes to your scene as a fully rendered image. Within the default Modo interface, a **Preview** viewport is embedded under the **Render** tab, providing you a fast, interactive means of determining lighting and surfacing settings, albeit some throttling to the quality settings. This command calls the same display technology, but from within anywhere in Modo. You can also open a **Preview Render** window by pressing the **F8** keyboard shortcut.

Open Last Render

This command opens the Render Display window, and displays the last successful full render without the need to invoke any of the render commands.

Add Per-Light Render Outputs

This command adds a **Light Path Expressions Output** item to the **Shader Tree** that allows users to enter Light Path Expressions character strings. These Light Path Expressions can be used to restrict which kinds of light paths contribute radiance to a particular render output image. See Rendering with mPath for more information on how to add Light Path Expressions.

Render Passes

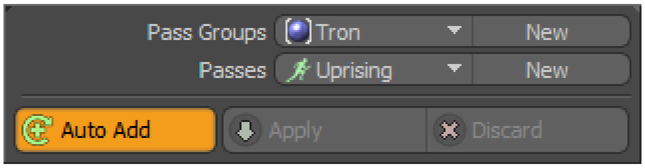
Render passes offer you a way to control any aspect of a scene that is, in-turn used to automatically produce multiple rendered images. Each rendered image reflects all the settings of an individual pass. In their most basic sense, Render Passes are simply containers for channel values. Once you add a pass, subsequent edits to the scene are stored within the current pass or passes.

When the **Render Passes** command is invoked, each particular pass of the pass group is rendered in succession and saved as specified. This way of working is simple, yet incredibly powerful, providing you a means to create rendered variations that were previously very difficult, if not impossible to do within a single scene.

Understanding Passes

Passes conceptually work as containers that only hold channel values. A channel is simply any attribute of an item that can be animated. For more information about channels, see the Channels Viewport topic of the documentation. Once these channel values are created, they sit in their pass container as a layer above the scene itself, called the **Setup** layer. When a pass is selected, any values held in the pass replace the ones of the **Setup** layer below it. For example, a cylinder could be colored red at the **Setup** level, and a pass could be created that changes the color to orange. Disabling the pass would return the color to red. If the passes were rendered, there would be two resulting images, one with a red cylinder and another with an orange one. The layering of the channel values is an important aspect to understand, because there is also a layer in-between a pass and a scene, called the **Edit** layer, explained further on.

Basic Pass Workflow

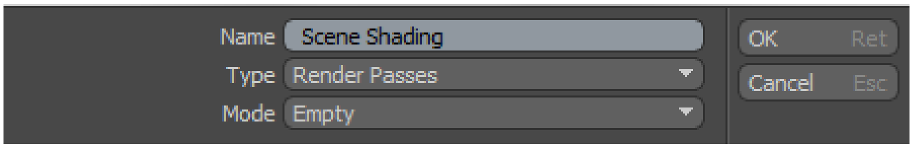

Pass Groups: Tron

Passes: Uprising

The **Render Passes** UI is found in the **Render** interface tab. It is controlled mainly by the buttons above the default **Properties** viewport. The workflow for passes moves top to bottom, first with the **Pass Groups** field with a button to select the group, and a button to create a new group. Next, the **Passes** field, also with a selector and a **New** button. Next are the **Auto Add**, **Apply** and **Discard** buttons that further control pass creation.

Before creating any passes, you need to first create a pass group. Pass groups are collections of different passes, meant strictly as an organizational device. You cannot create individual passes until a pass group is defined and selected.

1. To create a group, click the **New** button and define a name in the pop-up dialog.

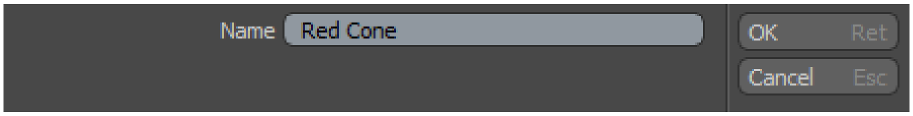

Name Scene Shading

Type Render Passes

Mode Empty

2. Click **OK** to accept all other defaults.

   When you create the pass group, it is automatically selected and any passes created are automatically added to that group.
3. The next step is to create the pass itself. Click the **New** button to the right of **Passes** to open the **Group Layer** dialog, where you can name the pass.
4. Once a name is entered, click **OK** to define the pass.


Name Red Cone

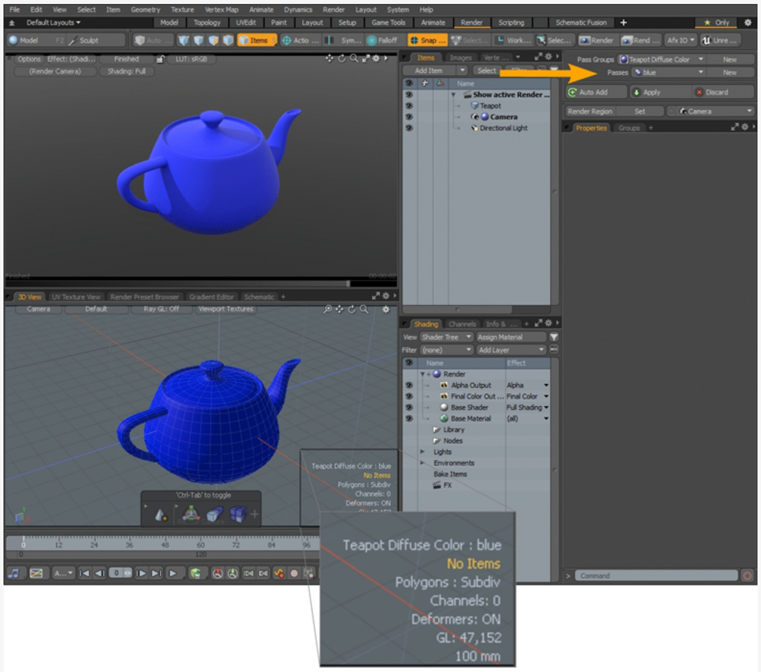
5. Next, edit the channel values.

   With **Auto Add** enabled (the default state), any changes made to any attribute automatically adds that attribute channel to the currently selected pass. This means that you can simply edit the scene and any modifications are automatically stored/saved in the pass. Disabling the pass, by returning the pass selection to **(none)** returns all the values to the underlying scene state.

Note: With the addition of passes to a scene, complexity can increase exponentially, therefore it is important to maintain a well-organized scene, especially in relation to the **Shader Tree**.

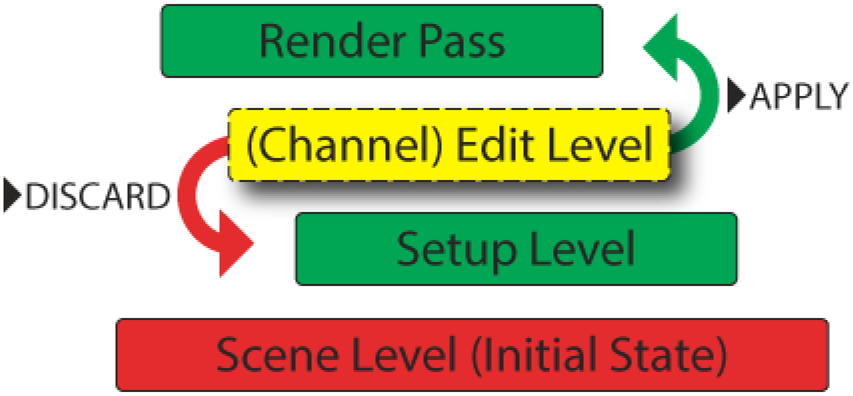
Tip: To quickly identify which render pass you are currently working on, the **3D Information Overlays**, at the bottom right corner of the 3D Viewport, displays the name of the selected **Pass Groups** item and the selected **Passes** item.

Click on the image below to view an animation.



Applying and Discarding

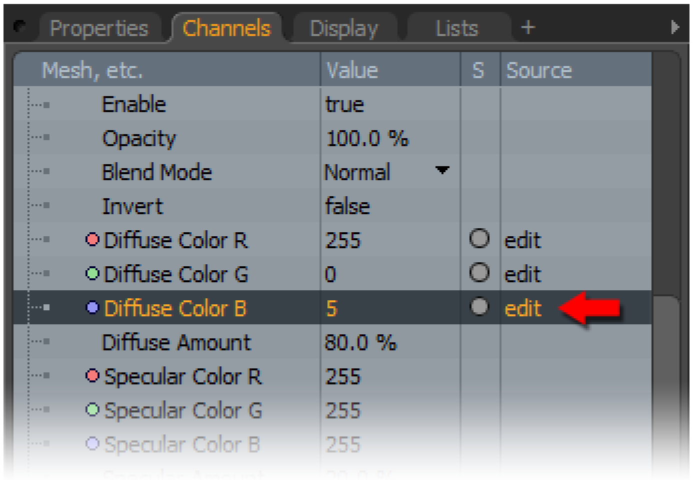
Modo's pass system conceptually works in layers, you have the bottom base layer called the **Scene** layer, this is the initial default state of any item as it is added to the scene, including Mesh Items, cameras, lights, and so on. The next layer is the most important, it is the **Setup** layer, all attributes from any item in the scene as they are modified are stored at this level. Each named pass exists as a layer above the **Setup** layer. For any channel values stored in the pass, when that pass is enabled, simply by selecting it, the stored values replace those of the **Setup** layer values as long as the pass is active. However, when you select a pass and begin to make changes, a new layer is inserted between the current **Pass** and **Setup** layers, called the **Edit** layer. This is where the 'Apply' and 'Discard' buttons come in to play. When a modification is made to a scene with a pass active, the values don't go directly into the pass. They can be pushed into the pass by either exiting the pass, or clicking the **Apply** button. Clicking the **Apply** button allows you to continue making modifications, clicking **Discard** returns the values to either their previous state when the Pass was entered or when Apply was last pressed.



Advanced Render Passes

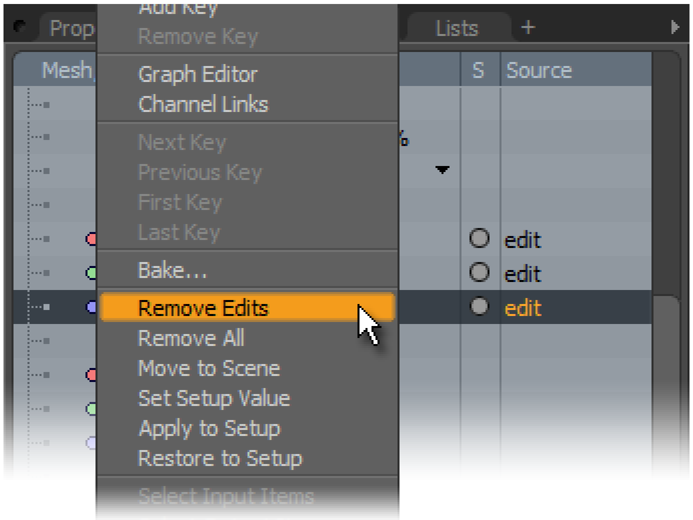
For successfully using Modo's Render Passes you don't need to be aware of what is going on in the background. Using the **Auto Add** function and then making normals scene modifications, Modo automatically takes care of all the technical details, as long as the proper passes are selected when attributes are modified. However, if something does go wrong, a basic understanding of the mechanics of it can be helpful in resolving any issues that may arise.

Editing scene attributes for passes can be a highly controlled manual process or completely automatic,depending on your preference. This is controlled by the **Auto Add** button, which is enabled by default. In order for a channel value to be stored in the pass container, the channel itself needs to be stored in the pass. The **Auto Add** button places the channel of any modified attributes in the pass. If you disable the **Auto Add** function, then you need to add the channels you wish to modify manually. This is done in the **Groups** viewport. Passes are actually a function of the Groups Viewport, used to store the collections of channels and their resulting values. Anyone familiar with groups understands why the dialogs that pop up are called **Create Group** and **Group Layer**. The channels can be added by selecting them in the **Channels** viewport and dragging and dropping them into the pass group in the **Groups** viewport, or by using the **Add Items** function found in the **Groups** viewport itself. For more information on working with groups, see the Groups Viewport


Properties
Channels
Display
Lists
+

Mesh, etc.	Value	S	Source
Enable	true		
Opacity	100.0 %		
Blend Mode	Normal		
Invert	false		
Diffuse Color R	255	○	edit
Diffuse Color G	0	○	edit
Diffuse Color B	5	●	edit
Diffuse Amount	80.0 %		
Specular Color R	255		
Specular Color G	255		
Specular Color B	255		

There is also some additional information provided regarding pass states in the Channels Viewport viewport. Down the right-hand side of the viewport is a **Source** column. When any pass is selected and channel values are modified, the source column shows **edit**, meaning that the current value is in the **Edit** layer, it has not yet become part of the pass, and has been modified from the **Setup** level. When you apply those changes, pushing the values into the pass, the **Source** column displays the name of the pass. When the pass is disabled, the name reverts to **setup**, letting you know that the current value for that channel is the base level. This can be very helpful in instances where values were mistakenly applied in a pass that were meant for the scene, or vice-versa. By right-clicking on the channel to open the contextual menu, a number of options are provided to give you some ability to push those values around between the different layers.


Prop
Lists
+
Mesh
S
Source
Add Key
Remove Key
Graph Editor
Channel Links
Next Key
Previous Key
First Key
Last Key
Bake...
Remove Edits
Remove All
Move to Scene
Set Setup Value
Apply to Setup
Restore to Setup
Select Input Items
edit
edit
edit

Remove Edits: This option reverts the values to the last saved state, such as if a Render Pass was selected and then several edits were applied, prior to exiting the pass or pressing **Apply**, you can use the **Remove Edits** command, returning the values to the initial state they were in when the pass was selected. This is the same function as the **Discard** command.

Remove All: This command removes all edits to the target channel, reverting the values back to the initial default (**Scene**) state.

Move to Scene: Takes the current **Edit** state value and pushes it to the **Scene** level.

Set Setup Value: Sets current channel values as the base **Setup** value and pushes value to the Render Pass.

Apply to Setup: Moves the current edit value to the **Setup** layer, leaving any selected pass values unchanged.

Restore to Setup: Reverts the current pass value to that of the **Setup**.

Adding New Items

An important detail to understand is that passes cannot contain items, only channels, therefore, any item that is added is placed at the **Scene** level and subsequently becomes a part of all existing passes. If an item is added to the scene, it needs to be added at the **Scene** level. If you accidentally add an item while a pass is active, the Channels Viewport offers functionality to move data in and out of the **Setup** level of the scene, so your work doesn't get lost.

Rendering Passes

Once the various passes have been created, in order to actually render the passes, you only need to use the **Render Passes** command found in the menu bar, under **Render**. You can select a pass group and all of the contained passes are rendered in succession and cached into the **Render Display** window. You can use the **Save As** function to save the image to your local hard drive. To automatically save images when rendered, use the **Render Animation** command and determine the pass group for rendering in the dialog where the files get saved according to the settings of the render outputs of each pass. File names conform to the settings of the render item's **Output Pattern**. Additionally, passes within a pass group may be enabled and disabled temporarily, controlling whether they render or not. You can disable a pass within a pass group in the **Groups** viewport by clicking the small camera icon in the **Render State** column of that viewport. Click the empty column slot to enable rendering for the pass.

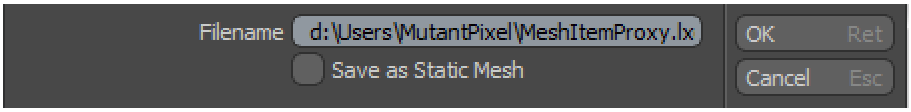
Render Proxy

The render proxy function allows you to completely remove one or more Mesh Item layers within a scene, offloading them to an external file that isn't loaded again until render time. The saved layers are replaced within the master scene by a simple bounding box, or if you need an alternate version of the saved file used just for display.

This is meant to facilitate two workflow processes. The replacement of high resolution meshes by a bounding box in the scene file produces a lighter file, making for smoother navigation and faster file save times. It also allows additional work to be applied to the saved proxy file externally. When the modified file is loaded at render time, any updates saved to the proxies are reflected in the final rendered image.

Usage

The recommended workflow for creating proxy items is that they originate in a given scene as a regular Mesh Item layer. You can convert the Mesh Item to a render proxy by right-clicking on the target item layer or layers in the Items (Scenes) List, opening the contextual menu. Select the **Convert to Proxy** command to open an OS-specific dialog and specify the Proxy file location and name. After clicking **Save**, the following dialog opens.


Filename d:\Users\MutantPixel\MeshItemProxy.lx
☐ Save as Static Mesh

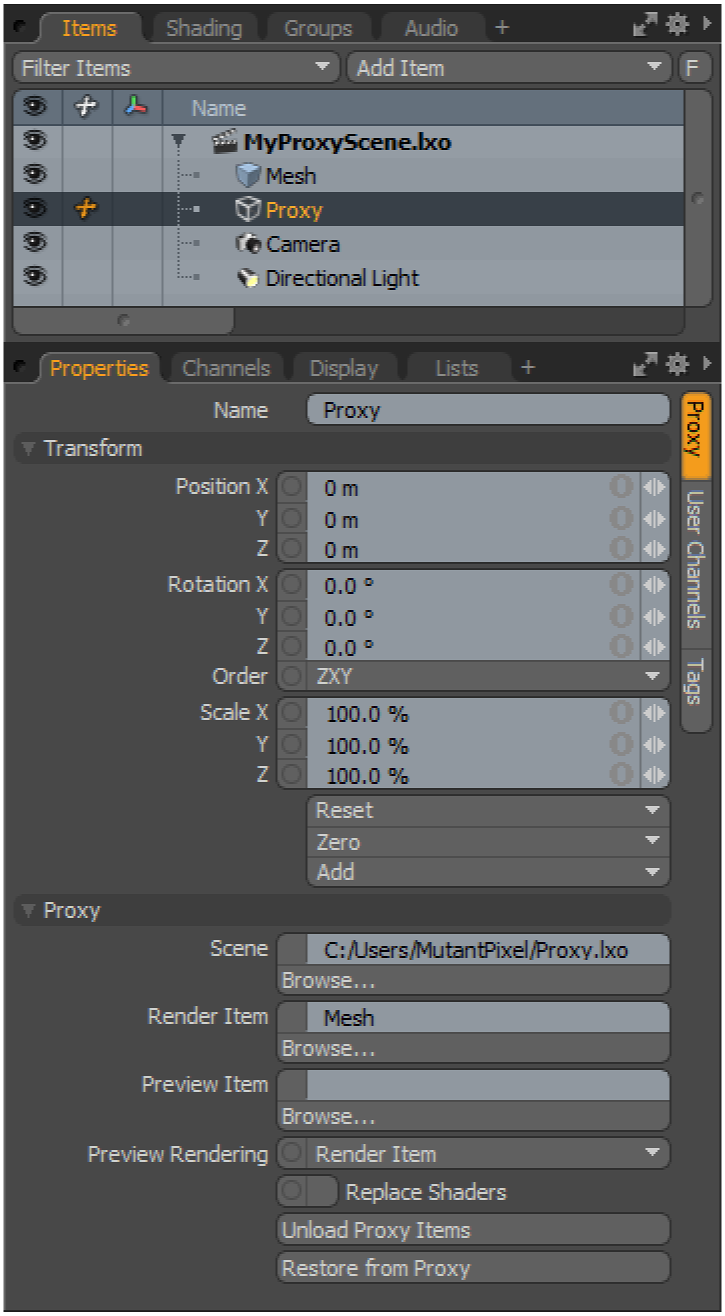
The **Filename** option always references the location defined in the previous OS dialog. There is also the additional option to convert the Proxy item to a Static Mesh Item on exporting. Generally, keep this option disabled, as converting a mesh to a static mesh is destructive. A static mesh cannot be directly edited, nor deformed. However it can be a good option to enable when it isn't necessary to edit the externalized file any further, because it creates a smaller file size. Click **OK** to write the Mesh Item out as an external **.lxo** scene file that contains the layers that were selected when the command was invoked. Each of the resulting layers are now represented by just the bounding box. When creating proxies, the resulting items inherit any transform values or keyframed animation that the original mesh had, keeping them in the master scene, but deformation-based animations are not applied, as the mesh is no longer part of the scene. Selecting the bounding box, the proxy item in the **Items** list provides a number of options in the **Properties** viewport panel.

Render Proxy Workflow Considerations

Once a layer is converted to a render proxy in a scene, the original geometry layers are only loaded at render time, the geometry itself is no longer part of the master scene. This excludes any associated Vertex Maps, and other relevant associated data is not accessible in the master scene. Because of this, render proxies cannot be deformed using any of Modo's deformers since the geometry isn't actually part of the scene until render time. Transform animations, such as move, scale and rotate, when applied to the proxy layer, render fine.

Due to the global nature of shading in Modo, when a proxy item is created, its related Shader Tree layers remain in the master scene, as they could still be a part of another surface. Nevertheless, further editing of these layers has no effect on the shading of the proxy item, only modified settings within the proxy file itself are reflected in the rendered image, unless the **Replace Shaders** option is enabled, acting as an override. If there is no need to further edit the proxy surfaces in the master scene and it is desirable to remove the extraneous Shader Tree layers, applying the menu bar command **Texture** > **Purge Unused Materials** removes any unused surfacing. If the surfaces remain, it is likely because they are still applied elsewhere or they are masked using the non-default (other than material) tags.

Render Proxy Properties


Items
Shading
Groups
Audio
+
Filter Items
Add Item
F
Name
MyProxyScene.lxo
Mesh
Proxy
Camera
Directional Light
Properties
Channels
Display
Lists
+
Proxy
User Channels
Tags
Transform
Reset
Zero
Add
Proxy
Browse...
Browse...
Browse...
Replace Shaders
Unload Proxy Items
Restore from Proxy

Name: Proxy

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Order: ZXY

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

Scene: C:/Users/MutantPixel/Proxy.lxo

Render Item: Mesh

Preview Item:

Preview Rendering: Render Item

Option	Description
Name	Displays the current item name. You can change it by clicking on the field and typing the new name.

Option	Description
Transform	
Position	An item transform that allows you to numerically position the item in XYZ space. By default, **Position** transforms originate from the center position.
Rotation	An item transform that allows you to numerically set the rotation of the item. By default, **Rotation** transforms originate from the center position.
Order	Allows you to set the order that rotations are applied to the camera item. Changing the order that rotations are applied can sometimes help to reduce or eliminate gimbal lock.
Scale	An item transform that allows you to numerically set the size of the item. By default, scale transforms originate from the center position.
Reset	Resets the selected transform values to **(0,0,0)** returning the items to their default state.
Zero	Resets the chosen transform property values to **0**, leaving the center position and mesh position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values' input fields). This action causes the particular transform item to be added automatically to the **Channels** viewport list. The **Add** function here can also be used to add the selected set of transforms to the **Channel** list while keeping the default **0,0,0** values (a necessary step for referencing, in order to override the channels, they must first exist).
Proxy	
Scene	Represents the path to the **.lxo** file containing the original proxy layers exported with the **Convert to Proxy** command. Should it be necessary to move the proxy file, you can use this option to reinstate the connection by specifying the file's new location with the **Browse** option.

Option	Description
Render Item	Represents the specific item layer name in the proxy **.lxo** file that is loaded at render time.
Preview Item	By default, a render proxy item is represented in the 3D viewport by its bounding box. When something more representative of the actual geometry is required (such as for animation purposes), the **Preview Item** option provides for an alternate Mesh Item layer from the proxy **.lxo** that is displayed in the 3D viewport of the master scene in place of the bounding box. This can be anything, including the actual proxy mesh - but that defeats the purpose of the proxy. Still, it is beneficial if the preview proxy is representational of the geometry that is rendered. This can often be created by opening the saved proxy **.lxo** separately, duplicating the render proxy item layer and deleting any unnecessary geometry in the new duplicated layer. The Polygon Reduce tool may also prove to be useful in generating reasonable preview geometry. Next, save the proxy **.lxo** file and then in the master scene, specify the reduced geometry layer with the **Preview Item** option.
Preview Rendering	Since a render proxy item isn't part of the actual master scene and is only loaded into memory at render time, it also doesn't show up in any of the **Preview** or **RayGL** viewports. This can be remedied for **Preview** by enabling the **Preview Proxies** option, but this keeps the render proxy loaded into memory, eliminating any memory saving benefits. The **Preview Rendering** option provides a means to specify an additional optional Mesh Item layer within the proxy **.lxo** to be used exclusively for preview rendering in place of the actual proxy mesh.
Replace Shaders	When creating a proxy item, all related shaders remain in the master scene, but they are also cloned into the proxy scene when it is saved. The cloned surfaces in the proxy item file are the surface settings that are referenced for rendering. When the **Replace Shaders** option is enabled, you can modify the shading information contained in the master scene to modify the surfaces, providing an override to the settings of the proxy.
Unload Proxy Item	Removes all proxy data from memory and re-initializes it, setting it to the initial state as when it was generated.
Restore from Proxy	Restores the Mesh Item layers as proxies to full-fledged mesh layers. This command doesn't remove the saved file from disc. Also, any changes made to the proxy file is reflected in the resulting geometry, including if the file is saved as a static mesh.

Deferred Meshes

In complex or geometrically-dense scenes the **Deferred Mesh** option works to convert an existing non-deforming mesh to an externally saved **.lxdf** file. This action removes the geometry from the scene, while remaining as a renderable element. The main benefits to using deferred meshes are a reduced memory footprint, better scene interactivity, and faster scene save times.

Memory is reduced because the element, when saved out to the **.lxdf** format, is diced up into multiple segments. At render time, only the required segments are loaded, allowing for the rendering of massive data sets. Scene interactivity is also greatly improved because the overhead required to edit the dense mesh is eliminated and the scene saving happens much more quickly, as the element doesn't need to be written to disc during each save operation (especially beneficial to the **Autosave** function).

The **Deferred Mesh** option appears functionally similar to a Render Proxy mesh, as both save Mesh Items to external files, but the **Deferred Mesh** has some important differences. The main one being that a **Deferred Mesh** is not user-editable. When the item layer is saved as a **.lxdf**, the mesh itself is essentially frozen and triangulated whereas a Render Proxy mesh is saved as a regular **.lxo** file that can be opened separately, edited, and updated. Another difference is that the shading of the Deferred Mesh remains in the original scene. Only the geometric data is stored externally. For Render Proxies, the shading definitions are moved into the **.lxo** file along with the geometry.

Converting Meshes

Any regular non-deforming mesh layer or static mesh layer can be converted to a **Deferred Mesh**. This means that the target mesh can be animated to move, rotate, or scale, but it must remain rigid in overall shape. For example, if a **Bend Deformer** were applied to the mesh, when converted to the **Deferred Mesh**, the bending deformation would be ignored.

Conversion to the **Deferred Mesh** is done in the **Item List** contextual menu.

1. With the layer selected, right-click over the layer to open the menu and select **Change Type** > **Procedural** > **Deferred Mesh**.

   This action opens an OS-dialog, asking you where to save the resulting **.lxdf** file.

2. Define a name and press **OK** to convert to the **Deferred Mesh**.

   This action freezes the mesh into triangles and converts the mesh. If any **Subdivision Surfaces** are present, the **Render Level** subdivision value is applied during the **Freeze** process.

Once completed, the Mesh Item is replaced in the **Item List** by the same named **Deferred Mesh** item. You can select the item and adjust its attributes in the **Properties** panel.

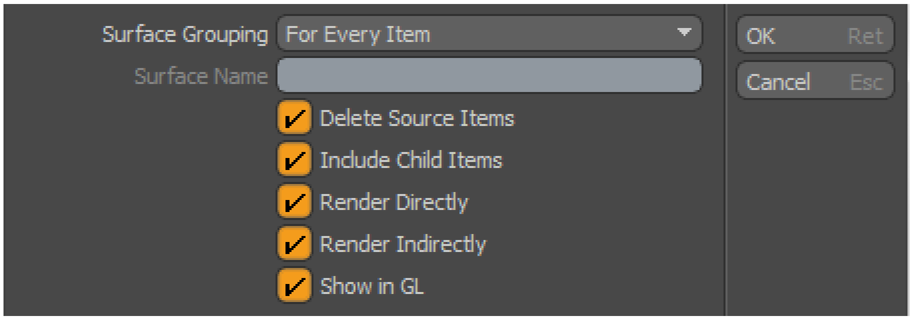
Warning: If you want to further edit a model that is converted to a **Deferred Mesh**, a separate version should be saved beforehand. Once a mesh has been converted to a **Deferred Mesh** and the scene file has been saved and closed (the undo stack is deleted), there is no way to return to the original unconverted version of the mesh.

Adding Meshes

Once a **Deferred Mesh** is created it is possible to add additional Mesh Items to the existing **.lxdf** file. To do this, follow these steps:

1. Select both the **Deferred Mesh** item layer in the **Item List** and the target mesh(es) to convert.
2. Then under the **Deferred Mesh** sub-tab of the **Properties** panel, select the **Add Geometry** button.

   This opens the **Add Deferred Geometry** dialog (items contained within a **Deferred Mesh** are individually referred to as **Surfaces**).



Surface Grouping: For Every Item

Surface Name:

☑ Delete Source Items

☑ Include Child Items

☑ Render Directly

☑ Render Indirectly

☑ Show in GL

This dialog allows you to define how the new geometry is added by its **Surface Grouping** setting and how it displays with its default display settings:

- **Surface Grouping** - defines how the additional item are treated within the **Deferred Mesh** file, once converted. Defining the proper **Surface Grouping** is important because it controls how the mesh can be affected in the future. There are three possible options:
  - **For Every Item** defines a separate grouping for every individual item that is added.
  - **For Selected Root Items** merges entire hierarchies of items into individual surface groups, named by their root item.
  - **One Surface** merges all items into a single surface group, named by the **Surface Name** input field. Defined Groups can be selected within the **Deferred Mesh** using the **Current Surface** pop-up in the **Properties** panel.

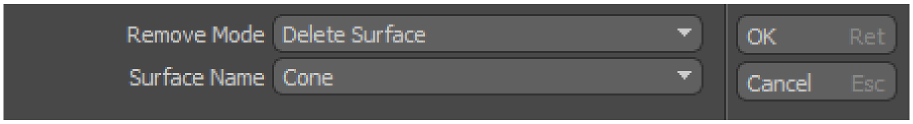
Note: Once combined as such, there is no way to undo the action or to then split individual surfaces out from the **Deferred Mesh**.

- **Surface Name** - when set to **One Surface**, this input field defines the name for the generated surface.
- **Delete Source Items** - when enabled, removes the item layers from the source scene, once they are converted.
- **Include Child Items** - when enabled, automatically includes all child items for selected root items.
- **Render Directly**/**Render Indirectly** - rendering of deferred surfaces can be split into two distinct passes, Direct and Indirect. **Direct** refers to light that is evaluated from direct light sources in the scene, such as a **Spot Light** or a **Distant** light. **Indirect** refers to global illumination rendering. This is the bounced light in the scene producing the color bleeding that happens between two surfaces. The reason for this distinction is that separate geometry can be used for each lighting pass. One high resolution surface for the direct rendering, and another lower resolution surface for the bounced global illumination rendering. When set-up as such, this method of rendering can increase the performance of capturing indirect illumination as detailed surfaces tend to slow down the evaluation of the bounced light. Once the mesh surfaces have been properly defined, the **Indirect LOD** option must also be enabled in the **Render Item** at the very bottom of the **Settings** sub-tab.
- **Show in GL** - controls visibility of the **Deferred Mesh** in the GL viewports. When enabled, the resulting Deferred Meshes are visible in the 3D viewport. When disabled, the Deferred Mesh shows up as an empty bounding box.

Note: When the **Surface Display** is set to **Direct Surfaces as Boxes**, the GL display is controlled by the **Render Directly** toggle, and not **Show in GL**. This is because the boxes shown are based on the render time geometry, so turning off render time geometry means that no boxes are created.

Removing Meshes

Removing of items stored in a **Deferred Mesh .lxdf** file can be done by selecting the target surface to be removed using the **Current Surface** option. Once selected, press the **Remove Surface** button. This opens the **Remove Deferred Surfaces** dialog:



Remove Mode: Delete Surface

Surface Name: Cone

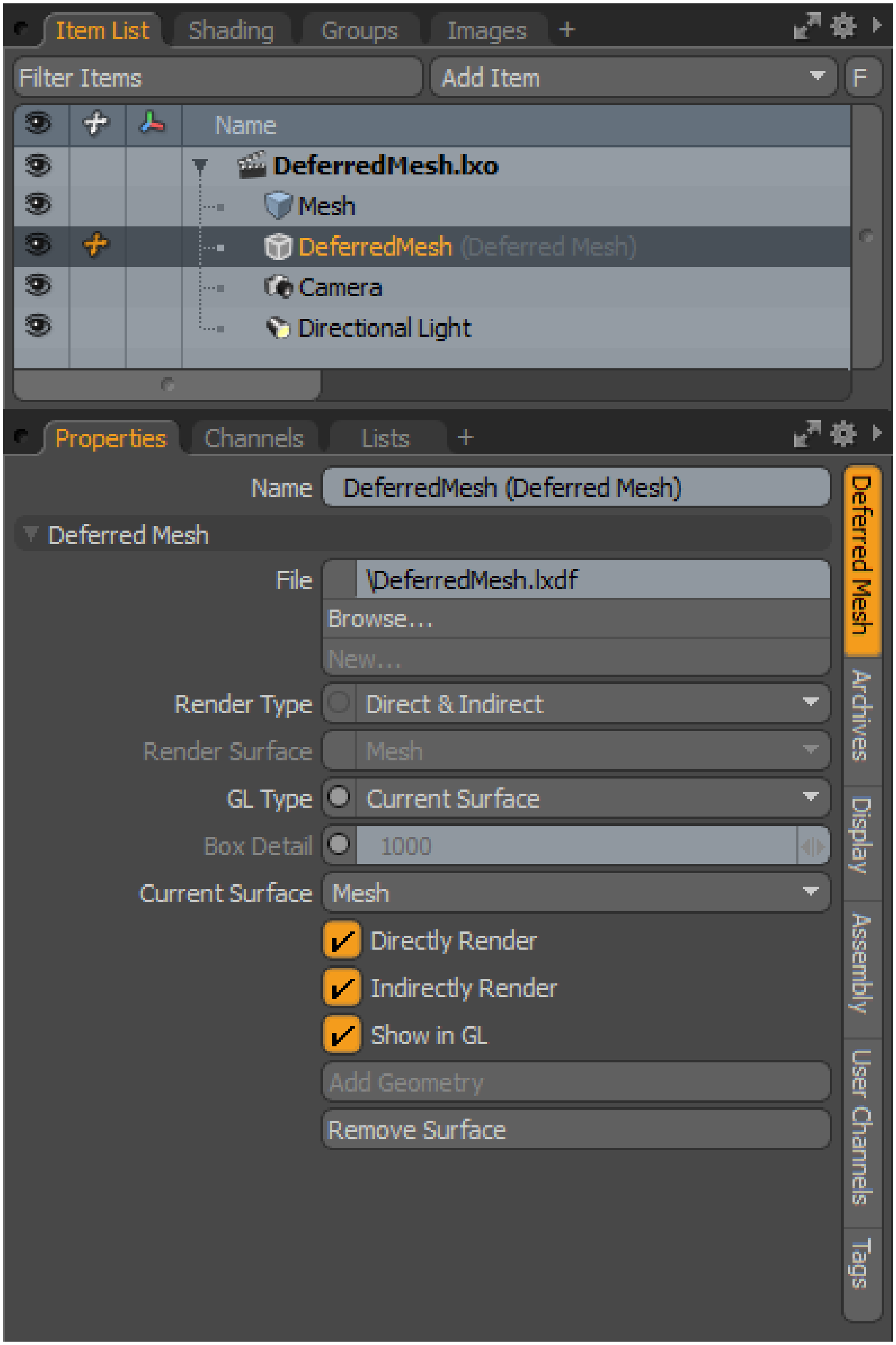
- **Remove Mode** - controls what action is performed on the selected surface:
    - **Delete Surface** - removes the surface definition from the **Deferred Mesh** files. (This action is not undoable.)

  - **Extract to Static Mesh** - extracts the surface definition and places it in a Static Mesh Item. (This action is not undoable.)
  - **Split to Independent Surface** - extracts surface definition to a **Deferred Surface** item that can be controlled independently (see **Independent Surfaces** below).
- **Surface Name**- allows you to select alternate surfaces for removal if the initial selection was incorrect.

Independent Surfaces

If a **Deferred Mesh** contains multiple surfaces, there may be cases where you want to break out individual surfaces so they can be either animated, shaded, or their visibility adjusted independently (or all three). This is done by creating a **Deferred Surface**. A **Deferred Surface** is an independent item in the scene, but it is still contained within its original **Deferred Mesh .lxdf** file. To create a **Deferred Surface** select the surface from the **Current Surface** pop-up. Once selected press the **Remove Surface** button. This opens the **Remove Deferred Surfaces** dialog (explained above). Select the **Split to Independent Surfaces** option to convert the selection to a **Deferred Surface**.

Deferred Mesh Tab Properties


Item List
Shading
Groups
Images
Filter Items
Add Item
F
Name
DeferredMesh.lxo
Mesh
DeferredMesh (Deferred Mesh)
Camera
Directional Light
Properties
Channels
Lists
Deferred Mesh
Archives
Display
Assembly
User Channels
Tags
Browse...
New...
Add Geometry
Remove Surface

Name: DeferredMesh (Deferred Mesh)

Deferred Mesh

File: \DeferredMesh.lxdf

Render Type: Direct & Indirect

Render Surface: Mesh

GL Type: Current Surface

Box Detail: 1000

Current Surface: Mesh

☑ Directly Render

☑ Indirectly Render

☑ Show in GL

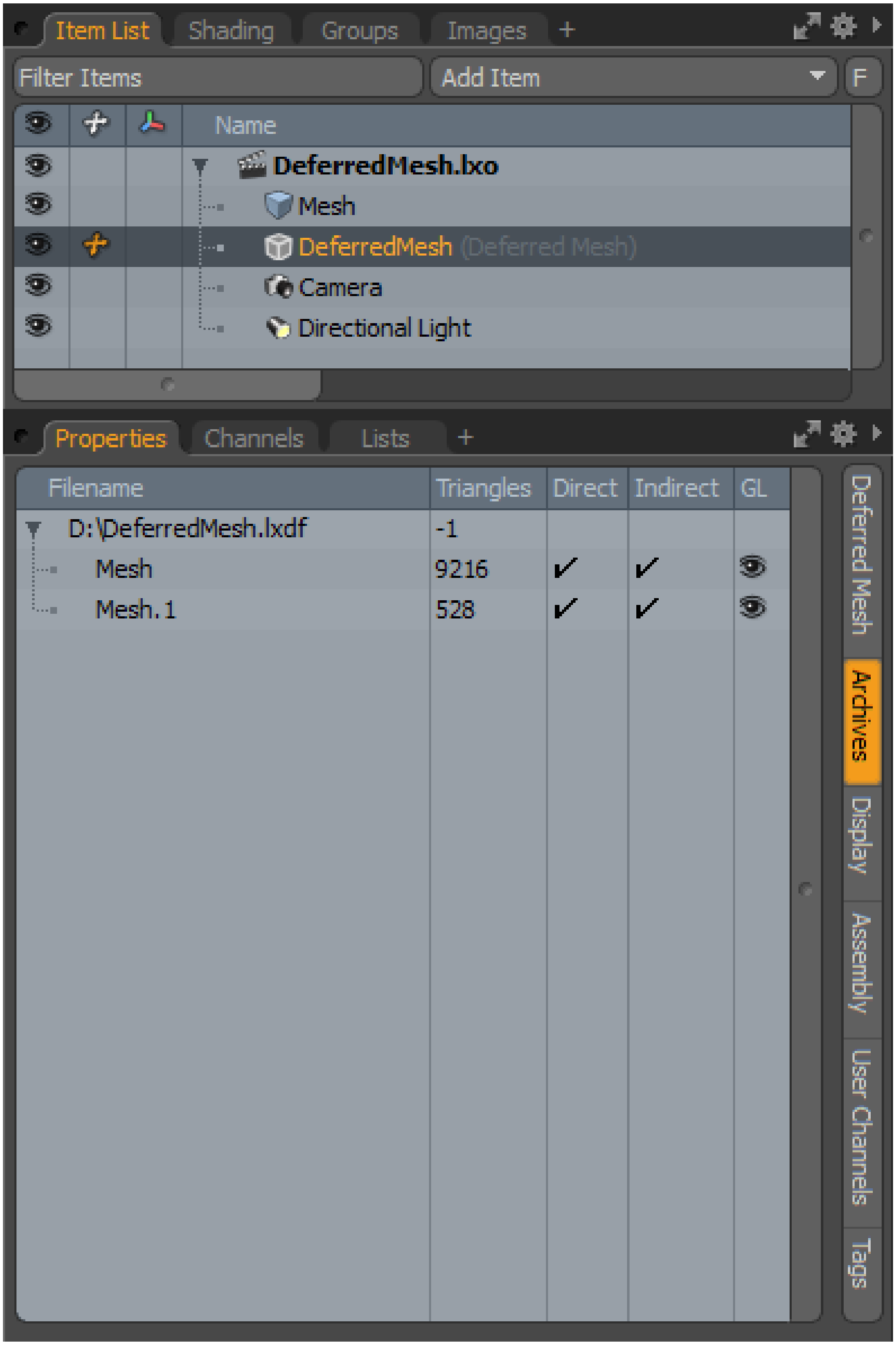
The following **Deferred Mesh** options are available for **Deferred Mesh**:

- **File** - lists the name and location on disc of the saved **.lxdf** deferred mesh file. The **Browse** option can be used to locate an alternate file as a replacement to the current file. The **New** option triggers the saving of a new **Deferred Mesh** file when an additional mesh layer is also selected.
- **Render Type** - determines the rendering evaluation of the contents of the deferred mesh:
  - **Direct & Indirect** - any surface with either the **Direct Rendering** toggle or the **Indirect Rendering** toggle is evaluated directly or indirectly, according to its setting. This is the default and probably most common choice.
  - **Direct Only** - only surfaces with the **Direct Rendering** toggle enabled render.
  - **Indirect Only** - only surfaces with the **Indirect Rendering** toggle enabled render.
  - **One Surface** - only a single named surface from within the **Deferred Mesh** renders, defined by the **Render Surface** option below.
- **Render Surface** - defines the surface that renders when the **Render Type** is set to **One Surface**.
- **GL Type** - determines how the contents of the **Deferred Mesh** file are displayed in the 3D viewport:
  - **None** - shows no surfaces in GL.
  - **Current Surface** - shows only the surface matching the name in the **Current Surface** channel.
  - **All GL Surfaces** - shows all the surfaces with the **Show in GL** toggle enabled.
  - **All Indirect Surfaces** - shows all the surfaces with the **Indirect Rendering** toggle enabled.
  - **Direct Surfaces as Boxes** - shows all surfaces with the **Direct Rendering** toggle enabled using a variable bounding box representation.
- **Box Detail** - defines an upper limit to the number of boxes used when the **Direct Surfaces as Boxes** option is set.
- **Current Surface** - allows you to select separate surface from within the **Deferred Mesh** file (added separately). Displayed surfaces can be modified for remove and to separate it out as a different surface using the other controls within the **Properties** panel.
- **Directly/Indirectly Render** - defines how the deferred surface is rendered. This evaluation can be split into two distinct passes: Direct and Indirect. **Direct** refers to light that is evaluated from direct light sources in the scene, such as a **Spot Light** or a **Distant** light. **Indirect** refers to global illumination rendering. This is the bounced light in the scene producing the color bleeding that happens between two surfaces. The reason for this distinction is that separate geometry can be used for each lighting pass; one high resolution surface for the direct rendering, and another lower resolution surface for the bounced global illumination rendering. When set up as such, this method of rendering can increase the performance of capturing indirect illumination as detailed surfaces can tend to slow down the evaluation of the bounced light. Once the mesh surfaces have been properly defined, the **Indirect LOD** option must also be enabled in the Render Item at the very bottom of the **Settings** sub-tab.
- **Show in GL** - controls visibility of the **Deferred Mesh** in the GL viewports. When enabled the resulting Deferred Meshes are visible in the 3D viewport. When disabled the **Deferred Mesh** shows up as an empty bounding box.

Note: When the **Surface Display** is set to **Direct Surfaces as Boxes**, the GL display is controlled by the **Render Directly** toggle, and not **Show in GL**. This is because the boxes shown are based on the render time geometry, so turning off render time geometry means that no boxes are created.

- **Add Geometry** - used to add additional item to an existing Deferred Surface. See Adding Meshes for more information.
- **Remove Surface** - used to remove a surface from a **Deferred Mesh** file or break it out into its own item layer, called a deferred surface. See Removing Meshes for more information.

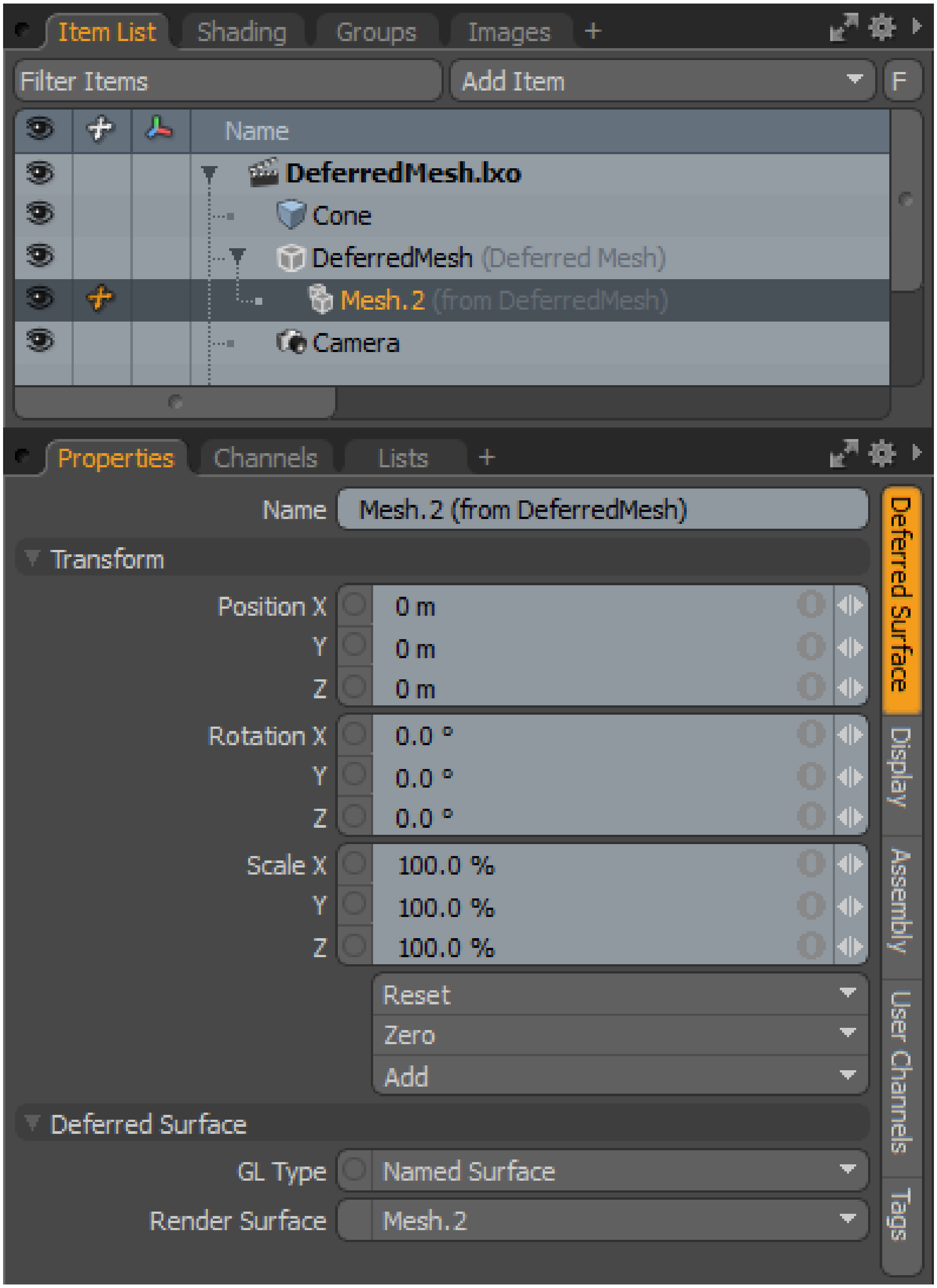
Archive Tab Properties


Item List
Shading
Groups
Images
+
Filter Items
Add Item
F
Name
DeferredMesh.lxo
Mesh
DeferredMesh (Deferred Mesh)
Camera
Directional Light
Properties
Channels
Lists
+
Deferred Mesh
Archives
Display
Assembly
User Channels
Tags

Filename	Triangles	Direct	Indirect	GL
D:\DeferredMesh.lxdf	-1			
Mesh	9216	✓	✓	👁
Mesh.1	528	✓	✓	👁

- **Archives** - displays all the separate surfaces contained within the current **Deferred Mes**h. You can view the surfaces and their polygon count statistics as well as the settings for the rendering and visibility toggles all at one time.

Deferred Surface Tab Options


Item List
Shading
Groups
Images
Filter Items
Add Item
F
Name
DeferredMesh.lxo
Cone
DeferredMesh (Deferred Mesh)
Mesh.2 (from DeferredMesh)
Camera
Properties
Channels
Lists
Deferred Surface
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Deferred Surface

Name: Mesh.2 (from DeferredMesh)

Position X: 0 m
Y: 0 m
Z: 0 m

Rotation X: 0.0 °
Y: 0.0 °
Z: 0.0 °

Scale X: 100.0 %
Y: 100.0 %
Z: 100.0 %

GL Type: Named Surface

Render Surface: Mesh.2

- **Name** - displays the current item name. You can easily change it by clicking within the field and typing the new name.

The following **Transform** options are available for the **Deferred Surface**:

- **Position** - an item transform that allows you to numerically position the item in XYZ space. By default, **Position** transforms originate from the item's **Center** position.
- **Rotation** - an item transform that allows you to numerically set the rotation of the item. By default, **Rotation** transforms originate from the item's **Center** position.
- **Scale** - an item transform that allows you to numerically set the size of the item. By default, **Scale** transforms originate from the item's **Center** position.
- **Freeze** - returns the item's **Center** position to the world space center (0,0,0) without changing the position of the Mesh Item itself.
- **Zero** - resets the chosen transform property values to **0**, leaving the **Center** position and mesh position intact. This is done by adding an additional transform item to the Mesh Item's channels with an inverted version of the current values. This is useful to allow, for example, a joint to have a base value of 0,0,0 but still be located away from the World Origin.
- **Add** - adds additional **Transform** items to the associated item, or if they do not yet exist it simply adds them. Transform items are the channel groups that store the actual transform values controlling any item's position, rotation, and/or scale. You can add as many transform items as desired for any transform property desired. Adding additional **Transform** items produces an additive effect, where each transform group is evaluated before the next and so on. Additional item transforms are evaluated in their order in the channels list, from the bottom upwards.

  It should be noted that, by default, new items do not have any transform items associated with them (even though they are visible here within the **Properties** panel). This is useful as an optimization as only the necessary transforms are created on an as-needed basis, reducing scene overhead. There are several actions that add these base transform items. One is by simply transforming the target item with one of the various transform tools or by editing the values' input fields. This action causes the particular transform item to be added automatically to the **Channels** viewport list. Due to this fact, you may need to specifically create item transforms when **Referencing**, because in order to override the channels in the Master scene, they must first exist.

The following **Deferred Surface** options are available for the **Deferred Surface**:

- **GL Type** - determines how the contents of the **Deferred Mesh** file are displayed in the 3D viewport:
  - **None** - shows no surfaces in GL viewport, just a bounding box.
  - **Named Surface** - shows only the surface matching the name in the **Render Surface** channel.
  - **Named Surface as Boxes** - shows the **Render Surface** definition using a variable bounding box representation.
- **Render Surface** - sets the **Render Surface** for either of the named surface options.

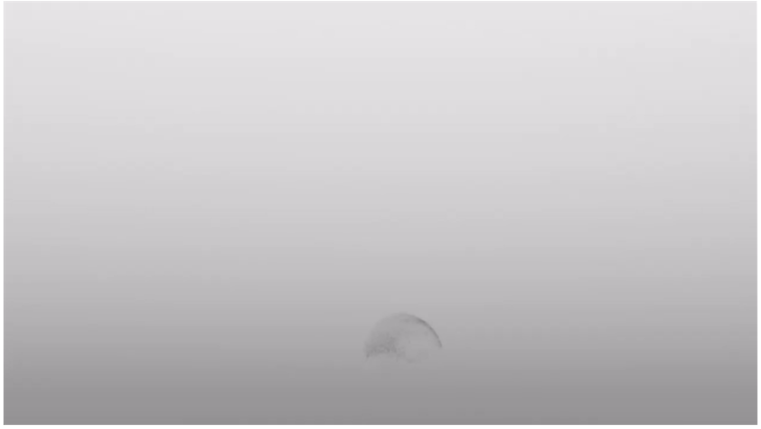
VDB Voxel

The VDB Voxel item allows you to render externally created simulations stored in the open-source VDB voxel format, as well as generate volume data from meshes or particles. The OpenVDB format is an efficient method of storing hierarchical data structures.

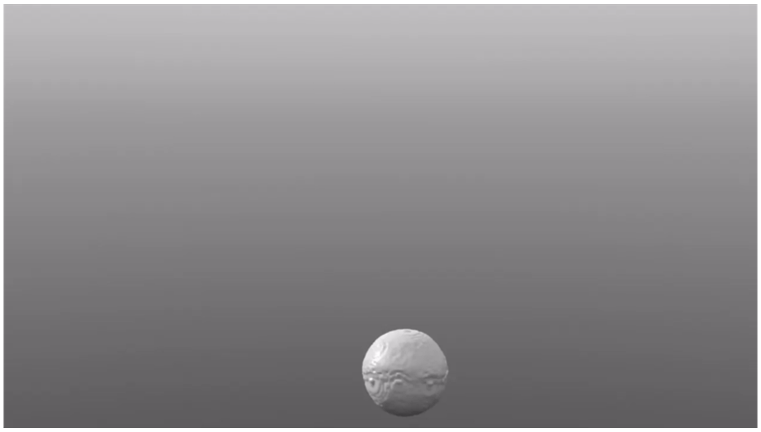
Simulations can be volumetric in nature, such as smoke and clouds, or represent a surface, such as water or other liquids. Volumetric effects must be connected to a Volume Item for rendering in Modo. Surfaces can be connected to Blobs, or use the internal meshing feature, when the **Draw Mesh Surfaces** option is enabled.

Note: VDB Voxel uses OpenVDB 3.0. VDB files encoded with Blosc are not supported.

Smoke example:



Fluid example:



Rendering Volumes

The VDB Voxel Item is added to the scene in the **Item List** under **Add Items** > **Volumes** > **VDBVoxel**. With the item selected, from the **Properties** panel, use the **Load File** button and locate the desired OpenVDB file (<filename>.vdb). Once loaded, an octree-like grid appears in the 3D viewport, representing the voxel shape. At this point the item won't yet render. To render using a Volume Item, the VDB Voxel Item and the **Volume** item must be connected in the **Schematic** viewport. This is done by selecting both items in the **Item List** and then, in a **Schematic** viewport, clicking the **Add Selected** button (or by dragging and dropping into the viewport). Once there, connect the **VDBVoxel** relationship connector to the **Voxel Source** input on the Volume Item. For information on working in the **Schematic** viewport, refer to Schematic Viewport.

Surfacing is applied by way of an item mask in the **Shader Tree**. Select the **Volume** item and right-click to open the contextual menu and select the **Create Item Mask** command. In the **Shader Tree**, selecting the resulting **Material** item within the generated mask and edit the **Volumetric Properties** under the **Material Rays** sub-tab.

Rendering Surfaces

The VDB Voxel Item is added to the scene in the **Item List** under **Add Items** > **Other** > **VDBVoxel**. With the Item selected, from the **Properties** panel, use the **Load File** button and locate the desired OpenVDB file (<filename>.vdb). Once loaded, an octree-like grid appears in the 3D viewport, representing the voxel shape. At this point the item won't yet render. To create a surface, simply enable the **Draw Mesh Surfaces** option. When enabled, the polygonized surface is visible in the 3D viewport, as well as in any renders (for large VDB files generating the surface may take some time). Surfacing is applied by way of an item mask in the **Shader Tree**. Select the VDBVoxel Item and right-click to open the contextual menu and select the **Create Item Mask** command. In the **Shader Tree**, select the resulting **Material Item** within the generated mask and edit any of the standard material properties available.

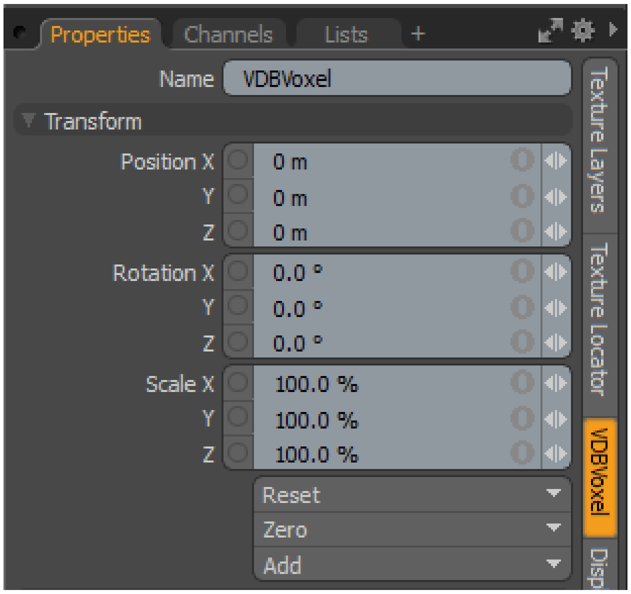
Meshing Particles

Load the particle simulation into the scene, or generate a particle simulation using any of the provided particle simulation components. Once completed, make sure to cache the simulation. The VDB Voxel Item is added to the scene in the **Item List** under **Add Items** > **Other** > **VDBVoxel**. With the item selected, from the **Properties** panel, use the **Source Item** selector and select the desired Particle Simulation as the source element. Next, enable the **Draw Mesh Surfaces** option to mesh the particles with a surface. From here the **Filtering** settings as well as the Voxel's **Radius** and **Resolution** settings come into play. They can be used to modify the look of the resulting surface. Surfacing is applied by way of an item mask in the **Shader Tree**. Select the VDBVoxel Item, right-click to open the contextual menu, and select the **Create Item Mask** command. In the **Shader Tree**, select the resulting **Material** item within the generated mask and edit any of the standard material properties available.

VDBVoxel Options

The following options are available for **VDBVoxel**:

Transform



Name VDBVoxel

Position X 0 m
Y 0 m
Z 0 m

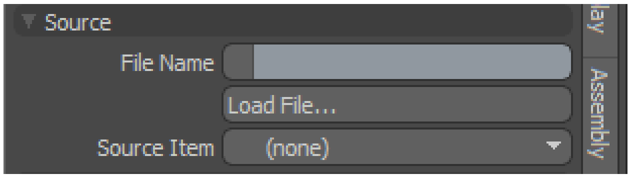
Rotation X 0.0 °
Y 0.0 °
Z 0.0 °

Scale X 100.0 %
Y 100.0 %
Z 100.0 %

Transform

Position	Use the X, Y and Z input fields to apply specific offset distance values, calculated from the position the object was in when the tool was activated. When adjusting an object interactively in the viewport, the Position input fields display the current offset distance applied, providing useful feedback.
Rotation	Use the X, Y and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, these input fields display the current offset distance applied, providing useful feedback.
Scale	Use the X, Y and Z input fields to apply specific scaling values. When adjusting an object interactively in the viewport, these input fields display the current scale factor applied, providing useful user feedback.
Reset	Resets the selected transform values to (0,0,0) returning the items to their default state.
Zero	Resets the chosen transform property values to 0, leaving the center position and mesh position intact.
Add	Transform items are the channel groups associated with an item that store its transform values, controlling its position, rotation and/or scale. By default, new items do not have any transform items associated with them (even though they are visible here within the Properties panel). This is useful as an optimization as only the necessary transforms are added on an as-needed basis, reducing scene overhead. There are several ways to add them. One is by simply transforming the target item with one of the various transform tools (or by editing the values' input fields). This action causes the particular transform item to be added automatically to the Channels viewport list. The Add function here can also be used to add the selected set of transforms to the Channel list while keeping the default 0,0,0 values (a necessary step for referencing, in order to override the channels, they must first exist).

Source



Source

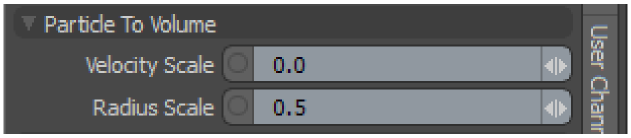
File Name

Load File...

Source Item (none)

Source	
File Name	Displays the currently-loaded path and file name. To update the current VDB file, use the **Load File** button.
Load File	Opens an OS-specific dialog, allowing you to locate and select the VDB file to be loaded.
Source Item	Serves two purposes: you can load Mesh Item layers and they are voxelized by the **VDB Voxel** item, or additionally a particle source can be specified, and with the **Draw Mesh Surfaces** option enabled, the particles can be "meshed" (meaning a polygonal surface is applied). When a Particle Source is defined, the **Filter** settings and the **Voxel** settings come in to play to control the look of the surfacing.

Particle To Volume



Particle To Volume

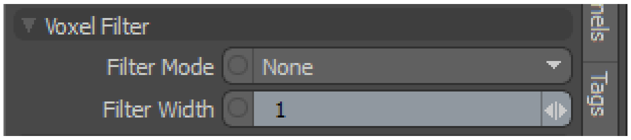
Velocity Scale 0.0

Radius Scale 0.5

Particle to Volume	
Velocity Scale	Controls the scale of particle stretching based on velocity. Values are additive to the base value, so a Stretch Particle value of 100% produces a particle that was stretched twice the length (200% altogether).
Radius Scale	Controls the amount of randomness added to the scale of each

	individual particle volume. The amount is determined by a scale percentage - a setting of 25% adjusts the scale of the base radius size by an amount of 0%-25% randomly across all particles, a setting of 50% adjusts the size by 0%-50% randomly across all particles and so on. **Tip:** When meshing a particle source (loaded through the **Source Item** control) the **Radius** value defines the rendered size of the virtual sphere shape formed around each particle position. The **Resolution** setting and the **Filtering** values determines how sphere-like the final result end up being.

Voxel Filter

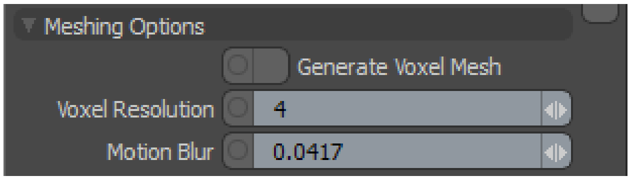


Filter Mode None

Filter Width 1

Voxel Filter	
Filter Mode	Determines a method of smoothing, applied to the surface generated from particle. The options possible are **None**, **Mean**, **Gaussian**, and **Median**. The **None** option disables filtering, while the **Mean**, **Gaussian**, and **Medial** all average the surface values differently, producing slightly different results. Results are also dependent on the **Width** setting and the **Resolution** of the Voxels.
Filter Width	Defines the size of the averaging filter applied to the generated surface values when a **Filter Mode** other than **None** is selected. Larger sizes produce a smoother surface but reduces detail overall.

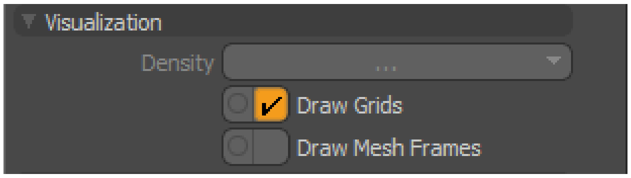
Meshing Options



Generate Voxel Mesh
Voxel Resolution 4
Motion Blur 0.0417

Meshing Options	
Generate Voxel Mesh	When enabled, a polygonal surface is applied.
Voxel Resolution	Determines how finely the particles are meshed. Low values can be easily smoothed with filtering, but produce limited detail. Higher values produce more detail, but may also require additional filtering to remove resolution artifacts.
Motion Blur	Defines the value for motion blur of particles for volume creation.

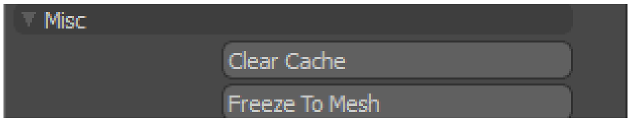
Visualization



Density ...
☑ Draw Grids
☐ Draw Mesh Frames

Visualization	
Density	Defines which stored features of the VDB file is used to represent the visualized form in the 3D Viewports.
Draw Grids	Toggles visibility of the representative grid-like display in the 3D Viewports based on the **Feature** definition.
Draw Mesh Frames	When loading a multi-framed sequence, enabling this option loads all the subsequent frames and draw them in the 3D Viewports.

Misc

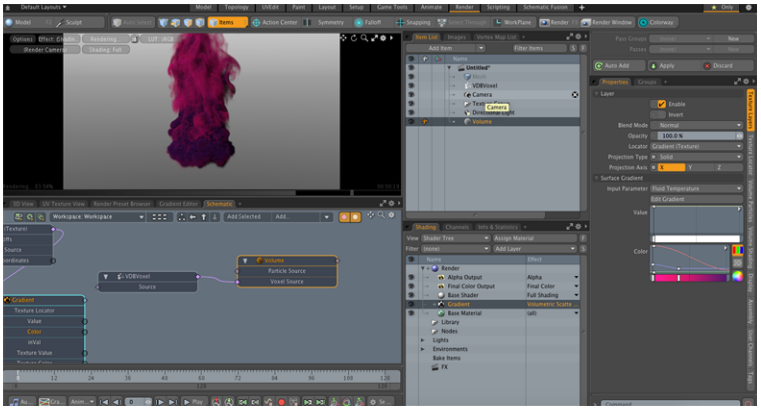


Misc	
Clear Cache	Used to remove any stored data from RAM. This resets the cache if the **Mode** is set to **Fast** or **Full**, or removes it if the **Mode** is changed to **None**. **Note:** You can access the additional cache options **Cache Mode** and **Cache Size** in the **Rendering** preferences. In the menu bar for Windows, click **System** > **Preferences...** > **Rendering** > **Final Rendering** and for Mac OS, click **Menu** > **Preferences...** > **Rendering** > **Final Rendering**. For more information, see Rendering Preferences.
Freeze To Mesh	Used to convert an item from type **VDBVoxel** to **Mesh**.

Applying Fluid Gradient Effects

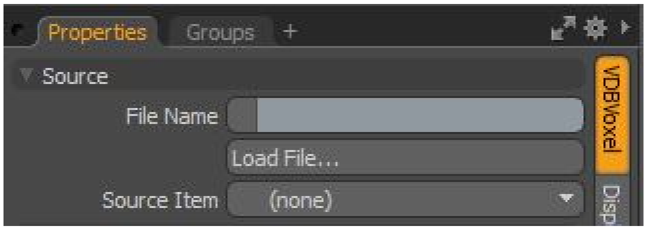
Modo has the ability to load fluid texture effects Pressure, Temperature, and Fuel from a VDB Voxel file. Voxel grids are converted into fluid gradient effects, allowing you to render externally created simulations and to generate volume data from meshes or particles with the imported texture effects.

Note: These texture effects are read only. You can only set them as input parameter, not a output parameter.



To add fluid gradient effects to your VDB Voxel file:

1. Add a gradient item to your scene. To do this open the Shader Tree by clicking on **Shading** tab and then click **Add Layer** and select **Processing** > **Gradient**.
2. In the **Properties** > **Texture Layers** pane, expand **Surface Gradient.**
3. Click the **Input Parameter** drop-down menu, expand **Volumetric Effects**, and select one of the following options:
    - **Fluid Temperature** - Import fluid temperature volumetric data from the VDB Voxel source file to convert it to a Modo gradient output.
    - **Fluid Fuel** - Import fluid fuel volumetric data from the VDB Voxel source file to convert them to a Modo gradient output.
    - **Fluid Pressure**- Import fluid pressure volumetric data from the VDB Voxel source file to convert them to a Modo gradient output.
4. In the **Mesh Operations** list, **Item List** tab, click **Add Item**.
5. Select **Volumes** > **VDBVoxel**.
6. In the **Properties** pane, under **Source**, click **Load File**.



7. Load a **.vdb** file which contains **Temperature**, **Fuel**, or **Pressure** effects.
8. Render your scene to see the results.

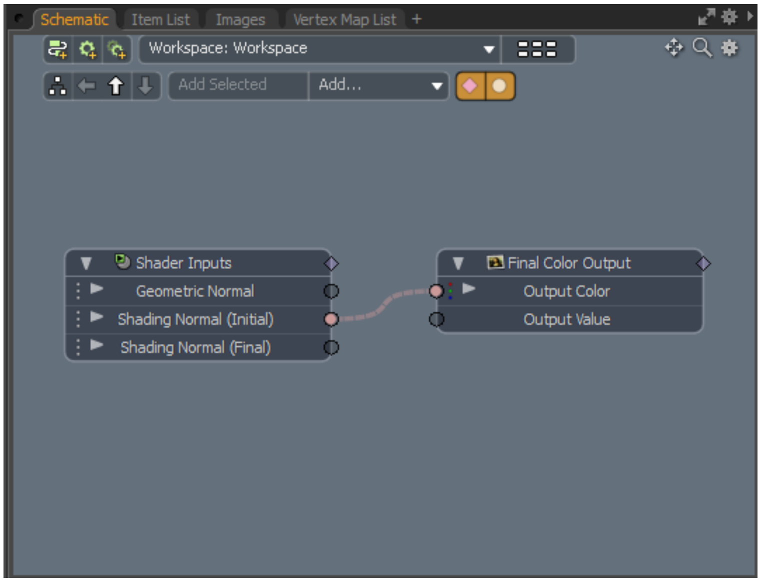
    For more information about rendering, see Rendering.

Driving Render Outputs Nodally

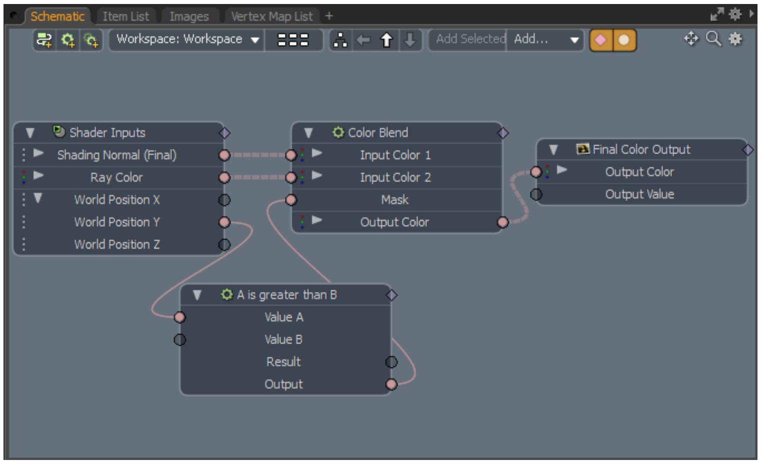
Render Outputs can be driven nodally, on a per-sample basis, through input channels for the color (**Output Color**) and value (**Output Value**) of each Render Output. This allows any shader or procedural texture to completely specify the pixels of the Render Output.

Note: The appropriate input (color or value) is selected based on the type of Render Output.

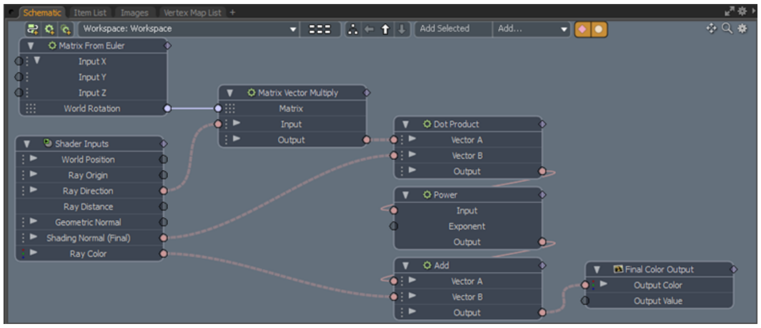
In the simplest case, the output is set directly, independent of its normal output:


Schematic
Item List
Images
Vertex Map List
Workspace: Workspace
Add Selected
Add...
Shader Inputs
Geometric Normal
Shading Normal (Initial)
Shading Normal (Final)
Final Color Output
Output Color
Output Value

You can also adjust the normal shaded output by blending the desired colors with the **Ray Color** of the **Sample Ray** from a Shader Input node. In this case, the original pixels of the image are shown, except when the World Position of the object is above Y=0, in which case it is blended with the RGB values representing the object's surface normals:


Schematic
Item List
Images
Vertex Map List
Workspace: Workspace
Add Selected
Add...
Shader Inputs
Shading Normal (Final)
Ray Color
World Position X
World Position Y
World Position Z
Color Blend
Input Color 1
Input Color 2
Mask
Output Color
Final Color Output
Output Color
Output Value
A is greater than B
Value A
Value B
Result
Output

Here's a less abstract example. You could add an artificial greenish specular highlight based on the angles defined in the Matrix From Euler node to an image that has no specular effects:


Schematic
Item List
Images
Vertex Map List
Workspace: Workspace
Add Selected
Add...
Matrix From Euler
Input X
Input Y
Input Z
World Rotation
Matrix Vector Multiply
Matrix
Input
Output
Dot Product
Vector A
Vector B
Output
Shader Inputs
World Position
Ray Origin
Ray Direction
Ray Distance
Geometric Normal
Shading Normal (Final)
Ray Color
Power
Input
Exponent
Output
Add
Vector A
Vector B
Output
Final Color Output
Output Color
Output Value

Network Rendering

Network rendering in Modo is designed to enable simple access to multiple machines across a home network or company-wide intranet for rendering single images or sequences. Using cross-platform Bonjour technology (developed by Apple and installed when you installed Modo), the number of steps have been greatly reduced, eliminating the overhead usually associated with setting up network rendering. Once activated, any render tasks on the host system are automatically spread out across all available workers.

Network Rendering Step-by-Step

Note: As of Modo 11.1, Modo's installer for Windows no longer installs Bonjour, which is used for automatic worker discovery in Modo's native network rendering. You can either install Bonjour on each Windows system you intend to use with network rendering, or use the Host List feature to specify the network IPs of the specific systems manually. For more information on how to install Bonjour, see this page.

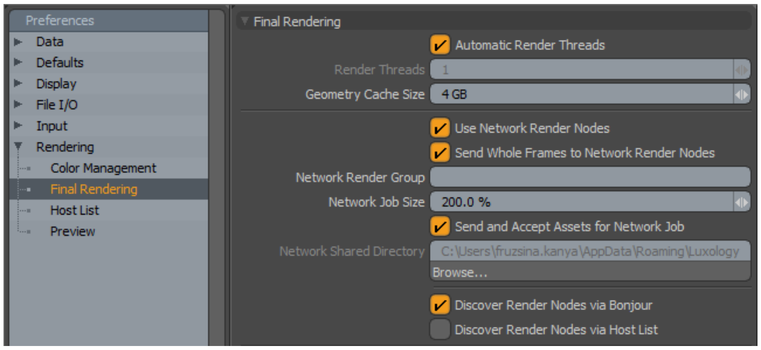
To set up network rendering with two or more systems, do the following:

1. Install the same version and build number of Modo on each system being used. See **Help** > **About MODO** to verify the build number. Worker systems do not need any content installed, but should have any third party plug-ins and/or kits installed, if scenes to be rendered contain those features.
2. Make sure that all the worker systems can see the master on the network and vice versa. Connected systems are visible in the **Network View** window, found in the menu bar, under **Render** > **Open Network View**.

Note: Hardware and software firewalls that can often block communications between systems, preventing network rendering.

3. Start Modo on the worker system(s) and enter into worker mode by selecting **Render** > **Enter Worker Mode** from the menu bar. Unlicensed installs can only enter worker mode.
4. Enable network rendering on the master system by enabling the **Use Network Render Nodes** toggle in **Preferences** > **Rendering** > **Final Rendering**.
5. If required, enable the **Send Whole Frames to Network Render Nodes** option to distribute entire frames to workers rather than buckets. This provides a significant performance improvement for rendering animations.

Note: For more information on rendering settings, see Rendering Preferences.



☑ Automatic Render Threads
Render Threads: 1
Geometry Cache Size: 4 GB
☑ Use Network Render Nodes
☑ Send Whole Frames to Network Render Nodes
Network Render Group:
Network Job Size: 200.0 %
☑ Send and Accept Assets for Network Job
Network Shared Directory: C:\Users\fruzsina.kanya\AppData\Roaming\Luxology
☑ Discover Render Nodes via Bonjour
☐ Discover Render Nodes via Host List

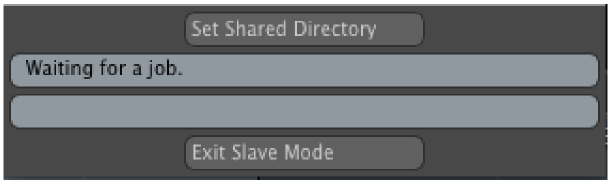
6. Open a scene or animation on the host and render with one of the Render Commands.

Note: Preview does not utilize multiple machines.

Modo has the **Send and Accept Assets for Network Jobs** option enabled by default. This means that the master system uses network sockets to transfer scenes and asset data to each worker system. The option of using a shared network folder is also available (see below). For simplicity of setup and use, Modo also has **Discover Render Nodes via Bonjour** enabled by default. This setting should work for most basic network setups. With **Use Network Render Nodes** enabled on the master and worker system in worker mode, any render started on the master sends scene and asset data to the workers, then assigns them buckets on each rendered frame (these buckets appear with blue outlines in the Render Window). When the worker systems complete those buckets, they return them to the master.

Using a Shared Network Folder

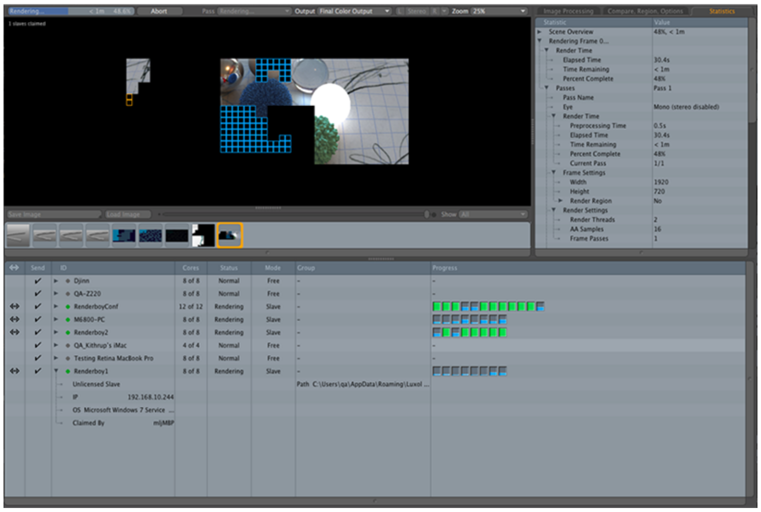
When the **Send and Accept Assets for Network Jobs** setting is disabled, the master system deposits the scene and assets to the specified network shared directory for access by all workers. To use this feature, each worker must be set up to use that same directory. This can be done by starting Modo on each worker and editing the **Preferences** setting **Network Shared Directory**, or by setting the directory through the **Render Worker** dialog (present only when a worker is in worker mode), by clicking the **Set Shared Directory** button. Make sure that all systems have proper access and permissions to read and write to the specified folder.



Waiting for a job.

Network View

The **Network** view can be accessed from the menu bar under **Render** > **Open Network View** or by dragging the divider handle up at the very bottom of the Render Display window. This allows you to view the connection status of all the systems on the network running Modo.



Statistic	Value
Scene Overview	48%, < 1m
Rendering Frame 0...	
Render Time	
Elapsed Time	30.4s
Time Remaining	< 1m
Percent Complete	48%
Passes	Pass 1
Pass Name	
Eye	Mono (stereo disabled)
Render Time	
Preprocessing Time	0.5s
Elapsed Time	30.4s
Time Remaining	< 1m
Percent Complete	48%
Current Pass	1/1
Frame Settings	
Width	1920
Height	720
Render Region	No
Render Settings	
Render Threads	2
AA Samples	16
Frame Passes	1

<--->	Send	ID	Cores	Status	Mode	Group
	✓	Djinn	8 of 8	Normal	Free	-
	✓	QA-Z220	8 of 8	Normal	Free	-
<--->	✓	RenderboyConf	12 of 12	Rendering	Slave	-
<--->	✓	M6800-PC	8 of 8	Rendering	Slave	-
<--->	✓	Renderboy2	8 of 8	Rendering	Slave	-
	✓	QA_Kithrup's iMac	4 of 4	Normal	Free	-
	✓	Testing Retina MacBook Pro	8 of 8	Normal	Free	-
<--->	✓	Renderboy1	8 of 8	Rendering	Slave	-
		Unlicensed Slave				Path C:\Users\qa\AppData\Roaming\Luxol ...
		IP 192.168.10.244				
		OS Microsoft Windows 7 Service ...				
		Claimed By mljMBP				

The **Network** view above shows eight systems on the network. The various columns of the **Network** view show the following information:

<--->: This is the communication column, it indicates if there is communication happening between the master and the worker with the back and forth arrow icon.

Send: This column shows if the connected system has the **Send and Accept Assets** option enabled. If it does, a checkmark appears, if not, the value is left blank.

ID: This column shows the info of any connected systems. The colored dots preceding the system names denote their present state.

- **Red**: The node is unreachable or not compatible.
- **Green**: The node is compatible and available for rendering.
- **Black**: The node is compatible but not available for rendering.
- **Gray**: The node's status is unidentified.

Clicking the triangle preceding each system name expands or collapses the system's information. When collapsed, each system takes up one row. When expanded, additional information can be seen, such as network folder path, IP address, OS, and whether or not it has been claimed by a master system for rendering.

Note: During the render, you can see how many workers have been claimed by the master for rendering. The count appears in the top left of the render window.

Cores: This column displays the number of cores available for rendering on each system. This setting can be changed on each system individually, by editing the option under **Preferences** > **Final Rendering** > **Render Threads**.

Status: This column displays the current status of each system on the network.

- **Normal**: For systems that are not running in **Worker** mode.
- **Idle**: The system is in **Worker** mode and waiting for a job.
- **Rendering**: The system is in **Worker** mode and is rendering.

Mode: This column displays the current mode of the system:

- **Master**: This system is a master in an active network render.
- **Worker**: This system is in **Worker** mode and ready to render.
- **Free**: This system is neither a worker or master, and is not currently rendering.

Group: This column displays which group the system is currently participating in. When expanded, the **Group** column also shows the network shared folder path set on each system.

Progress: This column is used to display the progress of each system that is rendering, showing vertical progress bars for each of its cores. As the bucket data is completed, a cyan bar rises from the bottom to the top and turns green upon completion. If the **Network Job Size** is set to a value greater than 100%, the progress bars cycle through one time for each bucket rendered (for the default 200%, the bar fills twice for each job). D

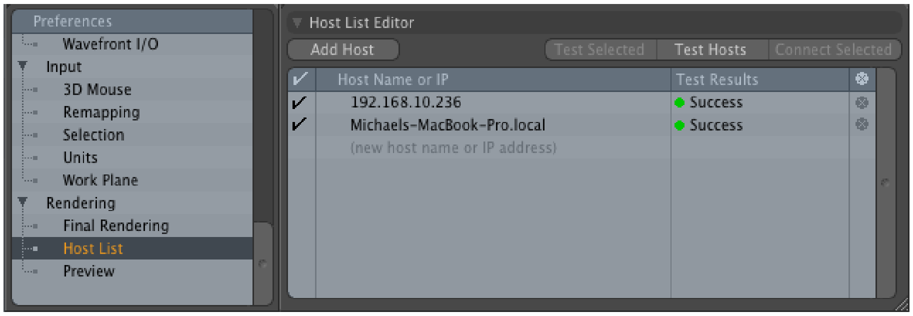
Note: Due to the modal nature of Modo's **Render** window, when it is open, only the **Network** view updates regularly during a render.

Advanced Options

The **Network Render Group** option in the **Preferences** > **Final Rendering** allows systems on a large network to be grouped. This allows you to specify which systems can actively participate in a network render. Any master and worker system with the same network render group name set in their preferences only participates with other systems that specify the same group name.

Host List

If the **Discover Render Nodes via Host List** option is enabled in **Preferences** > **Rendering** > **Final Rendering**, then worker systems can be manually added to a network render configuration by using the **Host List** preference option under **Preferences** > **Rendering** > **Host List**.


Preferences
Wavefront I/O
Input
3D Mouse
Remapping
Selection
Units
Work Plane
Rendering
Final Rendering
Host List
Preview
Host List Editor
Add Host
Test Selected
Test Hosts
Connect Selected

✓	Host Name or IP	Test Results
✓	192.168.10.236	● Success
✓	Michaels-MacBook-Pro.local	● Success
	(new host name or IP address)	

Hosts can be added by hostname or IP, by clicking the **Add Host** button. Once a host has been added to the list and is running Modo in worker mode, you can test the connection to it by selecting it in the **Host List** and clicking **Test Selected**. All systems in the Host List can be tested at once by clicking Test **Hosts**.

When adding Mac hosts, system names usually need to have **.local** added to the computer name.

Modo command line

Modo_cl can now be set to run in worker mode. The **worker** command line flag starts Modo_cl in worker mode and it then operates like any normal worker on the network, but no progress data is displayed in the terminal/console when running in worker mode.

Tip: When using a network firewall, make sure that Modo has access to the necessary ports for network rendering. Blocked ports is a common cause of network rendering issues, such as idle workers.

Best Practices

Here are a few best practices to help ensure that network rendering is a smooth process:

- When rendering scenes with slow irradiance pass calculations (the part of an irradiance cache enabled global illumination render that takes place before the actual buckets are rendered), the most powerful system (fastest at rendering) in your network should be used as the master, as it is doing all of the IC pre-pass calculations.
- In general, using the most powerful system as the master (that is, the fastest at rendering) yields the best results.
- Network speed has a direct effect on network rendering performance. We recommend having at least 100Base-T wired Ethernet; 1000Base-T Gigabit Ethernet is even better.
- Hosting the assets via a shared network drive as opposed to using send and receive assets option helps to speed up transfer times when rendering big scenes (file size-wise) with many assets.

Known Issues

There are a few considerations you should note when using network rendering.

- The use of the CAD item from the CAD loader plug-in is not supported; convert the CAD item to a mesh prior to network rendering.
- Rendering with **Write Buckets to Disc** is not supported for network rendering.

If you are having difficulties with network rendering, see the Network Rendering Troubleshooting topic.

Network Rendering Troubleshooting

Our support department has come up with a brief guide that may prove useful to troubleshoot issues with network rendering.

If the machines can see each other with Bonjour (show up in the **Network** view) but cannot connect, there are three possible issues:

- Firewall is blocking the connection
- Unable to resolve the name of the machine into an IP address
- Unable to reach the machine through TCP/IP

When the firewall is blocking the ports, you need to ensure the necessary ports are available for Modo. Please reference the documentation for the network firewall that is in use for information on opening ports.

The other two possibilities can be tested with the **ping** command in a terminal or an MS-DOS prompt:

```
ping machineName.local
```

Ping must be able to get through the firewall for this to work. This command attempts to resolve the IP address for the machine with the name **machineName** on the LAN. It should return something like this:

```
PING machineName.local (192.168.1.184): 56 data bytes
 64 bytes from 192.168.1.184: icmp_seq=0 ttl=128 time=1.487 ms
 64 bytes from 192.168.1.184: icmp_seq=1 ttl=128 time=0.732 ms
 64 bytes from 192.168.1.184: icmp_seq=2 ttl=128 time=4.758 ms
```

Press **Ctrl**+**C** to stop the command. This indicates both that the machine can be reached (the "64 bytes from..." lines) and that the machine name can be resolved into an IP address (in this case, 192.168.1.184).

If the machine name can't be resolved, you get back something like this:

```
ping: cannot resolve machineName.local: Unknown host
```

This indicates that you have a problem with your network. If the machine name can be resolved but not reached, you get the following:

```
PING machineName.local (192.168.1.184): 56 data bytes
 Request timeout for icmp_seq 0
 Request timeout for icmp_seq 1
 Request timeout for icmp_seq 2
 Request timeout for icmp_seq 3
```

This means that the name was resolved to 192.168.1.184, but the packets aren't getting through to the other machine. This also suggests that there is a problem with your network. If you can successfully ping the other machine, then it might make sense to contact your network support specialist.

Render Display

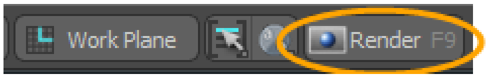
The **Render Display** window provides a variety of useful options and feedback for working with the resulting rendered pixels for any Modo scene.

Accessing Render Display

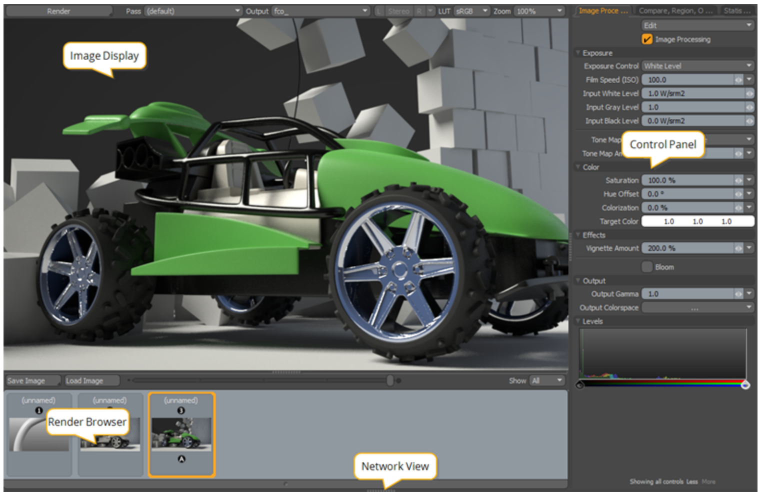
There a number of ways to access the Render Display viewport.

- On the main menu bar, click **Render** > **Render** to open the Image Display and render pixels.

- On the main menu, click **Render** > **Open Render Window** to open the Image Display without rendering any pixels.
- With items selected in the **Items List**, invoke any of the Render Commands.
- With the **Render** layout open, click the start arrow in the Image Display area.
- With any layout open, under the layout menu, click **Render**.


Work Plane
Render F9

Once open, the window provides a view of the rendered scene, along with a number of useful statistics, comparing tools and image adjustment controls for fine tuning an image. The window is broken into four main areas that are all inter-related: Image Display, Control Panel, Render Browser, and the Network View (minimized by default).


Render
Pass (default)
Output fco_
Stereo
LUT sRGB
Zoom 100%
Image Display
Image Proce ...
Compare, Region, O ...
Statis ...
Control Panel
Save Image
Load Image
Show All
(unnamed)
Render Browser
Network View
Showing all controls Less More

Edit
☑ Image Processing
Exposure
Exposure Control White Level
Film Speed (ISO) 100.0
Input White Level 1.0 W/srm2
Input Gray Level 1.0
Input Black Level 0.0 W/srm2
Color
Saturation 100.0 %
Hue Offset 0.0 °
Colorization 0.0 %
Target Color 1.0 1.0 1.0
Effects
Vignette Amount 200.0 %
☐ Bloom
Output
Output Gamma 1.0
Output Colorspace ...
Levels

Image Display

This is the main area of the **Render Display**. This is both a final image display and a source of useful visual feedback to the progression of the rendering frame. Directly above this window are the controls that determine what is actually displayed:

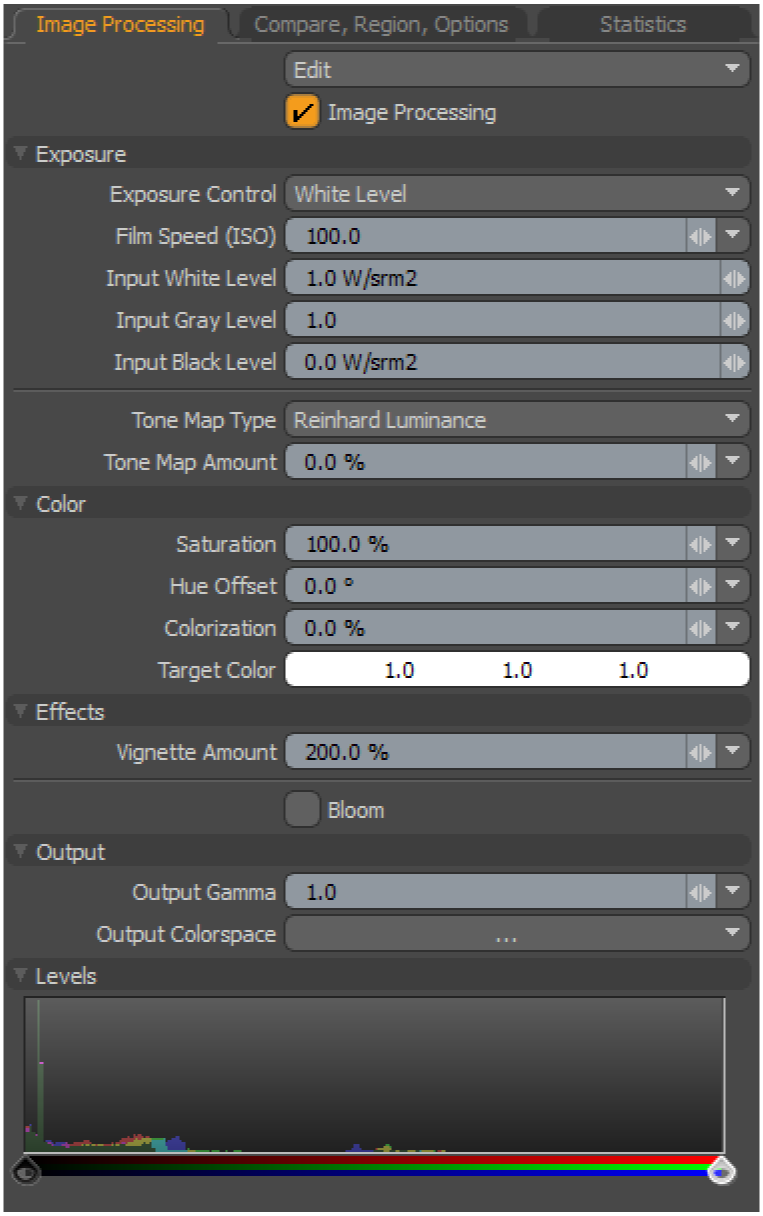
- **Render Type** - Sets what is rendered in the **Render Display**, such as **Render All**, **Render Passes**, and **Bake Texture**. Click and hold the **Render Type** button to display the available options.
- **Render Pass** - Sets which pass or passes are rendered from the specified **Render Type**.
- **Render Output** - Sets the output type rendered from the selected **Render Type**, such as Full Color Output (**fco_**).
- **Stereo Options** - Sets which individual view to display from stereo renders, or how to display both views simultaneously, such as anaglyph or side-by-side.

- **LUT** - The LUT currently selected for display purposes on your monitor (not to be confused with the output colorspace).
- **Zoom** - Controls the size of the rendered frame. You can also zoom in the window using **Ctrl**/**Cmd**+**Alt**+click and drag. Once zoomed in, click and drag to navigate across the image. Pressing **Ctrl**/**Cmd**+**A** fits the entire image to the viewable area.

Control Panel

The **Control Panel** is organized into three tabs, each with a dedicated function:

- **Image Processing** - Provides non-destructive controls to adjust, correct, and tone the rendered image.
- **Compare, Region Options** - Provides tools for rendering limited regions and comparing subsequent and prior renders.
- **Statistics** - Provides dynamic information regarding the frame during rendering, such as memory usage, elapsed time, and the final numbers of polygons.



Edit

☑ Image Processing

Exposure Control: White Level
Film Speed (ISO): 100.0
Input White Level: 1.0 W/srm2
Input Gray Level: 1.0
Input Black Level: 0.0 W/srm2
Tone Map Type: Reinhard Luminance
Tone Map Amount: 0.0 %
Saturation: 100.0 %
Hue Offset: 0.0 °
Colorization: 0.0 %
Target Color: 1.0 1.0 1.0
Vignette Amount: 200.0 %
☐ Bloom
Output Gamma: 1.0
Output Colorspace: ...

Note: The **Levels** histogram is hidden by default. Click the **Showing Standard Controls More** button at the bottom of the panel to display **Levels**.

Image Processing

The **Image Processing** section provides controls to adjust the look of the final rendered image. It can replace an external image editor, but best of all, it's non-destructive. Adjusting the values in the panel updates the **Image Display**, providing real-time feedback to the changes, making it easy to experiment and try different combinations.

Option	Description
Edit	Provides options for moving image processing settings between the render output and the **Render Display** window. • **Copy From Scene** - If adjustments were made to the render output options after the rendered image completes, the **Copy from Scene** option grabs those settings, updating the options of the **Render Display** window to match. • **Paste to Scene** - Any adjustments made to the rendered image within the **Render Display** window do not apply to subsequent renders, as the **Render Display** window reverts to the settings of the **Render Item**. The **Paste to Scene** option copies the settings to the render output back to the scene, applying it to all subsequent renders. • **Reset** - Returns all image adjustment options to their default values.
Image Processing	Toggles the effect of the **Image Processing** controls on and off, allowing you to quickly switch between the previous image and adjusted image.
Exposure	
Exposure Control	Sets the exposure preset to apply to the image: • **White Level** - Defines the amount of luminance (illumination brightness) that corresponds to pure white in the rendered image, adjusted with the **Input White Level** value. • **Photographic** - Produces similar results to **White Level**, but uses the **Film Speed (ISO)** setting more familiar to photographers.
Film Speed (ISO)	Sets the sensitivity of the film or sensor when **Exposure Control** is set to **Photographic**. Higher values simulate increased exposure time, but without any added noise.
Input White Level	Specifies the radiance or luminance level that corresponds to pure white in the final rendered image (a pixel color value of 1.0), when **Exposure Control** is set to **White Level**. Higher values tend to darken the image, requiring a much brighter area to represent the whitest white, but can also be used to bring out detail in overblown areas.

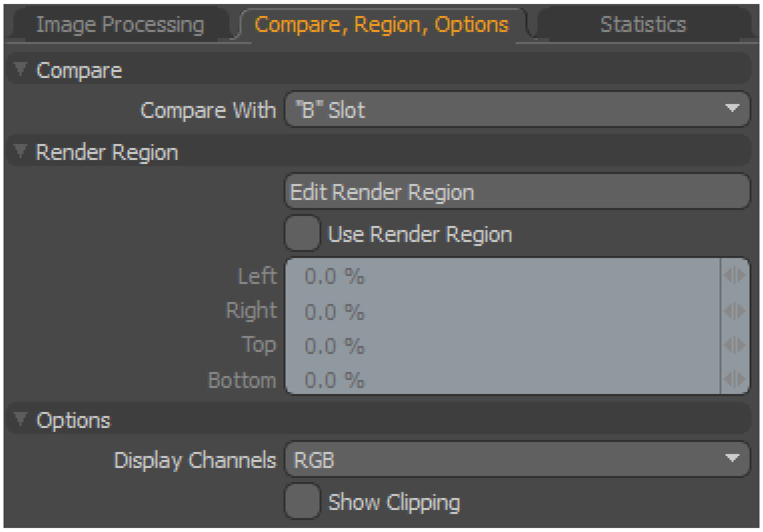
Option	Description
	Note: To use the input level controls effectively, **Clamp Colors** must be disabled in the **Shader Tree** > **Render Output** sub-tab.
Input Gray Level	A non-linear luminance adjustment that applies a curve-like function to the pixels, modifying the mid-range the most while attenuating the adjustment amount for values moving toward the defined **Black** and **White** points. Values above 1.0 lighten the mid-range, while values below 1.0 darken the mid-range.
Input Black Level	Specifies the radiance or luminance level that corresponds to pure black in the final rendered image (a pixel color value of 0.0). **Note:** To use the input level controls effectively, **Clamp Colors** must be disabled in the **Shader Tree** > **Render Output** sub-tab.
Tone Map Type	Sets the mapping type used to control tonal balance. Select the required type and then use the **Tone Map Amount** to control the range.
Tone Map Amount	Controls the overall tonal balance of the rendered image, compressing the rendered dynamic range into something viewable, where shadows and highlights can reveal previously obscured details. Often, it takes a combination of settings for white level, tone mapping, and gamma to achieve the best results. For further examples, see Tone Mapping. Dynamic range is measured as the difference between the darkest shadow to the brightest highlight in an image. Most image formats' dynamic range pale in comparison to that of the world around us. In the same way that a photographer struggles to capture all of the range our eyes can see, you may find your renders have shadow areas that are too dark, and highlight areas that are too bright. Luckily, Modo renders in full floating point accuracy, providing dynamic range well beyond what any monitor is capable of displaying. **Note:** To use the input level controls effectively, **Clamp Colors** must be disabled in the **Shader Tree** > **Render Output** sub-tab.

Option	Description
Color	
Saturation	Controls the concentration or amount of color in an image independent of the luminosity of brightness. At 100%, colors are fully saturated as defined in the item's material settings. Reducing the value reduces the overall color saturation to a value of 0%, producing an image only with gray shades. **Tip:** Modo renders in full 32-bit floating point accuracy, so values above 100% increase saturation without introducing the color banding and artifacts generally associated with over-saturating images.
Hue Offset	Adjusts the color values of the rendered image independently of the luminosity or brightness values, shifting them across the entire spectrum in a sequential fashion. Hue values are arranged on a standard colorwheel, where a rotation of 180° inverts all the color values. For example, shifting a red color in a positive direction adjusts reds toward an orange hue, then yellow, and so on.
Colorization	Introduces an overall color tint into an image. This can be purely for artistic reasons, but can also be helpful in reversing color casts introduced by image-based lighting among other things. Define a tint color using the **Target Color** control and then adjust the **Colorization** amount to determine the strength of the color in the image. A value of 0% produces no colorization, attenuating toward 100% where the image is replaced fully by the **Target Color**.
Target Color	Specifies the tint color when the **Colorization** control is set above 0%.
Effects	
Vignette Amount	Setting **Vignette Amount** to 100% creates a physically accurate vignette simulation, a natural reduction of the brightness of an image toward the outer edges of the frame, based on the camera's focal length and film size. Values above 100% darken the resulting vignette, and values below 100% reduce its visibility.

Option	Description
Bloom	When enabled, two additional options display, **Threshold** and **Radius**. Adjusting these controls provides near real-time feedback in the **Render Display** window. In photography, when a very bright part of an image neighbors a very dark part, the bright part appears to glow. This phenomenon is known as **Bloom**. Modo's **Bloom** simulates this effect by removing excess energy from a pixel and distributes it around to its neighbors.
Bloom Threshold	Sets the lower threshold to which pixels are effected, when **Bloom** is enabled. When set to 100%, only the hottest specular highlights bloom, whereas at 0%, every pixel in the scene receives the effect. For examples, see Basic Render Outputs. **Note:** This control is only available when **Bloom** in enabled.
Bloom Radius	Sets the distance that energy is distributed, in effect controlling the size of the glow. Higher values can impart an ethereal look to the rendered image. For examples, see Basic Render Outputs. **Note:** This control is only available when **Bloom** in enabled.
Output	
Output Gamma	Controls the amount of gamma applied to the rendered image, for both display and when the image is saved. It can be used as an image adjustment control to lighten or darken the image without the need to externally adjust it in an external bitmap editor. The gamma adjustment is non-linear and affects the mid-tones of the image more than the shadows and highlights. For best results, leave the gamma value at the default 1.0.
Output Colorspace	Defines the colorspace of the rendered frame. When set to **default**, the colorspace definition is derived from the Scene Item settings. Changing the value here results in an override that can be set per output.

Option	Description
	The default settings generally produce the correct results, but you can experiment as required due to the non-destructive nature of the Render Display. For more information on color management, see Color Management.
Levels	
Levels	Slides all color values up or down the value scale, shifting them identical amounts in the rendered image. For example, if an image has a 0% black pixel and a 50% gray pixel, offsetting the values 50% results in the black pixel changing to 50% gray and the gray pixel changing to 100% white with all other values changing with the corresponding amounts. **Note:** The **Levels** histogram is hidden by default. Click the **Showing Standard Controls More** button at the bottom of the panel to display **Levels**.

Compare, Region, Options



Compare With "B" Slot

☐ Use Render Region

Left 0.0 %

Right 0.0 %

Top 0.0 %

Bottom 0.0 %

Display Channels RGB

☐ Show Clipping

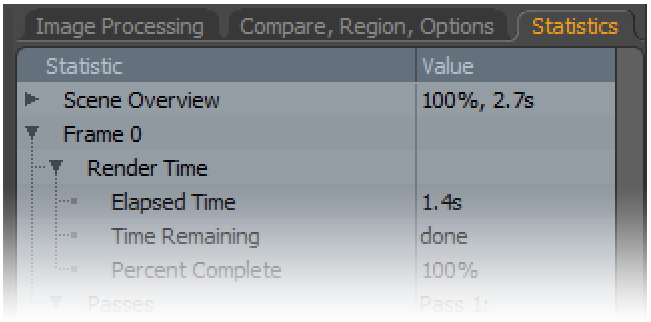
Option	Description
Compare - These options allow you to examine two images in a variety of ways, making it easier to contrast any differences between them.	

Option	Description
Compare with	Determines which opposing images are compared: • **None** - No image comparison • **"B" Slot** - Two separate images are compared. **Slot A** is determined by the current image selected in the browser and **Slot B** is determined by holding the **Ctrl/Cmd** key and clicking on a second image slot in the browser. A small **A** and **B** icon appears beneath each image, signifying which is which. • **"A" Slot Output** - Compares passes or outputs associated with a single image, selected using the **Pass** and **Output** controls.
Pass	When the **"A" Slot Output** option is selected, determines which **Pass** to compare against the default pass.
Output	When the **"A" Slot Output** option is selected, determine which **Output** to compare against the default output.
Apply Image Processing	Toggles the effect of the **Image Processing** controls on and off, allowing you to quickly switch between the original image and adjusted image.
Compare Mode	Several modes define how the two images are compared, each with additional options. • **Wipe** - The images are overlaid and the top image is wiped away, revealing the lower image. Use the **Wipe Method**, **Position**, and **Opacity** Controls to set the wipe behavior. • **Spotlight** - Reveals the **B** images as a circle around the pointer (move over the image to see the results). The size of the circle is determined by the **Spotlight Radius** control. • **Side by Side** - Displays two half-width images next to each other.
Wipe Method	When **Compare Mode** is set to **Wipe**, allows you to select **Horizontal**, **Vertical**, or **Dissolve** wiping.
Position	When **Wipe Method** is set to **Horizontal** or **Vertical**, sets the position of the wipe widget in the viewport.
Opacity	When **Wipe Method** is set to **Dissolve**, sets opacity of the top image to reveal the image underneath.

Option	Description
Spotlight Radius	When **Compare Mode** is set to **Spotlight**, allows you to control the radius of the spotlight.
Stack Vertically	When **Compare Mode** is set to **Side by Side**, enable this control to stack the images vertically instead of horizontally.
Difference	Calculates the pixel variance between the two images and displays it as a grayscale image. Black indicates no difference between the two pixels, attenuating toward white, which indicates the maximum difference. The amount of difference displayed can be scaled using the **Difference Scale**. Each of the available options displays the difference between the two images for the chosen aspect, such as the difference between the **Hue** of both images, or the difference of the **Saturation** amount. Utilizing the **Difference** option can be especially helpful in pinpointing subtle differences between two similar images. **Note:** This is mostly for high dynamic range images (HDR), but in some cases it provides a better indication of dramatic or subtle differences. The difference calculation is the same math used to determine the difference between two numbers (subtracting one value from the other).
Difference Scale	Scales the amount of difference for the selected **Difference** option. **Note:** This control is only available when **Difference** is set to something other than **none**.
Render Region - Allows you to selectively render a smaller segment of the entire image, making the iterative process of creating a final rendered image faster and more fluid. For convenience, these settings duplicate the functionality of the render item's **Render Region** functionality. For more details on the **Render Region**, see Render Item: Frame	
Edit Render Region	When enabled, you can drag directly over the rendered image to define the limited region area interactively. When the initial area is defined, edges and corners of the region are highlighted, allowing you to further refine the region. Click and drag the highlighted border to adjust.

Option	Description
	If edits are made to the same scene, the area outside the defined region is used as a background to successive renders until disabled. When switching between scenes or image resolutions, the background area outside the region renders black.
Use Render Region	Temporarily enables and disables the **Edited Render Region** without losing any of the settings.
Left/Right/Top/Bottom	Allows you to specify the total rendered region width and height manually.
Options	
Display Channels	All resulting rendered images are generated as standard RGB bitmap format images. For diagnostic purposes, it may be useful to see the individual **Red**, **Green** or **Blue** color channels of an image.
Show Clipping	When enabled, a diagonal black and white stripe pattern (called a zebra pattern) displays over overexposed or blown out areas of the image, providing useful feedback for adjusting image white point and tone mapping an image to compress as much dynamic range information as possible.

Statistics

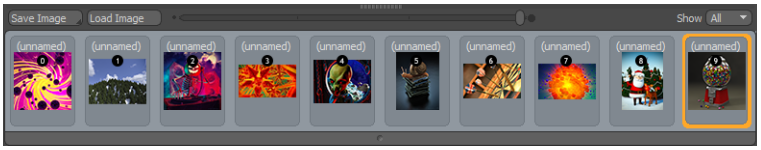


Statistic	Value
▸ Scene Overview	100%, 2.7s
▾ Frame 0	
▾ Render Time	
Elapsed Time	1.4s
Time Remaining	done
Percent Complete	100%
▾ Passes	Pass 1:

The **Statistics** panel provides a huge amount of statistical information related to rendering the scene. Included in this information is the elapsed time, as well as an approximate remaining time for in-progress frames. For finished frames, the total rendering calculation time is displayed, along with total memory usage, and total number of polygons generated.

Navigation of the list works the same as other viewports. You can open or close a specific section by clicking the preceding arrow (▸). To open or close all sub-sections of a specific topic, press **Shift** and click preceding arrow (▸) to view all the information related to a single frame.

Render Browser



The **Render Browser** allows for an unlimited number of rendered frames. The default is set to show the past 10 rendered frames, but you can change this in the **Options** panel. The browser itself serves a couple of purposes, the most important being that you can select the current image to display in the display area by clicking on any of the icons. The thumbnail of the selected image is highlighted in orange; only the selected image can be modified using the image adjustment controls. The render order of images is also displayed using the small number icons above each image. You can use the keyboard numeric keys to select a specific render slot as well as easily swap between different renders, simply by pressing its corresponding numeric key.

When you hover over an icon, two additional controls appear: an **X** that removes the image from the browser (deleting it from disk) and a small lock icon you can click to stop items being deleted accidentally. Click the lock icon again to unlock. You can select multiple images by holding the **Ctrl**/**Cmd** key, allowing you to remove, lock, or unlock multiple images at once.

The **Render Browser** is also used to load and save images. Images in the **Render Browser** are held in the render buffer until you save them, but you can still view and modify them in the **Render Display**. Using the **Save Image** button directly above the browser allows you to define a format and location for the image. For more information concerning images and formats, see Saving Images.

You can also load images into the browser area, which is very helpful when you want to apply image adjustments, or use previous renders to compare a current render, such as when the render buffer no longer exists. To load an image:

1. Click the **Load Image** button above the browser area.

   An OS-specific browser displays.
2. Navigate to the stored image location and select the target file.
3. Click **Open** to load the file into the browser.

Note: If any modifications are applied to the file, it must be saved again to retain the changes.

Network View

Below the **Render Browser** area is a minimized **Network View** panel. Open the panel by clicking on the small group of tiny squares and dragging it upward. This browser indicates the number of cores available on each machine, machine name, user name, mode, status and message. You can also tell if a machine is connected to your active host by the color of the text:

- **Green** - Indicates that the machine is connected and ready for rendering.
- **Black** - Indicates systems that are connected, but not enabled as workers.

Any other color indicates a problem with the connection or availability.

General View Shortcuts

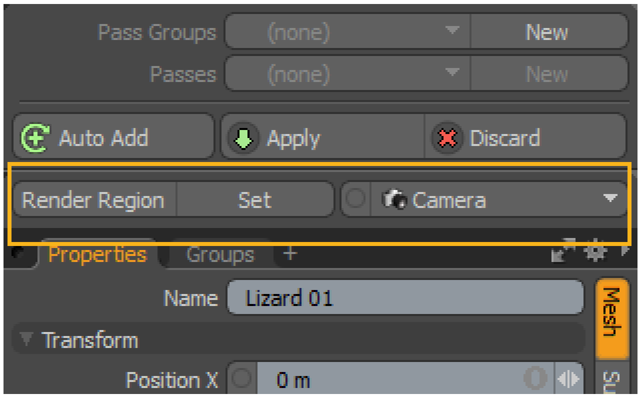
The **Render Display** provides standard Modo controls to adjust size and minimize frames, all the while automatically adjusting the interface, giving you control over what can be seen. Dragging the splitters between the panels scales the contents of each. There are also some additional controls that provide an easy way to quickly make certain panels within the display visible.

- **Spacebar** - Minimizes everything except the image pane.
- **T** - Toggles thumbnail panel visibility.
- **N** - Toggles the network view.
- **I**/**C**/**S** - Toggles the right-hand panel tabs (**Image Processing**/**Compare**/**Statistics** respectively).

Render Region

The 3D process is one of constant iteration and refinement. Rendering entire images repeatedly is a significant waste of time and a disruption to the creative workflow. To alleviate some of this pain, Modo provides the Render Region rendering functionality. You can use the **Left**/**Right**/**Top**/**Bottom** functions to specify the region manually, or use the Render Region tool to specify an area by dragging in any 3D viewport.

You can find the tool in the left panel of the **Render** layout, in the Render Region toolbar. Using the toolbar, you can activate and set a render region in preview, and change the camera being used to render it.



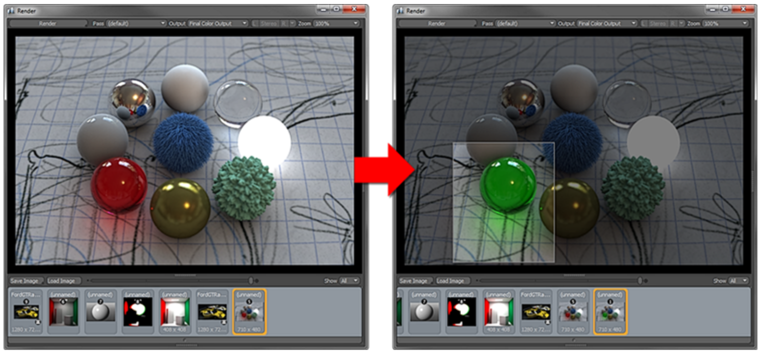
Name Lizard 01

Position X 0 m

To use the tool:

1. Click **Render Region** to activate the tool.
2. To set a region, click **Set**, then click and drag in the 3D viewport to specify area.

   When a render command is invoked (such as by pressing **F9**), only the area within the region specified is rendered.



To change the render camera, click **Camera**, and select the camera you need from the dropdown.

Additionally, the function can be activated through a button of the **Render Display** window, under the **Options** tab.

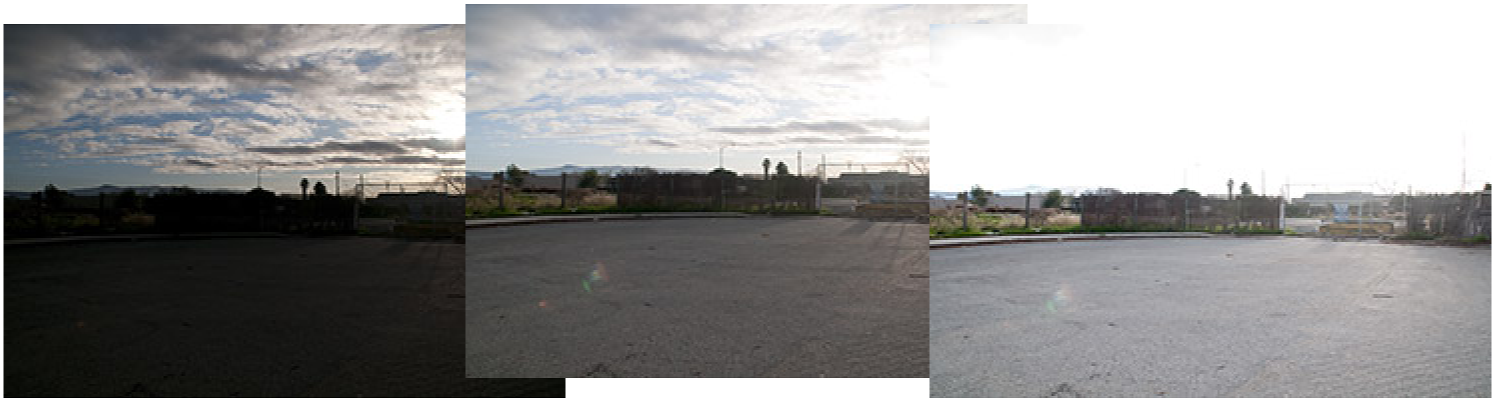
Tone Mapping

Tone mapping, for computer graphics, is simply the adaptation of contrast across a range of values. The reason for the manipulation can range from strictly aesthetic to purely technical. Tone mapping as it relates to Modo is often meant for the compression of dynamic range in the rendered image; dynamic range being the ratio between the brightest value and the darkest value.

Let's stray for a moment to propose an example to illustrate dynamic range. If you look at the world around you, there are areas that are extremely bright, such as a window with a good view of the sky, and areas that are very dark, possibly under your desk. The total range of values between the darkest area to the brightest in any scene is known as the dynamic range.

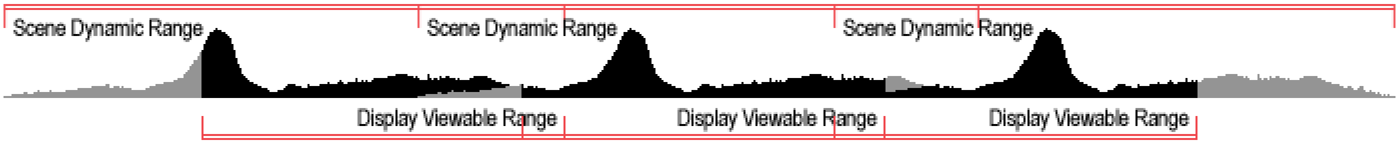
While your eyes are able to adjust appropriately to see detail in both areas, most image capture and display devices cannot. If you take a photo of this scene that has the window and the shadow area beneath the desk both visible, the shot can be exposed either for the area under the desk, overexposing any detail visible through the window, or exposed for the window, underexposing the shadow area, losing any detail in total blackness. Even if the capture device could capture the entire possible range, no means of viewing it could display it properly. A photograph of the sun would never sting your eyes when viewed on a computer monitor or printed in a book the same way the real sun does.

Lets revisit this example from the Image Based Lighting topic. Various exposures of the scene were captured and combined producing a new image that has an approximation of the total illumination in the scene (the full dynamic range of light to dark).

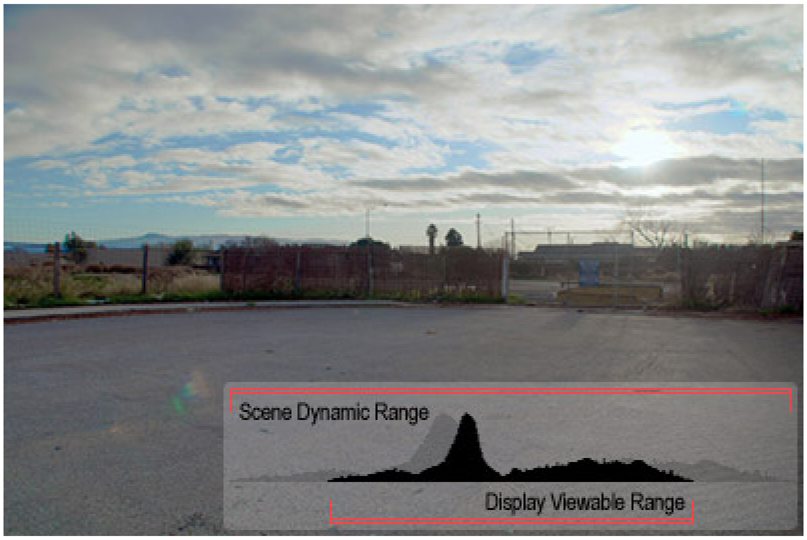


Sky exposure (-2 EV) Default Exposure (0 EV) Foreground Exposure (+2 EV)

Once the HDR image is created, all that extra information is still, in a way, hidden until you tell the system how to display it properly. This is where tone mapping comes in. Most modern displays are based on technology that greatly limits the visible dynamic range. As with the exposure examples above, areas outside this range are simply clipped, or cut off. Consider these histograms (graphs of the brightness values); a normal full range HDR can only display a select range of values at any one time, producing a similar effect as the original exposures above.

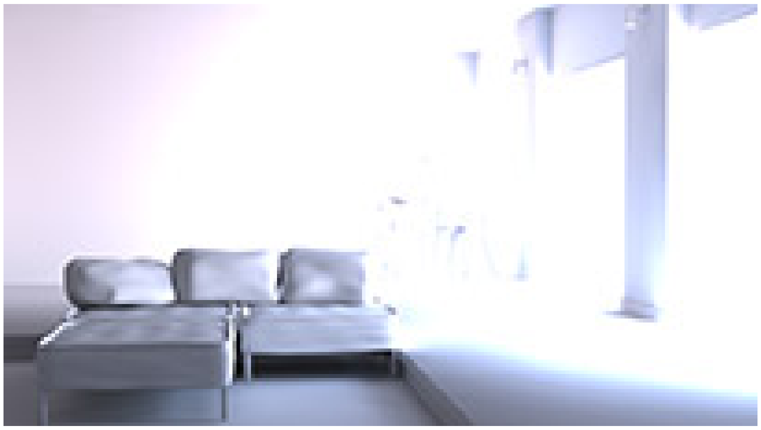


Tone mapping can compress the total dynamic range of the image into a range that is viewable on your screen. Here is a simple example of the combination of the above sequence of images tone mapped with an example of the compressed histogram.



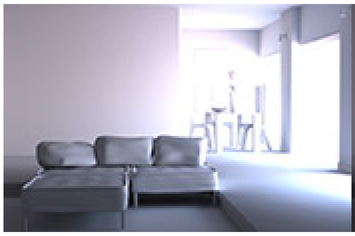
Unfortunately, it's not a simple push-button process, as the result of the tone mapping is highly dependent on the image. Each image reacts differently; some becoming too dark and/or bright, while others look dull and gray. Modo offers a variety of functions that can help resolve these issues. Now that you, hopefully, have a better understanding of dynamic range, lets visit them.

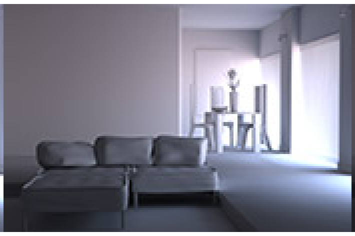
Modo renders in a full floating-point precision format. That sounds very technical, but what it really means is the rendering engine is just very precise with the values it generates, including capturing the full brightness range from the scene producing HDR images with a much higher degree of accuracy than those generated from multiple photographic exposures. Among other benefits, this affords you a lot of leeway in manipulating the tones, avoiding any of the common drawback of typical low-dynamic range image manipulations, such as banding due to posterization. More importantly, the changes are all non-destructive meaning you can make adjustments without compromising this extended amount of information (assuming levels aren't **Clamped**).

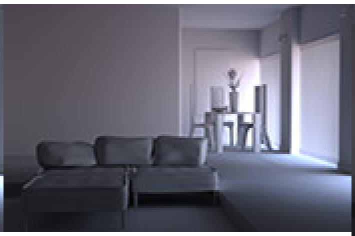


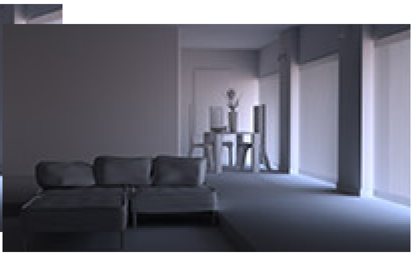
Lets consider the above example, rendered using the Physical sky feature, unclamped, with default settings. The image is of a small interior room with an outside facing window, but it's pretty tough to see anything outside of the couch in this render. It's pretty blown out and looks completely unusable, but hidden in those overly bright pixels is the detail we want.

The first setting to consider is the **White Level**, specifically this sets what pixels are pure white in the image, producing a result similar to how exposure functions in a camera. Lower values would seem overexposed, though increasing values won't underexpose the shadow areas. Below, raising the white level value increasingly reveals the details in the windows and rear of the room but, as the other pixels adjust accordingly, it has the unfortunate side effect of darkening the mid-tone values. These can be adjusted with Gamma.

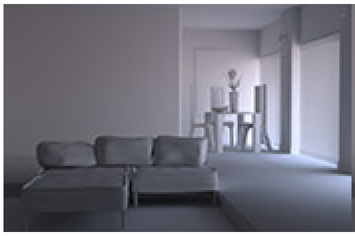

White Level 2.0

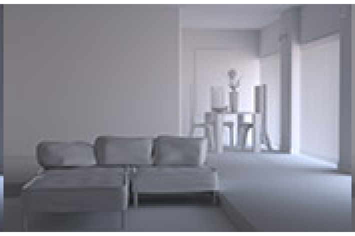

White Level 4.0

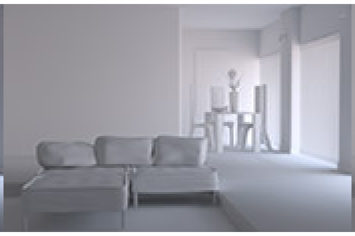

White Level 6.0

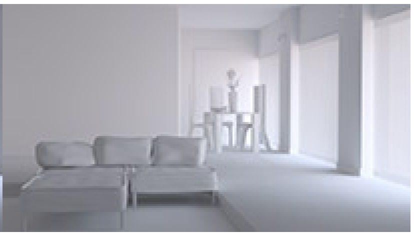

White Level 8.0

The gamma function adjusts the distribution of tones, modifying the mid-tone values to a much greater degree than the lightest and darkest values in the image. Increasing the curve of the gamma (by raising the value) you can see it lighten up those mid-tones while keeping the darkest shadows and the lightest areas produced by the white level adjustment made earlier.

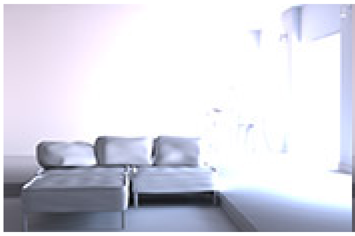

Gamma 2.2


Gamma 3.0

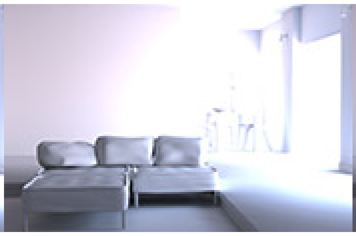

Gamma 3.8


Gamma 4.6

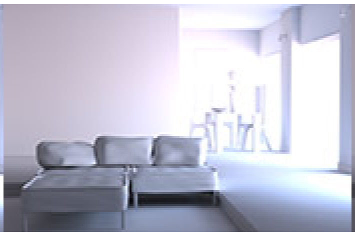
Now, if we take a quick step back to the default render above, we can see the addition of just the **Tone Mapping** setting to our raw render. When set at 100%, this function compresses the full dynamic range of the unclamped render into the same limited range that's visible on our monitors, revealing all the detail of the blinds in the window and the rear of the room but, unlike setting the **White Level**, the mid-tones are relatively untouched. Only problem is, at 100% the room also looks relatively flat. Including some of the contrast in the initial render helps create a more interesting image, and takes adjustments from all three settings for the best overall result.



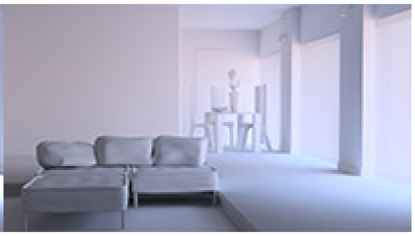
Tone Mapping 25%



Tone Mapping 50%

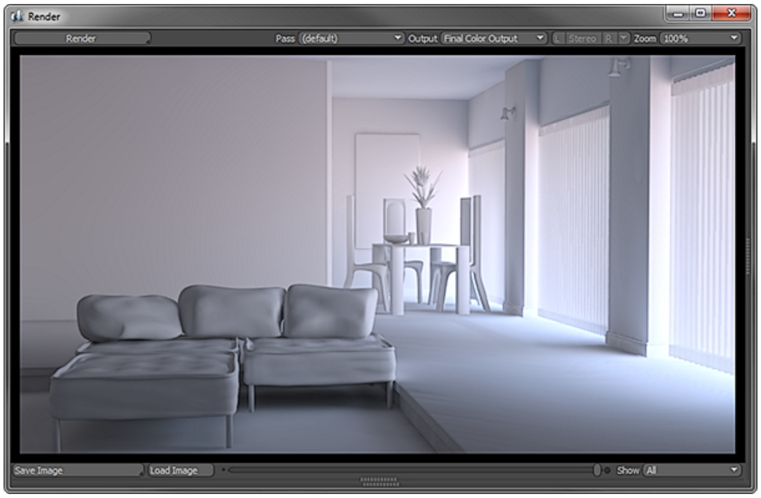


Tone Mapping 75%



Tone Mapping 100%

The perfect image is always a matter of personal preference, but below the white level has been adjusted to 3.0, the gamma to 2.2, and tone mapping amount set to 80% for the following result.



It keeps some of the subtle coloration introduced by Physical Sky, resolves a lot more detail in the vertical blinds and windows, while retaining a pleasing shadow under the foreground couch. All of this without a single second of additional rendering time either. All these settings are applied effectively in real time. Compare the image below to the raw render output, you'll quickly realize the power of these three seemingly simple settings.

Tone mapping isn't just for global illumination, image based lighting, or Physical Sky either; it can help in lots of instances. It won't solve all lighting problems, but it can add finesse and refinement to seemingly difficult situations, and save you a trip to your image editor. The best way to build familiarity with the settings is to experiment.

Tip: Many modern image editing programs are beginning to include functionality for working with HDR images, with full complements of tone mapping functions. If you prefer to work in another tool, simply save the rendered image unclamped (and undithered) to a format that preserves the extended information, such as EXR or HDR, and open in your favorite HDR image editor.

Color Management

Color management is a process that controls the conversion of colors when moving between different formats, systems, and devices. The primary goal of color management is to maintain accurate colors across the span of inputs and outputs. It is built using the academy award-winning OpenColorIO technology that is geared toward film and video production with an emphasis on visual effects and computer animation.

Color Management Overview

Color management in Modo works by applying a profile to a color input. A profile is a set of instructions that describe colors relative to a known standard, and the color input specifically would be color bitmap images imported into Modo (textures, backdrops, environments, and so on). Based on the assigned profile, the colors in the image are transformed to a linear colorspace format or to a numeric value colorspace format for internal processing and rendering.

A colorspace is a mathematical model that describes how colors are represented for a given format. Converting the bitmap image colors to a linear colorspace or to a numeric value colorspace ensures that image color fidelity is fully preserved during the rendering process.

The next part of the color management process is the output, where rendered frames (still or animated) are then displayed in the Render Display, where a destination transform is applied. The rendered image is stored in memory unaffected in the linear format, with the transform applied non-destructively by the viewport using a LUT (Look Up Table) which is a pre-defined set of translation operations from one colorspace to another. This allows you to properly preview how the final image should look, without adversely affecting the actual rendered frame. This also allows the image to be saved linearly using a high dynamic range format for additional processing in another application further down the pipeline, ensuring continued color fidelity. Colorspace transforms can be applied to the image when saving to a low dynamic range format for direct use or processing in a non-color managed application, retaining the look of the image from the **Render Display** window.

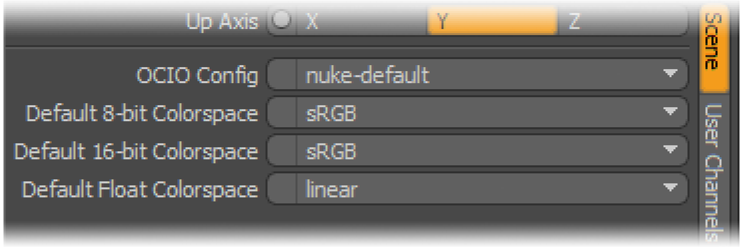
Note: Modo ships with default color management settings that should work perfectly fine for most artists without further adjustment. The following controls are likely to be only edited by advanced artists with specific needs.

Note: When using color management, it is no longer necessary to de-gamma an image for rendering, this is handled automatically by the color management system. Defaults for all Gamma settings are now set to 1.0.

Modo Workflow

A color profile is assigned automatically to any RGB bitmap image imported into Modo, single channel grayscale images are always treated linearly. Determining which profile is assigned to which image type starts with the color management settings found in the Preference Settings. These values specify the initial, default settings assigned to a new scene. Changing them doesn't directly affect older scene definitions or any scenes that are currently open. It simply determines the default values for newly created scenes.

When a new scene is created, you can access the scene-specific color management settings by selecting the scene item (topmost item in the **Items** list with the movie clapperboard icon). When selected, options are available in the **Properties** panel.



Up Axis: X Y Z

OCIO Config: nuke-default

Default 8-bit Colorspace: sRGB

Default 16-bit Colorspace: sRGB

Default Float Colorspace: linear

With these settings, you can globally modify the profiles for all images within the same scene using a single setting, instead of having to manually change each and every image. Of course, it's also possible to override this setting on individual images. This is done by changing the colorspace control on the image map item from default, which means it references the scene item setting, to a specific colorspace. The setting is found under the **Image Still Properties** sub-tab when the image map item layer is selected. Once defined this way, the image map retains the setting regardless of changes to the scene item colorspace settings.

Once the proper colorspaces have been defined, processing for color management is all done in the background, and requires no further input. The proper colorspace definitions allow Modo to accurately convert from the source colors to a linear colorspace for internal processing and rendering. For image display, you may need to adjust the **LUT** settings in the **Preview** and **Render Display** windows, depending on monitor type. In most cases, the default option is fine as most monitors use the sRGB colorspace to display the output from the video card.

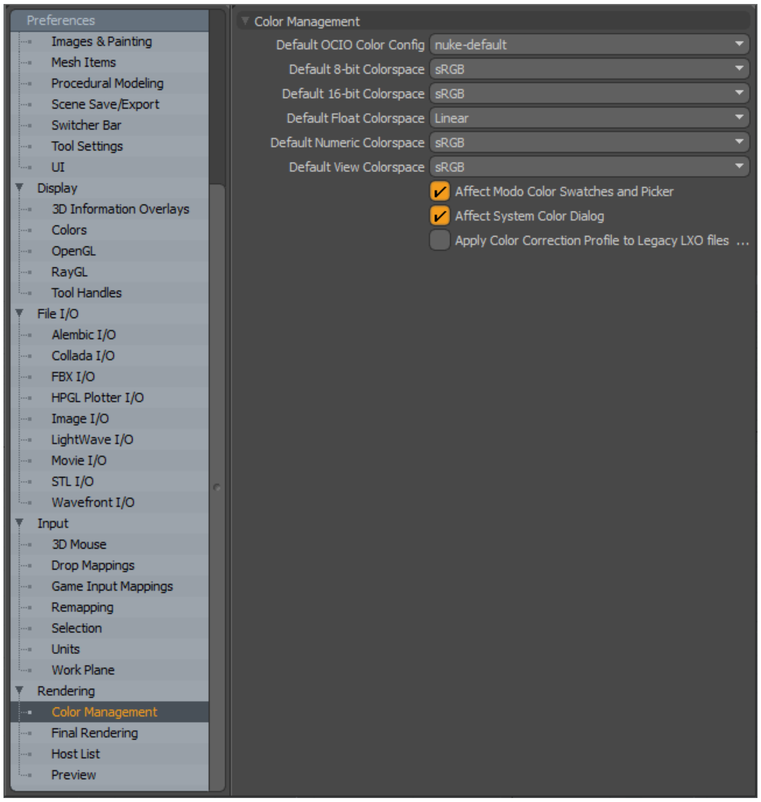
Saving images assigns color profiles using the same scene-based preference settings that are initially defined as the default value. When images are saved from the **Render Display** or from the **Preview** window, the **Colorspace** option from the bottom of the **Image Processing** tab controls which profile is assigned and embedded, should it ever be necessary to override the default value. Overrides can also be applied when saving rendered animations directly from a render output with the setting of the render output itself. Both the **Render Display** control and the render output setting are normally set to **Default**. The profile chosen depends on the bit depth of the target format.

> **Note:** To ensure consistent rendering, Modo opens older scene files (pre-801) with color management disabled. If color management is desired, you need to manually enable it for all items in the scene.

The following section explains the controls specific for color management and where they are located in the interface.

Preference Settings

On Windows, found in the menu bar under **System** > **Preferences...**. On Mac OS, found in the menu bar under **Menu** > **Preferences**. Open **Rendering** > **Color Management** section toward the bottom of the window. The **Color Management** preference settings defines the default values that are inherited by any new scene that is created. Adjusting these settings don't affect older saved scenes, or any open ones.


Preferences
Images & Painting
Mesh Items
Procedural Modeling
Scene Save/Export
Switcher Bar
Tool Settings
UI
Display
3D Information Overlays
Colors
OpenGL
RayGL
Tool Handles
File I/O
Alembic I/O
Collada I/O
FBX I/O
HPGL Plotter I/O
Image I/O
LightWave I/O
Movie I/O
STL I/O
Wavefront I/O
Input
3D Mouse
Drop Mappings
Game Input Mappings
Remapping
Selection
Units
Work Plane
Rendering
Color Management
Final Rendering
Host List
Preview
Color Management

Default OCIO Color Config: nuke-default
Default 8-bit Colorspace: sRGB
Default 16-bit Colorspace: sRGB
Default Float Colorspace: Linear
Default Numeric Colorspace: sRGB
Default View Colorspace: sRGB
☑ Affect Modo Color Swatches and Picker
☑ Affect System Color Dialog
☐ Apply Color Correction Profile to Legacy LXO files ...

Color Management

Default OCIO Color Config: This option controls which configuration is the default for new scenes. Each separate config controls which color spaces are available for use by items in the scene. Each different config contains a different set of profiles. You can also designate a config for individual scenes manually, as an override in the scene item. See Custom OCIO profiles for more information.

Default 8-bit/16-bit/Float Colorspace: Designate the scene's default profile assigned when reading and writing the associated bit depth of image for any items that have their colorspace control set to **Default**. You can also designate colorspace profiles individually on the Image Map item under the **Image Still** sub-tab.

Default View Colorspace: This option defines the colormapping assigned to UI elements, including the Color Picker Viewport, Color Swatches, the Render Display window and the interactive Preview Viewport. The **Perform Color Correction on OpenGL Object** must be enabled in order to see the results of this setting on the 3D Viewports.

Default Numeric Colorspace: Numeric control color correction applies color correction to the numeric values displayed in color fields. This allows you to, for example, copy and paste the hex code for a color from Photoshop into Modo and have the colors match. Previously, Modo supported color correction for the drawn representation of a color in the UI but the numeric values were always drawn in linear space.

Affect Modo Color Swatches and Picker: When this option is enabled, the **Default View Colorspace** option adjusts the display of individual color input fields in the **Properties** form to match the colors of the **Render Display**.

Affect System Color Dialog: When this option is enabled, the **Default View Colorspace** option adjusts the display of the Color Picker Viewport pop-ups and windows to match the colors of the **Render Display**.

Apply Color Correction Profile to Legacy LXO files: Enable this to apply the default color correction profile to pre-Modo 801 **.lxo** files. These files pre-date the use of OCIO. Consequently, it may be desirable not to use Modo's default color correction on loading the files as they may already have been corrected using an alternative method. For this reason it is off by default.

Note: Modo normally initializes scene files if OCIO color configurations are not stored correctly in the file.

Colorspace on Image Maps/Sequences

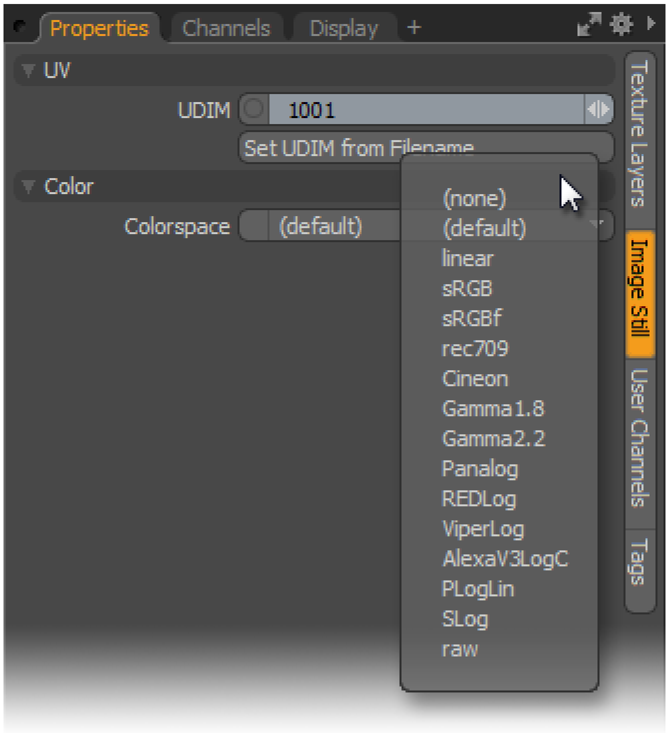
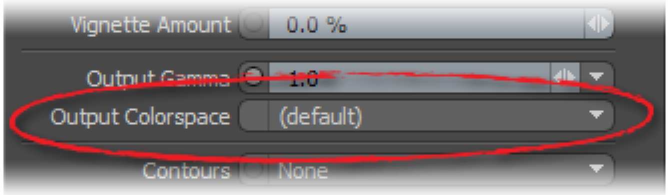

Properties
Channels
Display
+
UV
UDIM
1001
Set UDIM from Filename
Color
Colorspace
(default)
(none)
(default)
linear
sRGB
sRGBf
rec709
Cineon
Gamma1.8
Gamma2.2
Panalog
REDLog
ViperLog
AlexaV3LogC
PLogLin
SLog
raw
Texture Layers
Image Still
User Channels
Tags

Image maps and image sequences that are loaded into the **Shader Tree** are assigned default color management profiles based on the scene item settings. You can override the settings per image, based on the **Colorspace** control in the **Image Still** sub-tab of the **Properties** panel when the target image map item is selected. The options illustrated here are located in the **Nuke-Default** color management config.

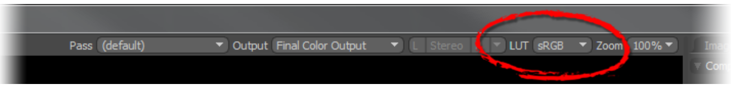
Render Outputs


Vignette Amount
0.0 %
Output Gamma
1.0
Output Colorspace
(default)
Contours
None

The **Output Colorspace** control in the **Render Output** determines the particular color transform (if any) applied to images when saved from Modo, either automatically by the render output when rendering an animation, or when saving from the **Render Display** window. The **Output Colorspace** basically has three different ways it can be set, the **None** option disables color management from affecting images when saved, '**Default** automatically applies the color profiles defined by those set in the scene item, based on the bit depth being saved out from Modo. The third and last option is dependent on the color profile config defined in the Scene, but allows you to override the default colorspace value to a specific color profiles by defining one of the select color profiles available in the drop-down.

Working independently from the render output, there is the identical control available in the **Render Display** window, at the bottom of the **Image Processing** tab, that provides the same functionality as the render output control.

Monitor Color Management



Pass (default)
Output Final Color Output
LUT sRGB
Zoom 100%

At the top of the **Render Display** there is a **LUT** setting (for Look Up Table) where you can specify your monitor's color profile (not to be confused with the render output's **Output Colorspace** profile). When working in a managed workspace, this control allows you to easily swap color table configurations, for instance if the **Render Display** is positioned on different monitors in a multi-monitor setup. In most cases, the default sRGB setting is fine, as most computer monitors default to working within this colorspace.

Tip: Color management can be completely disabled, where Modo works as in the previous version. This is done by setting the color management preference setting to **None** and returning the default **Gamma** value in **Preferences** to **2.2** before creating any new scenes. When this is done, you need to manually de-gamma imported images for proper results (as in previous version of Modo.

Custom OCIO profiles

From Modo 17.1v1 custom OCIO profiles can be added as kits.

For example, to add the ACES 1.0.3 color profile:

1. Create a folder in your configs directory and name it. In this example, the folder is named "myociocfgs". Any additional custom color profiles can be stored in this folder.

Note: You can navigate to the configs folder through Modo by going to **System** > **Open User Configs Folder**.

2. If you already have the ACES 1.0.3 color profile, copy it inside the **myociocfgs** folder. Any color profile folder should contain a folder of LUTs and a **config.ocio** file.

Tip: Public color profiles, including ACES 1.0.3, are readily available from online repositories, such as https://opencolorio.org/configurations/.

3. Copy the following snippet into a text editor.

```
<configuration><atom type="ocio">

<hash type="ociopath" key="aces_1.0.3">user:Configs/myociocfgs/aces_
1.0.3</hash>

</atom>

<configuration>
```

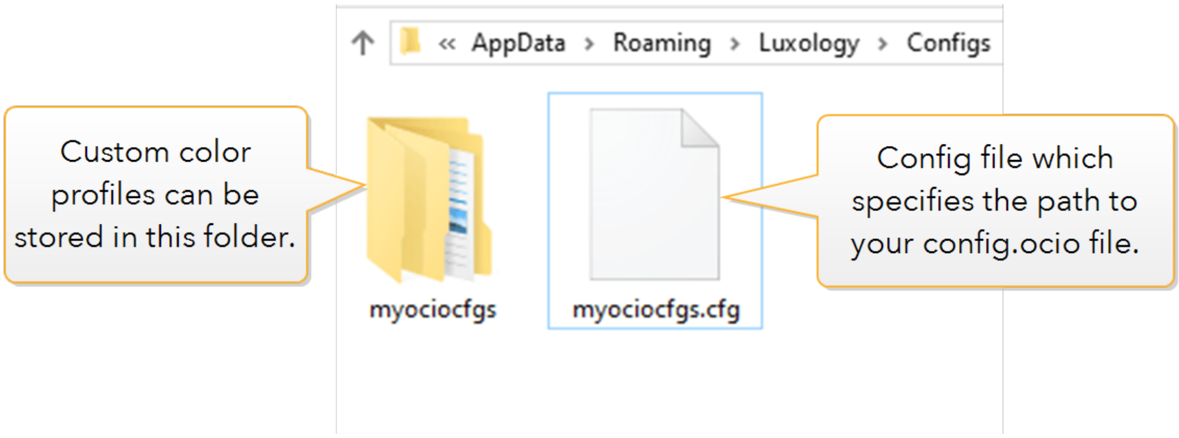
Note: If you copy and paste example code from this user guide into a text editor, line indentations and spacing may not be preserved. If this is the case, correct the indentations manually.

Note: You can embed multiple configs in a group within this text file if you want to add other color profiles.

4. Save the text file as plain text and store it in the configs directory. Config files need to be plain text for Modo to read them. You can give the text file any name but it's good practice to name it the same as the folder containing your custom color profiles.
5. Change your text file's extension to **.cfg**.

   The text file is now a config file, which Modo uses to reference your custom profiles.

Note: Ensure the name of the color profile folder matches the path specified in your **.cfg** file. In this example, the color profile is named **aces_1.0.3** so the path is **user:Configs/myociocfgs/aces_1.0.3**. This is also the name that appears in the **Default OCIO Color Config** options in Modo.


« AppData › Roaming › Luxology › Configs
myociocfgs
myociocfgs.cfg

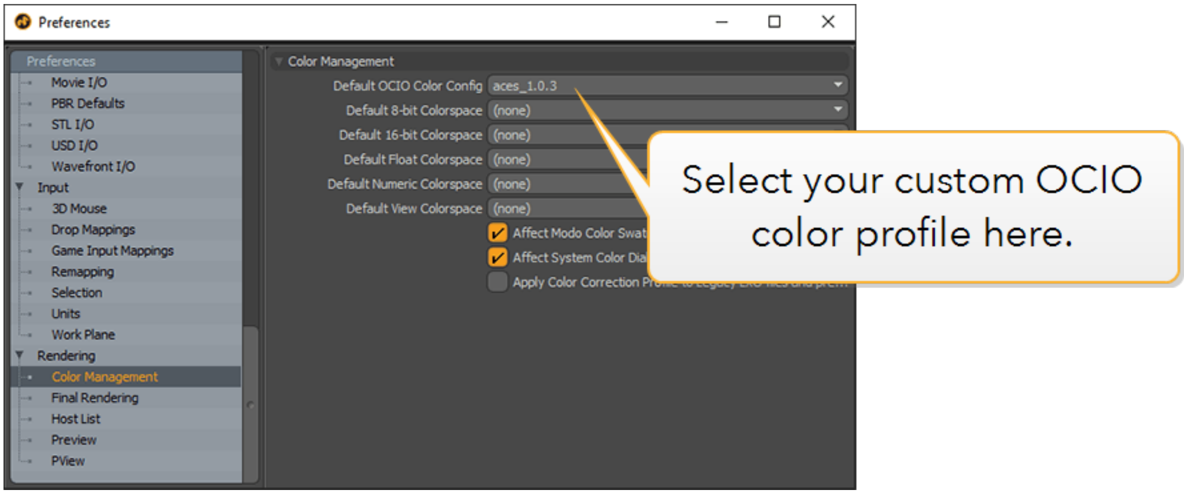
Custom color profiles can be stored in this folder.

Config file which specifies the path to your config.ocio file.

Your configs folder should look like this.

6. Restart Modo, and navigate to **System** > **Preferences**> **Rendering** >**Color Management**.
7. Click the **Default OCIO Color Config** dropdown menu.

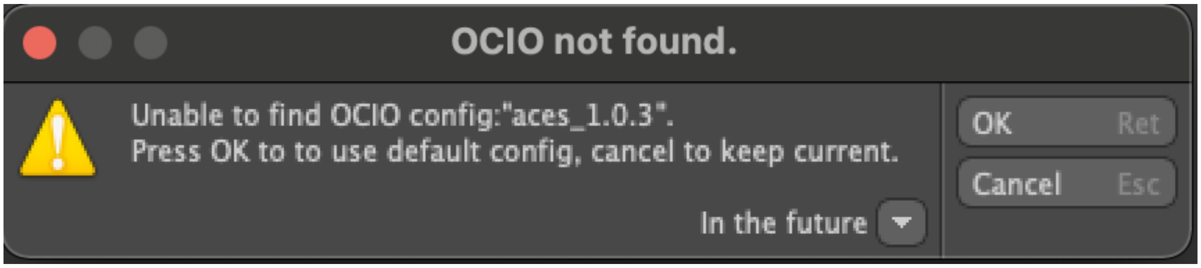
   You should now see the option to select **aces_1.0.3**.


Preferences
Preferences
Movie I/O
PBR Defaults
STL I/O
USD I/O
Wavefront I/O
Input
3D Mouse
Drop Mappings
Game Input Mappings
Remapping
Selection
Units
Work Plane
Rendering
Color Management
Final Rendering
Host List
Preview
PView
Color Management

Default OCIO Color Config: aces_1.0.3
Default 8-bit Colorspace: (none)
Default 16-bit Colorspace: (none)
Default Float Colorspace: (none)
Default Numeric Colorspace: (none)
Default View Colorspace: (none)
☑ Affect Modo Color Swat
☑ Affect System Color Dia
☐ Apply Color Correction Pr

Select your custom OCIO color profile here.

If a scene that contains a custom OCIO Config that isn't present on your system the following warning (or similar) appears:


OK Ret
Cancel Esc
In the future

OCIO not found.

Unable to find OCIO config:"aces_1.0.3".
Press OK to to use default config, cancel to keep current.

If OK is pressed, the missing OCIO config is removed from the scene and the default OCIO config is used instead.

If cancel is pressed, the scene will be loaded and any references to the missing OCIO config are preserved.

Note: Please be aware, that color management may not be correct if references to a missing OCIO config are preserved. However, the original color management will be preserved is the scene is saved and then loaded again with the missing OCIO config present.

When using Modo in headless mode, a warning is also presented if a scene with a missing OCIO config is attempted to be opened. Due to modo-cl often being used non-interactivly, the following environment variable can be set to force Modo to exit the scene if a custom OCIO config is found to be missing:

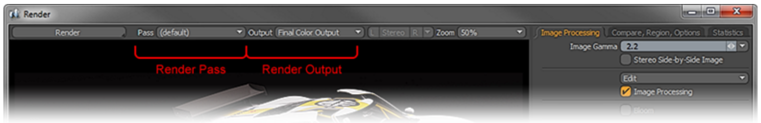
OCIOMISSINGQUIT=1

Saving Images

Modo saves images in various ways, depending on how the image is generated and/or applied. Understanding the individual instances ensures that no image goes unsaved, or is inadvertently saved either.

Saving Rendered Images

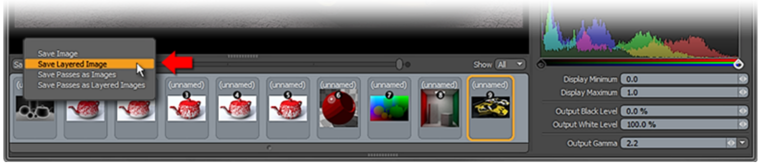
When rendering with the **Render** command (menu bar option **Render** > **Render** or by pressing **F9**), Modo obviously renders the image, but once finished, the image is in a limbo state within the Render Display window. It is cached to disk temporarily for you to preview. This is useful for the iterative process of adjusting scenes. However, the image itself is not truly saved, that is, it isn't accessible to other applications until you specifically tell Modo to do so, using any of the commands necessary for saving; the **Save Image** and **Save Layered Image** functions. Each time any render command is invoked, Modo calculates all render outputs and Render Passes. Once rendered, they appear in the **Pass** and **Output** lists of the **Render Display** (as seen below).



Pass (default)
Output Final Color Output
Zoom 50%
Image Gamma 2.2

When using the **Save Image** option, the currently selected render output and/or pass is saved. Clicking the **Save Image** button found in the upper left of the **Render Browser** area of the Render Display opens an OS-specific dialog, allowing you to specify a location and filename for the image, and most importantly, a format. Modo supports a wide variety of bitmap image formats, each with their own benefits and limitations. By itself, Modo supports the following formats:

Format Name	Extension	Bit Depth per Channel	Compressed	Type
Flexible Precision Image Format	.flx	32 bit	Yes	Lossless
Targa	.tga	8 bit + Alpha	No	
Windows BMP	.bmp	8 bit + Alpha	No	
Radiance High Dynamic Range	.hdr	HDR	Yes	Lossless
JPEG	.jpg	8 bit	Yes	Lossy
Portable Network Graphic	.png	8 bit + Alpha	Yes	Lossless
Portable Network Graphic 16 bit	.png	16 bit + Alpha	Yes	Lossless
PSD (Photoshop Document)	.psd	8 bit	No	
SGI RGB	.sgi	8 bit	No	
Tagged Image File Format	.tif	8 bit + Alpha	Yes	Lossless
16-Bit TIF	.tif	16 bit + Alpha	Yes	Lossless
16-Bit Uncompressed TIF	.tif	16 bit + Alpha	No	
OpenEXR	.exr	HDR	Yes	Lossless



Display Minimum 0.0
Display Maximum 1.0
Output Black Level 0.0 %
Output White Level 100.0 %
Output Gamma 2.2

When using the **Save Layered Image** option (found by clicking and holding the **Save Image** button to open the pop-up menu), Modo saves all available render outputs simultaneously. It opens an OS-specific dialog where you can specify a location a file name and a format for the image. The following format options are available:

Format Name	Extension	Bit Depth per Channel	Compressed	Type
Layered PSD	.psd	8 bit	No	
Multi-File PNG*	.png	8 bit	Yes	Lossless
Multi-File PNG 16 bit*	.png	16 bit	Yes	Lossless
Layered OpenEXR	.exr	HDR	Yes	Lossless
* Multi-File PNG saves all outputs with an appropriate output name appended to each				

Saving Render Passes

Modo's flexible Render Pass system allows you to control virtually any aspects of a scene, and then render the results consecutively, producing multiple frames that can then be composed in an external image editing application if needed. The **Render Display** offers two additional commands allowing you to automatically save multiple Render Passes simultaneously. The **Save Passes as Images** and **Save Passes as Layered Images** commands. Both work identically to their single image counterparts, but save all passes automatically, using the naming conventions determined by the **Output Pattern** options found in the **Frame** sub-tab of the render item (for more information, see Render Item: Frame). Selecting either option opens an OS-specific dialog, allowing you to specify a location, file name and format. The output pattern naming options are automatically appended to the specified name when you click **Save**. For more information on Render Passes, see Render Passes.

Saving Animation Sequences

Saving an entire animation of still images individually is pretty tedious. Thankfully, there are several options for saving sequences automatically. You can acces these in the menu bar, under **Render** > **Render Animation**. Three options are available: **Image Sequence**, **Movie** and **Layered Images**.

Choosing **Image Sequence** is highly recommended; this allows you to render entire animations of individual frames that can be assembled in a video editing application that accepts them. It also provides an easy means of updating frames without having to re-render the entire sequence, should the need arise. Additionally, it offers the ability to recover from system crashes, allowing you to pick up where the system left off, without the fear of a corrupted video file. When you click this option, you can specify the file name, location and format.

The **Image Sequence** and **Layered Images** options offer identical file format options as their corresponding commands in the **Render Display**, listed above. The **Movie** option provides choices specific to your platform and operating system.

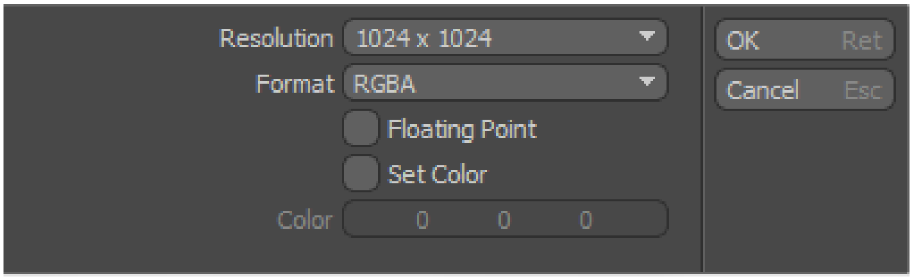
Render Outputs

Individually, file location, name and format can also be specified within the Render Outputs. This is useful, because it is saved within the file and is therefore portable and persistent across Modo sessions. Modo only utilizes the **Output Filename** option when the **Render Animation** command is invoked. Once specified, and the command initiated, a dialog asks if you wish to use the filename as specified in the render output. Select **Yes** to use it, and **No** to specify an alternate name.

Warning: Pressing **F9** or utilizing any of the other various render commands does not automatically save images as specified in a render output. Only the **Render Animation** command has this option.

Saving User-created Images

Modo offers the ability to create bitmapped images directly in Modo, eliminating the need for an external image editing application. Within nearly every image loader and image browser viewport, you have the options to create a new image or image sequence. These commands are also available in the menu bar under the **File** option. When using either command, an operating system-specific dialog prompts you to choose a filename, location and format. Once specified, another dialog opens, allowing you to choose various options related to the image.



Resolution 1024 x 1024

Format RGBA

☐ Floating Point

☐ Set Color

Color 0 0 0

Once you click **OK** or press **Enter**, the image is saved to disc.

Note: This file is not embedded into the Modo file itself, but acts like any other imported image, remaining linked to the saved external file as specified.

The newly created image is effectively a blank canvas for you to use for painting, sculpting and baking. The image appears within the **Images** viewport and is immediately available as a texture layer for placement anywhere within the Shader Tree. However, any modification made to the image, once created, are not saved unless you specifically save them. Files can be saved several ways.

- Right-click on the image in the **Image Viewer** palette, and choose either **Save** or **Save As**.
- In the menu bar, click **File** > **Save Image** or **Save Image As**.

In the case of scenes with multiple images, the highlighted or selected image in the **Image Viewer** palette is the image saved when utilizing either command. In addition, if any modifications have occurred to images present in Modo when exiting the application, you are prompted to save the changes.

Saving 3D Images

PNG Anaglyph Optimized mode

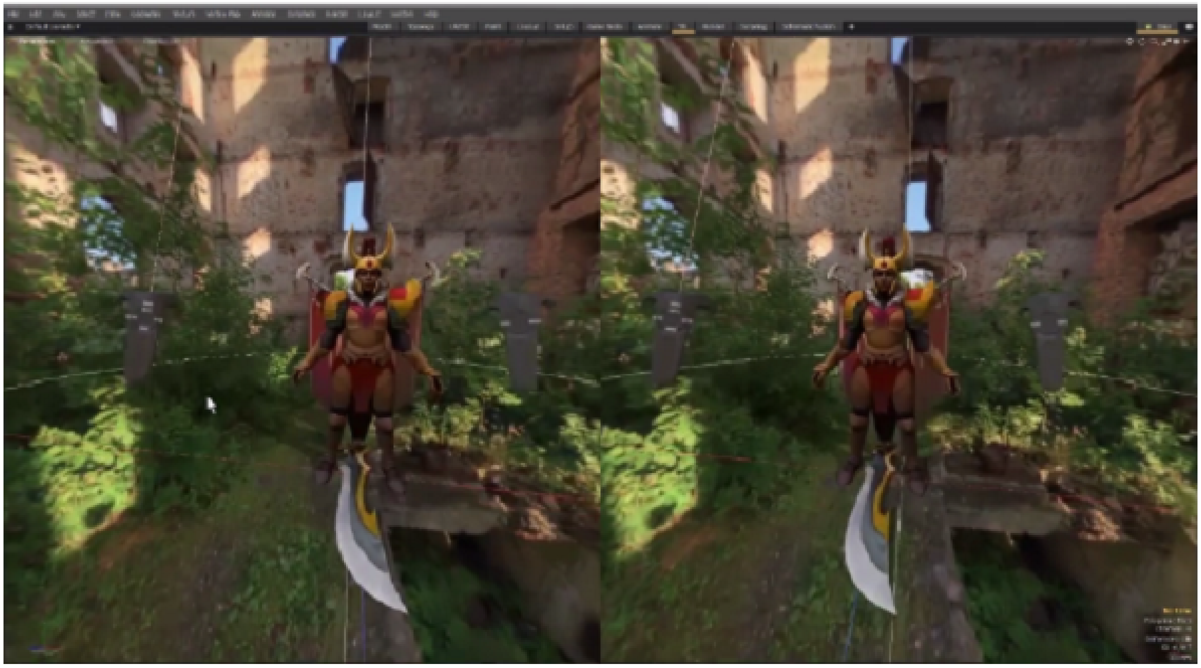
PNG Side By Side mode

Modo has the ability to save scenes rendered in stereo into any of the supported bitmap formats. To do this, you first have to enable the camera item's **Stereo** option has to be enabled in the **Properties** viewport under the **Camera Effects** sub-tab. Adjustments can also be made to the **Interoccular Distance** and the **Convergence Distance**, if necessary. The **Stereo Composite** option then determines how the two rendered images are combined to produce the final stereo image, covered fully in the Camera Items topic. Once enabled, each time a render command is invoked, the two images are rendered consecutively, representing the left and right eye views. These two images are combined as defined, and displayed as a single slot in the Render Display. Using the controls above the display, you can choose to view the **Left**, **Right** or combined **Stereo** image. As with the other options, the currently visible image is saved.

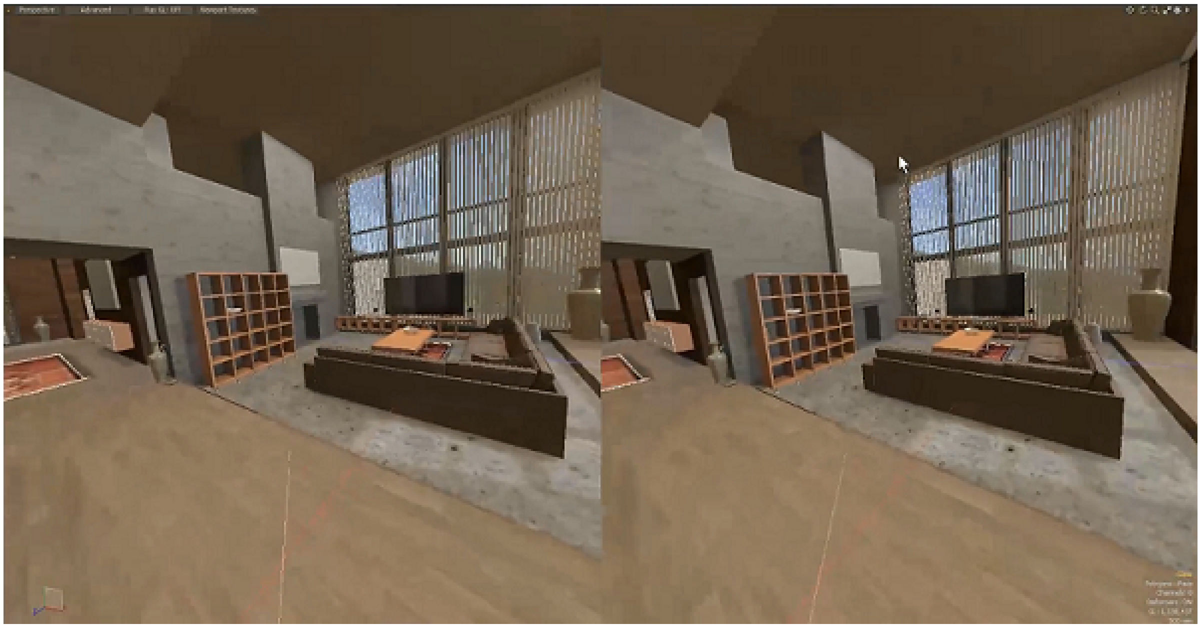
Tip: A frequently asked question is which is the best format to save in. Unfortunately, there is no one correct answer, as it is highly dependent on the intended use; particular workflows impose specific requirements. Visual effects work is enhanced by the additional information afforded by **HDRI** formats such as **.hdr** and **OpenEXR**, but these formats also require the greatest amount of storage space. Advertising and visualization are often further edited in imaging applications such as Photoshop. The layered **.psd** format can be a good choice, but the 16 bit flavors of **.tif** and **.png** can afford additional editing leeway with their extended bit depth. Any of the compressed formats helps to reduce the amount of disc space necessary, especially when storing long image sequences. However, lossy formats may also introduce compression artifacts that can reduce quality of the final deliverable. For best results, thoroughly test your chosen format to ensure you receive the intended outcome.

Modo VR

Modo VR allows first person manipulation of layout, annotations, playback, and review directly in Modo in VR. Modo VR is built on the Modo Advanced Viewport technology, using a VR headset to let you model in real life scale and perspective.



Reviewing content in VR directly via Modo greatly reduces time needed to switch applications, and greatly improves the overall immersion and connection an artist has with an object's scale and perspective.



Modo VR is ideal for 3D content creators preparing data VR and AR experiences, and also provides accurate scale and context for designers developing items for real-world construction.

Modo VR Requirements

VR is a performance demanding operation when done correctly. There is a much higher minimum machine specification required than is needed to run the default installation of Modo.

Modo VR is supported on Windows, Mac, and Linux.

Note: Oculus only runs on Windows.

The following machine specifications are recommended:

Windows

- Windows 10
- Intel i7 or Xeon, 3.5 GHz or higher CPU
- 8 GB DDR4 Memory
- NVIDIA GeForce 1080 or Quadro M6000 Graphics card
- HTC Vive or Oculus Rift HMD with Oculus Touch controllers

Mac

- macOS 10.14 (Mojave)
- iMac Pro with Vega 56 or Vega 64 GPU

Other machine specifications are possible but the above has been officially tested for the Modo VR.

Alternate hardware options have been tried and should work, including AMD graphics cards. If any issues for any machine configurations are found, please direct questions to the Foundry's Public Modo forum.

Linux

- Ubuntu 18.04
- Intel i7 or Xeon, 3.5 GHz or higher CPU
- 8 GB DDR4 Memory
- NVIDIA GeForce 1080 or Quadro M6000 Graphics card

SteamVR

Modo VR can use the SteamVR (or OpenVR) system to talk with your supported HMD. This requires Steam to be setup and installed on your system, using a Steam account. For more information, see SteamVR Support.

Oculus Home

Modo VR can use the Oculus Home system to talk with your Oculus HMD. This requires the Oculus Home system to be setup and installed on your system. For more information, see Oculus Setup. Once installed and running, click **Settings** > **General** and enable Unknown Sources.

Note: For all HMDs, please make sure the installation is working correctly in the SteamVR or Oculus Home applications before trying to use Modo VR.

Using Modo VR

To start and use Modo VR with an HMD, do the following:

1. Start the SteamVR or Oculus Home application.

Tip: Type SteamVR in the Windows Start search menu to locate the application or run it from within the Steam client.

Tip: Type Oculus in the Windows Start search menu to locate the application.

2. Click the VR layout tab in the switcher bar directly below Modo's menu bar.

   You should now see a single viewport with no side panels and a split left/right view. If you lift and move the HMD around, the view in the viewport should change.
3. Move the mouse over the 3D viewport and click in the view to ensure the application has focus.
4. Put on the HMD and activate the controllers. In the HMD, you should see an empty scene and a ground grid.

This VR layout can be switched to or from at anytime to allow you to review or edit your loaded Modo scene.

Tip: Modo VR will favor Oculus Home over SteamVR, if detected. If this is an issue, run Modo VR with -nooculus on the command line to stop Oculus Home from being used. Conversely, use -noopenvr on the command line to stop SteamVR from being used.

Options, Scenes, and Performance

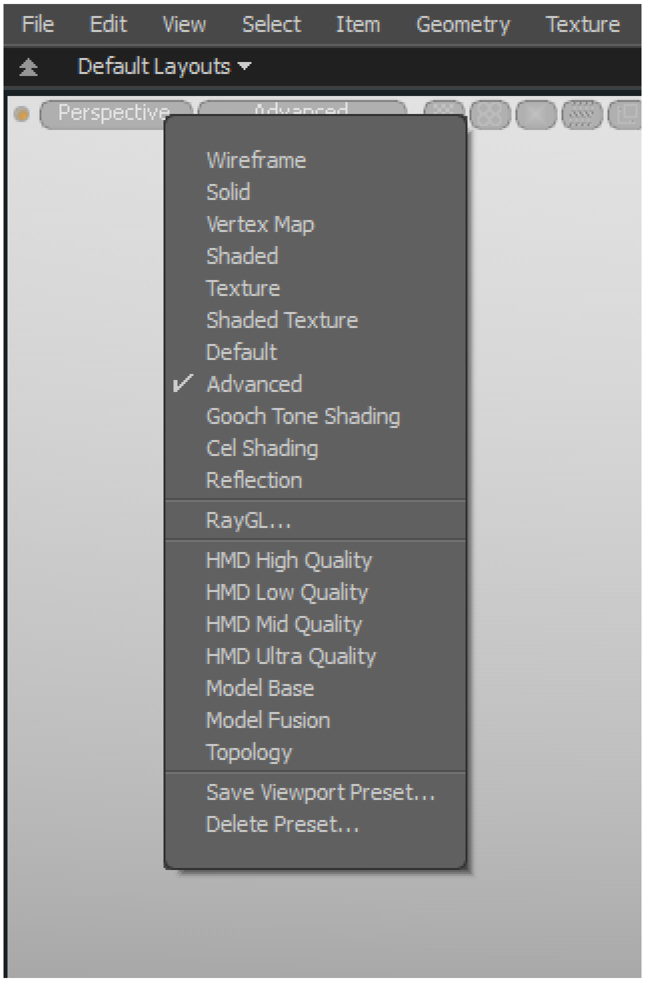
VR requires a much higher resolution and a higher frame rate for rendering than a normal Modo viewport session. This makes it easy to push Modo beyond the minimum required performance for an acceptable VR experience. To help control Modo's performance, there are a number of HMD Quality display setting presets that are predefined in Modo VR to allow you to configure your display.

Selecting a HMD Preset

Modo VR provides a number of predefined HMD Quality display settings.

To select a HMD quality preset:

1. In the top-left corner of the viewport, click the second capsule next to the Thumb to open a dropdown menu.


File
Edit
View
Select
Item
Geometry
Texture
Default Layouts
Perspective
Advanced
Wireframe
Solid
Vertex Map
Shaded
Texture
Shaded Texture
Default
✓ Advanced
Gooch Tone Shading
Cel Shading
Reflection
RayGL...
HMD High Quality
HMD Low Quality
HMD Mid Quality
HMD Ultra Quality
Model Base
Model Fusion
Topology
Save Viewport Preset...
Delete Preset...

2. Select a preset you would like to try. The following predefined HMD Quality display settings are available:
   - HMD High Quality
   - HMD Low Quality
   - HMD Mid Quality
   - HMD Ultra Quality

If you would like to adjust the display settings manually, either use the gear settings icon on the top-right corner of the viewport or press the **O** key when the cursor is over the viewport. This opens the 3D Viewport Properties dialog. Open the Advanced Options tab to customize the display of elements within a viewport. For more information, see 3D (OpenGL) Viewport.

Performance Tips for Modo VR

If using the presets or manually adjusting the display settings does not produce adequate performance, try the following:

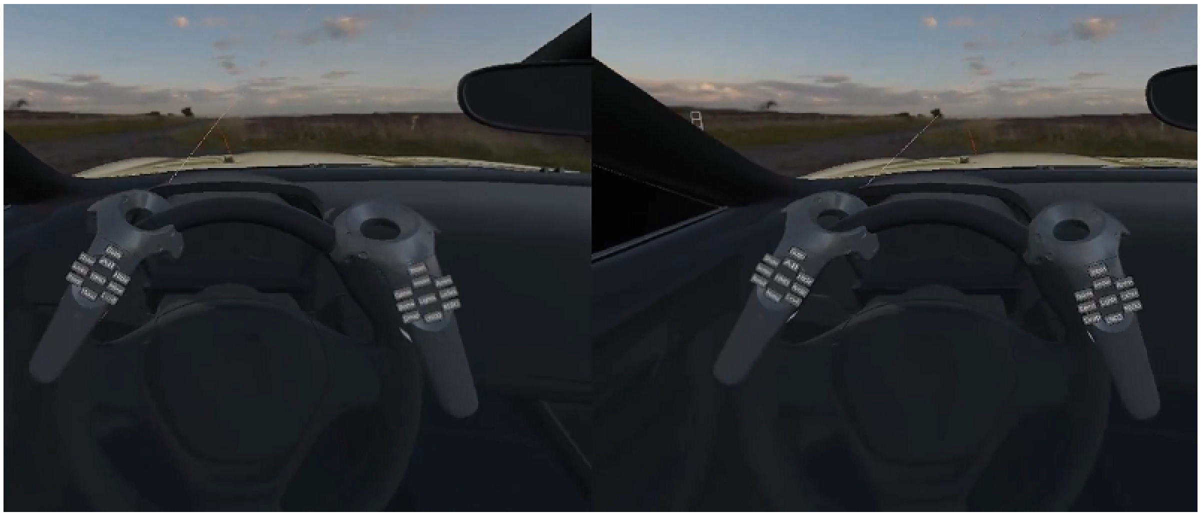
- Hide objects and lights in the scene that you are not presently interested in.
- Make sure Modo has focus while using the HMD.
- Make sure there are no other applications that are producing high GPU or CPU loads.

Using Modo VR

Controllers

The main way to interact with Modo VR is through hand controllers. When wearing the HMD you should see the two controllers.

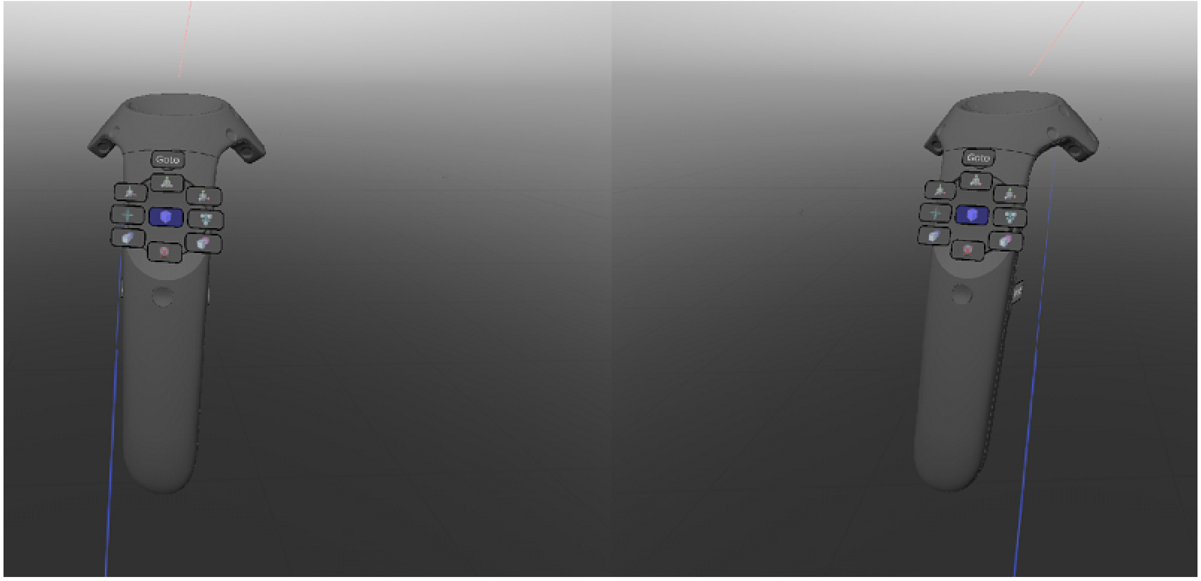
Tip: If the controllers are missing, make sure they have not powered down.



Each of them should look slightly different. One will have a laser line emitted from the front. This is the Pointer controller and is used as the master controller.

Each controller has a trigger on the underside that works like the left or right mouse button to activate tools or selection using the laser pointer. Any buttons with actions assigned will have labels on them.

If a button is touch sensitive, the label will change color when touched (blue) and again when pressed (orange).



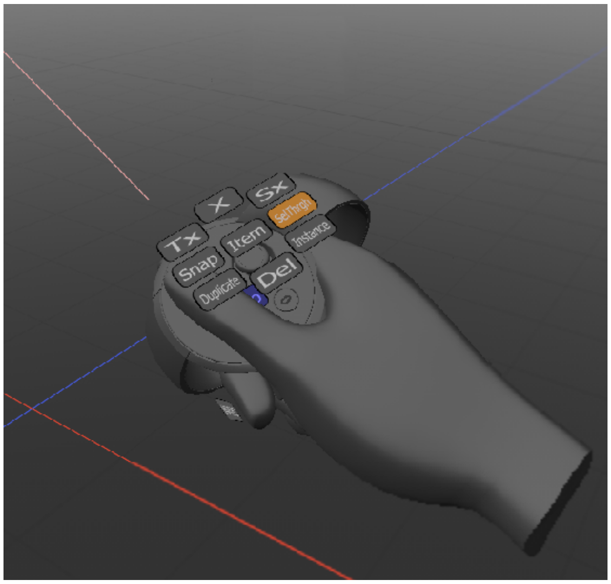
This is most noticeable when a controller has a touchpad or joystick that is segmented into button regions; the label highlights and vibrates when you move your finger or thumb into that button’s region, letting you know that clicking at that point activates that action.

Note: The touchpad or joystick on a controller can be used without having to look at it. When your finger moves over a button the controller will vibrate to let you know a new button is active and ready to be clicked.

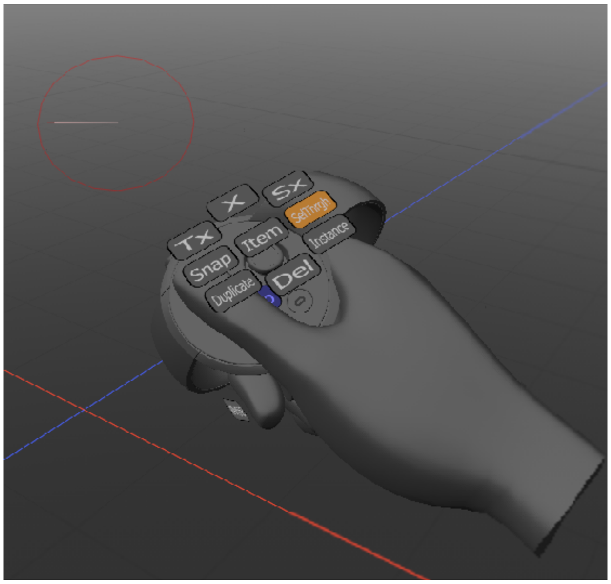
Tip: Buttons on diagonals can be easily found by first moving your finger in a vertical or horizontal motion and then sliding around the touchpad or joystick until you feel the next vibration.

Laser Pointer and Target Modes

The laser pointer has two modes of operation: Pointer mode and Target mode; which you can switch between using the Laser Mode options.



Pointer Mode



Target Mode

Pointer Mode

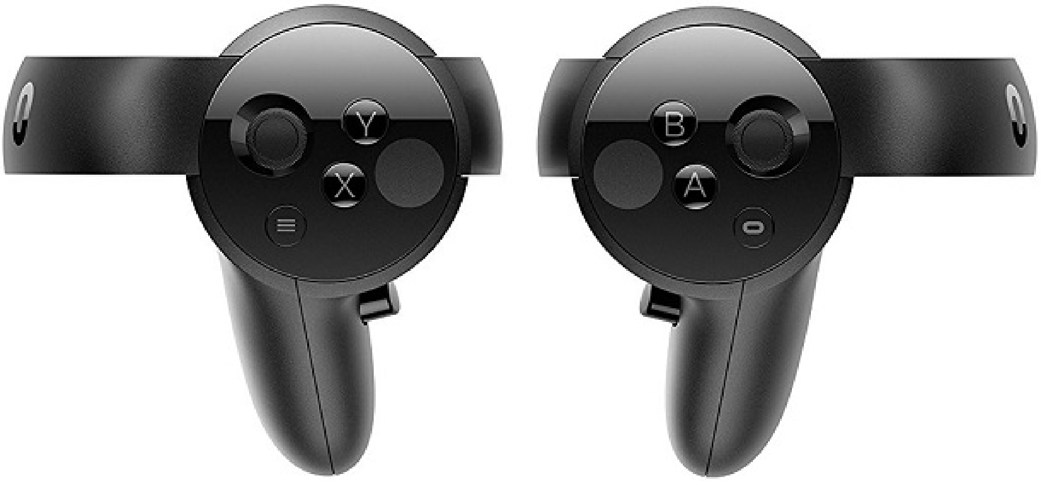
The pointer mode is the default mode of operation and can be seen as a line that emanates from the controller and points in the controller's forward direction. Point the laser line at the items you would like to interact with, as you would a laser pointer on a screen.

Target Mode

The target mode is shown as a circle with a small line within it. This works by looking through the line in the circle at the items you would like to interact with. There is still a laser line (the short line you can see inside the circle) but it always points in the direction of your head. This method is harder to learn, but improves on some of the issue that affect the Pointer mode, like hand shake.

Oculus

Oculus Touch controllers use a finger pointing mode. This is the same as target mode (above), but uses the end of your index finger as the focus point.



Note: Although this section refers to the A and B buttons, the instruction equally applies to the left-hand X and Y buttons.

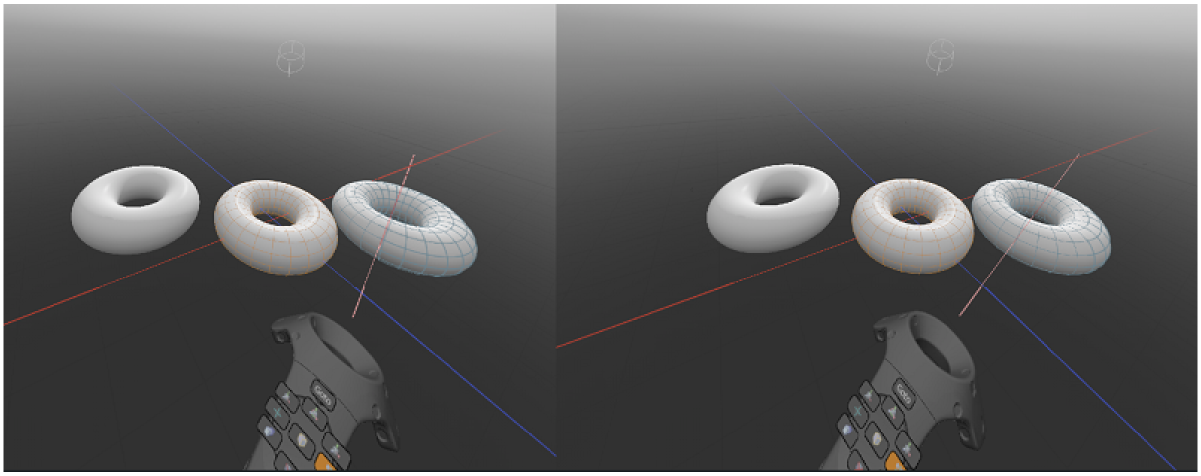
To activate this mode use the point action: touch the A+B+Grip, do not touch the Trigger. The controllers will vanish leaving just the hand in the pointing pose. You can use either hand.

To trigger an action or piece of UI press the Grip button: touch the A+B, do not touch the Trigger, press the Grip.

When using Oculus Touch controllers a number of hand actions will trigger different modes of operation, i.e. finger pointing and grab actions. To stop these from happening, reset your thumb on the Thumb Stick instead of on the controller casing or other buttons.

Selection

The default mode for the controllers is a selection mode. If you point the laser pointer at an object, its wireframe will change color (blue) to show it is highlighted. If you then use the pointer’s trigger, the object will be selected (yellow).



Tool Handles

When tool handles are active you can use the laser pointer to highlight and drag different parts to perform operations. Pointing at a part of the tool handle will highlight it, changing its color to yellow.

While highlighted, you can hold the trigger and move the controller to drag the tool handle. For example, highlighting and dragging one of the transform tool handles' arrows will move the selection on that axis.

Use the Drop button to deactivate a tool handle and go back to Selection mode.

Note: For a full description of the different tool handles, see Transform Item Tools.

Teleport and Goto

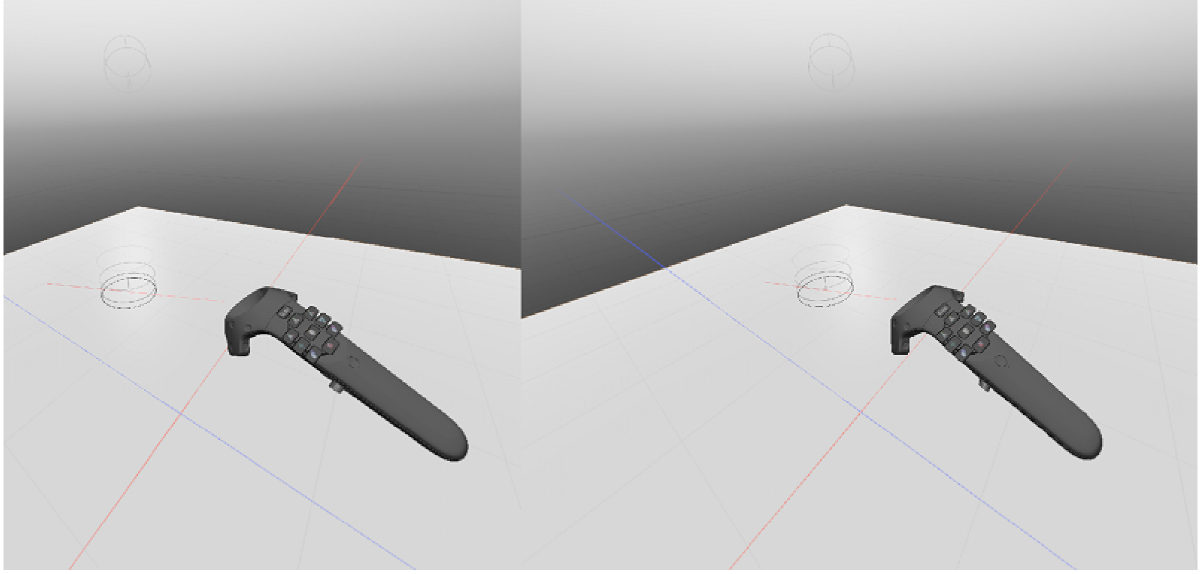
There are two main ways to teleport yourself around a large scene: Teleport tool and Goto action.

Goto Action

The Goto action works by pointing the laser pointer at a location and pressing the Goto button. This moves you to the new location. This will work while any other tool mode is active without deactivating that tool mode.

Teleport Tool

Pressing the Teleport button activates the Teleport tool. This changes the pointer from its previous mode (usually the Selection tool) to the Teleport mode. Pointing the laser line at a location now shows you an indication of where you will appear once teleportation has happened. To perform a teleport, use the trigger button.



Once you have finished with the Teleport tool, use the Drop button to go back to the default tool.

Pinch to Navigate

An alternative to the Teleport and Goto options is Pinch to Navigate (P2N). This involves holding down the navigation buttons on each controller.

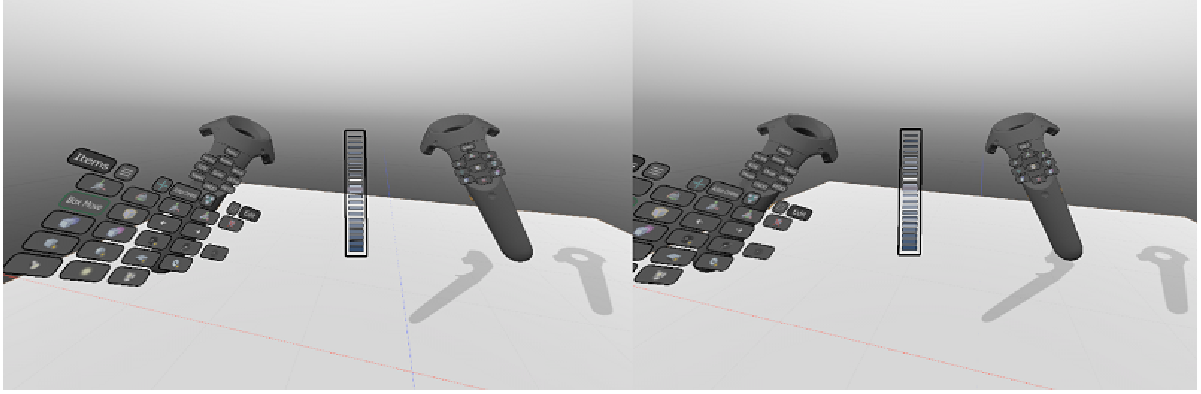
Note: For the HTC Vive controllers, these are the grip buttons on the underside of the controllers.

There is also a label on both sides of each controller. Only one button per controller has to be held down. Once held down, the scene around you can be translated, rotated, and scaled by moving both controllers as follows:

- Scale - move the controllers closer together or further apart to scale the scene.
- Rotate- move the controllers around a center point to rotate the scene around the point.
- Translate - move the controllers in the same direction to translate the scene.

Tip: P2N needs some practice to get used to as it is easy to lose track of the present scale of the scene.

To help with scale, there is a UI element that appears between the controllers when P2N is activated.



The UI element represents the present absolute scaling of the scene. By default the scaling is x1, which is shown by the single white marker on the UI. For every 4 marks, the scale goes up or down by a factor of x10. Use this UI element to check that you are only changing the scene scale when you intend to.

Note: The menu item **P2N Reset** will place you back at the origin with a normal scale.

When using Oculus Touch controllers it is possible to translate and rotate using either hand. To activate this mode use the grab action (touch the A+B+Trigger and press the Grip button) , the controllers will vanish leaving just the hands.

Button Sets

Some of the buttons on the non-pointer controller and menu items will change the actions assigned to the buttons on the pointer controller. These are called Button Sets and represent different groups of actions. Once activated you will see the labels on the pointer controller change.

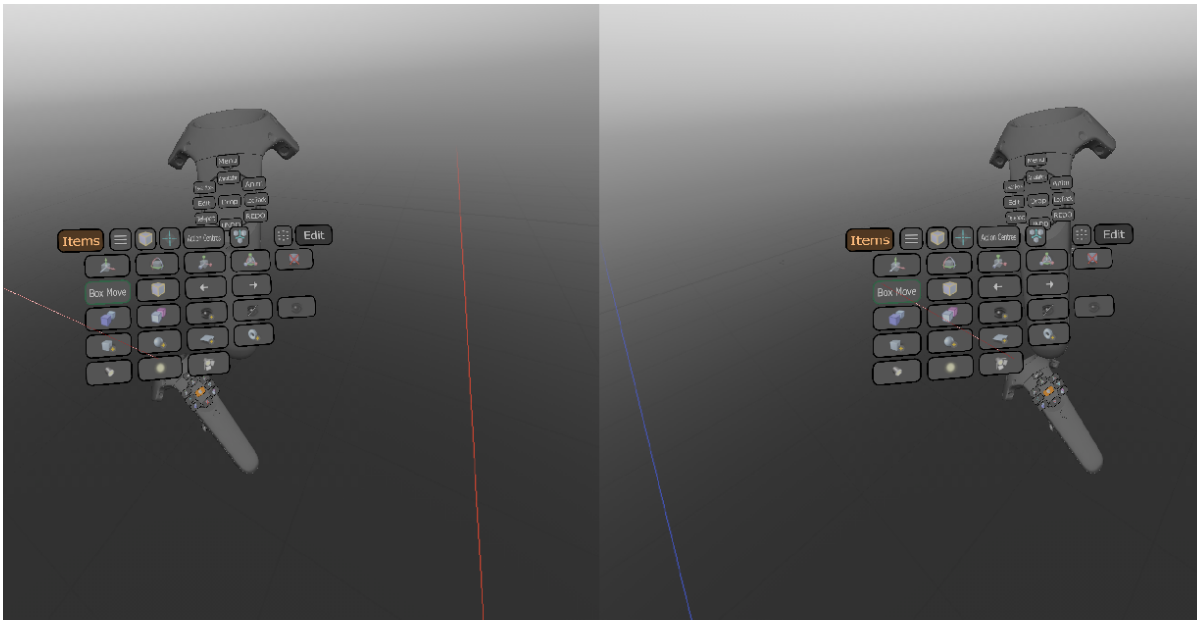
The table below lists all available button sets.

Options	Display options for tweaking performance from within the HMD.
Anim	Animation controls.

Annotations	Annotation creation tools and UI.
Items	Item move tool handles, duplication and deleting.
Lights	Light creation and move tool handles.
LocTrack	Locator Tracking controls.
Navigate	D-Pad style navigation.

Menus in Modo VR

The non-pointer controller has a number of floating labels over your wrist area. These make up the Menu system. Use the pointer controller to point at and highlight an item, and then use the trigger to select it.



Some items produce pop-up selections, hold the trigger down to show the pop-up items, highlight the desired item, and then release the trigger.

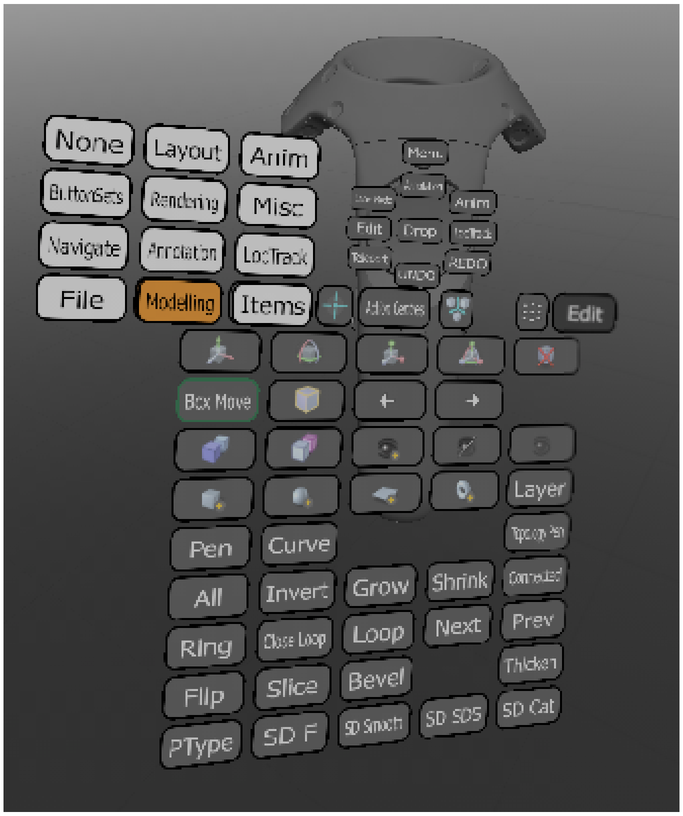
If you are not sure what a menu item is meant to do, highlight it for a couple of seconds (without triggering it) and a tooltip should appear.

The first row of items are the top menu and do not change. Clicking the leftmost one of these will change the menu items farther down to show sub-menus tailored to different workflows.

Top Menu Items

Menu	Show and changes the active sub-menu.
Selection Mode	Shows and changes the present selection mode.
Snapping	Toggles Snapping.
Action Centres	Selects the current Action Center mode.
Select Though	Toggles Select Through, when a transform tools is active.
Button Sets	Show and changes the active Buttons Set.

Menus


None
Layout
Anim
ButtonSets
Rendering
Misc
Navigate
Annotation
LocTrack
File
Modelling
Items
Menu
Anim
Edit
Drop
REDO
UNDO
Edit
Box Move
Layer
Pen
Curve
All
Invert
Grow
Shrink
Ring
Close Loop
Loop
Next
Prev
Thicken
Flip
Slice
Bevel
PType
SD F
SD Smooth
SD SDS
SD Cat

Anim	Show Animation sub-menu.
Button Sets	Show Buttons Sets sub-menu.
Rendering	Show Rendering options sub-menu.
Annotation	Show Annotations sub-menu.
LocTrack	Show Locator Tracking sub-menu.
Items	Show Item creation and manipulation sub-menu.
Layout	Show Item advanced manipulation sub-menu.
Misc	Show miscellaneous sub-menu.
Modeling	Show modeling sub-menu.

Common Menu Items

Undo	Undo operation.
Redo	Redo operation.
Drop	Drop present tool, and return to selection mode.
Item	Item selection mode.
SelThrgh	Toggle Select Through state.
Duplicate	Create duplicates of selected items.
Instance	Create instances of selected items.
Hide	Hide selected items.
Hide Invert	Invert hidden items.
UnHide All	Unhide all items.
Tx	Transform tool.
Rx	Rotation tool.

Sx	Scale tool.
X	Transform, Rotation and Scale tool.

Action Centre Modes

Automatic	Action Center mode.
Selection	Action Center mode.
Sel Border	Action Center mode.
Sel Auto	Action Center mode.
Element	Action Center mode.
Screen	Action Center mode.
Origin	Action Center mode.
Local	Action Center mode.
Pivot	Action Center mode.
Auto	Action Center element mode Auto.
Autoc	Action Center element mode Auto Center.

Layout Menu Items

Snap	Toggle snapping on/off. Start of snapping options.
Grid	Toggle grid snapping.
1 cm	Set grid snapping to 0.01 meter.
10 cm	Set grid snapping to 0.1 meter.
1 m	Set grid snapping to 1 meter.

V	Vertex snapping or action centers.
E	Edge snapping or action centers.
Ec	Edge Center snapping or action centers.
P	Polygon snapping or action centers.
Pc	Polygon Center snapping or action centers.

Rendering Menu Items

HMD Low	Use HMD Low Quality viewport preset.
HMD Mid	Use HMD Mid Quality viewport preset.
HMD High	Use HMD High Quality viewport preset.
Simple	Set material mode to Simple.
Basic	Set material mode to Basic.
Full	Set material mode to Full.
Bump	Toggle bump mapping on/off.
Shadows	Toggle shadows on/off.
Wire	Cycle through Wireframe combination: Wireframe on all meshes.Wireframe only on select meshes.No wireframes.
Mirror	Toggle mirroring the HMD view in the viewport on/off.
Res Up	Increase resolution rendered for HMD.
Res Down	Decrease resolution rendered for HMD.
MSAA Up	Increase Multisample value.
MSAA Down	Decrease Multisample value.

Misc Menu Items

Menu	Toggle the menu on/off.
UI	Toggle UI elements, like Annotations, on/off.
Move UI	Toggle which controller the UI elements appear on.
Swap Hands	Switch which hand is the pointer and non-pointer controller.
Laser	Toggle the laser pointer on/off.
LaserCone	Show the laser's collision cone.
LaserNarrow	Make the laser's collision cone angle narrower than default.
LaserMid	Make the laser's collision cone angle the default size.
LaserWide	Make the laser's collision code angle wider than default.
Laser Mode	Toggle between laser pointer and laser target modes.
Laser Tilt	Cycle through the laser pointer tilt angles.
P2N Reset	Move the user back the center of the environment with a normal scale.
Menu Gaze	Toggle between 'point controller at menu' and 'look at menu' interaction modes. The 'look at' mode requires you to look at the menu item you are interested in instead of pointing at it with your hands.

Modeling Menu Items

Layer	Add a new layer.
Pen	Activate the Pen tool.
Topology Pen	Activate the Topology Pen tool.
All	Select all.

Invert	Invert selection.
Grow	Grow selection.
Shrink	Shrink selection.
Connected	Select connected components.
Ring	Select ring.
Close Loop	Close loop selection.
Loop	Select loop.
Next	Move selection to next loop.
Prev	Move selection to previous loop.
Flip	Flip polygon.
Slice	Activate the Slice tool.
Bevel	Activate the Bevel tool.
Thicken	Activate the Thicken tool.
PType	Toggle polygon types.
SD F	Subdivide Flat.
SD Smooth	Subdivide Smooth.
SD SDS	Subdivide Subdiv surface.
SD Cat	Subdivide Catmull Clark surface.

Indicators in Modo VR

The pointer controller can have a number of labels above the buttons. These are temporary indicators for user information, akin to status bar icons.

The appearance of an indicator will vibrate the controller to draw your attention. Any simple modal dialogs that Modo produce will show up here for a limited time, i.e. “Out of Undo - Ok” or “Nothing to Save - Ok” dialogs.

Note: At present, eye tracking is not supported, so looking at the menu items requires you to move your head until the menu item is in the center of your view.

The table below lists available indicators.

Lost Focus	The Modo application has lost focus to another application. This can happen due to notifications, new applications starting, or any number of other triggers. When this happens Modo VR will not respond until it has been made the active application again.
Threading Disabled	Modo VR ideally uses a background thread to render the HMD view. This can be disabled by the user. The performance and interaction penalty for doing this is severe, this indicator is to remind you it is disabled and to re-enable it. **Warning:** There is also a ‘Threading Not Supported’ indicator that can be seen in some situations. Some graphics card drivers have issues when running Modo VR in threaded mode and have to have the feature disabled.
GPU Slow	The rendering of the HMD view is slower than is needed for a comfortable VR experience. There are many reason why this will happen. Some reasons are external like another application in your system is slowing your computer down, but the main course is your GPU is not fast enough to render the present scene or point of view, in this case try turning down the rendering options or hiding parts of the scene you are not concerned with at present. Some of the other indicator are hints as to why you are seeing bad performance and will alway be in combination with this indicator. **Note:** The GPU Slow hints are only guesses at why your rendering is slower than normal. There are many issues that can contribute to slow rendering, and a hint might not be the best reason, but a good place to start looking for a solution.
Reduce Shadows	GPU Slow Hint: Calculating the shadows is taking too long. Try disabling the shadows.
Reduce Geometry	GPU Slow Hint: Rendering the geometry is taking too long. Try turning down the Multisampling, drawing less objects or drawing less vertices.

Simplify Pixels	GPU Slow Hint: Rendering the materials is taking too long. Try turning down the Multisampling, reducing the resolution, disabling bump-mapping or disabling shadows.

Modo VR Additional Features

Annotations

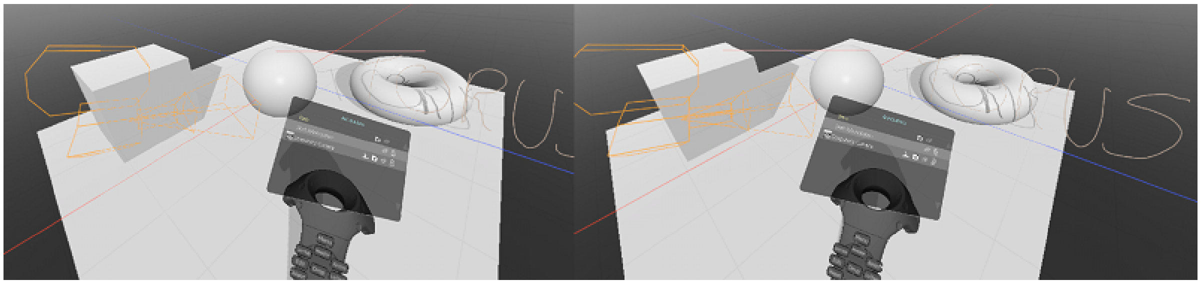
Annotations are a way of leaving temporary notes in a Modo scene for later action.

The items used for annotations are:

- Sketched meshes.
- Snapshot cameras. These can contain a sub-mesh to hold sketched meshes.

Sketch Annotations

When in HMD mode it is not easy to use the keyboard. So to allow notes to be taken the sketch tool has been used to allow sub-meshes to be added for leaving notes. The sketch tool will leave a trail behind when the trigger is held down and the controller moved around. This can be used to place arrows, highlights, or roughly drafted text in the 3D scene.

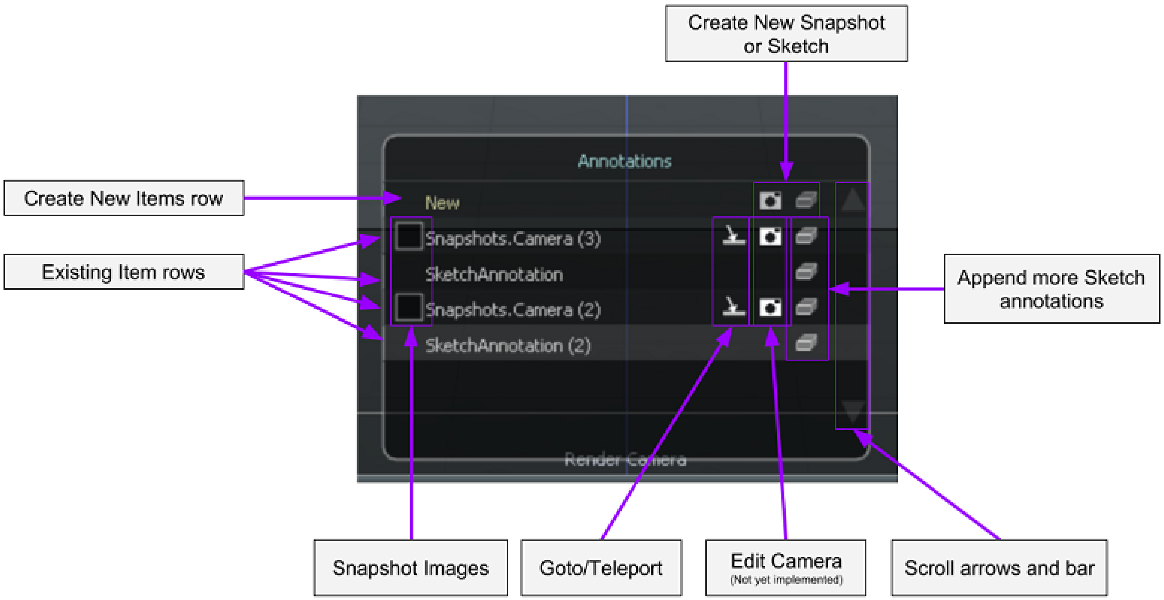


Snapshot Annotations

Snapshots are cameras that can be placed down in the scene to allow for notes to be made for different viewpoints.

Annotations UI

The Annotations UI allows you to create, find, navigate, and amend annotations in the scene.


Create New Snapshot or Sketch
Annotations
Create New Items row
New
Existing Item rows
Snapshots.Camera (3)
SketchAnnotation
Snapshots.Camera (2)
SketchAnnotation (2)
Append more Sketch annotations
Render Camera
Snapshot Images
Goto/Teleport
Edit Camera
(Not yet implemented)
Scroll arrows and bar

There are four main parts of the interface:

Scrollbar	Scrolls through the list of items when the list is larger than the panel space.
New Row Icons	Clicking these creates one of the following: • New camera Snapshot annotation - Creates a camera at the present location. • New Sketch annotation - Immediately goes into Sketch mode, holding the trigger to leave the sketch trail.
Teleport Icon	Teleports you to the location of the item/camera.
Mesh Icon	For the item on that row, adds/appends more sketched annotation. Immediately goes into sketch mode.

Select these by pointing the other controller at the UI and using its trigger to activate.

Locator Tracking

Locator Tracking allows your Modo scene to be affected by your current hand and head positions. This is done by placing named items into your scene and then rigging them to other parts of the scene.

To place and rig a named item:

1. Create an item in the root of the Item Tree, preferably a Locator.
2. Rename the item to one of the designated names. See the table below.
3. Go to the LocTrack button set on the hand controllers.
4. Use the corresponding button to toggle the desired item to actively track.

   See the table below for items and their corresponding buttons.

Item Name	**Button Name**	**Description**
LocatorHMDControllerLeft	Secondary	Position of the Secondary controller
LocatorHMDHead	HMD	Position of your Head
LocatorHMDControllerRight	Pointer	Position of the Pointer controller

Multi-GPU Support

Machines capable of NVIDIA SLI or AMD Crossfire are supported. This allows multiple GPUs to be used together for faster VR rendering.

To use this feature:

1. Install multiple SLI or CrossFire compatible graphic cards into a compatible system.
2. Install a graphics card driver that supports SLI or CrossFire in OpenGL.

> **Note:** The OpenGL driver must expose either GL_NV_gpu_multicast or GL_NVX_linked_gpu_multicast extensions. Some drivers only support SLI or CrossFire in DirectX and not OpenGL.

3. Enable SLI or CrossFire in your graphics cards settings.

Note: Instruction for enabling SLI or CrossFire can be obtained from your graphics driver supplier. This is different for each vendor and can change between driver versions.

4. Run Modo using -gpumulticast on the command line.

Quick Start: Coffee Mug

Follow these Modo written tutorials to get a good understanding of how Modo works. As you become familiar with Modo, you'll find there are different ways of achieving the same results.

These tutorials demonstrate the key features you'll reuse in other projects.

- Model the Mug - Learn how to model a Modo coffee mug.
- Render the Mug - Learn how to render the Modo coffee mug.

Model the Mug

To get started, let's model a simple object: a Modo coffee mug. Don't worry if you haven't used Modo before, this is a good starting point.

Video: This video shows you how to model the mug. It covers the creation of a basic cylinder through to advanced polygon manipulation. The written tutorial includes background information for those new to 3D modeling, although you can choose to follow the video instead.

Here's a picture of the mug that we'll model.

This part of the tutorial shows you how to model the shape of the mug, including the handle. If you've already done this, you can skip to the Render the Mug to create the surface detail and produce the finished render.

As you become familiar with Modo you'll find there are different ways of achieving the same result. However, the methods described here demonstrate key features that you'll reuse in other projects.

If you're brand new to Modo, it may help to take a look at Exploring the Layout to discover how the interface is organized. To find out how to interact with Modo, you can read User Interface Conventions. You can also find a quick introduction to modeling concepts in Modeling Overview and Techniques. Reading these aren't compulsory, but recommended if you're new to 3D modeling.

Setting Workspace, Layout, and Viewport

Reset the Workspace

If you've just opened Modo, the you can start the tutorial with the default configuration. If you've been trying out the interface, it's a good idea to reset the workspace to the default (base) configuration. This guarantees that we're starting from the same place.

You can also reset the workspace if you run into trouble during the tutorial and want to start afresh.

To Reset the Workspace:

1. Select **File** > **Reset**.
   You'll see a prompt asking if you want to restore the layout to its base version.
2. Click **OK**.

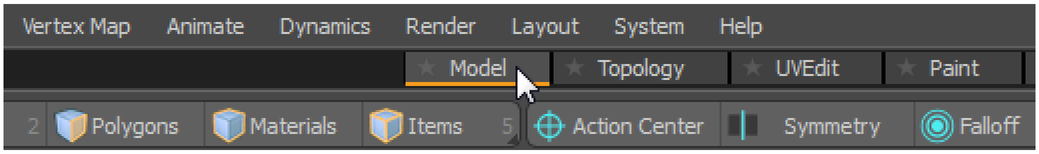
Choosing the Model Layout

Modo offers multiple interface layouts, each with a toolset targeted at a particular task. Layouts are selected using the tabs directly under the menu bar. For this tutorial, we'll use the **Model** layout. This layout is designed for creating 3D objects using object primitives (simple shapes), and it provides a large perspective view of the scene.

Note: In the **Perspective** view, you see objects in three dimensions as you would in the real world. It's just like viewing the world through a camera viewfinder; you see the world in a rectangular window (viewport) and you can move the camera in space to see objects from a different position.

To Choose the Layout:

1. Click the **Model** tab on the switcher bar. The switcher bar is below the menu bar at the top of the application window.


Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Model
Topology
UVEdit
Paint
2
Polygons
Materials
Items
5
Action Center
Symmetry
Falloff

2. If you can't see the switcher bar, it may be hidden. Click the dark line below the menu to reveal the switcher bar.

OR

1. Select **Layout** > **Layouts** > **Model**.

Tip: You can also select a layout using Modo's layout switcher. Press and hold **Shift** + **Ctrl/Cmd**, then press **Tab** to open the switcher. Keep pressing **Tab** until you highlight the chosen layout, and release both keys to use the layout.

Introducing the 3D viewport

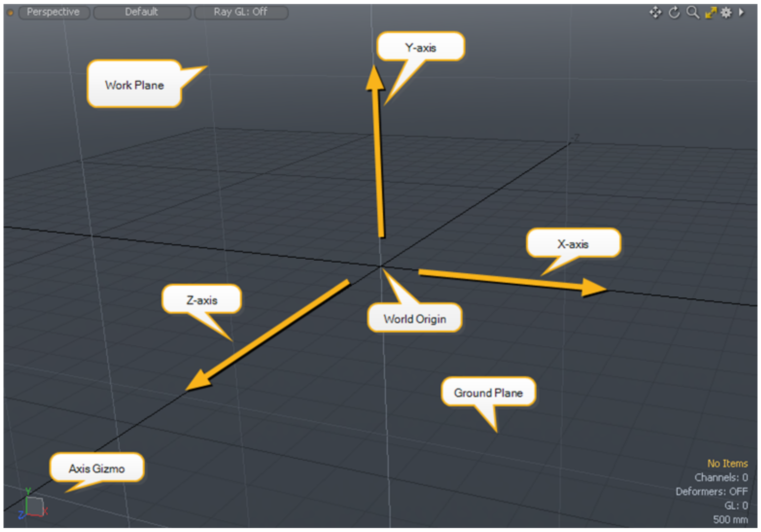
Most of your work in Modo is done in the 3D viewport. This is a window onto 3D space that you can explore by rotating, panning and zooming. The viewport has several visual cues to help you build objects and find your way around.

Any position in the 3D world is defined using values along three orthogonal axes: X, Y and Z. The X-axis runs horizontal from left to right, and the Y-axis runs up and down. These axes cross at the World Origin, where X = 0 and Y = 0. The addition of a third dimension introduces the Z-axis, which runs forward and backward through the World Origin.

An object's position in 3D space is identified by a given distance along each of the axes from the world center (or World Origin). For example, a sphere may be placed at X = 2 m (2 meters over to the right), Y = 1 m (1 meter off the ground) and Z = -3 m (3 meters behind the World Origin).

The diagram below shows the location of the axes and the World Origin as they appear in Modo.

Note: The yellow arrows are for illustration purposes only.



The image above shows the following elements:

- X, Y and Z axes: Modo draws the X and Z axes using thick black lines. The Y-axis rises perpendicular (at 90 degrees) to the point where the X and Z axes intersect. The X-axis values increase to the right, the Y-axis increases upward, and the Z-axis increases forwards. (The increase in values is shown by the direction of the yellow arrows.)
- World Origin: The point at which the axes intersect is known as the World Origin. This is the point where X = 0, Y = 0 and Z = 0.

- Ground Plane: Modo draws a dark-gray grid on the ground to help you orient the 3D viewport. This surface is the ground plane, or the XZ plane. (A plane is a flat surface in the 3D space.) The ground plane goes through all points in X and Z, where Y = 0.

  You can think of a plane as a slicing through the 3D space. The ground plane slices through the space along all points where Y=0.
- The Work Plane: This is a light-gray grid that shows the plane on which new objects are drawn (see below). The Work Plane snaps to the facing plane as you rotate the view.
- Axis Gizmo: This graphic shows the current orientation of the axes in 3D space. The Gizmo rotates as you rotate the view. By checking the Gizmo, you can see at a glance how the axes are oriented and in which direction you are facing.

  The Gizmo also shows the position of the Work Plane using a small square that is aligned to the appropriate plane.

Tip: To rotate the viewport, hold **Alt**, click and drag in the viewport.

To pan the viewport, hold **Alt**+**Shift**, click and drag in the viewport.

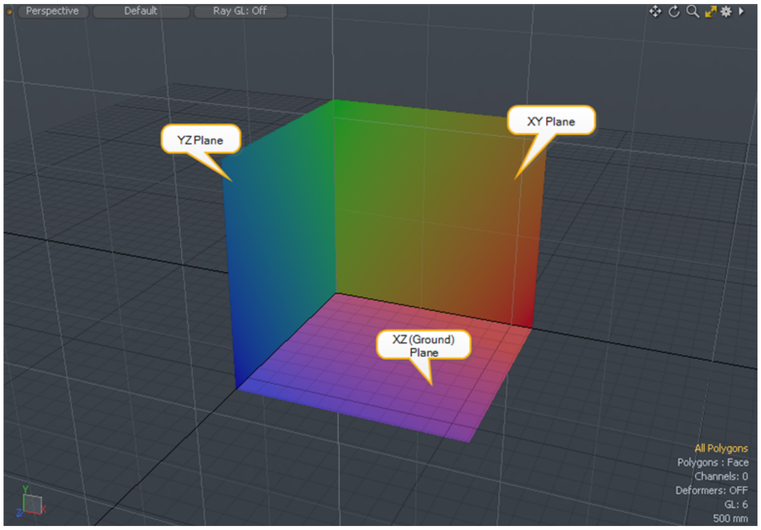
Try rotating and panning the viewport to become familiar with navigating the 3D space.

Introducing the Work Plane

The work plane is the surface on which you draw objects. This is an important tool in Modo as it helps you place objects in the world as you work. As you add objects to 3D space, it's useful to have control over how they are first placed, and how they are translated. For example, if you're modeling the inside of a room and want to add a poster, then it would make sense to first add the poster to the wall, and move it across the wall into the right position. In this case, we'd align the work plane to the wall. If you're adding a rug to the room, then it would be better to drop the rug onto the floor and move it along the floor plane, so the work plane should be aligned to the floor.

In Modo, the Work Plane automatically snaps to become parallel with one of the three dominant planes. A dominant plane is the flat surface that aligns with two of the three axes. For example, the XY plane is the flat surface that covers all points that extend left/right along the X-axis, and up/down along the Y-axis. The image below shows three squares, each aligned to a dominant plane.

Note: By default, the Work Plane snaps parallel to a dominant plane but it does not mean that it is always exactly in-line with each axis. For example, the Work Plane may snap parallel to the ground plane, but its height above/below the ground is adjusted to suit your viewpoint.



In Modo the work plane is shown as a light-gray grid. In the image above, the Work Plane is aligned to the XY plane. Try rotating the viewport and see how the work plane changes.

Drawing a Cylinder

Often in 3D modeling, we start with a simple shape and adjust it to resemble the main body of the object that is being modeled. Different parts of the object are then modeled separately to achieve the overall look.

If we look at the photo of the coffee mug, we can see the mug has two main parts, the body of the mug and the handle. We'll start by creating the mug body. The body itself resembles a tapered, hollow cylinder. So let's start by creating a cylinder that's standing on the ground plane. To do this, we'll first need to position the work plane.

Orientating the Work Plane and Drawing a Cylinder

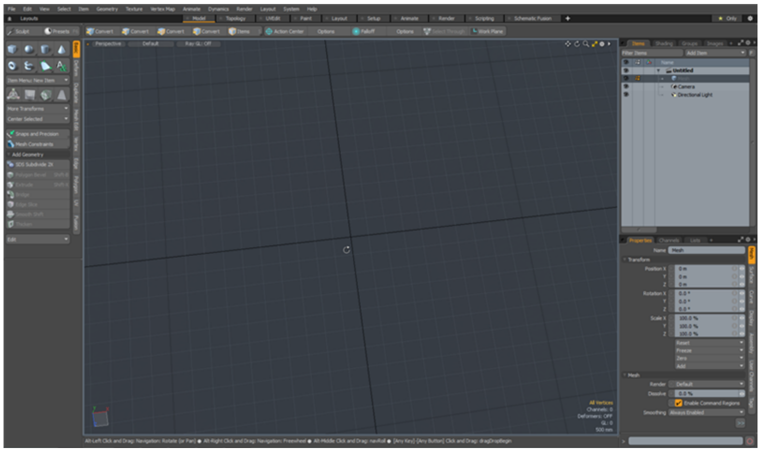
To Orient the Work Plane and Draw a Cylinder:

1. First we need to make sure that we draw the cylinder base parallel to the ground plane. With the mouse in the 3D viewport, hold **Alt** and click and drag to tumble the view so that you are facing down toward the ground plane, with the origin close to the center of the viewport (as shown in the image below).

   The work plane snaps to the ground plane.

Note: The origin is at the intersection of the X and Z black axis lines.

Tip: The Modo window may lose focus if you interact with another window (such as this help). If so, you may find Modo doesn't respond to key presses. Click on a blank part of the Modo interface to regain focus.



Note: The Work Plane is used when you add or transform a primitive. It has no effect in interactive mode when you adjust the handles to change the shape or dimensions. You can also freely tumble and zoom viewports, and the movement of the Work Plane does not alter existing geometry.

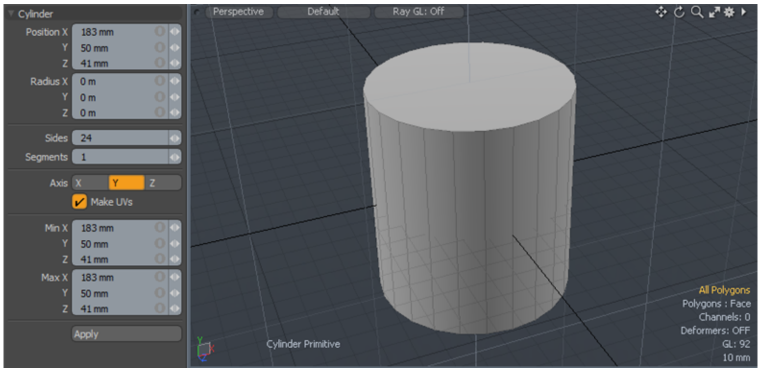
2. Click on the **Cylinder** tool in the **Basic** sub-tab of the toolbox on the left side of the screen.

   When selected, the cylinder icon is highlighted in orange and its attributes appear in the tool **Properties** panel in the lower-left of the Modo window.

Note: If you see properties in the panel in the lower-left of the Modo interface (the **Properties** panel), it means that a tool is active. When active, you can edit the property values or use the mouse to manipulate the tool.

3. Hold **Ctrl/Cmd**, click at the origin (in the viewport) and drag outward.

   Holding **Ctrl/Cmd** constrains the cylinder to be circular.



Cylinder

Position X 183 mm
Y 50 mm
Z 41 mm
Radius X 0 m
Y 0 m
Z 0 m
Sides 24
Segments 1
Axis X [✓ Y] Z
☑ Make UVs
Min X 183 mm
Y 50 mm
Z 41 mm
Max X 183 mm
Y 50 mm
Z 41 mm

4. In the tool **Properties** panel, set **Sides** to 24 and **Segments** to 1 (as shown above).

Tip: To change a property value, edit the value and press **Tab** or **Enter**.

This creates a cylinder made up of 24 polygonal strips around the circumference, where each strip is a single polygon.

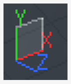
Tip: Try altering the value of **Segments** to see how the cylinder changes. As you increase the value, you'll see how more polygons are used to build the sides.

Sizing and Placing the Cylinder

Once placed, a primitive enters interactive mode and you can see the red, green and blue editing handles on the object. You can use the handles or edit property values to modify the primitive, and it updates immediately.

At the center of the cylinder there are three arrows for moving the cylinder, one for each axis. The red, green and blue arrows move along X, Y and Z respectively. The small, colored boxes on the surface of the cylinder are for adjusting the size along each axis.

Tip: You can see the axis colors and the current orientation in the bottom-left of the viewport:



We'll adjust the size of the cylinder to roughly match that of the mug, then we'll sit the base on the ground (XZ) plane. The mug is approximately 90 mm wide and 100 mm tall. For sizing the cylinder, Modo uses the **Radius** property, which is half of the full length.

Tip: Use **Ctrl/Cmd**+**Z** to undo. If you want to redo an operation, press **Ctrl/Cmd**+**Shift**+**Z**.

To Change Size and Position:

1. Set the **Radius X** and **Z** to 45 mm and the **Radius Y** to 50 mm. Enter the values in the **Properties** panel or use the handles to resize the cylinder.

Tip: If the cylinder appears small, use the mouse wheel to zoom. Use **Alt**+drag to rotate and **Alt**+**Shift**+drag to slide the viewport.

2. Click the green arrow handle and move the cylinder upward onto the ground plane. The **Min Y** setting is the height of the lowest edge of the cylinder; drag the handle so that **Min Y** equals zero. Alternatively, you can enter 0 in the **Min Y** property.
3. To make sure the cylinder is at the world origin, set **Position X** and **Position Z** to 0.
4. Once you have set the values, press **Q** to drop the tool.

   The properties and the editing handles disappear.

Tip: If a key press doesn't work when a tool is active, it may be that the tool properties intercept the key press and you edit a property unintentionally. If this happens, click anywhere on a blank area of the interface (not in the viewport) to shift focus away from the **Properties** panel, then retry the key press.

Drawing, Sizing and Placing the Cylinder - Quick Method

The instructions above are good practice for placing and sizing geometry. However, there is an alternative method which cuts out a few steps and is ideal if you want to place a primitive quickly. To do this, you first define the properties of the cylinder, including size and location, and draw the cylinder using the **Apply** button.

To Predefine and Draw the Cylinder:

1. If you've been using Modo already, close the existing scene using **File** > **Close Scene** (or **Ctrl/Cmd**+**W**).
2. Click on the **Cylinder** tool icon in the **Basic** sub-tab of the toolbox on the left side of the screen .
3. In the cylinder's **Properties** panel in the lower-left of the Modo interface, edit the properties as follows:

Position:

- **Position X** : 0 mm
- **Position Y** : 50 mm
- **Position Z** : 0 mm

Tip: The **Position** parameters refer to the center of the cylinder. To put the cylinder on the ground plane, we need to set the Y-position to half the height, in this case 50 mm.

Size:

- **Radius X** : 45 mm
- **Radius Y** : 50 mm
- **Radius Z** : 45 mm

Polygons:

- **Sides**: 24
- **Segments**: 1

Orientation:

- **Axis** : Y

4. Click the **Apply** button at the bottom of the **Properties** panel.

   If you can't see the **Apply** button, it may be that part of the **Properties** panel has been hidden due to lack of screen space. If so, you'll see a small double arrow >> button . Click this to reveal the rest of the **Properties** panel.

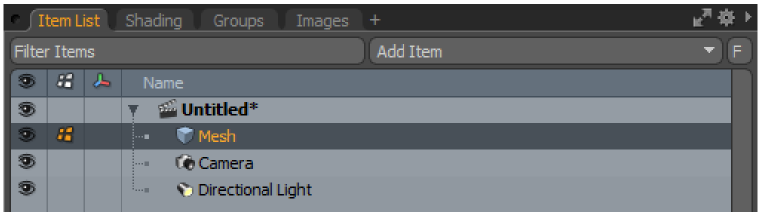
Tip: If your cylinder is too far away or you are too close, press **A** to fit your object to the viewport. If **A** doesn't work, click on a blank area of the interface to shift focus away from the primitive properties and try again.

Working with Meshes, Layers, and Scenes

Introducing Meshes, Layers and Scenes

In the field of 3D modeling, you'll often hear the word 'mesh' used to describe 3D objects. A **Mesh** is the term given to the collection of vertices, edges and polygons that fit together to make up a 3D model.

Whilst working in Modo, your 3D objects are stored in a mesh layer, which is shown in the **Item List** to the right of the workspace. A **Layer** acts as a container for each type of item (such as a mesh), and allows you to focus your work on the contents of the layer. Layers may contain items such as lights, or cameras, and if you want to edit these items, you can switch to the appropriate layer.



The collection of all the layers in a file is called a **Scene**. When you render your work, Modo draws the entire scene using the items in the scene's layers.

Tapering the Cylinder

To shape the body of the mug, we need to taper the cylinder so that the base is smaller than the top. The simplest way to do this is to shrink the polygon at the base of the cylinder.

So far we've been working in **Items** selection mode, which applies operations to all of the geometry in the current mesh layer.

To manipulate individual polygons, we need to go into **Polygons** selection mode. You can then select a polygon by clicking on it.

Note: In any selection mode, unless you select something, Modo uses the implied selection, which is everything in the layer. For example, if you are in **Polygons** selection mode and nothing is selected, any transform you apply affects all polygons.

Note: Selections are highlighted yellow. You can deselect everything by clicking on empty space in the 3D viewport (when no tool is active) or by pressing **Esc**.

To select multiple elements, click and drag over the geometry. This is known as paint based selection.

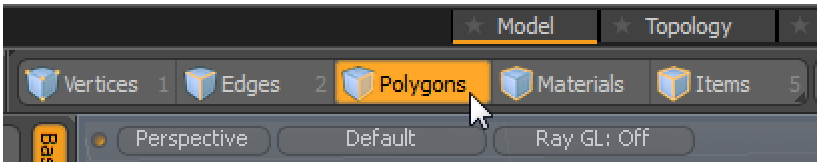
To add to the current selection, hold **Shift** and click to select the element.

To remove something from the current selection, hold **Ctrl/Cmd** and click on the selected element.

Transforming Polygons

To Select and Transform a Polygon:

1. Choose polygon selection mode by clicking **Polygons** on the modes toolbar. Alternatively, press **3** on the keyboard.


Model
Topology
Vertices 1
Edges 2
Polygons
Materials
Items 5
Perspective
Default
Ray GL: Off

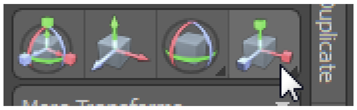
2. Tumble the viewport so that you can see the bottom of the cylinder.

> **Tip:** Before the next step, make sure there is no active tool. If you see parameters in the **Properties** panel (lower-left), press **Q** to drop the tool.

3. Click on the bottom of the cylinder to select the polygon.

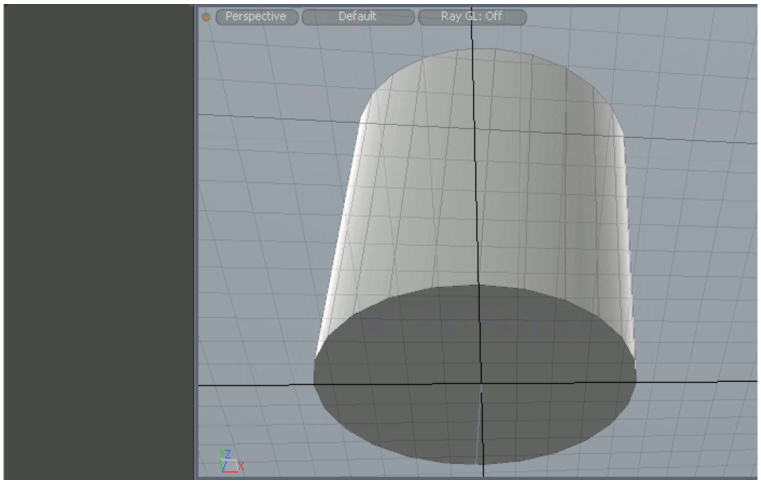
   The polygon turns yellow.

4. Click on the Scale tool or press **R**.



   The Scale tool handles appear in the center of the selected polygon.

5. Click and drag the small green circle to scale both the **Scale X** and **Z** values simultaneously. Scale them to 70%.


Perspective
Default
Ray GL: Off

Tip: You can apply a transform equally in all dimensions by holding **Ctrl** and dragging one of the transform axes.

The small circles that appear in the scale tool also constrain the transform to be equal, but along a single plane.

6. Press **Q** to drop the Scale tool and return to selection mode.
7. Deselect the base by clicking away from the geometry in the 3D viewport.

Beveling and Hollowing-Out the Cylinder

To create the interior of the mug, we need to add more polygons to the top of the cylinder so that we can drag down to create a hole. If we were to drag the top polygon as it stands, the entire mug would be flattened. To remedy this, we can use the Bevel tool to add an inner ring of polygons to the top.

We'll need to bevel twice. Once to create the rim of the mug, and again to create the inside base of the mug.

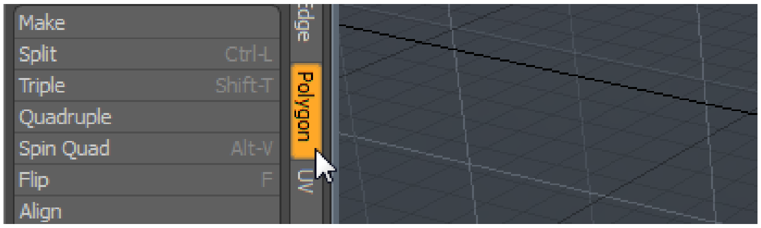
Note: You may have noticed that when the mouse hovers over a particular element, its color changes. This is called pre-highlighting, and it indicates that the element is selectable.

Tip: To apply a second bevel, we can drop the Bevel tool and add it again. However, Modo provides a shortcut. With the first bevel active, simply **Shift** click in the viewport to reset the tool and apply a new bevel.

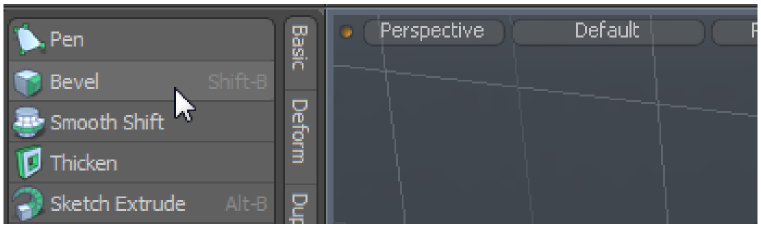
Beveling and Adding an Interior

To Bevel and Add an Interior:

1. In **Polygons** mode, tumble the viewport and select the top surface of the mug.
2. Select the **Polygon** sub-tab of the toolbox.


Make
Split Ctrl-L
Triple Shift-T
Quadruple
Spin Quad Alt-V
Flip F
Align
Edge
Polygon
UV

3. Click on the Bevel tool or press **Shift**+**B**.

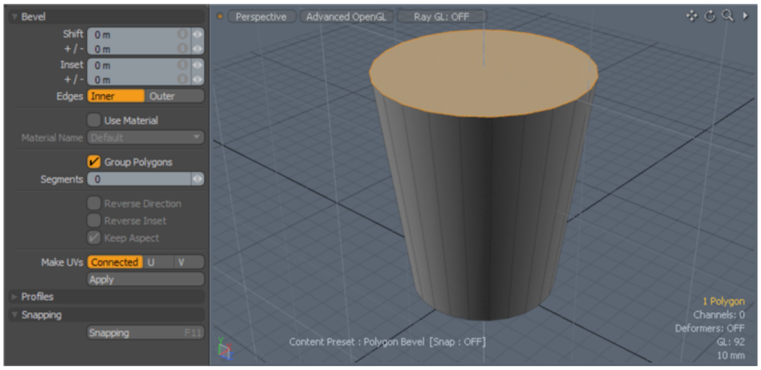

Pen
Bevel Shift-B
Smooth Shift
Thicken
Sketch Extrude Alt-B
Basic
Deform
Dup
Perspective
Default

4. With the tool active, click in the 3D viewport to enter interactive mode for the tool.

   The bevel tool has two handles, a blue one to offset the new polygon from the surface, and a red one to change its size.

5. Click and drag over the red handle to scale the inner polygon until the first **Inset** value reads 5 mm.

   This creates the rim of the mug.



Bevel
Shift 0 m
+/- 0 m
Inset 0 m
+/- 0 m
Edges Inner Outer
☐ Use Material
Material Name Default
☑ Group Polygons
Segments 0
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
Make UVs Connected U V
Apply
Profiles
Snapping
Snapping F11

Content Preset : Polygon Bevel [Snap : OFF]
1 Polygon
Channels: 0
Deformers: OFF
GL: 92
10 mm

Perspective
Advanced OpenGL
Ray GL: OFF

6. While the tool is still active, hold **Shift** and click anywhere in the 3D viewport (away from any tool handles), to reset the Bevel tool.

   This adds another polygon within the rim.

7. Click and drag the blue handle down to a **Shift** value of -95 mm.
8. Press **Q** to drop the tool.
9. Activate the Scale tool by pressing **R** and use the green circle to scale the selection to 70%.
10. Press **Q** to drop the tool.

Note: You may have noticed that we could use **Bevel Inset** to scale the base polygon instead of the **Scale** tool. Since the bevel specifies the inset as a measurement, it's easier to match the inside base of the mug to the outside base by giving it the same scale value.

Beveling and Rounding Corners

The mug in the photo has nice rounded edges, unlike our model, so we need to round them using the **Edge Bevel** tool.

Since we only want to affect the edges at the corners of the mug, we need to select only those edges. The corners occur at the rim of the mug, inside around the base, and around the outside base. Once the edges are selected, we can apply the **Edge Bevel** tool to add more edges and create a rounding effect.

Selecting edges manually can be time-consuming. Luckily, Modo provides a number of ways to speed up selections.

To select all the edges around the border of an area of polygons, first select the polygons, then hold **Ctrl/Cmd**. You'll see that the **Edges** mode button becomes **Bounds**. With **Ctrl/Cmd** still held, select **Bounds** mode and the bounding edges are selected automatically.

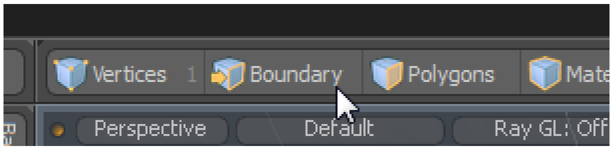
Tip: To select all the edges in a loop, in **Edges** mode, select two adjacent edges and press **L**. Selecting two elements determines the direction of the loop. Loop selection also works for polygons and vertices.

To Select Edges and Apply a Bevel:

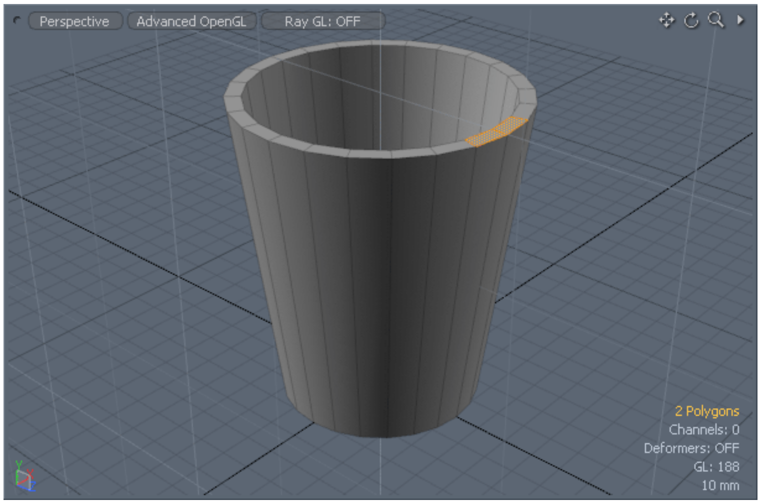
1. In **Polygons** selection mode, select two adjoining polygons on the top rim by clicking and dragging over the polygons.
2. Press **L** to select the entire loop.
3. Rotate the view to see down inside the mug and **Shift** select the inside base of the mug.
4. Rotate to see the outside base of the mug. **Shift** select the bottom polygon.

Tip: If you accidentally lose your selection, use **Ctrl/Cmd** + **Z** to undo and restore it.

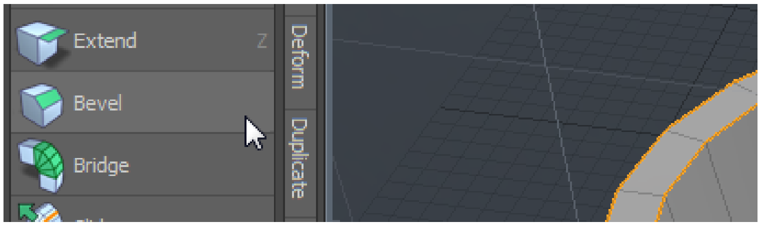
5. Hold **Ctrl/Cmd** to change the **Edges** selection mode button into the **Boundary** button.


Vertices 1
Boundary
Polygons
Mate
Perspective
Default
Ray GL: Off

6. Click the **Boundary** button to convert the selection and automatically enter **Edges** mode.

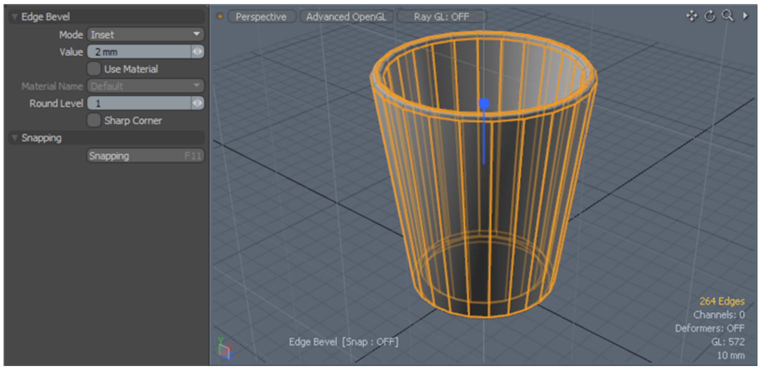

Perspective
Advanced OpenGL
Ray GL: OFF
2 Polygons
Channels: 0
Deformers: OFF
GL: 188
10 mm

7. Select the **Bevel** tool from the **Edge** sub-tab of the toolbox.


Extend
Z
Bevel
Bridge
Deform
Duplicate

8. Click in the 3D viewport to activate the tool's interactive mode.
9. Set **Value** to 2 mm with a **Round Level** of 1 as illustrated below.

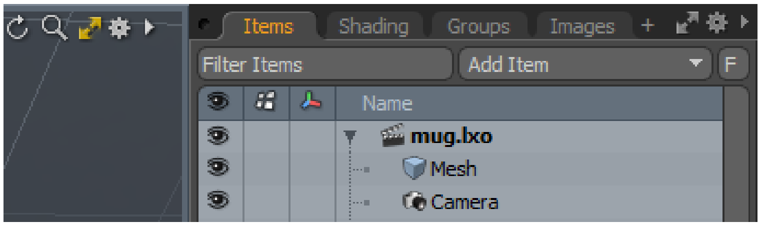
   This makes rounded edges, similar to the original mug image.
10. Press **Q** to drop the tool and **Esc** to drop the selection.


Edge Bevel
Mode: Inset
Value: 2 mm
☐ Use Material
Material Name: Default
Round Level: 1
☐ Sharp Corner
Snapping
Snapping F11
Perspective
Advanced OpenGL
Ray GL: OFF
Edge Bevel [Snap : OFF]
264 Edges
Channels: 0
Deformers: OFF
GL: 572
10 mm

Saving Your Work

To save your work, select **File** > **Save** or press **Ctrl/Cmd** + **S** to open the save dialog. Name your file 'mug', and save it to a directory.

When you save, the scene name in the **Items List** in the upper-right of the Modo interface changes to the filename.



Note: When you open Modo, the scene is called **Untitled** by default, and a small asterisk is shown by the scene to indicate that it hasn't been saved.

Preparing the Surface

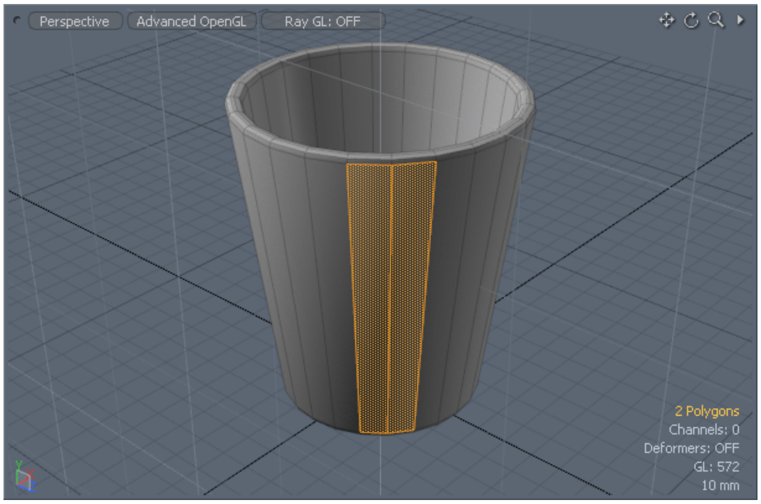
Now let's start to build the handle. You'd be right in thinking that we could model the handle as a separate object and join it to the body of the mug. Though possible, one of the drawbacks with that approach is that it makes it harder to guarantee one continuous smooth surface across the object. Instead, we'll add edges to the side of the mug so that we can grow the handle outwards from the body and preserve continuity.

A quick way to add edges is with the **Loop Slice** tool. To use this we first need to select the surface polygons that we want to slice.

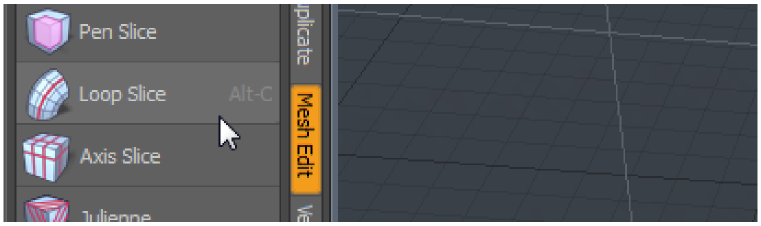
Note: We could have added the extra edges when making the cylinder. However, this would have made shaping the cylinder more difficult. If we started with more edges on the side, shaping the bottom polygon would only taper the single row of polygons above it, and not the whole cylinder.

Add More Edges:

1. In **Polygons** mode, select two adjoining polygon faces, this time on the side of the mug, and press **L** to select the loop.


Perspective
Advanced OpenGL
Ray GL: OFF
2 Polygons
Channels: 0
Deformers: OFF
GL: 572
10 mm

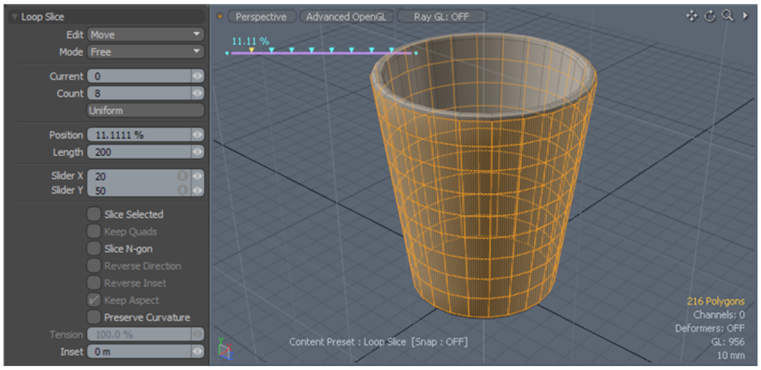
2. Activate the **Loop Slice** tool from the **Mesh Edit** sub-tab of the toolbox, or press **Alt**+**C** to select the tool.


Pen Slice
Loop Slice
Alt-C
Axis Slice
Julienne
uplicate
Mesh Edit
Ve

3. Click in the 3D viewport to activate interactive mode.

   By default, this draws a single slice halfway up the side of the mug.

4. We want 8 slices; in the tool properties, set **Count** to 8.
5. Drop the tool and the selection.



Loop Slice
Edit Move
Mode Free
Current 0
Count 8
Uniform
Position 11.1111 %
Length 200
Slider X 20
Slider Y 50
☐ Slice Selected
☐ Keep Quads
☐ Slice N-gon
☐ Reverse Direction
☐ Reverse Inset
☑ Keep Aspect
☐ Preserve Curvature
Tension 100.0 %
Inset 0 m

Perspective
Advanced OpenGL
Ray GL: OFF
11.11 %
216 Polygons
Channels: 0
Deformers: OFF
GL: 956
10 mm
Content Preset : Loop Slice [Snap : OFF]

Adding Action Centers

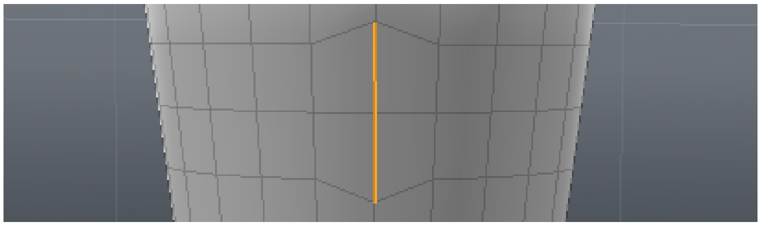
In this stage we'll introduce the Action Center feature. Put simply, the Action Center defines the origin around which a transformation occurs. To make sure an edge grows outwards from its center, we need to set the Action Center to **Local**. This places the Action Center at the center of a selection.

Note: To understand how an action center works, consider the hands on a clock face. The action center of rotation for each hand is at the center of the clock face. Any changes to the action center change the rotation path of the hand. If, say, you put the action center at the number six on the bottom of the clock face, the hand rotates around a circle whose center is located on the number six.

Creating the Handle Buds

You've made the mug body, now let's make the attachment points (or buds) for the handle.

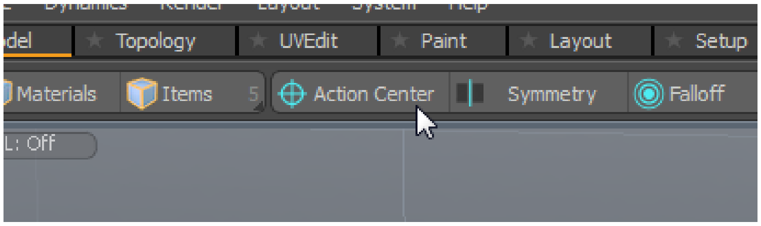
In order to give the handle its slightly rounded shape, we need to shape the area where the handle attaches to the body to be hexagonal. We can do this by lengthening a vertical edge between two polygons, as shown below.



Note: Later, we'll use Modo to round the hexagon into a smooth oval.

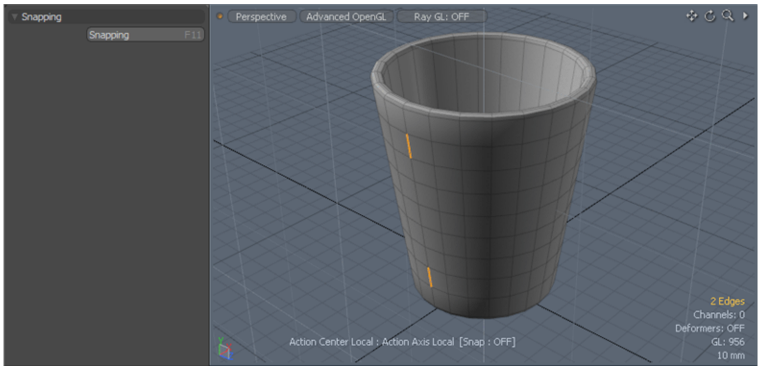
To Shape The Attachment Edges:

1. Set the selection mode to **Edges**.
2. Rotate the view to see the left side of the mug body looking down the x-axis.
3. Select the two edges one span away from both the top and bottom of the mug, that are aligned to the XY plane, as shown in the image below.
4. To the right of the modes toolbar is the **Action Center** selector button. Click on it to open the dropdown menu and select **Local**.



5. Press **R** to activate the **Scale** tool.
6. Scale only the Y (green) axis to 180% by clicking on the green handle and dragging it upward.

   Both edges scale.
7. Drop the scale tool.


Snapping
Snapping F11
Perspective
Advanced OpenGL
Ray GL: OFF
2 Edges
Channels: 0
Deformers: OFF
GL: 956
10 mm
Action Center Local : Action Axis Local [Snap : OFF]

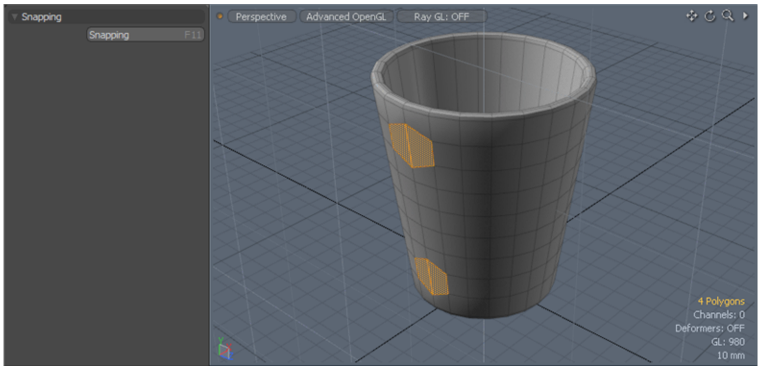
To Extend the Connection Buds:

1. In **Polygons** mode, select each polygon at either side of the edges you just scaled (see below).

   Four polygons should now be selected.
2. Activate the **Polygon Bevel** tool by pressing **Shift**+**B** and then clicking in the 3D viewport.

Note: The **Polygon Bevel** tool handles may not appear in the correct location, but they still work correctly.

3. Adjust the **Inset** (red) handle to 3 mm.
4. Hold **Shift** and click somewhere in the viewport to reset the **Bevel** tool, then bevel outward (with the blue arrow) to a **Shift** of 5 mm and drop the tool.


Snapping
Snapping F11
Perspective
Advanced OpenGL
Ray GL: OFF
4 Polygons
Channels: 0
Deformers: OFF
GL: 980
10 mm

Adjusting Bud Size

The height of each bud is a little too tall, so let's scale them down.

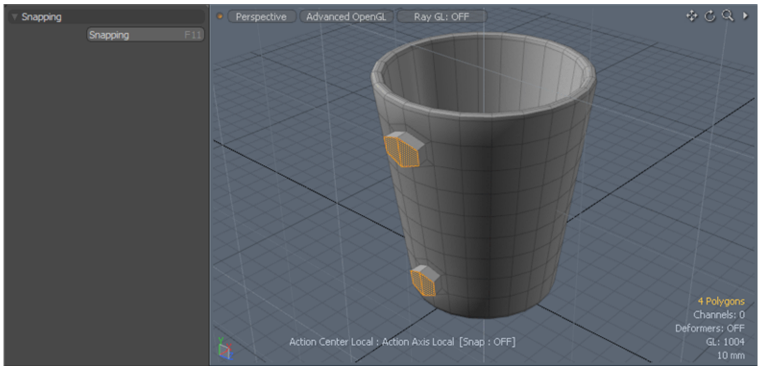
Tip: To speed up the selection, Modo offers some shortcuts. By holding **Shift** and pressing the **Up** arrow, you can extend the selection to include the neighboring polygons.

To Select Neighboring Polygons and Resize:

1. With the 4 polygons still selected, press **Shift** + **Up**.

   The selection is extended to all adjoining polygons and both buds are selected.
2. Press **R** to activate the **Scale** tool and scale the height of the handle buds to 75% on the Y (green) axis.

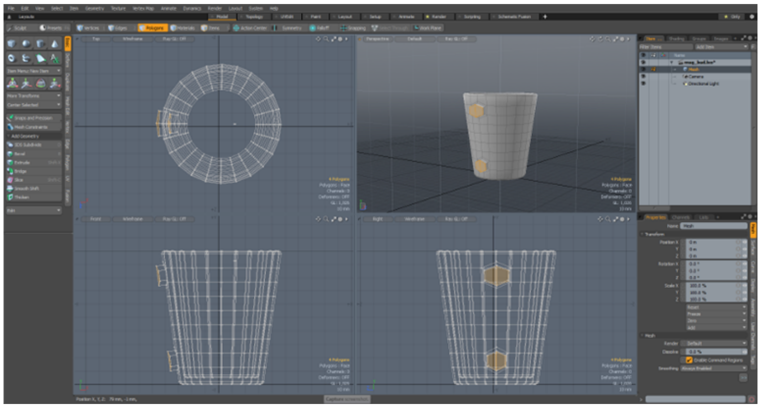
Note: **Local Action Center** is still active from the earlier procedure, so you can scale each one independently with the one action.


Snapping
Snapping
F11
Perspective
Advanced OpenGL
Ray GL: OFF
4 Polygons
Channels: 0
Deformers: OFF
GL: 1004
10 mm
Action Center Local : Action Axis Local [Snap : OFF]

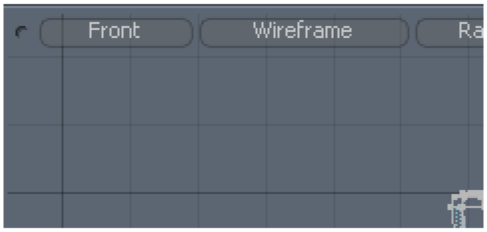
Introducing Orthographic Views

So far we've been working in the **Perspective** view, which shows a 3D view of the object and allows us to explore the space in all axes by 360 degrees. This view is great for most of our needs, but sometimes it's easier to switch to one of the **Orthographic** views to do some detail work. Orthographic simply means that we view the object looking from one side, and see it in two dimensions.

There are several orthographic views in Modo, including **Top**, **Front** and **Right**. Working in an orthographic view is ideal if you want to focus on a specific side, or limit transforms to two dimensions. You can see the default orthographic views in a four-view layout by clicking on the double arrow icon in the top-right corner of the perspective view .

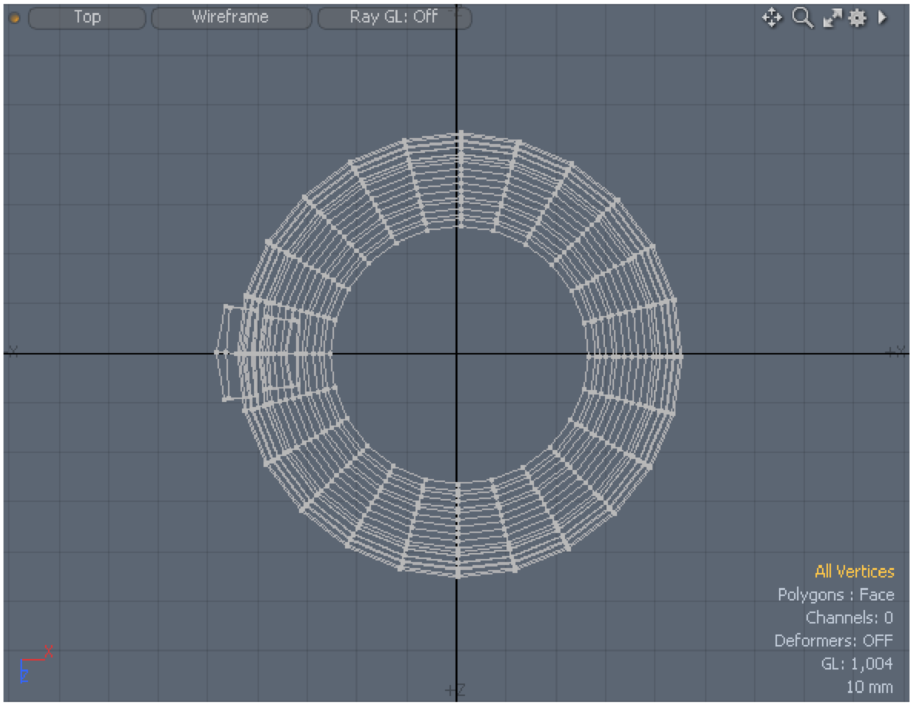


The window on the top-right shows the perspective view. By default, the orthogonal views are drawn in **Wireframe** mode. Wireframe mode shows objects as wire meshes, drawn using only vertices and edges. You can see the type of view and the draw mode in the top-left of each window.



Centering the Model

The next steps rely on the buds being aligned with the X-axis, such that if you look from the **Top** view, the buds extend to the left with the X-axis at their center, as shown below.



All Vertices
Polygons : Face
Channels: 0
Deformers: OFF
GL: 1,004
10 mm

To Rotate the Top View:

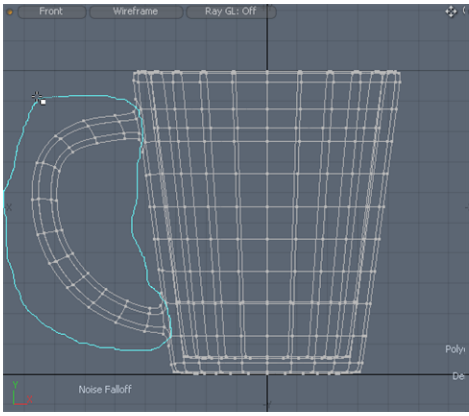
1. In **Polygons** mode, deselect all polygons.
2. Click the multi-viewport button to open the four-view layout.
3. Set the **Action Center** to **Origin**.

   This puts the center of rotation at the world origin (X=0, Y=0, Z=0).
4. Press **E** to activate the rotate tool.
5. In the **Top** view, click and drag to align the buds with the X-axis (as above).
6. Press **Q** to drop the **Rotate** tool.

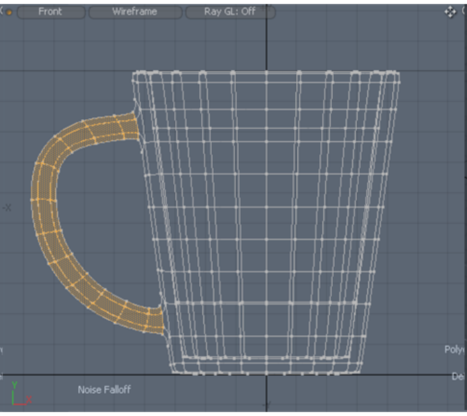
Introducing Lasso Selection

The lasso selection method lets you select regions of an object by drawing a boundary around the region. To use the lasso, press and hold the right or middle mouse button and drag the selection lasso.

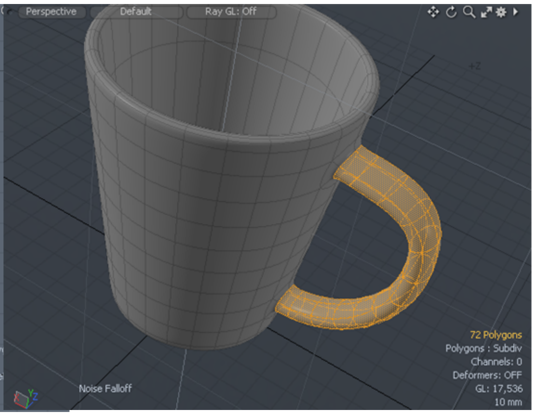
Note: A polygon is only included in the selection if the entire polygon is within the lasso boundary.


Front
Wireframe
Ray GL: Off
Noise Falloff

An orthogonal view. The mug handle has been selected using a right-mouse lasso selection.


Front
Wireframe
Ray GL: Off
Noise Falloff

The result of the lasso selection as seen in the orthogonal view.


Perspective
Default
Ray GL: Off
Noise Falloff
72 Polygons
Polygons : Subdiv
Channels: 0
Deformers: OFF
GL: 17,536
10 mm

The selection as seen in the perspective view. Note how the selection includes both sides of the handle.

Note: You can use either the right-mouse or middle-mouse when drawing a lasso with different results. In **Wireframe** view, right-mouse + drag selects elements at the front and any that are occluded, middle-mouse + drag selects only elements in front.

In a non-wireframe draw mode, such as **Default**, the reverse applies.

Connecting the Handle Buds

The next step is to connect the two buds using the Bridge tool. This tool creates new tube geometry that connects two selected areas. To help Modo make a good bridge, the connection surfaces must be slightly inclined towards each other.

Note: You can use the **Bridge** tool with polygons or edges.

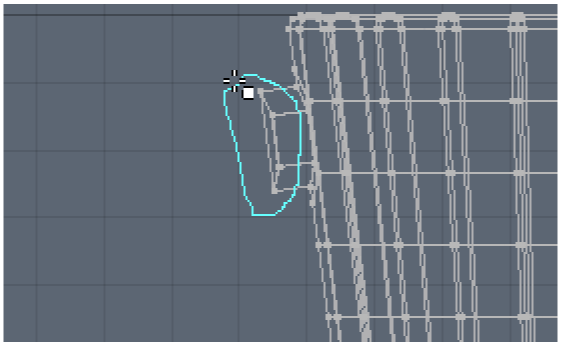
To Orient to Bridge:

1. Drop any active tools and deselect any polygons.
2. If you haven't already, click the multi-viewport button to open the four-view layout.

Tip: Click in a viewport and press **A** to fit the object to the viewport.

3. Set the **Action Center** type to **(none)**, to use the default action center.
4. Locate the orthographic view of the upright mug with the handle buds extending out to the side.
5. Right-click+drag around the polygons on the outside surface of the top handle bud.

Tip: Remember, a polygon must be completely within the lasso to be selected.

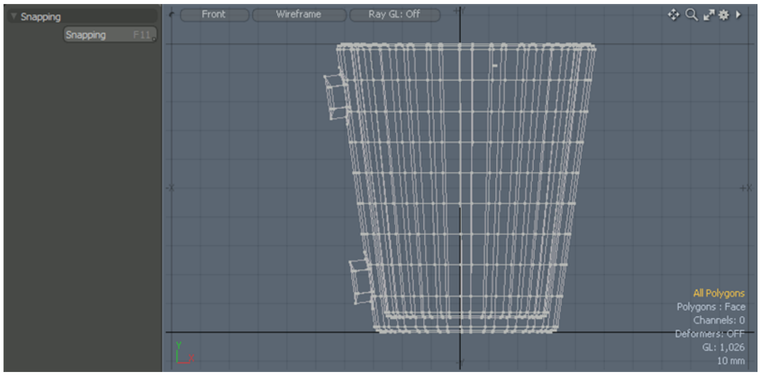


Tip: Adjust the **Perspective** view so that you can see the buds and make sure the selections are correct.

6. With the two polygons now selected, press **E** to activate the **Rotate** tool.
7. Use the rotation handle to rotate the polygons downward 20 degrees.

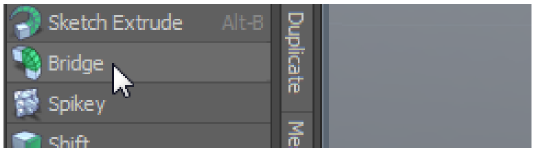
Tip: **The Offset Angle** is shown at the bottom of the Modo window, below the viewport.

8. Drop the **Rotate** tool.
9. Repeat steps 3 to 7 with the lower handle bud, but rotate the lower bud upward 20 degrees.


Snapping
Snapping F11
Front
Wireframe
Ray GL: Off
All Polygons
Polygons : Face
Channels: 0
Deformers: OFF
GL: 1,026
10 mm

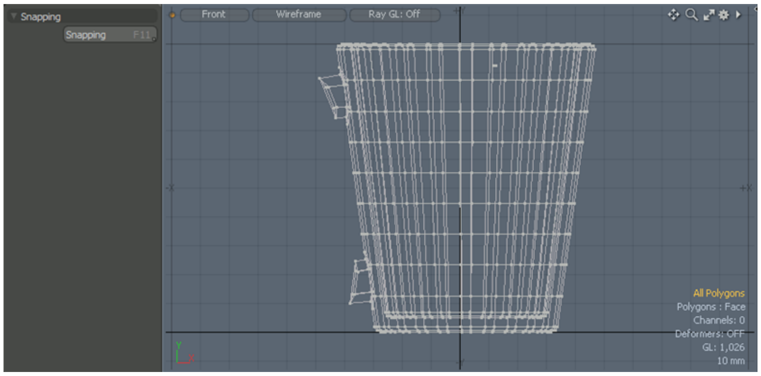
To Apply the Bridge Tool:

1. Press **Q** to drop any active tools.
2. Select the outward faces of the handle buds on both the upper and lower bud. Hold **Shift** when selecting to add to an existing selection.
3. Activate the **Bridge** tool from the **Polygon** sub-tab of the toolbox.


Sketch Extrude
Alt-B
Bridge
Spikey
Shift
Duplicate
Me

4. Click in the **Perspective** viewport to enter interactive mode.

   A single set of polygons is drawn as a straight line between the groups.
5. In the **Properties** panel, set the **Segments** value to 10 to add polygons to the bridge.
6. Adjust the **Tension** value to 200%, curving the handle out.


Snapping
Snapping
F11
Front
Wireframe
Ray GL: Off
All Polygons
Polygons : Face
Channels: 0
Deformers: OFF
GL: 1,026
10 mm

7. Drop the tool.

Creating Falloffs

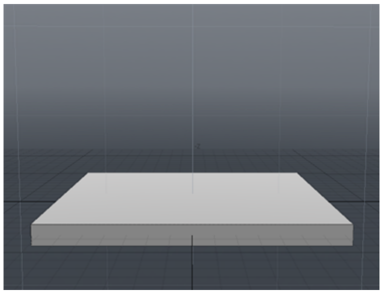
Instead of moving the geometry as a single rigid piece, the falloff lets you move one area more than another, with a soft fade between the two extremes. This lets you apply bulges or slopes to a surface easily.

A falloff is often drawn as a wireframe in the viewport (depending on the type), with editing handles to change the shape of the falloff and adjust its intensity.

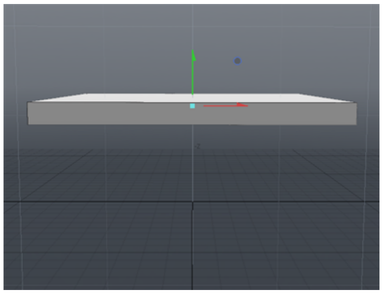
You can add falloffs to a tool separately, or, in the case of some tools (including the **Soft Move** tool), a falloff is added automatically to give the tool its characteristic behavior. In both cases, the falloff and the transform are shown as two separate tools in the viewport.

There are several types of falloff, each describing a different shape for the falloff region. For example, a Linear falloff is drawn as a wedge shape, where the geometry closer to the narrow end of the wedge is affected less than the geometry at the wide end. The amount of the affect has a gradual, linear change along the length of the falloff.

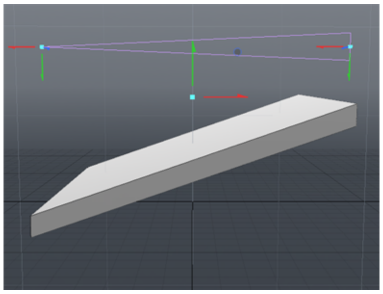
The images below compare the effect of a vertical (Y-axis) translate with and without a Linear falloff.



The original object.



The object is translated in the Y-axis, without a falloff.



The object is translated in the Y-axis using a linear falloff. You can see the falloff above the object.

The **Soft Move** tool uses a spherical falloff. The geometry at the center of the sphere is affected the most, and the effect softens to zero at the edge of the sphere. This lets us move the geometry with minimal changes at the edges of the drag region.

Shaping the Handle

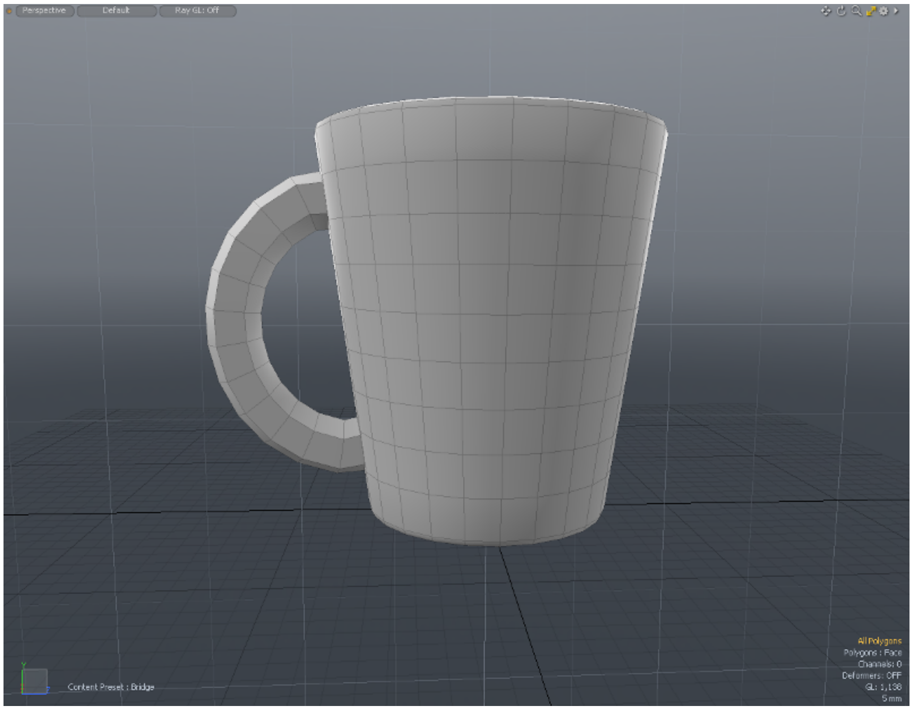
We've made the geometry for the handle, but it doesn't match the shape in the original photo. To fix this, Modo has a palette of tools to deform shapes and bend geometry.

In this case, we'll use the Soft Move tool. The **Soft Move** tool lets us drag geometry in any direction. This is similar to the regular translate tool (**W** key) but the **Soft Move** tool uses a falloff.

To Soft Move the Handle:

1. Return to single view perspective mode by clicking the double arrow in the **Perspective** view.
2. Drop any active tools.
3. Orient the viewport so that you are looking at the side of the mug with the handle extending to the left.

   The Work Plane snaps to the facing plane.

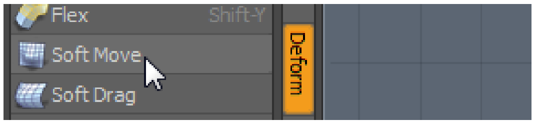

Perspective
Default
Ray GL: Off
Content Preset : Bridge
All Polygons
Polygons : Face
Channels: 0
Deformers: OFF
GL: 1,138
5 mm

4. Select the entire handle (front and back) using middle-mouse and drag.

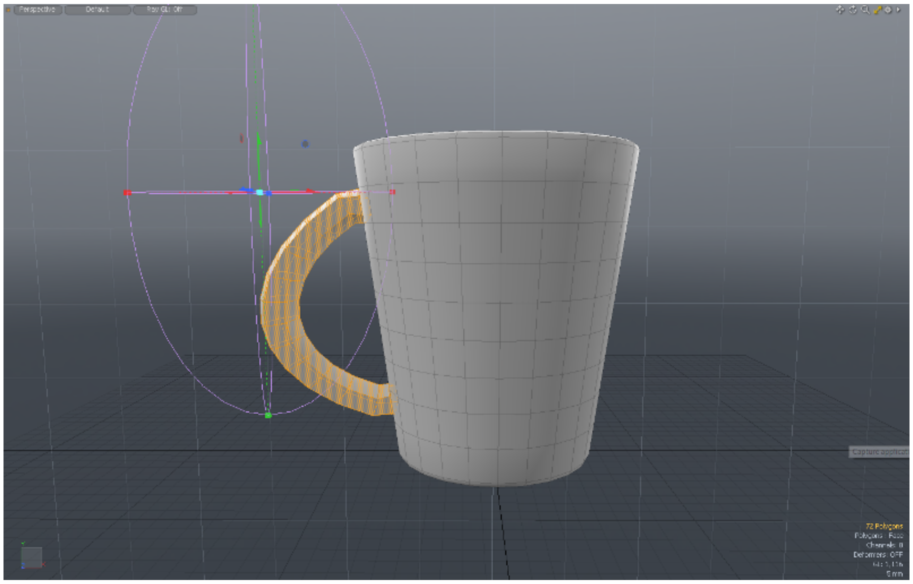
   You'll have to move the viewport to select missed polygons or deselect unwanted polygons.

Tip: Remember to add to a selection using **Shift** + click and to remove using **Ctrl/Cmd** + click.

5. Under the **Deform** sub-tab of the toolbox select the **Soft Move** tool.

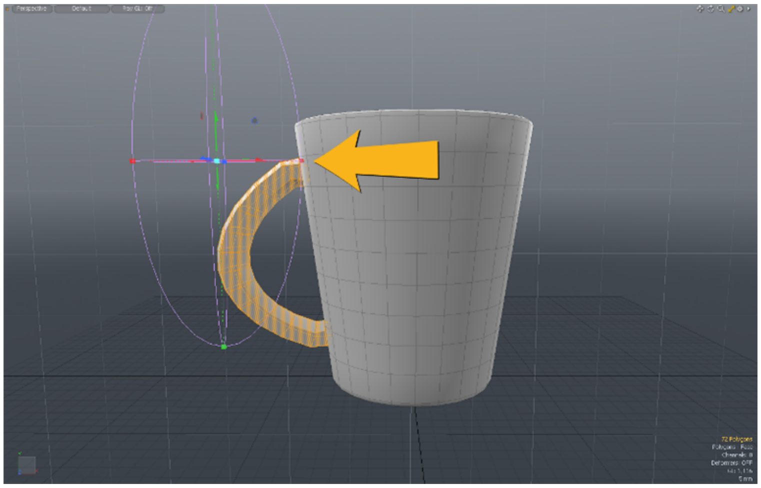

Flex
Shift-Y
Soft Move
Soft Drag
Deform

6. With the mug facing you and the handle to the left, click above the left edge of the mug handle, level with the top of the handle, to position the tool as shown below.

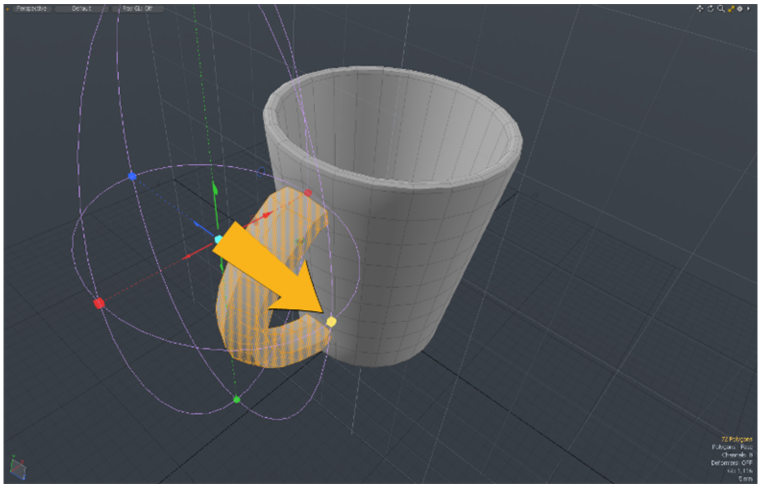


The tool has two components, the falloff and the move control. At first they are in the same position.

7. Resize the width of the falloff so that it only covers the handle. Drag the red (X) box at the edge of the falloff to the left, up to the edge of the handle.



8. Rotate the viewport to see more of the mug handle.
9. Use the blue box, at the intersection of the pink lines, to widen the falloff, so that it resembles the one below.



Tip: When dragging the blue box, be careful not to select the blue arrow that drags the move tool. if you drag the move tool the mug handle moves. If this happens, **Ctrl** + **Z** and try again.

10. Orient the viewport to look at the side of the mug with the handle to the left (see below).

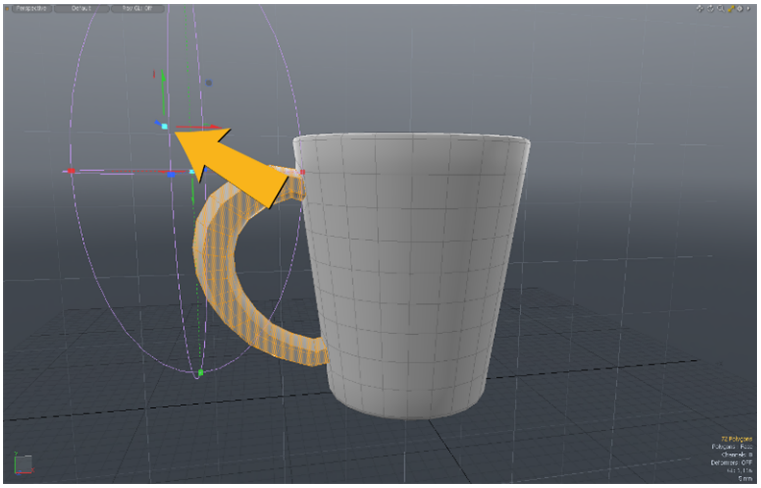
    The work plane will align to the handle.

11. Click and drag the upward green arrow to lift the move control.

    Now the falloff control and the move control are in different positions and can be controlled separately.

12. Try dragging the move control up and to the left to raise the top of the mug handle.

Tip: If you use the light blue box at the center of the Move control, the Move control shifts in the Work Plane, as does the handle geometry.



13. By careful adjustment of with move tool and the falloff, you should be able to reshape the handle to look roughly similar to the original (it doesn't have to be perfect!).

14. When you're happy with the handle, press **Q** to drop the tool.

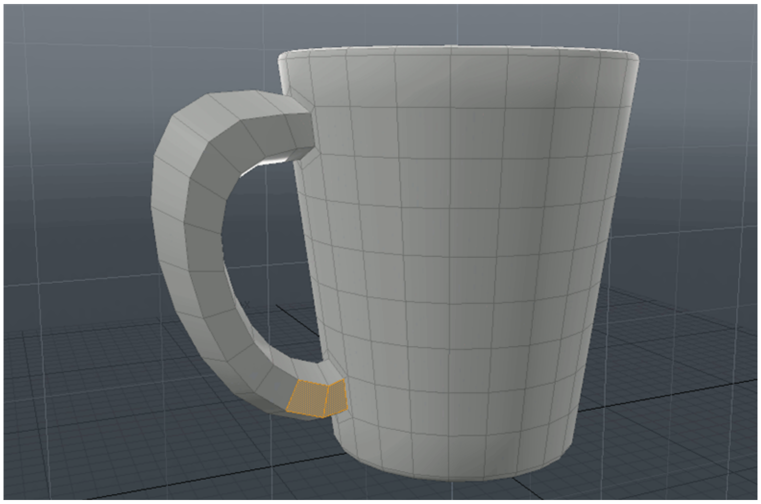
Evening-Out the Handle

In our model, the top of the handle is wider than the top, but in the original mug, the handle is an even width along its length. This can be fixed with the **Rotate** tool. We just need to rotate each side of the handle slightly to straighten them up.

Tip: To quickly select elements in a strip, select the first two elements and then keep pressing **Up** to select the rest.

To Rotate the Sides:

1. With the side of the handle facing you, paint-select the first two polygons at the base of the handle.
2. Press **Up** a few times to select the entire side.
3. Rotate the viewport so that the handle is at the center of the mug.



4. Activate the **Rotate** tool (**E** key) and adjust the facing rotate circle until the side of the handle is straight.
5. Drop the **Rotate** tool.
6. Repeat steps 1-5 for the other side of the handle.
7. Press **Esc** to drop the selection.

Smoothing the Surface and Adding Color

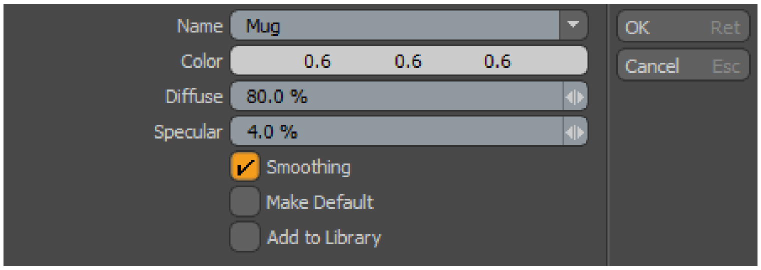
Although similar in shape to the original, the model of the mug is made of flat polygons and has a blocky appearance. To fix this, we can apply subdivision surface smoothing to automatically add detail to the mug and give it a smooth, rounded appearance.

Finally, we'll give the mug a color similar to the one in the photo. In Modo, color and other surface details of an object are stored in a **Material**.

To Subdivide and Add Color:

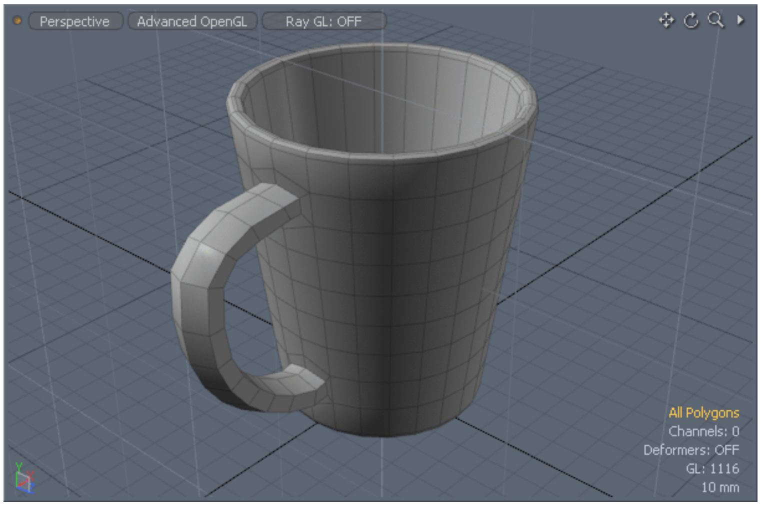
1. Deselect any polygons by clicking away from the mug in the viewport.
2. Subdivide the model by pressing **Tab** or select **Geometry** > **Toggle Subdivision**.

   This adds additional polygons to the model that are formed from the base low-resolution mesh.
3. Press **M** to open the **Polygon Set Material** dialog.



Name: Mug
Color: 0.6 0.6 0.6
Diffuse: 80.0 %
Specular: 4.0 %
☑ Smoothing
☐ Make Default
☐ Add to Library

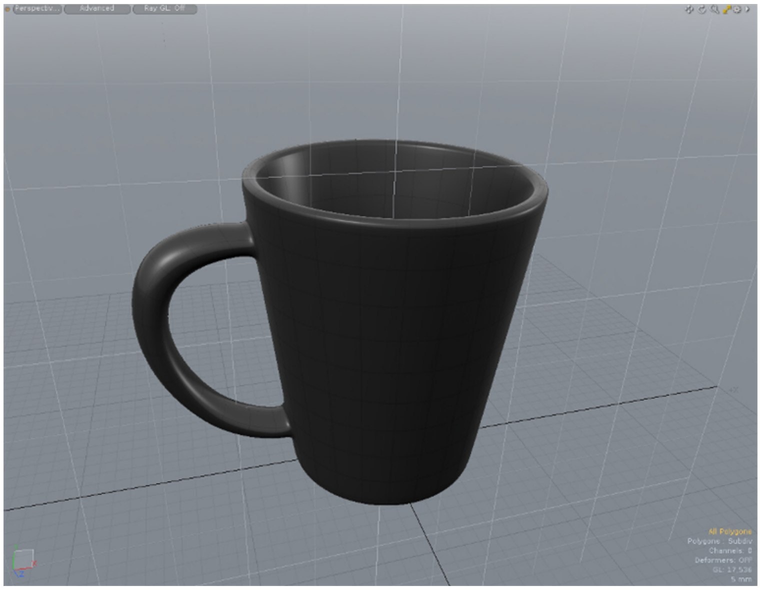
4. Call the material 'Mug'.
5. Click on the color values to open the color picker.
6. Set the **Red**, **Green** and **Blue** values to 45. Click **OK**.
7. Save your model.



Next Steps

Well done, you've successfully made a 3D model from scratch!

To see your mug using a more sophisticated rendering mechanism, change the drawing method from **Default** to **Advanced**, using the center button in the top-left of the viewport.

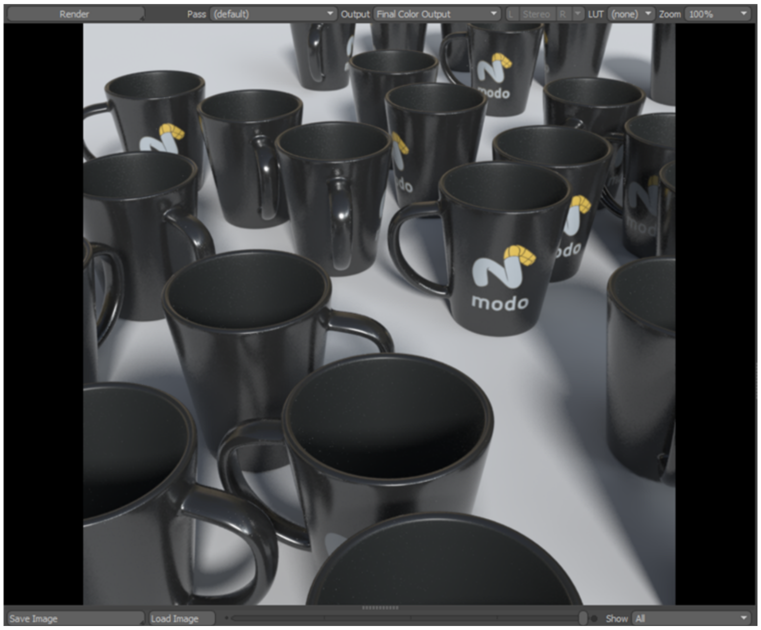

Perspectiv...
Advanced
Ray GL: Off
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: OFF
GL: 17,536
5 mm

The next step is the Render the Mug. Here, we'll set up a simple scene with a collection of mugs, add the logo, and render out our scene into a finished image.

Render the Mug

This is a continuation of the Model the Mug tutorial. The information presented here builds on the lessons learned in that tutorial. Learn how to render the mug to produce a final image. You can start this tutorial by downloading the coffee mug model from here. You'll need **CoffeeMugTutorialModel.lxo**.

Here's a picture of the a final image we'll render.



Pass (default)
Output Final Color Output
LUT (none)
Zoom 100%
Show All

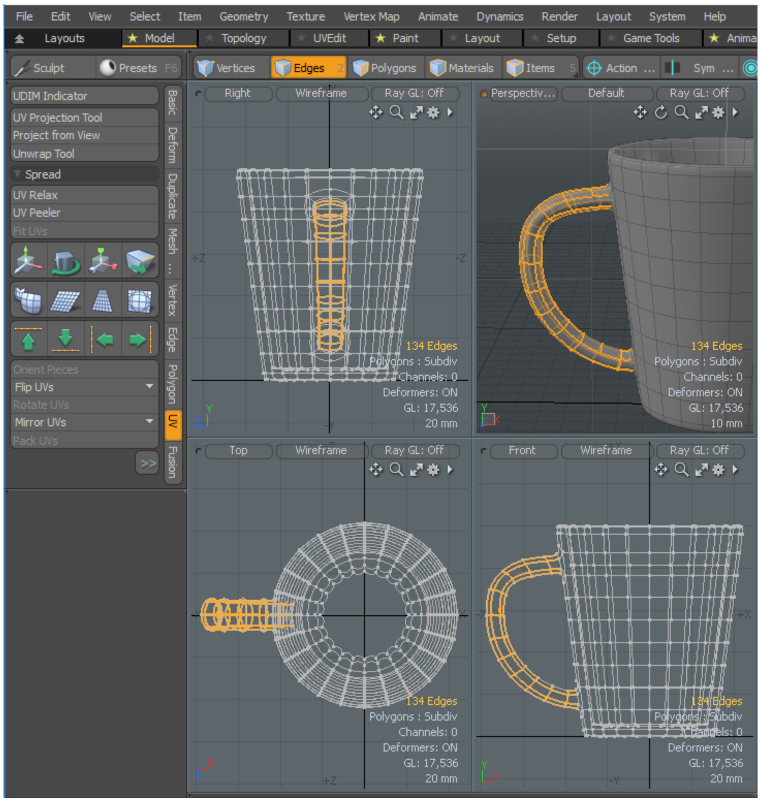
It would be simple enough to just press **Render** and have an image of the coffee mug at this point, but it wouldn't be that interesting of an image, especially with the default settings. While rendering a single object has its place, in most cases you render a scene in Modo, with multiple objects, a background, and some distinct lighting. We'll assign a UV map, adjust the surface of the mug and clone it, using a preset to make a basic scene, and then adjust camera and render item settings to produce a final rendering.

Making a UV Map

Applying UV maps is used for assigning how a flat texture gets applied to a 3D surface at render time. To make things a little easier, let's hide the handle first.

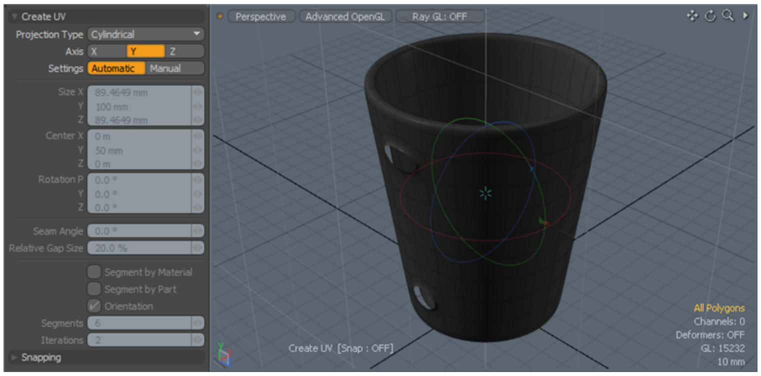
1. With the mug model open, click the **Model** tab to ensure that you are working in the Model layout.
2. Using the lasso selection method, covered in the previous tutorial, from a side view, right-click and drag out a shape, selecting just the handle.

Tip: To avoid selecting some of the body polygons, select the main part of the handle, then press **Shift**+**Up** to add the adjoining polygons where the handle meets the body.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Layouts
Model
Topology
UVEdit
Paint
Layout
Setup
Game Tools
Sculpt
Presets
Vertices
Edges
Polygons
Materials
Items
Action
Sym
UDIM Indicator
UV Projection Tool
Project from View
Unwrap Tool
Spread
UV Relax
UV Peeler
Fit UVs
Orient Pieces
Flip UVs
Rotate UVs
Mirror UVs
Pack UVs
Basic
Deform
Duplicate
Mesh
Vertex
Edge
Polygon
UV
Fusion
Right
Wireframe
Ray GL: Off
Perspective
Default
Top
Front
134 Edges
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 17,536
20 mm
10 mm

3. Once the entire handle is selected, press **H** to hide the geometry.

   Hiding geometry is a great way to concentrate on just a group of polygons in an object layer, and to ensure that geometry remains unmodified, as hidden geometry cannot be selected or edited.


Create UV
Projection Type: Cylindrical
Axis: Y
Settings: Automatic
Size X: 89.4649 mm
Y: 100 mm
Z: 89.4649 mm
Center X: 0 m
Y: 50 mm
Z: 0 m
Rotation P: 0.0 °
Y: 0.0 °
Z: 0.0 °
Seam Angle: 0.0 °
Relative Gap Size: 20.0 %
☐ Segment by Material
☐ Segment by Part
☑ Orientation
Segments: 6
Iterations: 2
Snapping

Perspective
Advanced OpenGL
Ray GL: OFF
Create UV [Snap : OFF]
All Polygons
Channels: 0
Deformers: OFF
GL: 15232
10 mm

In the bottom panel of the toolbox there are some UV tools, not as many as in the **UVEdit** layout, but enough to do what we need to do here.

4. In the toolbox, in the left pane, click the **UV** sub-tab.
5. Select the **UV Projection Tool** and click in the 3D viewport to activate the tool.
6. In the **Tool Properties** panel, set the **Projection Type** to **Cylindrical**, making sure the axis is set to **Y** so the map wraps around the mug's height. Leave the **Settings** on **Automatic**.

This sizes the cylinder volume to match that of the selected, or in this case visible, the polygons' outer bounding box volume. A bounding box is simply a rectangular volume of space that represents an object's total height, width, and depth.

The map is made.

7. Press the **spacebar** to drop the tool accepting the UV map values.

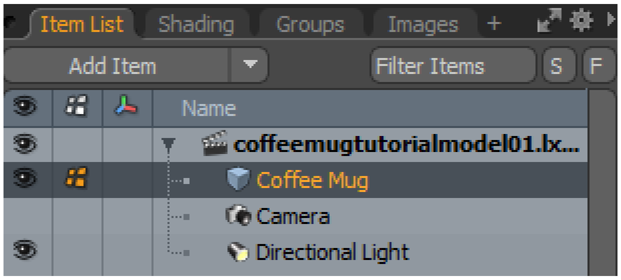
Working in Items Mode

We now need to assign an image map that matches the logo on the original reference image.

Before we continue, there is a rather important concept you must understand to use the tools to your best benefit. So far you've used the **Polygons** and **Edges** selection modes to modify the geometry. For laying out the scene work in the **Items** selection mode. In this selection mode you can only modify an item's position, rotation, and scale, collectively known as its **transform values**. The previous modes, **Vertices**, **Edges**, and **Polygons** are collectively known as the **component editing modes** and are strictly for editing the geometry itself. Two distinct operations take place in either the **component** or **Items** modes, though their results are identical. Each item layer in the **Item List** is a container for geometry. While in **Items** mode, you transform the container, but while in any of the **component** modes, only the contents of that container can be edited.

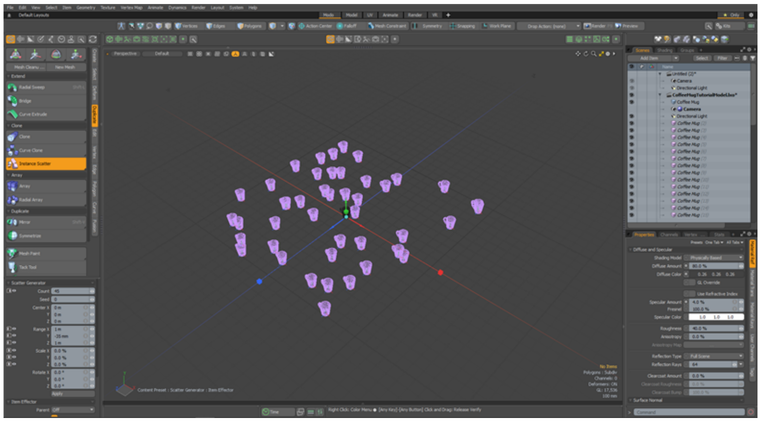
1. Press **U** to make the handle visible again.
2. Click on the layer name, called 'Mesh' by default or 'Mug' if you grabbed the included file, to highlight the layer.

   Now any operations applied to the scene only affects that selected layer.



3. Click and hold on the **Scatter Clone** button, available under the **Duplicate** tab, to reveal the alternate options. Click **Instance Scatter**, then click in the 3D viewport to activate interactive mode.

   You might not see anything in the viewport right away, aside from a tool handle.



4. Hover your cursor over the 3D viewport and press **A** to view all items in the scene.

   Eight mugs appear, the default clone number, scattered all around. In the **Tools Property** panel, you can adjust the **Count** value to any number you like.
5. Set the **Range** for the vertical axis **Y** at **0 m** to set all coffee mugs on the ground plane.
6. To randomly rotate the clones, set the **Rotate Y**value to **360°** and then adjust the **Range** on the **X** and **Z** axes to bring the mugs closer together.

   Some mugs may overlap. If you're not happy with the placement of the clones , you can adjust the **Seed** value until you receive a result you are happy with.
7. Once finished, drop the tool by pressing the **spacebar**.

Propagating updates to Clones

In addition to the original mug, you now have a number of pink wireframe clones. The pink color denotes that the duplicates are Instances, virtual copies that reference the original Mesh Item. They contain no real geometry, just a reference to their source, so they keep the scene file size small. Any modifications made to the source item propagate to the clones automatically, making it easy to modify them later on.

1. Unlike other types of duplicated objects, such as Replicator Item, you have individual control over each mug instance, so while still in **Items** mode, click on any mug to select it, then press **W** to move the mug's position.
2. Click on the green circle handle to constrain the movement parallel to the ground plane, making it easy to modify each item's position.
3. While the **Move** tool is active, click over another mug instance to select it, and then use the tool handles to move it around. Keep doing this until all mugs are positioned so they are no longer intersecting.

Tip: Tumbling the viewport helps you get a better view of the scene when making adjustments. Press **Shift**+**A** to zoom the currently-selected items to fill the entire viewport, making it a little easier to navigate around them.

Grouping Layers

The **Item List** now displays individual duplicated mug layers. Lets group these layers to organize the **Item List**.

1. Click on the topmost Instance layer, scroll to the bottom and hold **Shift**, and click on the lowest layer, selecting all the duplicate mug layers.
2. Press **Ctrl**+**G** to group all the layers together into a **Group Locator**. Double-click the Group Locator name, type 'Duplicate Mugs' and press **Enter**. Toggling the ▶ icon before the name hides all the layers in the list, making it easier to navigate.

Using a Preset Mesh

Now we need only one additional piece of geometry to fill out the scene and for that we use a **Preset** mesh.

1. In the **Preset Browser** across the bottom of the screen, select the **Meshes** tab.
2. Click on the Preset Mesh named **Ground Plane**.

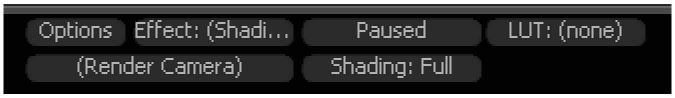
   This serves as the ground in our scene. It catches cast shadows, so the rendered mugs do not look like they are floating in space.
3. Save your scene.

Setting Up the Render

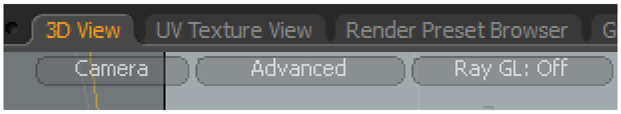
For the rendering portion of this tutorial, switch to the **Render** layout. In the upper right viewport, expose the **Shader Tree** panel. All the surfacing and render settings are set through the items in this list.

First, let's position the camera. This can be done in the main viewport area. There are three 3D viewports to work with in the **Render** layout:

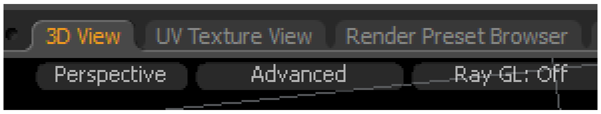
- the **RenderPreview** in the top pane that gives a quick reduced quality sample of the final image,



- the **Camera > Render Camera** view in the bottom pane that shows exactly how the camera sees the geometry,


3D View
UV Texture View
Render Preset Browser
Camera
Advanced
Ray GL: Off

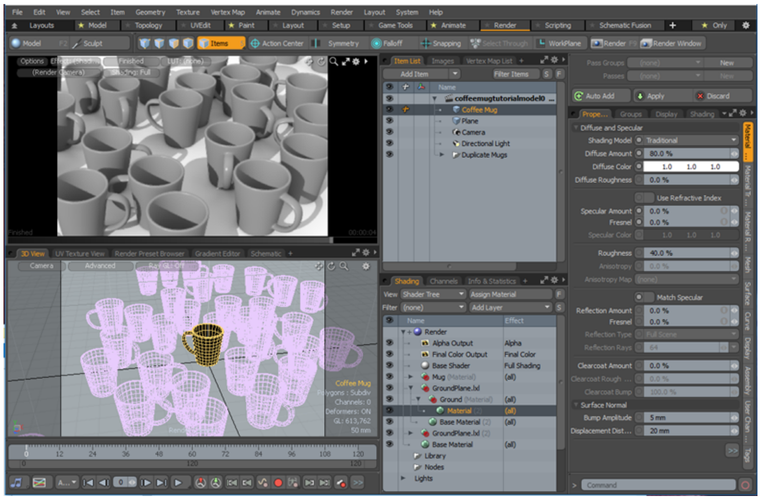
- and the **Advanced Perspective** view in the bottom pane that is similar to the one in the **Model** layout.


3D View
UV Texture View
Render Preset Browser
Perspective
Advanced
Ray GL: Off

By clicking in the upper right corner of the viewport, you can navigate around, using the standard viewport controls.

1. Use **Alt**+click and drag to rotate the view and **Ctrl**+**Alt**+left-click and drag to zoom in and out. With these actions you actually move the **Camera** item, there is no need to select the **Camera** item and then activate a transform tool.

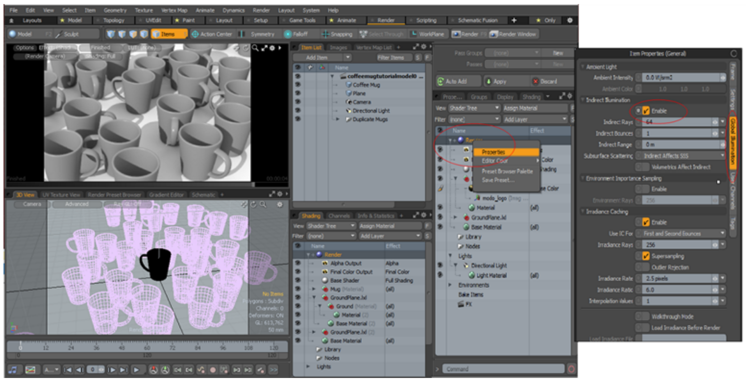
   As you change the view, you can see the preview update in real time.
2. To set a wider angle of view than the default settings provide, hold the **Ctrl**+**Alt** keys and drag in the viewport.
3. Feel free to move any of the mug items once the camera is set, using the **Perspective** viewport.
4. Once the **Camera** position is set, save your scene.



Surfacing the Mugs

Before we start surfacing the mug, we need to adjust the lighting, as it plays such an important role in the rendered scene. We work in a facsimile of the lighting used for the final render when adjusting surfaces. The **Shader Tree**, which is the panel labeled **Shading**, contains the various layers that represent the shading and rendering of our scene. Selecting one of the layers displays its attributes in the **Properties** panel. If it is not visible, click the **Shading** tab, exposing the viewport. The very top item in the **Shader Tree** is called the

Render item. Right-click it and select **Properties**. This item holds the settings related to rendering the scene, such as frame size, anti-aliasing and other quality-related settings, and also **Global Illumination**.



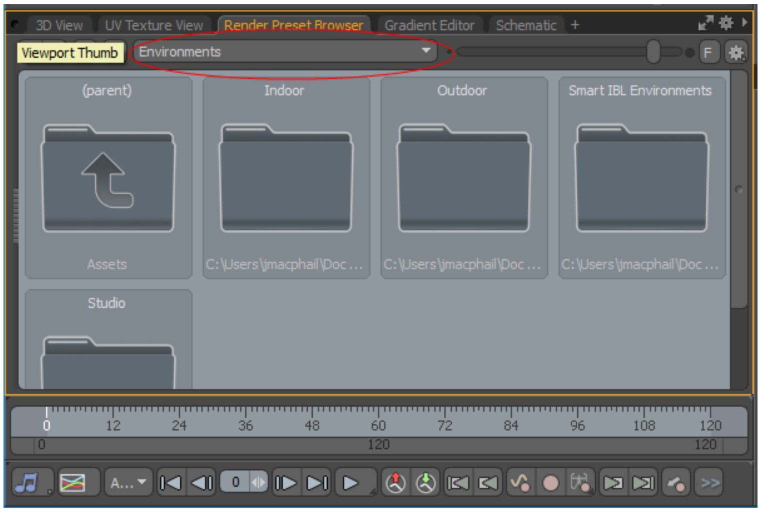
Global Illumination is a technique used in 3D graphics to mimic the effects of real world light bouncing around in a scene, producing extremely realistic results. It is turned on by default in a new scene.

1. To toggle this setting, select the **Render** item in the **Shader Tree**, under the **Properties** tab, and then select the **Global Illumination** tab.
2. Check the box next to **Indirect Illumination** to enable it.

Note: When enabling **Global Illumination**, Modo lights the scene using the settings of the Environment Item. By default, this comes from a gradient applied to the Environment Material.

If you prefer the look of **Image Based Lighting**, add an HDRI environment. This can be done using the presets Modo provides.

1. In the bottom left pane, click the **Render Preset Browser** tab.
2. Using the controls, click to twirl open the small arrows, locate the **Environments** > **Outdoor** folder.


3D View
UV Texture View
Render Preset Browser
Gradient Editor
Schematic
Viewport Thumb
Environments
(parent)
Indoor
Outdoor
Smart IBL Environments
Assets
C:\Users\jmacphail\Doc ...
C:\Users\jmacphail\Doc ...
C:\Users\jmacphail\Doc ...
Studio
0
12
24
36
48
60
72
84
96
108
120
0
120
120
A...
0

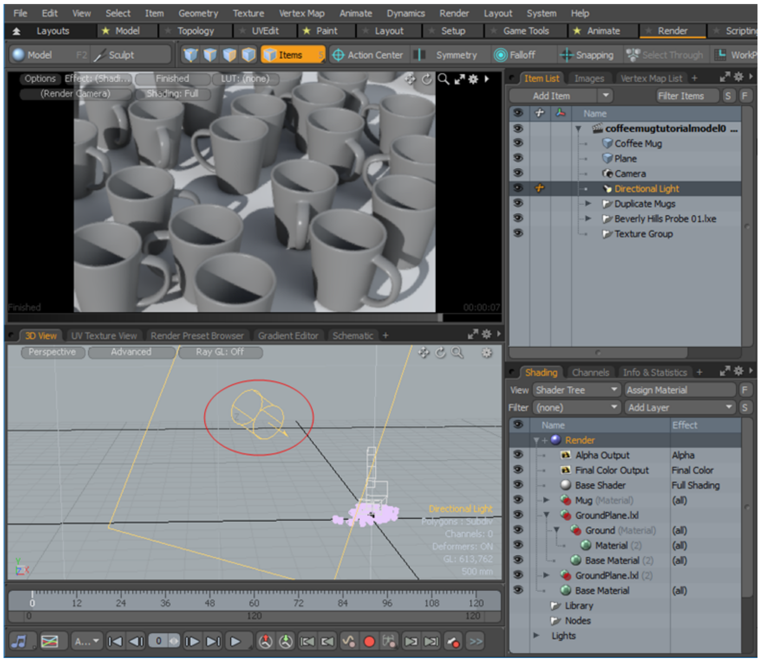
3. Double-click the **Beverly Hills Probe 01** preset to add the image to the Environment item.

4. Click the **3D View** tab to view your model in a perspective view.

   The lighting dramatically changes for the scene in the **Render Preview**, producing a nice outdoor look to the rendered image.

The lighting in the Beverly Hills image is coming from the left-hand side for our image (-X), so let's rotate the **Directional Light** to match that. This can be best accomplished in the **Perspective** view.

1. Hold **Ctrl**+**Alt** and drag in the viewport to zoom out and then select the **Directional Light** item in the viewport by clicking over it when it highlights.


File
Edit
View
Select
Item
Geometry
Texture
Vertex Map
Animate
Dynamics
Render
Layout
System
Help
Item List
coffeemugtutorialmodel0
Coffee Mug
Plane
Camera
Directional Light
Duplicate Mugs
Beverly Hills Probe 01.lxe
Texture Group
3D View
Perspective
Shading
Render
Alpha Output
Final Color Output
Base Shader
Mug (Material)
GroundPlane.lxl
Ground (Material)
Material
Base Material
Library
Nodes
Lights

2. With the item selected, in the menu bar, click **Edit** > **Action Center** and set it to **Origin**.

   This allows you to easily rotate the light from the world's center.

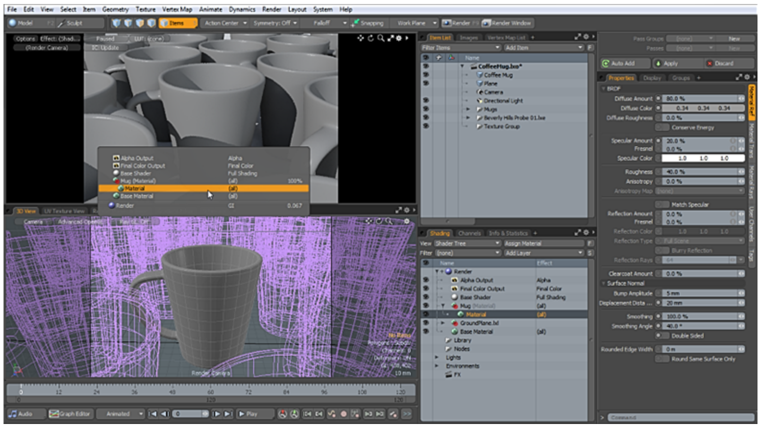
3. Press **E** to activate the Rotate tool and then grab the **Y** axis handle (the green circle), and rotate the light about 45° to the right.

4. Press the **spacebar** to drop the tool.

The next part is surfacing. Earlier in the modeling tutorial, a **Material** tag was applied to the model, named 'Mug'. This action also created a Material Group. This particular item creates a procedural mask that limits any surfacing contained under that group in the **Shader Tree** to just surfaces tagged with that name. For more details about this, see Shader Tree.

Each **Material Group** mask contains a **Material** item that controls most of the attributes of any surface, such as color and luminosity. We need to select this item to edit it in the **Properties** panel. While it's easy to select the proper **Material** item in the **Shader Tree** in this case, Modo allows you to quickly see what layers are attributed to any part of the scene, by using a shortcut in the **Preview** viewport.

1. Right-click over one of the coffee mugs in the **Preview** panel to open a context menu with a list of all the layers that apply to the mug.
2. In the menu, click on the **Shaders** > **Mug Material** > **Material** item. The item is selected in the **Shader Tree**.

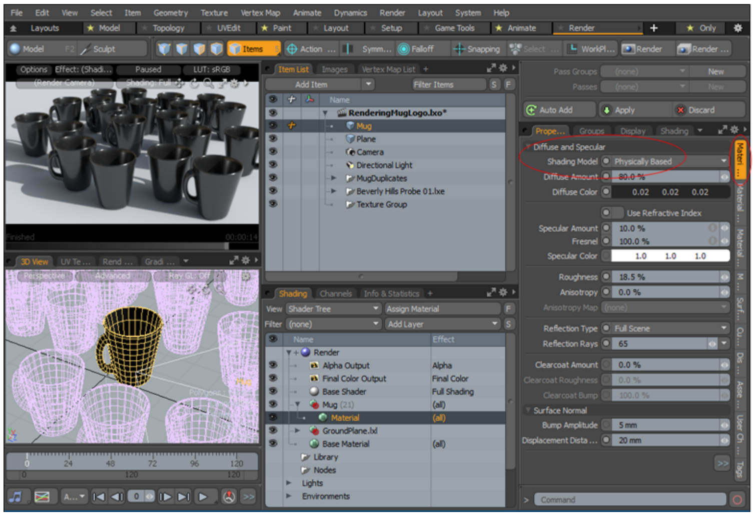


Looking over the attributes in the **Properties** viewport, we need to adjust the color, as it is much lighter than the mug in the original image.

1. In the **Shading > Shader Tree** pane on the right, right-click and select **Properties**.
2. Click anywhere on the **Diffuse Color** swatch to open the Modo **Color Picker**, and adjust the value to a much darker gray color without going completely black. In this example the values are set to 0.02 respectively.
3. To make your surface more realistic, set **Shading Model** to **Physically Based** and set the following settings:
   - Change the **Diffuse Amount** to 80% .
   - Change the **Specular Amount** and **Fresnel** settings to 10% and 100%, respectively.
   - Change the **Roughness** setting to 18.5%.

   You can see now its color is much closer to the original mug.

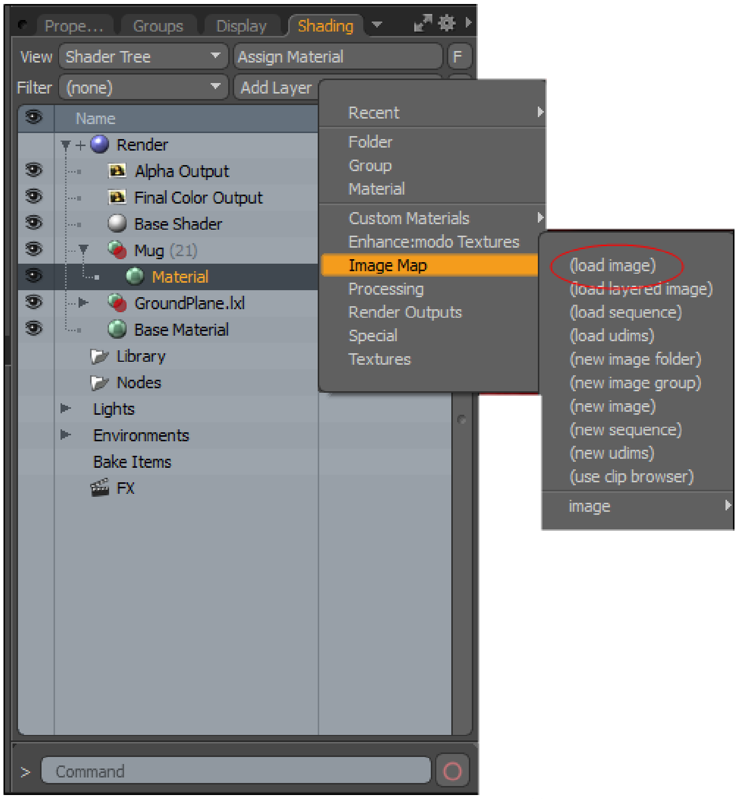
> **Note:** These attributes are render intensive, so if your system specifications are modest, keep them at the default disabled state. For more information, see Material.



Applying an Image on the Mug

Next, we use the UV map we made earlier to apply an image to the mug. Grab this image of a Modo logo and save it, so you can apply it to the model.

1. Press the down arrow in the **Add Layer** option at the top of the **Shader Tree**, select the **Image Map > Load image** option from the pop-up menu.


Prope…
Groups
Display
Shading
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
Name
Render
Alpha Output
Final Color Output
Base Shader
Mug (21)
Material
GroundPlane.lxl
Base Material
Library
Nodes
Lights
Environments
Bake Items
FX
Recent
Folder
Group
Material
Custom Materials
Enhance:modo Textures
Image Map
Processing
Render Outputs
Special
Textures
(load image)
(load layered image)
(load sequence)
(load udims)
(new image folder)
(new image group)
(new image)
(new sequence)
(new udims)
(use clip browser)
image
Command

2. Navigate to where you saved the image and select **Open**.

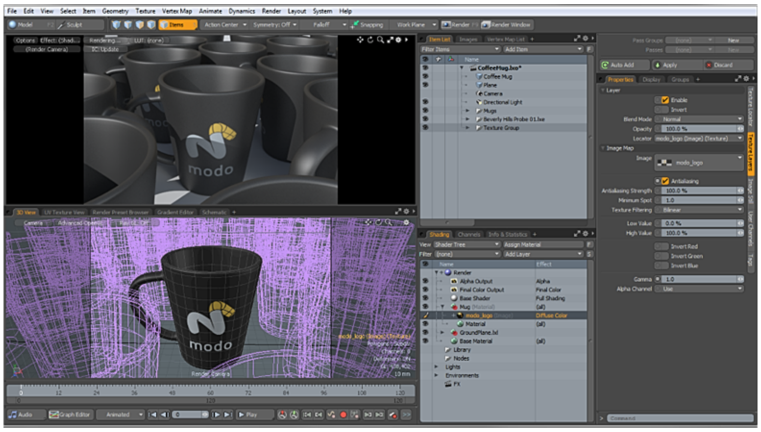
   The image appears right away on the surface.

If the image does not show up correctly, it may not assigned to the UV map properly.

3. Right-click on the small **+** icon on the left of the layer name and select **Properties** to open the **Texture Locator**.

   This controls how the image maps onto the mug surface.
4. Make sure the **Projection Type** is set to **UV Map**, and if there is more than one map listed under the **UV Map** option, select the alternate map.

The image we just applied is a transparent **.png**. Because of the layered approach of the **Shader Tree**, the transparent edges show the regular material color underneath our image instead of making the surface transparent. This makes it easy to add labels, markings, and other non-rectangular images to a surface.



Next, we are going to position the logo to center on the side view of the mug.

1. Press the **spacebar** to drop any tools and then **Esc** to drop any selections.

2. In the **Shader Tree** view, click the down arrow beside the mug and select the **modo_logo**. A cylinder is displayed and selected in your scene.

Tip: To turn on the Texture Locator, type the letter **O** to display your viewport properties, open the **Visibility** tab, and enable **Show Texture Locators**.

3. In the **Properties** panel, open the **Texture Locator** tab.

4. Under **Transform**, change the **Rotation Y** axis to -69 degrees.

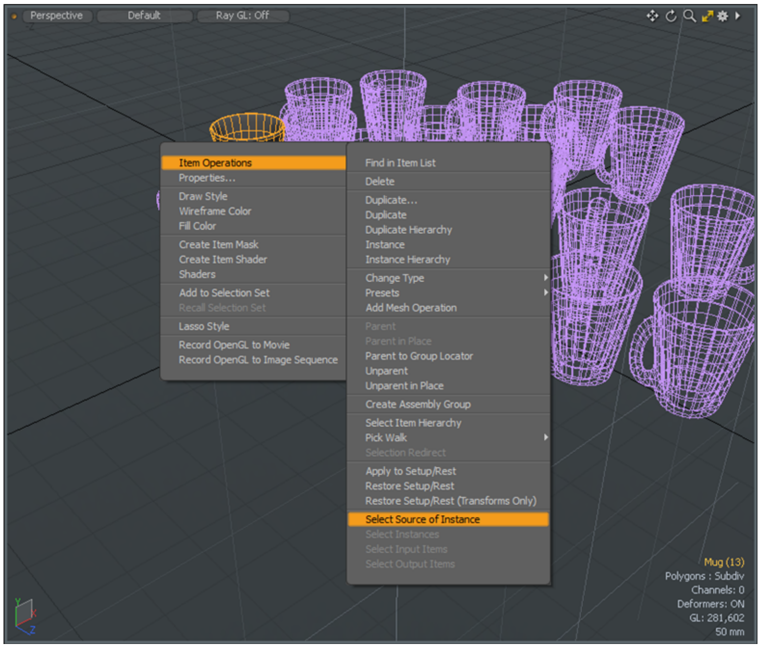
Note: For more information about shading model types and other shading options, see Material.

Now, you can see that the inside of the mug has a logo as well. Since the mug is one continuous surface, the logo projects all the way into it. The easiest way to remove the unwanted logo is to apply a different surface to the interior of the mug. Because we've used Instances, you only need to modify the source mug, and the changes you make propagate to all the clones.

1. Press the **spacebar** to drop any tools and then **Esc** to drop any selections.

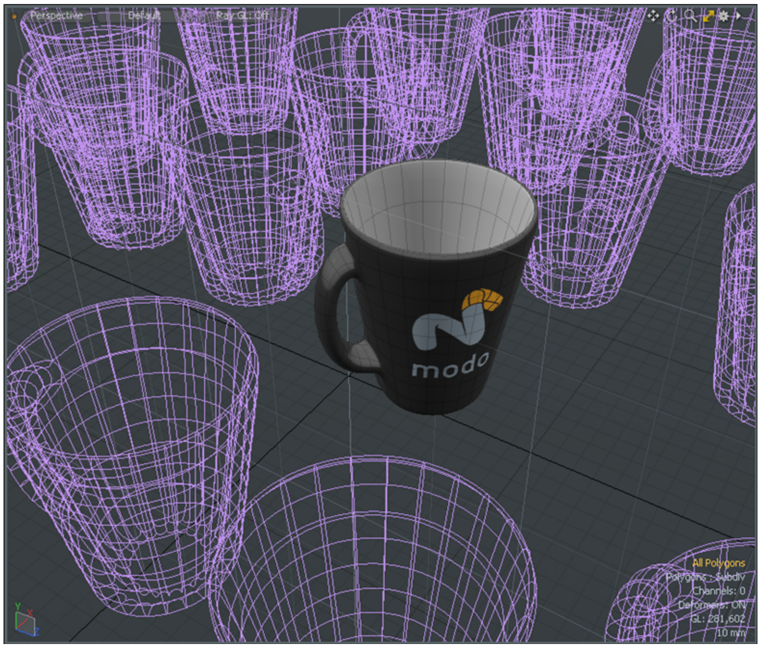
Tip: To turn off the Texture Locator displayed in your viewport, press **O**, open the **Visibility** tab and remove the check mark for **Show Texture Locator**.

2. Make sure to switch to **Polygons** selection mode, since we're now editing the model itself.
3. In the **Perspective** viewport, right-click on any of the pink mugs and from the pop-up context menu that opens, select the **Item Operations** > **Select Source of Instance** option and press **Shift**+**A** to zoom to that model.


Perspective
Default
Ray GL: Off
Item Operations
Properties...
Draw Style
Wireframe Color
Fill Color
Create Item Mask
Create Item Shader
Shaders
Add to Selection Set
Recall Selection Set
Lasso Style
Record OpenGL to Movie
Record OpenGL to Image Sequence
Find in Item List
Delete
Duplicate...
Duplicate
Duplicate Hierarchy
Instance
Instance Hierarchy
Change Type
Presets
Add Mesh Operation
Parent
Parent in Place
Parent to Group Locator
Unparent
Unparent in Place
Create Assembly Group
Select Item Hierarchy
Pick Walk
Selection Redirect
Apply to Setup/Rest
Restore Setup/Rest
Restore Setup/Rest (Transforms Only)
Select Source of Instance
Select Instances
Select Input Items
Select Output Items
Mug (13)
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 281,602
50 mm

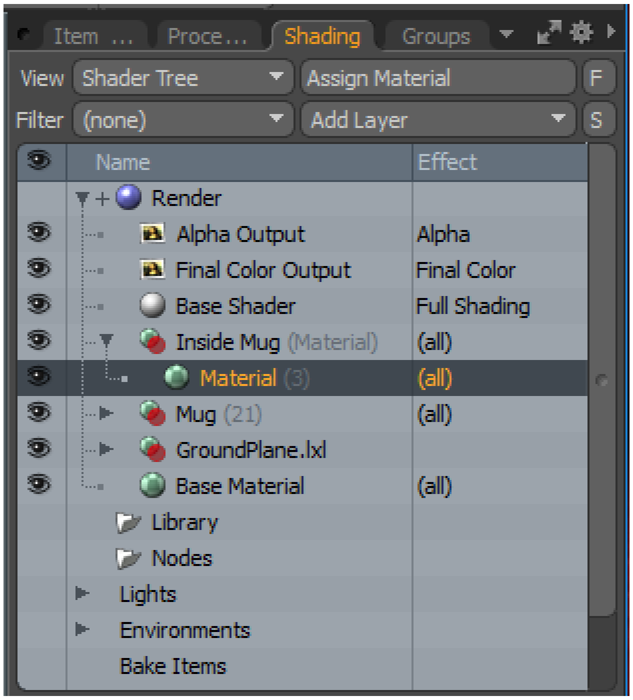
4. Rotate the view to see the inside of the mug and click to select the base of the interior.
5. Press **Shift**+**Up** to grow the selection until it encompasses the entire inside of the mug.
6. Assign a new material tag, overwriting the previous, by pressing **M** to open the **Assign Material** dialog and name the surface 'Inside Mug'.
7. Press **OK** to accept the rest of the default settings.

   You can see in the **Preview** that the insides of all the mugs are now back to the default values, but the logo is gone.


Perspective
Default
Ray GL: Off
modo
All Polygons
Polygons : Subdiv
Channels: 0
Deformers: ON
GL: 281,602
10 mm

The inside material has to match that of the exterior. It is useful to instance the material, so that changes to it also change the instance later on if you need to edit them:

1. In the **Shader Tree**, select the mug exterior **Material**, right-click on it and select the **Create Instance** option from the dropdown menu.


Item ...
Proce...
Shading
Groups
View
Shader Tree
Assign Material
F
Filter
(none)
Add Layer
S

Name	Effect
Render	
Alpha Output	Alpha
Final Color Output	Final Color
Base Shader	Full Shading
Inside Mug (Material)	(all)
Material (3)	(all)
Mug (21)	(all)
GroundPlane.lxl	
Base Material	(all)
Library	
Nodes	
Lights	
Environments	
Bake Items	

There are now two **Material** items in the group, one of them has its name in italic.

2. Click on the italicized item and drag and drop it above the material inside the Inside Mug group.

3. Click to select the material that was already in there and press **Delete** to remove it from the tree as it is no longer necessary.

Now the insides have no logo, the inside surface matches the exterior and since the interior mask contains an instance, any changes made to the original propagate to the instances, so subsequent edits effectively change all surfaces.

Review the Rendering

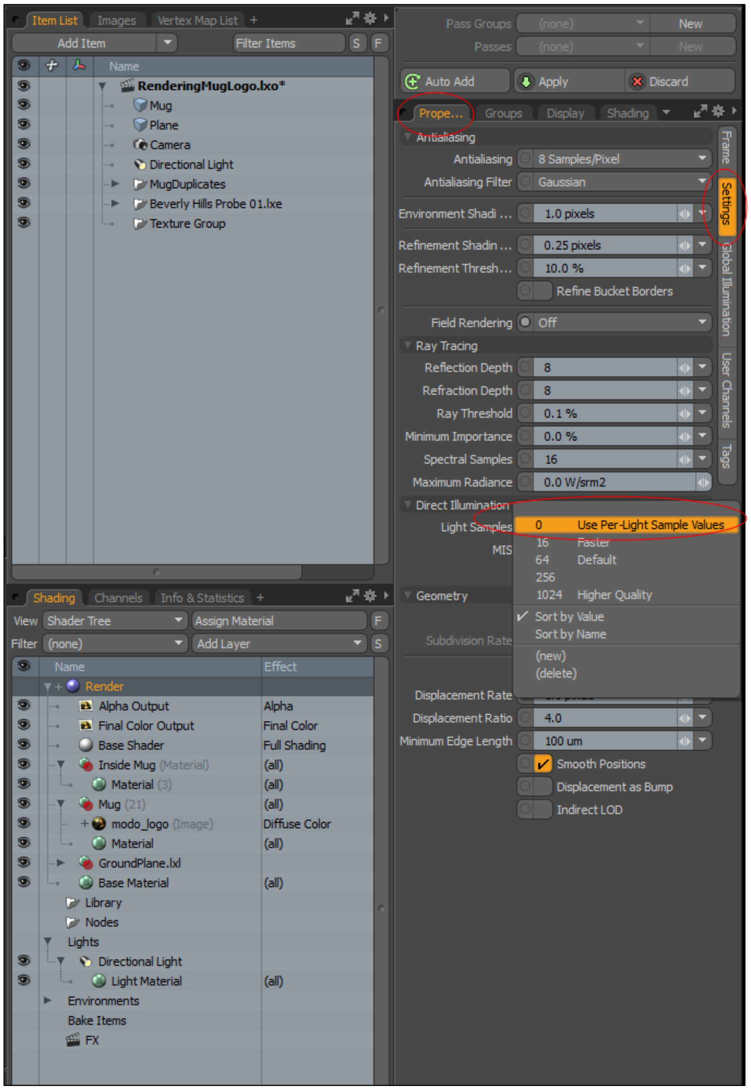
You can see your progress in the menu bar: select **Render** > **Render Current View** to render the scene. Feel free to make adjustments to the **Shader Tree** to get a feel for what does what. Once you have the scene the way you like it, make sure to save it.

You can add more effects to make your scene even more realistic.

Note: Some of these settings may take a long time to render, so if your machine specifications are modest or you don't like waiting for renders to complete, skip these steps.

The first settings you can adjust are those of the light. The shadow the mugs cast are a little more render process-heavy.

1. To soften the shadows, select the **Render** item in the **Shader Tree**.
2. Open the **Properties** > **Settings** tab, under **Directional Light**, set **Light Sampling** to **0 Use Per-Light Sample Values**.


Item List
Images
Vertex Map List
Add Item
Filter Items
Name
RenderingMugLogo.lxo*
Mug
Plane
Camera
Directional Light
MugDuplicates
Beverly Hills Probe 01.lxe
Texture Group
Shading
Channels
Info & Statistics
View
Shader Tree
Assign Material
Filter
(none)
Add Layer
Effect
Render
Alpha Output
Alpha
Final Color Output
Final Color
Base Shader
Full Shading
Inside Mug (Material)
(all)
Material (3)
(all)
Mug (21)
(all)
modo_logo (Image)
Diffuse Color
Material
(all)
GroundPlane.lxl
Base Material
(all)
Library
Nodes
Lights
Directional Light
Light Material
(all)
Environments
Bake Items
FX
Auto Add
Apply
Discard
Prope…
Groups
Display
Shading
Frame
Settings
Global Illumination
User Channels
Tags
Antialiasing
Ray Tracing
Direct Illumination
Geometry
16 Faster
64 Default
256
1024 Higher Quality
Sort by Value
Sort by Name
(new)
(delete)

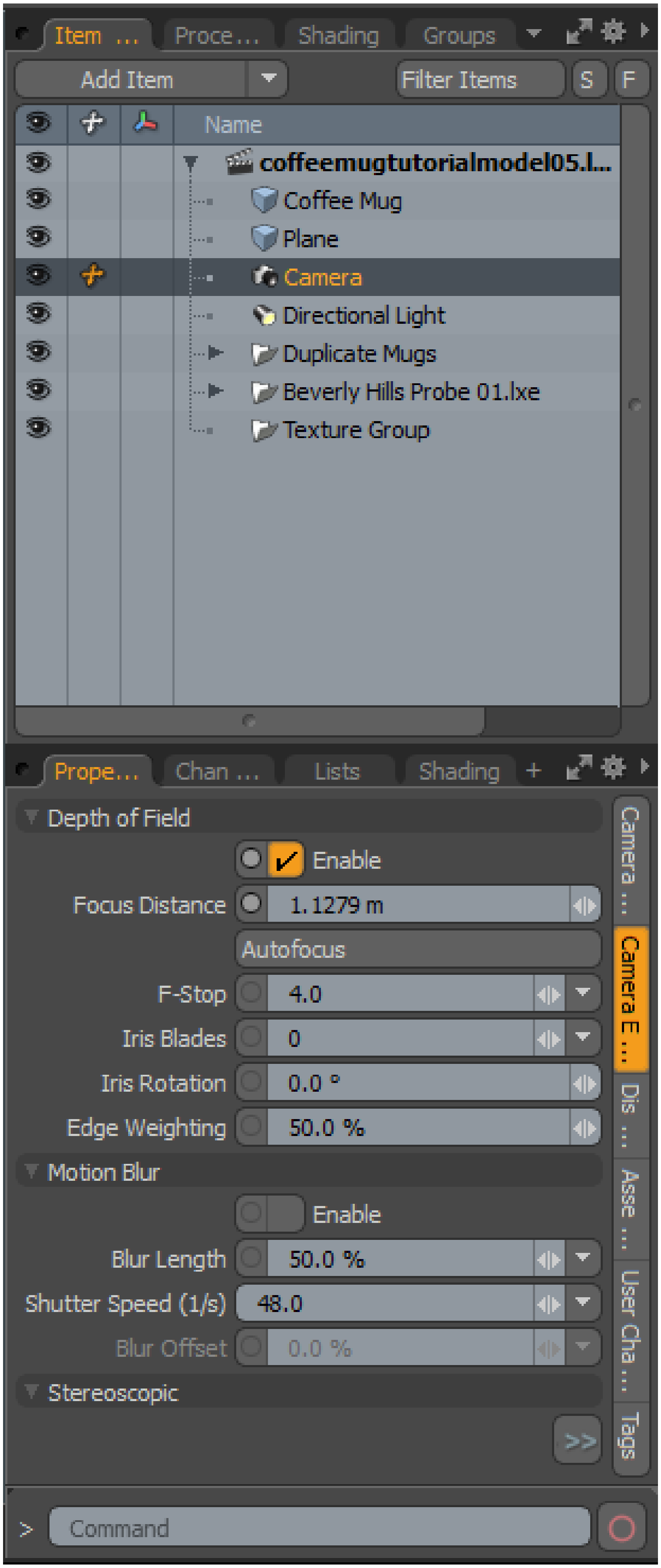
Pass Groups (none) New
Passes (none) New
Antialiasing: 8 Samples/Pixel
Antialiasing Filter: Gaussian
Environment Shadi…: 1.0 pixels
Refinement Shadin…: 0.25 pixels
Refinement Thresh…: 10.0 %
☐ Refine Bucket Borders
Field Rendering: Off
Reflection Depth: 8
Refraction Depth: 8
Ray Threshold: 0.1 %
Minimum Importance: 0.0 %
Spectral Samples: 16
Maximum Radiance: 0.0 W/srm2
Light Samples: 0 Use Per-Light Sample Values
MIS
Subdivision Rate
Displacement Rate
Displacement Ratio: 4.0
Minimum Edge Length: 100 um
☑ Smooth Positions
☐ Displacement as Bump
☐ Indirect LOD

3. Select the **Directional Light** item in the **Shader Tree**.
4. In the **Properties** panel, change the **Spread Angle** value to 10° and the **Samples** setting to '128' to make the light look a lot more realistic.

For the Camera, select the **Camera** item in the **Item List**. The shallow depth of field look is very popular, so let's enable that feature as well.

1. Go to the **Camera Effect** sub-tab of the **Properties** panel and under **Depth of Field**, check the **Enable** checkbox.

   You can also select the **Autofocus** button, this causes Modo to fire a single ray from the center of the camera and set the **Focal Distance** at the first surface it encounters.


Item ...
Proce...
Shading
Groups
Add Item
Filter Items
S
F
Name
Prope...
Chan ...
Lists
Shading
+
Camera ...
Camera E ...
Dis ...
Asse ...
User Cha ...
Tags
>>
>
Command

coffeemugtutorialmodel05.l...
- Coffee Mug
- Plane
- Camera
- Directional Light
- Duplicate Mugs
- Beverly Hills Probe 01.lxe
- Texture Group

Depth of Field

☑ Enable

Focus Distance 1.1279 m

Autofocus

F-Stop 4.0

Iris Blades 0

Iris Rotation 0.0 °

Edge Weighting 50.0 %

Motion Blur

☐ Enable

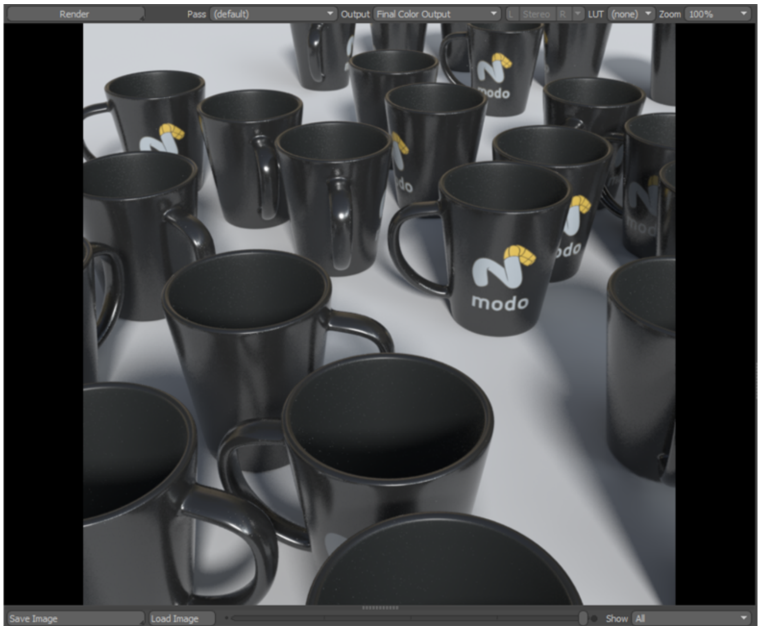
Blur Length 50.0 %

Shutter Speed (1/s) 48.0

Blur Offset 0.0 %

Stereoscopic

2. Click **Render** > **Render Current view** to see the results.


Render
Pass (default)
Output Final Color Output
Stereo
LUT (none)
Zoom 100%
modo
Save Image
Load Image
Show All

This tutorial has given you an overview of working from start to finish in Modo and the steps required to complete a project, including setting up the scene, the difference between the component and **Items** selection modes and how to enable and use Modo's **Global Illumination** rendering engine. If you'd like to explore further, you can assign some preset surfaces to the **Ground Plane**, select a material in the **Preset Browser**, drag and drop it somewhere in the 3D viewport, and drop it onto the **Ground Plane**.

If the image looks too grainy for you, you can increase the anti-aliasing level in the **Render Item** under the **Settings** sub-tab. For the **Soft Reflections** and **Soft Shadow** options of the **Material** and **Light** items respectively, increasing the **Samples values** can also help. Keep in mind that as the number of samples increase, image quality increases, but so does the render time.

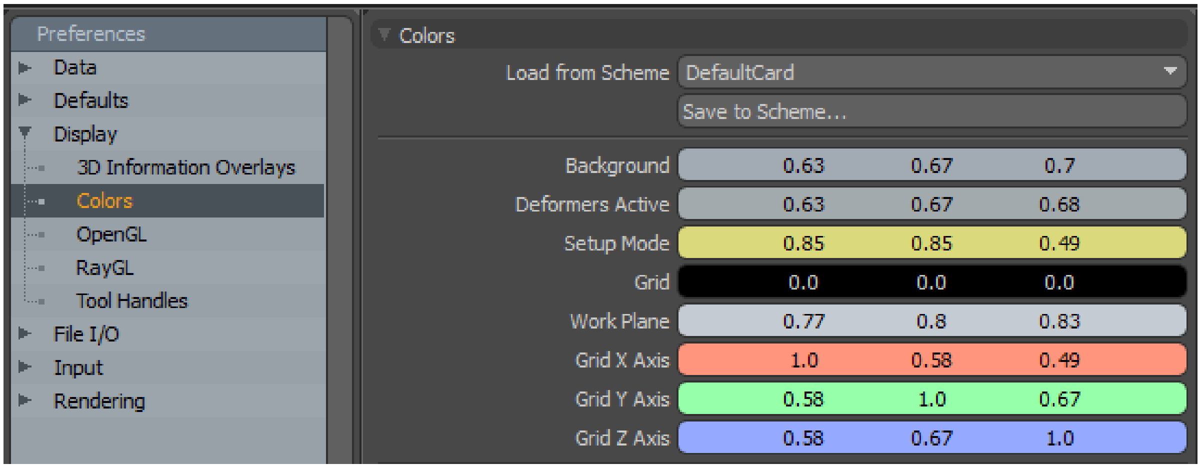
Resources

Use the following supplemental reference information to assist you in using Modo.

Preferences

Modo's **Preferences** menu contains the settings that define the default behavior for a variety of options and functions in Modo. You can customize the application to suit your particular needs and workflow.

The **Preferences** are organized into multiple categories, each dealing with specific areas of customization for the application.


Preferences
Data
Defaults
Display
3D Information Overlays
Colors
OpenGL
RayGL
Tool Handles
File I/O
Input
Rendering
Colors
Save to Scheme...

Load from Scheme: DefaultCard

Background	0.63	0.67	0.7
Deformers Active	0.63	0.67	0.68
Setup Mode	0.85	0.85	0.49
Grid	0.0	0.0	0.0
Work Plane	0.77	0.8	0.83
Grid X Axis	1.0	0.58	0.49
Grid Y Axis	0.58	1.0	0.67
Grid Z Axis	0.58	0.67	1.0

Data Preferences

The **Data** preferences determine how Modo handles file data.

To access these preferences:

- On Windows, open **System** > **Preferences**.

- On Mac OS, open **Menu** > **Preferences**.

Item Tag Definitions

The **Item Tag Definitions** provides names, descriptions and tooltips for items, which show up in the **Tags** sub-tab in the **Item Properties** viewport. The tags are four-character identifiers associating arbitrary strings. Each item can have any number of tags, although only one of each type of tag. The editor also allows the number of lines to be set (for a multi line tag like **Comment**) and allows tags to be hidden (for internal tags used by scripts/plug-ins). The **Item Tag Editor**, which is actually an embedded viewport, is intended for use mostly by scripting and plug-in developers. They can be helpful to you in that they allow arbitrary information to be associated with specific item layers, such as adding instructions within a scene that applies to a certain item.

Path Aliases

Path Aliases provide a means to easily move Modo scenes between disparate computer hardware. You can define an **Alias** that represents an absolute path on the local system. Any files within a scene that contain the defined path, use the alias instead. When the scene file is moved to the next system, if the same aliases are defined with appropriate paths, when Modo comes across any alias in the scene the provided path is used.

For example, Machine 1 path is **C:\Animations\LostSkeleton\Scenes\**. This path is defined as an alias called **LostSkeleton**. When files are loaded into the scene from this directory, their path is **LostSkeleton** internally.

Machine 2 path is **/Users/Lighting/Scenes/LostSkeleton**. This path is also defined with the alias **LostSkeleton**. You can copy the files from Machine 1 into this directory. When Modo opens the sceneand it encounters images saved in the folder **LostSkeleton**, the path alias defined on machine 2 is used to search for the images for loading. Only the alias itself needs to be identical, the paths do not. The content needs to exist within each location, as well.

There are also options available by right-clicking on the **Alias** and **Path**. You can rename, change the path, duplicate or delete an alias.

Preset Markup Definitions

The **Preset Markup Definitions** are definitions assigned to assets within the Preset Browser for the purpose of adding metadata to the individual files. Adjustments to these settings appear when the **Edit Markup** pop-up panel is opened.

Referencing

These options define the default referencing behavior of imported references. These behaviors can be modified on a per-scene basis in the **Reference Override Settings** dialog box. For additional information, see Import Reference. The operations include:

Allow Channel Overrides	Changes made to any of the associated attributes of the item itself, such as changes to the transform values or item specific changes such as changing a mesh item's subdivision level.
Allow Name Overrides	Changes to the referenced items' names (overrides are tracked by an internal ID assigned when the items are imported).
Allow Structural Overrides	Changes to the hierarchies of the referenced elements, meaning their ordering in parent child relationships can be changed, or their positions within the **Items** lists.
Allow Deletions	Control whether or not referenced items can be removed from the master scene.
Allow 3D Selections	Controls the selectability of the referenced elements in the 3D viewports.

Each of these operations have four possible states governing it:

- **Always Allowed** - Override changes can always be made of the designated type.
- **Never Allowed** - Overrides cannot be assigned, editing of the referenced items is locked.
- **If Allowed by Item** - Overrides cannot be applied, unless explicitly set as **Enabled** by the reference item setting.
- **Unless Disallowed by Item** - Overrides can always be applied, unless explicitly set as **Disabled** by the reference item setting.

The latter two settings can be defined directly on the items, either within their source scenes, or as referenced items within the master scene. To do this, select the target element to control, and then in the menu bar, select **Items** > **References** > **Item Override Settings**, which opens a dialog box.

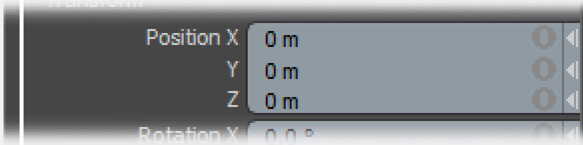
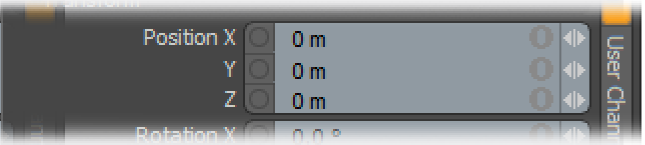
Defaults Preferences

You can edit basic behaviors of Modo.

To access these preferences:

- On Windows, open **System** > **Preferences**.
- On Mac OS, open **Menu** > **Preferences**.

Animation

Animation	
Channel Controls	This option toggles Modo's ability to set keyframes on specific attributes within the viewport panels. When disabled, it removes the keying fields, the default enabled state allows you to set a keyframe directly next to the attribute, using the small circle control preceding each input field.  Channel Controls disabled  Channel Controls enabled
Default Settings	
Time System	This setting determines how Modo displays time in both the **Timeline** and input fields related to timing. This option also determines the default time system for any new scene. You can modify these settings independently for each scene in the Scene Item. You can also input alternate time styles into dialog for Modo to convert to the chosen format, such as entering **3.25s** is displayed as **00:03:06** with SMPTE format time system. The following options are available. • **Frames** - Indicates frames as sequential numbers, using standard numbering format. • **Seconds** - Indicates frames as a passage of time, using the **0.0s** format. • **SMPTE** - Specifies frames utilizing a shorthand method, **00:00:00:00** representing hours:minutes:seconds:frames • **Film Code** - Specifies frames by the length of the film in feet (linear measurement) using the **0.0 + 0** format.
Frames per Second	Also referred to as **FPS**, this setting defines the number of still images that play back in sequence in one second of real time. You can specify a frame rate using the dropdown menu, applying standard definitions, such as **Film (24 fps)**, **PAL (25 FPS)**, and **NTSC (29.97 fps)**. This option determines the default FPS value for new scenes. You can modify these settings independently for each scene in the Scene item. Choose **Custom...** to

Animation	
	specify alternate custom values using the **Custom Rate** option.
Custom Rate	If **Custom** is selected as the default **Frames Per Second**, you can define the custom value in this field.
Play Real Time	When this option is enabled, Modo attempts to play back the sequence at the true speed, skipping frames if necessary. When disabled, Modo plays back every frame in sequence as fast as the system can handle it, which may be faster or slower than the specified **Frames Per Second** value.
Cache Deformations	Enable/Disable deformer caching. When enabled, Modo caches the resulting meshes from deformers and mesh operations into memory for each frame. As a result, playback and stepping through frames becomes very fast, even with complex deformers.
End Behavior	Determines what happens when playback reaches the end of the play range. • **Hold**: Stops animation at the end of the playback range. • **Repeat**: The default setting, loops playback indefinitely. • **Wrap**: Repeats the playback until the current time matches the initial time, and then it stops.
Preserve Key Timing	When enabled, Modo retains the timing of keyframes when the **Frames Per Second** value is modified, adjusting their position in the Timeline as necessary. When disabled, frames retain their Timeline position, which changes their position in time. For example, an animation is timed with a keyframe at frame 120 of the Timeline. At 30fps, the key appears at the 4 second mark. If the **Frames Per Second** value is changed to 24fps after the keyframe was set with **Preserve Key Timings** enabled, the keyframe moves to frame 96, retaining its position in time at the 4 second mark. If disabled, at 24fps the keyframe remains at frame 120, which is at the 5 second mark, slowing down the animation speed by 20%.
Scene Range	This value specifies the default time duration for new scenes. You can change the duration for each scene in the Scene item.
Auto Key	When items are moved in a scene, the **Auto Key** function can automatically create keyframes eliminating the need to manually create them. The different settings specify how keyframes are created. **All** generates keyframe data for all channels associated with the modified item, **Animated** only keyframes values that have been edited or modified, and **None** disabled the auto keyframe functionality. This option determines the default **Auto Key** behavior for new scenes. You can modify these settings independently for each

Animation	
	scene in the Scene item.
Auto Update Existing Keys	This option is only available when **Auto Key** is off. In that case, any edits that are made get discarded when the current time is changed, unless they have been manually keyed. This option allows existing keyframes to be updated automatically, if their values are changed without requiring manual keying. Un-keyed changes still get discarded when the current time is changed.
Time Snap	When enabled, this toggle snaps the current time indicator and any keyframes to whole frame values. To define sub-frame keyframes, disable this option.
Slope Type	This option determines the default slope type applied to new keyframes. See Graph Editorfor slope definitions.
Initial Weight	This option determines the default weight type applied to new keyframes. See Graph Editor for weight definitions.
Channel Behavior	This option specifies how a channel acts outside of the keyframed range. See Graph Editor for information on pre and post channel behaviors.
Channel Interpolation	This option determines the default channel interpolation method. SeeGraph Editor for information on curve interpolation behavior.
Link Keys	When link keys are enabled in the **Graph Editor** (its background changes to orange), associated keyframes remain linked together in time, meaning that, for properties like Position, Rotation and Scale that each have associated X, Y, and Z values, editing a single axis value, by moving it forward or backwards in time, keeps all the keyframe positions linked together as if all three were selected. When inactive, only the selected channels are modified. This option determines the default link keys behavior.
Rotation Order	Changing the order that rotations are applied can sometimes help to reduce or even eliminate gimbal lock (gimbal lock is when two axes rotate into a parallel angle and lose a degree of rotation). This option determines the default rotation order for all new items. Rotation order can also be adjusted on a per item basis in its associated **Property** panel when selected.

Application

Application	
Initial Selection Type	Determines the default selection mode when Modo is first opened. For more information on selection modes, see Selecting Items.
Max Undo Levels	The undo history is viewable within the Command History viewport. This option controls the maximum number of steps backwards you can go in the history. The default value of '100' is generally very good at allowing leeway to step backwards through the history. Increasing the value allows you to go further back in time, but also increases memory usage. For memory intensive scenes, it may be beneficial to reduce this value to reduce total memory overhead.
Group Selection Cmds	When enabled, Modo groups multiple consecutive selection actions into a single undoable action. The grouping size is determined by the **Max Sub-Commands Recorded** Value.
Max Sub-Commands Recorded	Determines the maximum number of consecutive actions grouped into a single undoable action when the **Group Selection Cmds** option is enabled.
Default Material	Determines the default material tag applied to newly created polygons. This is also the default name that appears in the **Polygon Set Material** dialog box when defining new material tags.
Default Part	This option determines the default name used for new part tags. Name appears in the **Polygon Set Part** dialog box.
Default Texture Map	This option determines the default name used for new and auto-generated UV maps. The name appears in the Vertex Map List viewport.
Default Vertex Normal Map	The default value is set to Vertex Normal.
Default Multiresolution Layer	This option determines the default name used for new Multiresolution Sculpting layers. The name appears in the Sculpt viewport of the Sculpting and Painting workspace layout.
Item Index Style	Scenes cannot have multiple items with identical names (much like program files in the same folder), when this occurs, Modo automatically renames the

Application	
	item, be it a Shader Tree layer, item layer or Vertex Map, appending a numeral to the end of the defined names in a style determined by this option.
Use OS Color picker	When enabled, Modo uses the operating system-specific color picker by default, duplicating the behavior of pre-501 versions. When disabled, Modo uses its own internal Color Picker Viewport.
Max Recents in Categorized Lists	Add menus, such as **Add Layer** in the **Shader Tree** have a **Recent** menu at the top of the list to hold the most recent items added during the present Modo session. You can adjust this setting to determine the maximum number of recent entries in the menu.
Max Recents in Searches	Several of the list viewports (like the **Item List** and the **Shader Tree**) offer a search function through the use of the small **F** button. Previous search terms can be cached in the drop-down menu when searching, allowing speedier subsequent searches for the same term. The **Max Recents in Searches** determines the number of search terms that are cached for this purpose.
Reset "In the Future...": Messages	Many pop-up dialogs in Modo provide a function to dismiss them in the future. For example, when welding pairs of points consecutively, you can dismiss future pop-ups of the same dialog using the **In the Future** setting in the pop-up itself. You can use this button to reset all these types of pop-ups to their factory installed state (meaning, if they have been dismissed permanently, this brings them back).
Automatically Check for Updates	This option can be used to determine how often Modo checks the update server to see if a newer version is available. You can disable this action with the **Never** option, or checking **Every Launch**, **Daily** or **Weekly**. When a newer build number is encountered, Modo alerts you to the new version. **Note:** Modo doesn't update itself, it is up to you to download the new version from your Foundry user account and install separately. For installation instructions, see Installation and Licensing.
Automatically Submit Usage Statistics	Usage statistics of Modo sessions can be submitted to Foundry. This helps us to determine application stability and in the case of instability, helps us to detect the source of the problem. You can help out in the efforts to improve Modo by opting to submit this information to us **Every Launch**, **Daily** or

Application	
	Weekly. Selecting the **Never** option disables the function completely. The information collected and submitted is the Modo build version, the current OS, the last 10 commands run prior to any crash/hang, the duration of the last session and if it ended successfully or terminated in a crash/hang event. **No personal information is collected**. We hope you'll help us in our efforts to improve Modo, but we also understand should you choose to opt-out.
Default Scene	This option allows you to simply store your favorite settings, default lighting setups and geometry into a scene that is loaded by default when a new scene is created or Modo is opened. Any file type that Modo supports can be loaded as a default scene, and the specified scene can contain any number of elements, including any of the items types Modo supports, such as lights, cameras, meshes and even deformers and falloffs. The most common usage is to store camera, frame and GI settings that are used as the new default values, and maybe even a lighting setup and a ground plane if desired. To use, simply create a scene, with all the items and settings desired and save it somewhere on the local hard disk drive. Then close the file and use the **Default Scene** preference option to locate the saved file and then exit Modo. The next time Modo is opened, the saved scene is present, rather than the regular Modo default. You can open and edit the specified **Default Scene** at any time should it be necessary to make any adjustments. Simply save the scene file to initialize the new settings. **Note:** When a default scene is specified, the settings of the designated scene override any scene specific **Preference** settings, such as those for **Framerate**, **Scene Length**, **Color Management**, **Default Gamma values**, and so on.
Preset Browser	
Content Location	This option allows you to change where preset files are saved (assets, kits, and samples). Changing the **Content Location** takes effect the next time the application is launched. The following system paths are the default locations: • Windows: **C:\Users\NAME\Documents\Luxology\Content** • MAC OS X: **/Library/Application Support/Luxology** • Linux: **/usr/Shared/Luxology**

Application	
	Note: Advanced users: The **NEXUS_CONTENT** environment variable is processed before the **Content Location** preference and always overrides it.
Dir Cache Updates	When using the Preset browser, this option determines how the thumbnail cache is built for viewing the various assets. The **Only When in Use** option will build the cache only once the Preset browser is opened or made visible. The **Always Keep Updated** option will refresh the cache each time Modo is opened. During the initial cache process, if significant changes have been made (such as installing a new version) this may take some time, and use processing resources. During this process things might seem slightly sluggish, depending on system specifications.
Scan For Changes Using	Determines how the dir cache decides whether the content of a directory on disk needs to be scanned for changes to the files within. The following options are available: • **Directory Modification Time Only** - The default option. Directory modification times are updated when a file is added or removed from a directory, but not when an existing file is changed. It is also not updated for some special directory alias types on some operating systems. If the modification time of the directory is newer than the last scan, the files in the directory are scanned. This method is very fast, but can miss some changes. • **File and Directory Modification Times** - Always scan the files in a directory for changes, even if the directory modification time hasn't changed. This method allows you to catch modifications to files that don't update the directory modification time, however, it is significantly slower than checking only the directory modification time.
Lock Preset Thumbnails	When this option is enabled, saving over any existing presets retains the existing thumbnail.
Foundry Link Settings	
Use these settings to configure the Modo Bridge plug-in. For more information, see Modo Bridge for Unreal Engine.	

<table>
<tr><th colspan="2">Application</th></tr>
<tr><td colspan="2">Tip: You can switch between multiple Modo scenes when the server connection is alive between Modo and the Unreal Engine client.

You can push Items, Textures, and Materials from multiple Modo scenes to the Unreal Editor client.</td></tr>
<tr><td>Address</td><td>Sets the IP address where Modo is installed, allowing you to push and pull a scene (including geometry, lights, cameras, materials, and textures) between Modo and Unreal Engine.

Note: This must be the same IP address that is used by the Unreal Engine Unreal Bridge feature. Once both the Address and Port are configured, updates made between both software applications is shared instantaneously.

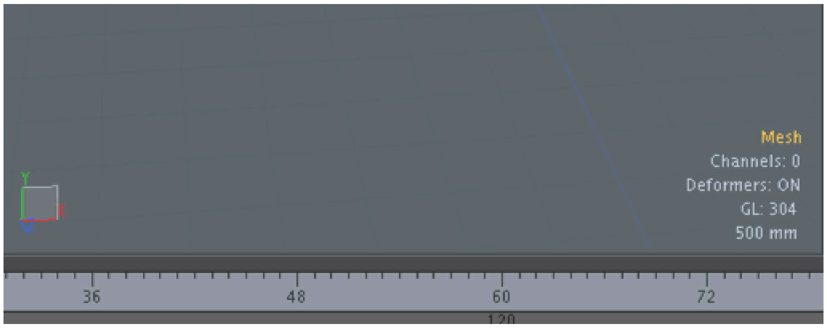
For more information, see Configuring the Server and Client.</td></tr>
<tr><td>Port</td><td>Sets the port number to use for the Unreal Engine Unreal Bridge feature. For more information, see Configuring the Server and Client.</td></tr>
</table>

Auto-Save

You can use the **Auto-Save** option to systematically save an alternate file at a specific time interval, with options to save multiple versions. This can be helpful if you tend to forget to save your files regularly and want some insurance against system crashes. In addition, **Auto-Save** also saves your full progressive render state in the Preview window when it completes a save. You can load the previous render state at a later time.

Note: Files are saved in a temporary directory that you have specified in the Auto-save **Back-up Directory** option. These files are deleted automatically once they are more than one month (31 days) old. The cleanup is done at shutdown.

In the 3D viewport, a dialog displays a 10 second countdown with an option to cancel. Clicking **Cancel** skips the current Auto-Save. Modo launches an Auto-Save at the next regularly scheduled Time Interval.



Note: Modo may slow down slightly as the file is being saved. It can also take some time to save large files which may cause a slowdown in workflow.

Note: The auto-save file is saved in an alternate location and is therefore a separate file from the original saved file. In the event of a unexpected exit, you can recover the last auto-saved version from the defined **Backup Directory** location.

Enable	When this option is enabled, Modo automatically saves a backup scene file and associated texture maps that have changes, such as images for sculpting or painting, at the specified location and interval.
Time Interval	This option determines the length of time in minutes between each auto-save operation.
Backup Directory	You can use this option to set the directory location for auto-save files. This location can be different from where you save files regularly.
Number of Revisions	You can set the number of revisions you would like to auto-save. Having multiple versions can be useful in reviewing how a file progresses or as insurance against file corruption.
Progressive Render Auto-Save	
Enable Image Auto-Save	This option will enable the ability for Modo to automatically save the generated progressing rendered Preview image at the defined interval as an EXR file. The image(s) is saved to the defined **Backup Directory** location (as defined above).
Write Image Sequence	When this option is enabled subsequent saves will have a number appended to the file name so the overall progress of the rendering is saved.

Save Layered Image	When enabled, Modo saves the combined render outputs into a single file.
Time Interval (minutes)	This option determines the interval of time between auto-saves.

Experimental

Multi-Threaded Drawing	This option allows **View Object** processing to **utilize multiple background threads**. This is **disabled by default**. When disabled, a single background thread is used to off-load geometry processing for viewport drawing, increasingModo performance on scene loading, timeline scrubbing of animations, Mesh Op evaluation and more. Performance improvements will be most noticeable when loading scenes and interacting with them, especially in cases where multiple individual meshes are present.

Images & Painting

Enable Image Antialiasing by Default	Modo provides an image antialiasing function to smooth the rendered result of bitmap images. In certain instances, lower resolution images may appear blurry when rendered. By default, Modo always enables this option found in the Image Map properties. You can set the default antialiasing to disabled with this option. You can override the default value on a per image basis in the **Image Map** properties.
Default New Image Resolution	When creating a new blank image for painting or baking, this value defines the default resolution. You can also change this at the time the image is created.
Always Display Images as Icons	When this option is enabled, Image maps are represented in palettes and menus as generic file icons instead of thumbnail images. In scenes with many textures, this preference saves the processing time required to build thumbnails when opening menus and palettes with images.
UDIM is	This option allows you to control how UDIM values are read from filenames

	when working with UDIM images. There is an option for preceding and following numerical UDIM values in a filename. For more information on working with UDIM images, see UDIM Workflow.
Threaded Painting	This option, enabled by default, threads the painting and sculpting tool into multiple processes, providing better performance. It is possible, however, that in some dual or single core systems this setting should be disabled.
Auto Create Textures	When enabled and an image map does not exist, one is created automatically when you paint on a mesh You can disable this feature to avoid automatic image map creation.
Default Foreground Color	Allows you to change the default paint color from white to black for the foreground while working in the **Paint** layout. The default foreground color is **White**.

Mesh Data Evaluation

Defer Smoothing During Interaction	When enabled, smoothing of vertex normals between polygons is automatically disabled during interaction with a mesh in the viewport and re-enabled on pen-up.
Compute Smoothing During Animation Playback	When enabled, smoothing of vertex normals between polygons is automatically disabled during animation playback in the viewport and re-enabled when playback stops.

Mesh Items

Auto Create Item Masks	When this option is enabled, an item mask is automatically generated for each new item layer that is created in the **Items** list. For more information on item masks, see Material Group.
Auto Create Preset Item Masks	When this option is enabled, an item mask is automatically generated for each new mesh preset item that is added to any scene. This can be especially helpful to limit Shader Tree layers to the specific preset, should there be identically

	named material tags between existing layers and the preset.
Auto Create Item Transforms	When new items are created in the **Items** list, they do not have specific transform channels in the channels viewport by default (**Position**, **Scale**, and **Rotate**). Enabling this option instructs Modo to create the 3 basic transform channels for all new items.
Auto Hide Child Items	You can create hierarchies of elements in a scene, and relate elements under **Group Locators** for organization. The **Auto Hide Child** option, when enabled, allows you to toggle visibility of all related child layers when any parent layer is toggled. When disabled, layer visibility in a hierarchy is toggled individually per layer.
Create Instances as Bounding Boxes	By default, Modo displays instanced items as full wireframe representations of the source item. When this option is enabled, Modo only displays bounding box representations of instanced items. This can improve display performance for heavily instanced scenes.
Save Vertex Maps Selection to Scene	When this option is enabled, any selected Vertex Map remains selected when the file is closed and later re-opened. When disabled, all Vertex Map selections are reset with no maps being selected when the file is re-opened.
New Patch Display Level	Determines the default Patch level for Subdivision Surfaces items. Higher values produce smoother results but generate more geometry. You can manually adjust the Patch display level for each item in the item's properties. For more information on this, see Mesh Item.
New Curve Display Level	Determines the default amount for how smoothly curves are displayed in the 3D Viewports. You can choose between **Coarse**, **Medium** and **Fine**.
New Curve Display Angle	Defines the default value that controls how curves are quantized for display and freezing. This is analogous to the patch resolution for subdivision surfaces.
New Spline Patch Display Level	Determines the default **Spline Patch Level** for patched surfaces. You can manually adjust this value in the item's properties.
New Catmull-Clark Display Level	Determines the default **Catmull-Clark Subdivision Surfaces** subdivision level. Higher values produce smoother results but generate more geometry. You can manually adjust the subdivision level for Catmull-Clark surfaces in the item's properties.
Default Primitive	Certain primitive mesh objects (such as the Sphere and Torus) have an option

Polygon	that sets the type of polygons the tool generates. The **Default Primitive Polygons** option defines the default state for these values.
Symmetric Primitive Creation	When this option is enabled in conjunction with the symmetry option, for any primitives that are created, such as a cube or a sphere, an identical, mirrored version is also created across the line of symmetry.
3D Editing	
Flatness Limit	Determines the threshold at which a polygon is considered to be planar or non-planar. Higher values designate fewer polygons as being non-planar. You can use the Statistics panel to select these types of polygons automatically.
Copy and Paste	
Deselect Elements after Copying/Select Pasted Elements/Deselect Elements before Pasting	By default, when you paste a copied element into Modo, the original selection remains active and the newly pasted geometry is inserted unselected. With these options, you can now have Modo automatically select/deselect elements, also providing a workflow familiar to many graphics applications.
Slip UV	
Startup State	Here you can automatically disable the **Slip UV** option each time a tool is dropped. **Off** means **Slip UV** is always off when a tool is activated; and is the default setting. **Last Selected** is the default behavior for the tool, where the **Slip UV** state is retained.

Procedural Modeling

Draw Tool Handles	You can enable/disable displaying tool handles in the 3D viewport.
Ghosted Mesh Drawing	
Drawing Style	Define how the ghosted mesh is displayed in the 3D viewport. You can choose **Shade** or **Wireframe**, or select **Hide**, not display it.
Color	Adjust the color of the ghosted mesh.
Opacity	Set the transparency of the ghosted mesh.

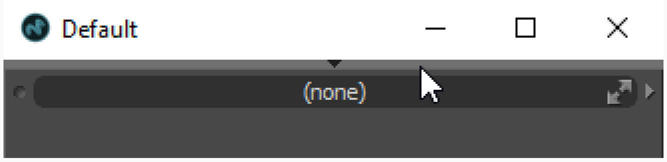
MeshFusion	
MeshFusion Wire Opacity	Defines the opacity of the wireframe Modo uses to indicate MeshFusion meshes.

Scene Save/Export

Author/Copyright	These values, when defined, are embedded within saved and exported files (where supported).
Absolute Path	When this option is enabled, Modo writes all link references (such as those to images, MDDs and IES files) as absolute paths, referencing their location to specific areas of a disc drive when saving files. When disabled, Modo uses relative paths to locate linked files.
Merge Reference Items	When utilizing referenced items, you can enable this option to embed the referenced items into the scene file when saving, instead of having them remain external entities.
Text Encoding	Determines the text encoding of the file when saved, using either UTF-8 or the OS systems default.

Switcher Bar

Autohide	When enabled, automatically hides the switcher bar when you move the mouse away from it.
New Window Defaults	
Window Edge	Specifies the edge the switcher bar is aligned to on new windows. You can set the switcher bar to appear along the top, bottom, left, or right edge.
Initially Collapsed	When enabled, the switcher bar is collapsed by default on new windows. To reveal the switcher bar, click the reveal strip at the top of the window, just below the header.

	
Show When Collapsed	When enabled, the reveal strip is displayed when the switcher bar is collapsed. When disabled, the strip is completely hidden.
Show Only Favorites	When enabled, only layouts marked as favorites are displayed on the switcher bar.
New Palette Defaults	
Palette Edge	Specifies the edge the switcher bar is aligned to on new palettes. You can set the switcher bar to appear along the top, bottom, left, or right edge.
Initially Collapsed	When enabled, the switcher bar is collapsed by default on new palettes. To reveal the switcher bar, click the reveal strip at the top of the palette, just below the header.
Show When Collapsed	When enabled, the reveal strip is displayed when the switcher bar is collapsed. When disabled, the strip is completely hidden.
Show Only Favorites	When enabled, only layouts marked as favorites are displayed on the switcher bar.

Tool Settings

The **Tool Settings** options allow you to define default states for different functions for both the component selection modes (vertices, edges and polygons) and for items selection mode.

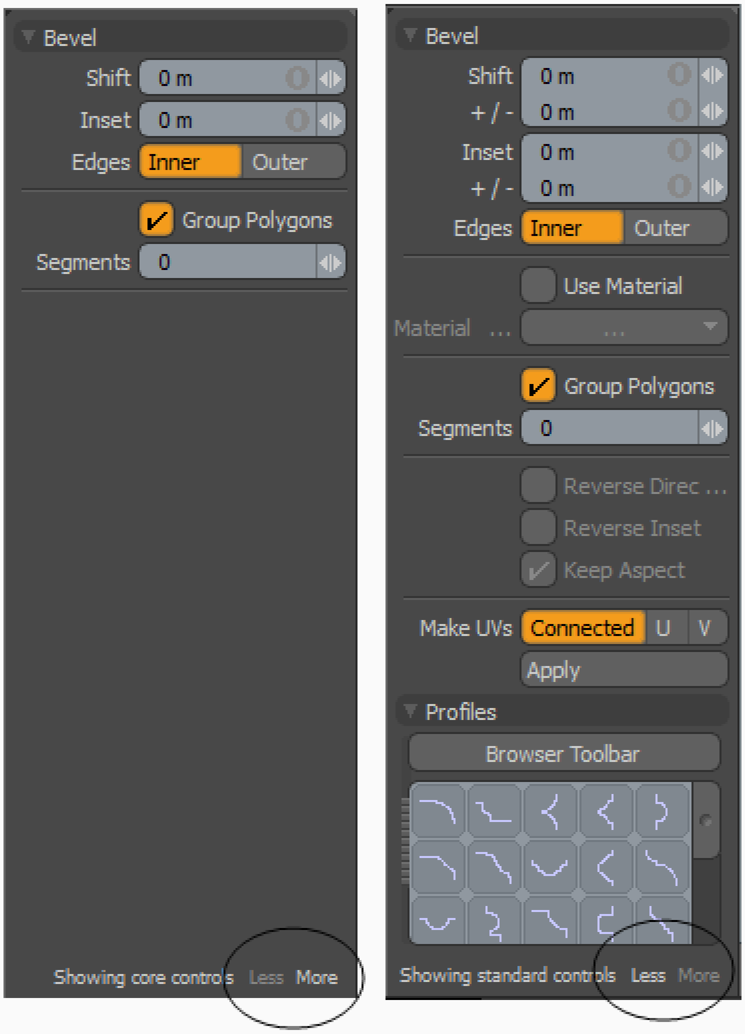
Default Tools - Component Mode/Item Mode	
Default Action Center	This option determines the default Specifying Action Centers and Falloffs when Modo is first opened.
Default Action Axis	This option determines the default **Action Axis** when Modo is first opened.
Default Brush	This option determines the default **Brush Tip** that is auto-selected when utilizing any sculpting or painting brush. **Last Selected** obviously returns the brush to its prior state.

Default Tools - Component Mode/Item Mode	
Default Constraint	You can use this option to determine the **Constraint** type when Modo is first opened.
Global Flags	
Auto Activate	When enabled, the **Auto Activate tool** control property of the Tool Pipe viewport becomes a global setting, rather than a per-tool option.
Curve Tool Auto Activation	When enabled, selecting a curve automatically activates the appropriate curve tool.
Select Through	When enabled, the **Select Through** tool control property of the **Tool Pipe** viewport becomes a global setting, rather than a per-tool option.

UI

Tidy Layouts	Modo's interface is made up of a collection of viewports. Viewports always have a frame you can use to resize the window. When multiple viewports are nested within each other, multiple viewport frame borders become unnecessary. The **Tidy Layouts** option, enabled by default, removes redundant viewport frames providing additional screen space and a cleaner UI. However, you may wish to disable the setting to better see how viewports relate to one another for the purposes of interface customization.
Popup Transparency	This percent value controls the transparency of pop-up windows, pop-overs and pie menus. Higher values make the pop-ups more transparent, allowing the background to been seen through them, making them less obtrusive, but also more difficult to read the contents. **Note:** This may affect performance on some systems. We do not recommend using this option for Windows users.
Popover Dismissal Behavior	This options can be used to override the dismissing behavior for popovers. The **Default** option retains the hard coded behavior. **Force Click-Off** makes sure that popovers remain open until explicitly clicked away from and **Force Roll-Off** dismisses the window when the cursor is simply rolled away from it. This

	only affect simple popovers, point at popovers are always click-off.
Base Form Proficiency Level	This option is used to change the level of detail displayed in Toolbox Properties of the controls. For example, the **Polygon Bevel** tool properties. This preference is used to set the state of the view when the **Form View Options** level is set to Default. **Less** and **More** buttons are displayed at the bottom of the Tool Properties pane. To save on space when the display window is not fully expanded, **-** and **+** buttons are displayed, preceded by the name of the level setting. The default level is set to **Standard**. The following level settings are available: • **Basic** - A short list of the common core properties. • **Standard** - A list of all of the core properties. • **Advanced** - A list of all of the core properties plus custom functions for special behaviors.

	 Bevel Shift Inset Edges Inner Outer Group Polygons Segments Showing core controls Less More Bevel Shift + / - Inset + / - Edges Inner Outer Use Material Material Group Polygons Segments Reverse Direc ... Reverse Inset Keep Aspect Make UVs Connected U V Apply Profiles Browser Toolbar Showing standard controls Less More
Default Form Icon Size	This option determines the size of the icons used in the interface. By default, this option is set to **Normal**. To save on real estate, select **Small Icons**. To enlarge the icons, select **Large Icons**.
UI Images	
Sort By	This option determines how images are sorted in UI viewports, such as the Clip Browser function. Images can be sorted **Alphabetically** or by **Age** with the most recently imported images appearing at the top of the list.
Maximum Images (per group)	This determines the maximum number of clips (stills or animations) that can be loaded into a Clip Browser.
Image Cache Size	This setting determines the maximum amount of RAM dedicated to holding all

	user interface thumbnails, such as those generated for imported images, image inks and brush tip images.
Thumbnail Size	This setting determines the pixel resolution for auto-generated user interface thumbnails.
Clear Images From	You can open this menu to choose from multiple user-determined options to remove images from the user interface. Once a particular command is invoked, images in that function's Clip Browser are removed from display, clearing memory.
Clear All UI Images	This option clears all latent images from the various Clip Browsers for the tools, for example, the Image Ink Browser, Brush Tip Browser, and so on.
Preset Browser	
Show Nice Names	Shows underscores as spaces for file names in the Preset Browser.
Show Extensions	Shows the file extension for file sin the Preset Browser.
Show Info Badge Only	When enabled, this option only shows the **i** badge icon over thumbnails in the Preset Browser. This setting only affects thumbnail views, as badges are always shown in **List** mode.
Tool Properties	
Modo Layout	The way tool properties are displayed in the **Modo** layout. The following options are available: **Popover** - You can access the tool properties by clicking a button at the bottom of each toolbar. It opens in a popover which can be repositioned and pinned. **Inline** - Tool properties are displayed below each toolbar. This option is intended for larger screen sizes with sufficient room below the toolbar controls. **Split** - An embedded **Tool Properties** form appears on the left panel while a tool is active. It is hidden when a tool is dropped or the tool's properties are opened in a popover, by pressing **k** on the keyboard).

Display Preferences

You can fine-tune different aspects of Modo's displays.

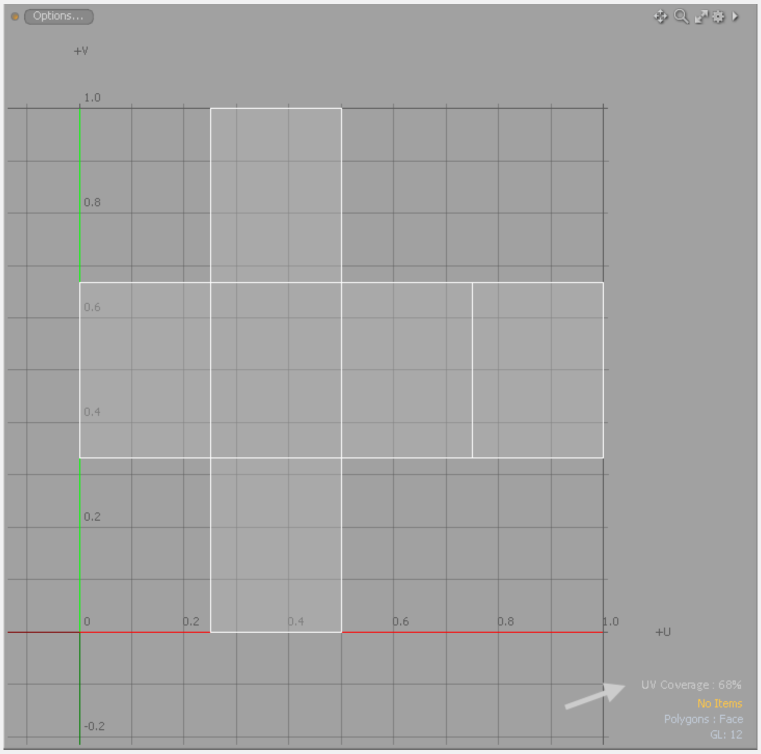
To access these preferences:

- On Windows, open **System** > **Preferences**.
- On Mac OS, open **Menu** > **Preferences**.

3D Information Overlays

Each 3D viewport displays a variety of information as a HUD (heads-up display) directly in the viewport. You can control what information is viewed by these toggles.

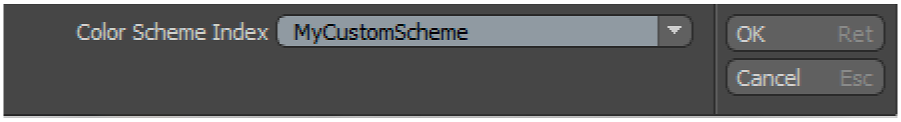
Morph Map	Displays the name of the currently selected morph map, if available.
Current Selection	Displays selected element information, based on the current mode (component, item or otherwise).
Reference Item	Displays the name of the source file for selected items that are part of a referenced scene.
Selected Channel Count	Displays the number of currently selected (active) channels in the Channels Viewport, useful for keyframing.
Deformers Enable State	Displays the current state of deformers. Animated deformers can be enabled in the 3D (OpenGL) Viewport options.
OpenGL Polygon Count	Displays the number of GL polygons, including those added by Subdivision Surfaces.
Grid Size	Displays the real-world measurement equivalent of the current grid divisions.
Active Passes and Actions	Displays any active Passes and/or Actions by name that are selected.
Polygon Type	Indicator displays polygon type for active mesh layers: **Face**, **Subdiv**, **Catmull-Clark** or **mixed**.
Current Tool	Displays current tool information as well as info for falloffs, Action Centers and

	Snap.
Snapping State	Displays the current enabled options for Applying Snapping.
UV Coverage	Displays a percentage value in the lower-right corner of UV viewport, representing the amount of uniform UV space of the total region that is covered by geometry. When you use an UV editing tool, the coverage is updated editing and after releasing the mouse button.  Options... +V 1.0 0.8 0.6 0.4 0.2 -0.2 0 0.2 0.4 0.6 0.8 1.0 +U UV Coverage : 68% No Items Polygons : Face GL: 12 **Tip:** This option is useful to game workflows when trying to maximize the amount of texture space being used for the target surface.

Colors

You can customize the colors of many aspects of Modo related to the 3D Viewports (application-specific viewports don't have directly editable color schemes).

Note: Modifying colors in **Preferences** does not affect the actual default viewport colors. In order to apply a color scheme to a viewport, you first need to adjust the colors in this section, then use the **Save to Scheme** function and assign a new name using this dialog box.


Color Scheme Index: MyCustomScheme

Once a scheme name has been set, you need to switch back to the particular 3D viewport you wish to apply the scheme to. Presets can be applied to each viewport separately, determined by its **Active** status. Active viewports are separated by the **Thumb** changing to an orange color. You can activate any viewport by simply clicking anywhere over the viewport window. Next, click **View** > **Viewport Color Scheme**, then select the previously saved scheme from the list. The following elements of Modo can be customized.

Background	The viewport background color when the **GL Background** option is set to **None** or **Gradient**.
Deformers Active	The viewport background color when the **Enable Deformers** option is enabled.
Grid	The main ground plane grid visible in every 3D perspective viewport.
Work Plane	The adjustable construction plane.
Wireframe	Default **Uniform** wireframe draw color, can be over ridden by individual item draw options. For more information, see
Color Wireframe	Default **Color** wireframe draw color, can be overridden by individual item draw options. For more information, see
Background Item	Inactive mesh color, also considered **Background** or **Unselected**.
Instance	Instanced item duplicates.
Solid	Geometry display color when using the **Solid GL** viewport display style.
Patch Cage	When **SubDivision** display cage option is enabled, this determines its color.
Discontinuous Pair	Color referencing discontinuous edges in UV mode.
Selection	Element selection color for all selection types, including component and item modes.

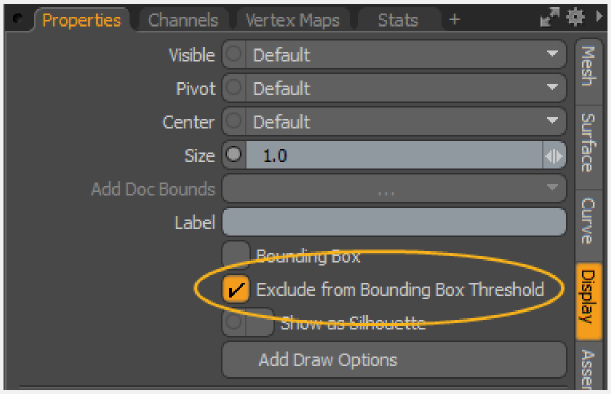
Selection Fill	Fill color of selected elements. It can be set different from selection itself.
Child Highlight	When a parent item is selected, child items can have a custom color referencing their relationship to the selected item.
Assembly Highlight	Highlighted mouse-over color of assemblies.
Lasso	Color of line drawn when drag selecting elements with a lasso.
Handles	
Handle	Unselected action area of tool.
Active	Selected Action area of tool.
Axis X/Y/Z	Colors for constrainable tool handles for each axis.
Guide	When using **Snapping** > **Constrain to** > **Guides**, this option determines guide color.
Cage	Color for Spline/Curve Cage edges.
Label	Item label as determined by **Display** viewport options.
Range Cool	When **View** > **Show Falloffs** is enabled, determines cool range of falloff.
Range Warm	When **View** > **Show Falloffs** is enabled, determines warm range of falloff.
Unsnapped	When snapping elements, a small pre-highlight displays when elements are in close proximity, defined by this color.
Text	
Selection	Color for on-screen informational text display of selected elements.
Info	Color for on-screen informational text display.
Weight Values	Color for text display when the **Display Weight Values** option is enabled and a Weight Map is selected.
Selection Rollovers	
Generic	Generic selection roll-over pre-highlighting color.
Face	Geometry face roll-over pre-highlighting color.
Subdivision	Modo Subdivision surface geometry roll-over pre-highlighting color.

Catmull-Clark	Catmull-Clark geometry roll-over pre-highlighting color.
Topo	
Topo Faces	Polygon faces while using the **Topo** display mode option.
Topo Vert	Vertices while using the **Topo** display mode option.
Topo Edge	Edges while using the **Topo** display mode option.
Topo Back Edge	Rear facing polygon edges while using the **Topo** display mode option.
PSUB Masks	
PSUB Mask	Color of sculpting mask overlay.
OpenGL Highlights	
UV Boundary	Determines the color of UV island borders displayed in the 3D viewport when the 'Show UV Boundaries' viewport option is enabled.
Geometry Boundary	Determines the color drawn over any open geometry edges when the 'Show Geometry Boundaries' viewport option is enabled.
SelOp Selection	Determines the color of selection operation selections.

OpenGL

Texture Resolution	This value (set as powers of 2) determines the largest bitmap texture size displayable in Modo 3D viewports before Modo resorts to mipmapping, which is a method of subsequent resamples of an image at a lower resolution that enable faster image drawing to the 3D viewport. Higher settings increase image quality, but require greater video RAM.
Environment Resolution	This value specifies (in pixels) the maximum image resolution for environment images. Unlike **Texture Resolution**, this option is used only for Environment lighting. The default setting is 4096 x 2048. The following resolutions are supported: • 64 x 32 • 128 x 64 • 256 x 128

	• 512 x 256 • 1024 x 512 • 2048 x 1024 • 4096 x 2048 • 8192 x 4096 • 16k x 8192 **Tip:** Suggested system requirements: For Environment Resolution 8192 x 4096, use 4G VRAM at least for a Graphic Processing Unit (GPU) For Environment Resolution 16384 x 8192, use 16G VRAM at least for a Graphic Processing Unit (GPU)
Flatness of Perspective	This option determines the amount of perspective in the 3D perspective viewport. Similar to the **Field of View** option in the **Camera** options panel. The higher the value, the flatter the perspective; the lower the value the more distorted and extreme the perspective becomes.
Max GL Polygon Count	This option defines an upper limit to the number of polygons visible in a 3D viewport. It triggers an error message allowing you to cancel the previous operation if it were to exceed the defined amount. This option was created mostly to help you avoid locking up a system when trying to subdivide a model too many times when using multiresolution sculpting.
Bounding Box Threshold	This option draws items as bounding boxes when rotating, panning, or zooming in the viewport, if the Open GL polygon count exceeds the **Max GL Polygon Count** value. This eliminates lag and makes it easier to work with large scenes. You can enable this option for **All Surface Items**, or **Only Inactive Surface Items**. You can also exclude individual items by enabling **Exclude from Bounding Box Threshold** in the **Display** tab of the item **Properties**. For more information, see Display Viewport.

	 Visible: Default Pivot: Default Center: Default Size: 1.0 Add Doc Bounds: ... Label: ☐ Bounding Box ☑ Exclude from Bounding Box Threshold ☐ Show as Silhouette Properties Channels Vertex Maps Stats + Add Draw Options Mesh Surface Curve Display Asset
Bounding Box Drawing in Animation	Only available when **Bounding Box Threshold** is not **None**. When enabled, surface items with a polygon count exceeding the **Max GL Polygon Count** value are displayed as bounding boxes during animation playback and when scrubbing the timeline.
Enable Displacement in GL	This option determines if displacement effects are displayed in the GL viewports as actual polygon displacement. When disabled, displacement displays in the 3D Viewports as a bump map.
Grid Visibility	This option determines the default visibility state of the ground plane grid visible in 3D Viewports. It is enabled (visible) by default.
OpenGL Point Size	This option determines the size of the on-screen display of vertices, specified in screen pixels. The default value is **3**. Consequently, it also controls the selection hit area for selecting vertices.
Selected Point Size	To make it easier to see their on-screen display, selected vertices can be displayed larger in the 3D viewport, an amount determined by this option. Scale is based on the original **OpenGL Point Size** option.
Viewport Fit Factor	You can automatically frame the view of all the elements of a scene to the bounds of the active viewport by using the menu bar commands **View** > **Fit All** or **View** > **Fit Selected** (or keyboard equivalents **A** or **Shift**+**A**, respectively). This option determines the scaling of the fit. **1** corresponds to 100%, fitting the bounding box of the elements to that of the viewport's shortest length. The default of **1.5** provides a little room around the elements and ensures that the viewport fits into view regardless of bounding box size.
Topology Vertex Backface Culling	When enabled, this option disables the display of vertices for rear facing polygons while using the **Topo** display mode. This makes distinguishing foreground vertices easier when editing.

Weight Value Precision	If a Weight Map and vertices are selected with the **Show Weight Values** option enabled, the **Weight Value Precision** option determines the number of decimal places shown for the weight values in the GL viewport.
Transparency	
Selection	When viewing a selection through an opaque surface, this option determines the opacity of the selection highlighting through the surface (similarly to an X-ray). This is especially helpful in differentiating front and rear facing selections.
Handles	When viewing tool handles through an opaque surface, this option determines the opacity of the handle itself, making it possible to see the handles through the surface.
Topology Polygons	This option determines the transparency of forward-facing polygons while in the **Topo** display mode.
Selection	
Selection Rollovers	Pre-highlighting of elements in a scene helps you to make selections, showing which elements are getting selected when the mouse button is clicked. The **Selection Rollover** option determines how the highlighting occurs. • **None** - No pre-highlighting of elements in the scene. • **Manual** - Pre-highlighting of elements is based on the current mode (vertices, edges, polygons, items, and so on). • **Closest** - Pre-highlighting the nearest geometric component element regardless of mode.
Mouse Regions Trigger Selections	With this option enabled, you can automatically switch between component modes (vertex, edge, polygon) by right-clicking the target element. **Tip:** Combine this with the selection rollover set to **Closest** to aid determining the proper element to click on.
Performance	
VBO Mode	VBO stands for **Vertex Buffer Object**, a method of working and displaying geometry in OpenGL. VBOs offer substantial performance gains over the older immediate mode rendering method, but may also cause problems with older graphics cards, or graphics cards with lacking drivers). Modo tries to determine

	the best mode when set to **Auto**, depending on the video cards driver. You can also manually enable/disable the option with the **On**/**Off** options to increase stability.
Only Draw Selected Procedural Outlines	When this option is enabled, Modo only draws the surface outline for procedural geometry (such as the Gear Item). When disabled, Modo draws any individual triangles that make up the surface.
Sync Display Refresh	Also known as **Vsync**. When enabled, this option helps with tearing in the GL viewports but may cause lag with larger meshes and lower framerates.
Sync Application Response	When enabled, this option syncs the application with the 3D OpenGL viewports reducing lag, but it can sometime also significantly reduce the 3D Viewports framerate, dependent on the scene's contents.
AVP Allow Material Texture storage	Controls how the 3D Viewport stores its data internally in the textures or uniform buffers. When disabled, Modo reduces the maximum size of the scene displayed and the framerate is higher to improve performance. **Note:** In order for your changes to take effect, you must restart your computer.
Performs Color Correction on OpenGL Objects	This option toggles the color management of the 3D Viewports. When enabled, the 3D Viewports are adjusted using the current Color Management settings. When disabled, the viewports are not corrected. **Note:** In complex scenes, corrected viewports may reduce display performance.
Backdrop Item	
Include Backdrops in Viewport Fitting	When using the **Viewport Fit Factor** function of a viewport, enabling the **Include Backdrops** option accounts for the size of any visible backdrop item when calculating bounding box size for the fit action.
Display Backdrops in Uniform Size	When this option is enabled, backdrop items ignore image size, and base the scale of the backdrop on the item's **Scale** value.
Pixel Size	When importing bitmap images for use as backdrops, this option determines

	the measurement of a single pixel in an image upon import, allowing you to place backdrop images scaled to real world size. For example, if **Pixel Size** is set at 10 mm, a 512x512 image is imported at 5.12 meters in size.
Item Drawing	
Default Size for New Scenes	Determines the default drawing size in the 3D Viewports for non-Mesh Items, such as cameras and lights. You can also adjust the scaling values manually in the **Display** viewport.
Default Size for Current Scene	Determines the default drawing size in the 3D Viewports for non-Mesh Items added after a scene is created. You can also adjust the scaling values manually in the **Display** viewport.
Schematic	
Schematic Grid Snapping	Specifies the grid size for node snapping in the Schematic. The default value is 2.0. A value of 0.5 produces the pre-Modo 11 behavior.

RayGL

Resolution	When **RayGL** in-viewport render preview is utilized, this option determines the resolution of the rendered image, initially based on the on-screen size of the 3D viewport. Lower resolutions update more quickly. There are three options: • **Full** - Renders at the same pixel resolution of the viewport. • **Half** - Renders at half the resolution of the viewport. • **Quarter** - Renders at one quarter of the resolution of the viewport.
Updates	This setting determines the way the **RayGL** preview updates (or renders) while working in a 3D viewport. Three options are available. • **Delayed** - This option updates the viewport preview when the mouse button or tool handle is released. • **Lazy** - This option updates the viewport each time a pause is encountered, regardless of mouse or tool state. • **Synchronous** - This option continuously updates the viewport, as best as the system is able to process the data, reducing resolution as necessary to keep up.

Quality	When **RayGL** in-viewport render preview is utilized, this option determines the quality of the rendered display. • **Draft** - Renders the image using a reduced number of ray samples, as specified in the **Render** preferences, in the **Preview** section under **Preview Draft Quality Settings**. This option produces a faster, though less accurate result. • **Final Render** - Renders the image using the same number of rays you specify for the final render. • **Extended Refinement Passes** - Renders the image with additional ray samples beyond any value specified for **IC**, **MC**, **Reflections**, **Refractions**, **Transparency** and **Subsurface Scattering** up to the number of rays specified within the **Render** preferences section, under **Preview**.
RayGL Toggles	For memory or calculation intensive scenes, you can disable certain effects from being considered in **RayGL** processing. The various options can be toggled independently: shadows, reflections, refractions, global illumination, displacements, fur and low fur density. You can also reduce the polygon count of fur texture layers without disabling fur entirely.
Draft Quality Settings	
When using a preview viewport with the Quality settings in the **RayGL** section set to **Draft**, the following options determine the quality settings solely for the preview viewport itself. You can throttle material and render setting values for more responsive preview performance.	
Displacement Rate Multiplier	Sets the maximum displacement rate for Micro Polygon displacement.
Quality	Sets the preview quality as a percentage. At 50%, all **Samples**, **Number of Rays** and other related quality setting are cut in half reducing rendering quality, but increasing performance.
Antialiasing Samples	Sets the maximum amount of antialiasing samples calculated for the preview.
Max GI Bounces	A general quality settings multipliers that throttles the total number of rays related to IC, transparency, reflections and refractions. The default setting of 50% reduces the total number of rays specified by half.
Max MC Rays	This option determines the maximum number of Monte Carlo global illumination rays used to calculate indirect lighting.
Max IC Rays	This option determines the maximum number of Irradiance Cache rays used to calculate indirect lighting.

Tool Handles

Large/Small Points	Determines the visible size, in pixels, of manipulator handles displayed on tools, such as those on the **Curve** drawing tools.
Large/Small Points Hit Size	Determines the hit area where a mouse click affects the particular tool handle. Typically, this value is larger than the handle size, to make it easier to select a particular handle, but making it too large can make it difficult to select a specific handle in a tight cluster.
Handle Line Width	Determines the width, in pixels, of the lines drawn for tool handles, such as the widgets for **Move**, **Rotate** and **Scale**.
Handle Line Hit Width	Determines the hit area where a mouse click affects the particular tool handle. Typically this value is larger than the handle size to make it easier to select a particular handle.
Handle Scale	
Standard	If your screen is very large or very small, you may prefer to set the default size of the tool handles larger or smaller using this setting. Handles stay consistent in size regardless of viewport scaling. You can also interactively Scale tool handles by pressing the **-** and **=** keys.
Small (% of Standard)	Some tools have a smaller secondary handle. You can adjust its size here based on a percentage of the **Standard** amount.
Plane Handle Ratio	Many tools have additional plane handles, that constrain actions to two axes simultaneously. You can adjust the plane tool handle sizes based on the **Standard** amount.
Draw Style	
Selected Tool Handles	Determines the default drawing behavior for tool handles. The following options are available: • **Invisible** - No handles display in the viewport. • **Basic** - Basic tool handles display allowing you to constrain certain functions when used, based on the particular tool. • **Advanced** - Additional controls or handles appear, where available, to aid in setting certain options for a tool or item. For example, when selecting the

	Camera item with any tool that has advanced handles, additional controls become visible for setting **Focal Distance** and **F-Stop**.
Unselected Tool Handles	Depending on the tool, additional handles display in the viewport, especially those related to setting falloff positions and sizes. The **Unselected Tool Handles** option determines the drawing options for these types of additional handles.
Auto-Haul Display	Some tools can be enhanced with additional on-screen controls. For example, advanced in-viewport sliders can specify **Bevel Shift** and **Inset**. This option determines whether or not to draw such additional helpers. • **Off** - Disables display of additional on-screen tool auto-haul displays. • **Basic** - Displays only basic tool handles. • **Advanced** - Displays only advanced tool handles when available. • **On** - Displays on-screen auto-haul handles defaulting the settings of the preset itself as determined by its options in the **Tool Pipe**.
Handle Snapping	
These are the default values, in pixels, for the distance from a snapping target where the cursor first gets snapped.	
Inner Range (pixels)	Determines the actual snapping range.
Outer Range (pixels)	Determines the range to highlighting elements for snapping to.
Rotation Snap Angle	Sets the angle of the fixed intervals for any rotation tools. Activated only when pressing **Ctrl** before rotating the target element.

File I/O Preferences

You can customize the behavior of Modo related to various file formats.

To access these preferences:

- On Windows and Linux, open **System** > **Preferences**.
- On Mac OS, open **Menu** > **Preferences**.

Alembic I/O

Alembic Export	
Animation Time Range	Exports specified animated frames or ranges into the Alembic file. This is a useful option when exporting scenes containing many keyframes, deep item hierarchies (for example ACS scenes), and replicators. The following options are available: • **Current Frame** - Selects the current frame to export. • **Current Range** - Selects the current range to export. • **Scene Range** - Selects the scene range to export. • **Custom Range** - Selects the custom range specified in the **Custom Start**, **Custom End**, and **Custom Step** options to export.
Custom Start/End	Allows you to set a custom start and end time to export.
Custom Step	Allows you to set a custom value for each animated step to export.
Export Type	Specifies whether to export all items or selected items (includes their parent hierarchy item) to the Alembic file. **Export All** - Exports the entire animated frame range to the Alembic file. When disabled, only the current frame is exported. **Export Selection With Hierarchy** - Exports selected items and their hierarchy parent to the Alembic file.
Save Meshes	Controls which items are exported. When enabled, meshes are saved to the Alembic file.
Save Procedural Items	Controls which items are exported. When enabled, procedural items are saved to the Alembic file.

Save Locators	Controls which items are exported. When enabled, Locators and groups of Locators are saved to the Alembic file.
Save Cameras	Controls which items are exported. When enabled, Cameras are saved to the Alembic file.
Save Particle Systems	Controls which items are exported. When enabled, particle systems are saved to the Alembic file.
Save Replicators	Controls which items are exported. When enabled, replicators are saved to the Alembic file.
Export Replicas As	Specifies the type of data exported to use in DCC applications. The following options are available: • **Geometry** - Exports only the fully baked geometry. Each replicator is baked as a separate Alembic mesh, subdivision, and other objects. Use this option for rendering purposes. This is the default setting. • **Particles** - Exports only the particle systems. Use this option to conserve space when exporting replicas to other DCC applications, for example Katana and Houdini. Particles and geometry can then be connected to native replicators and easily modified. • **Particles and Geometry** - Exports both the particles and geometry. For more information, see Exporting Alembic Files.
Save UV Sets	Controls which parts of geometry are exported. When enabled, mesh UV sets are saved to the Alembic file.
Save Normals	Controls which parts of geometry are exported. When enabled, vertex normals are saved to the Alembic file.
Save Material Tags	Controls which parts of geometry are exported. When enabled, material tags are saved to the Alembic file. Material tags are exported as a standard face set with a **material_*** prefix.
Save Part Tags	Controls which parts of geometry are exported. When

	enabled, part tags are saved to the Alembic file. Part tags are exported as standard Alembic faces.
Include Unique Suffix In Item Names	When enabled, Modo's fully unique item names are used, including the suffix (usually numbers in parentheses, which are grayed out parts of the names in the Modo's **Item List**). Disabling this option, omits the suffix whenever possible, allowing you to keep the same names of the Alembic file you imported into your scene. **Note:** Alembic has a built-in limitation of not accepting two children items of the same parent with exactly the same item names.
Force Letters and Digits Only Item Names	Removes all spaces, parentheses, and other non-standard characters that cause problems with certain importers, making it compatible with other applications (for example, Katana).
Scale	Modo's default unit is 1 m, whereas some applications, such as Maya, Houdini, or Nuke use 1 cm as default. This can causes problems with size when exporting/importing files from Modo to other applications. If you frequently move scenes between Modo and other applications, you can use this setting to scale your scene. Set this to 100.0 to load scenes from Modo correctly in other apps. To load files into Modo with the correct size, set this to 0.01.

Collada I/O

COLLADA Export

For user convenience, the **Absolute Path**, **Merge Reference Items**, **Unit System**, and **Up Axis** settings are included in both their original locations on the **Accuracy and Units**, and **Scene Export** panels, as well as in the **COLLADA Export** panel. Changing any of these settings in the **COLLADA Export** panel also changes the value globally, so the new setting value also updates in the original preference group panels.

COLLADA Export	
Absolute Path	When this option is enabled, Modo writes all link references (such as those to images, MDDs and IES files) as absolute paths, referencing their location to specific areas of a disc drive when saving files. When disabled, Modo uses relative paths to locate linked files.
Merge Reference Items	When utilizing scene reference items, enable this option to embed the referenced items into the scene file when saving, instead of having them remain external entities.
Unit System	You can choose your preferred measurement units system based on several options. Once set, this is the measurement input method Modo uses for all numerical input values. This value is also exported in the COLLADA scene's Asset tag. The following options are available: • **SI** - The International System of Units (abbreviated 'SI') is the modern form of the metric system, complies to universal base units. • **Metric** - A universal system of measurement based on powers of 10 - millimeter, centimeter, meter, kilometer. • **English** - A historical measurement system based on Imperial Units -mils, inch, foot, yard and mile • **Game Units** - An arbitrary unit of measurement (defined by the **Meters per Game Unit** setting). • **Unitless** - An arbitrary decimalized unit of measurement based on cubic meters. 1 unit = 1m.
Default Unit	This option is dependent on the **Unit System** and determines the default unit used when no unit is specified.
Meters Per Game Unit	When utilizing the arbitrary **Game Units** system, use this value to determine the scale of a single unit. This allows you to work with even whole numbers. So if a 1 m equivalent is really 1.375 real world meters in Modo, you can enter 1.375 here and then specify units normally in the numerical input fields.
Up Axis	This option determines the major axis that is considered the default up direction for Modo. 3D programs typically use Y as up, whereas CAD applications typically use Z as the up direction. When importing geometry, if it is always 90° off, changing the up axis to match the originating application may resolve the issue. This can also be set separately for each scene in the Scene Item.

COLLADA Export	
Save Hidden Items	Saves items that are hidden in the **Items** list (that the eye icon next to the item's name is toggled off).
Save Cameras /Lights/Locators	This controls whether or not cameras, lights or locators are saved with the scene.
Save Triangles as Triangles	Saves Mesh Items that are constructed of triangles using the COLLADA **<triangles>** element, instead of the more general-purpose **<polylist>** element.
Order Vertex Maps Alphabetically	Enabled by default, this option organizes Vertex Maps into alphabetical order.
Bake Matrices	The COLLADA format can represent Modo's stack (or layers) of multiple transform elements, so that more than one rotation, position, or scale can be applied to a single item in series. For compatibility with other applications, **Bake Matrices** reduces such a transform stack down to a single transform by pre-multiplying the results before export.
Save Vertex Normals	Enables the saving of normals for each geometry vertex.
Save UV Texture Coordinates	Enables the saving of one or more UV texture coordinate maps for each Mesh Item.
Save Vertex Colors/Weights	Enables the saving of the **Color** and **Weight** Vertex Maps.
Save Animation	This option enables or disables the output of animation channels. When **Save Animation** is disabled, all item values are sampled at the first frame of any animated channels.
Sample Animation/Sample Start Frame and Sample End Frame	Creates a keyframe value at every frame within the range specified by the **Start** and **End Frame** settings. This option is useful when exporting to applications that can't interpolate all of the animation channels in a given file.
Z-near and Z-far	The Z-buffers on 3D graphics cards have limited resolution, and so these settings describe the closest and furthest points that may be displayed for a given scene. Z-near and Z-far are not applicable for the types of rendering performed by Modo, but they are useful for interactive 3D applications where

COLLADA Export	
	this information can be used to optimize the resolution of the hardware Z buffer.
Save modo Profile	This option provides a human-readable form of all of Modo's items and their internal channels and animation envelopes, and also allows the Modo COLLADA importer to re-import assets with the same Modo-specific items and their channels and envelopes. A full description of the Modo profile is beyond the scope of this document, so please refer to the *Collada I/O documentation* included with the File I/O SDK for additional details (available in Modo's help directory as part of the SDK).
Save Maya/3ds Max/Okino/XSI Profile	Determines whether application-specific tags are exported or not.
Formatted Arrays	This option breaks up long lists of values into related groups on separate lines, which makes the output text easier to read.
COLLADA Import	
Import Units	This option, when enabled, reads and applies the unit's information from a COLLADA file to the imported Modo scene.
Import Up Axis	This option, when enabled, reads and applies the Up axis information from a COLLADA file to the imported Modo scene.

FBX I/O

In Modo, mesh instances depend on a source mesh item to define the geometry. This is different to FBX, where a mesh cannot be an item, but mesh is an optional attribute of an item, and different items can then reference the same mesh attribute. This means that FBX uses mesh instances only, not a mesh source with optional mesh instances. This has the following consequences:

- When importing FBX, the first mesh instance is converted into the source mesh item.
- You cannot use **Export Selected Layers** in the Item Context Menu for **.fbx** export. Instead, in the export preferences, set the **Export Type** to **Export selection** or **Export selection with hierarchy**, and then use **File** > **Export as...** to export the scene.

FBX Import

Modo offers a number of FBX loader plug-ins. The most commonly used are:

- **Compatibility loader**: Autodesk FBX 2015 plug-in.
- **Normal loader**: Autodesk FBX 2018 plug-in.

The 2018 plug-in is backwards compatible and should be able to read prior versions of **.fbx** files. However, we cannot say that it is 100% reliable, and so other plug-in versions are included just in case.

Note: The Autodesk FBX 2015 plug-in does not support newer versions of the FBX format.

To select the loader type:

Windows: Open **System** > **Preferences**.

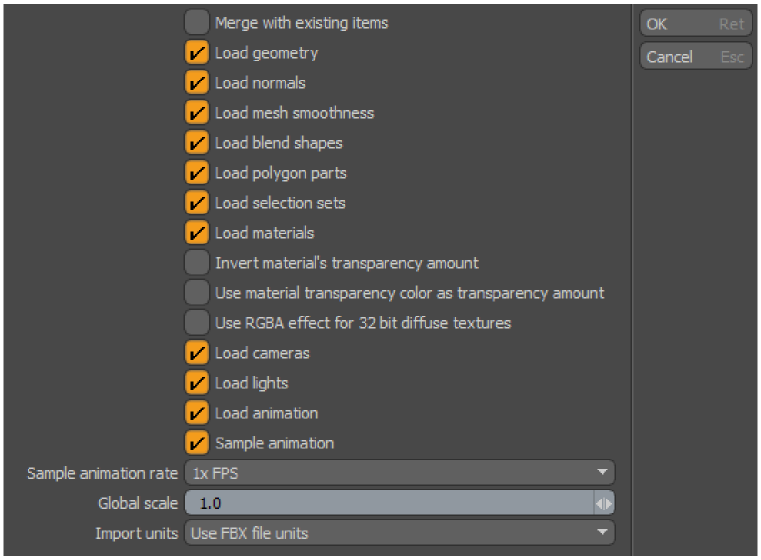
Mac OS: Open **Menu** > **Preferences**.

1. Select **FBX I/O** from the **Preferences** list on the left.
2. Check **Use legacy plugin for file import** to use the Autodesk FBX 2015 plug-in. Leave this unchecked to use the latest version of the Autodesk FBX 2018 plug-in.

Both loaders share the same settings, though they are written slightly differently on the import window. The image below shows the window for the normal loader. The property descriptions apply to both the normal loader and the compatibility loader.

Note: Image maps have their color correction turned off after file import. This means that images whose pixels represent vectors should not have color correction applied to them. **Bump**, **Vertex Normal**, **Displacement**, and **Vector Displacement** maps all have their **Colorspace** set to **(none)** after import.

Note: Animation from **.fbx** files can be imported onto current scene items. To do this, select **Merge with existing items** before the import. Items in the scene are matched by name.



☐ Merge with existing items
☑ Load geometry
☑ Load normals
☑ Load mesh smoothness
☑ Load blend shapes
☑ Load polygon parts
☑ Load selection sets
☑ Load materials
☐ Invert material's transparency amount
☐ Use material transparency color as transparency amount
☐ Use RGBA effect for 32 bit diffuse textures
☑ Load cameras
☑ Load lights
☑ Load animation
☑ Sample animation
Sample animation rate: 1x FPS
Global scale: 1.0
Import units: Use FBX file units

The FBX texture loader reads all available FBX material properties and imports all available textures, even if the texture effect is not recognized by Modo. Textures that have known texture effects are set to Modo's texture effect counterparts once imported For example, FbxSurfaceMaterial::sDiffuse maps to a **Diffuse Color** texture effect in Modo. Textures that have unknown texture effects are still imported as Modo textures but without a texture effect.

Importing all available textures makes sure all available embedded textures inside an **.fbx** file are extracted and loaded in Modo. An additional benefit of this import method is that you don't need to find and manually add any textures that were outside of the **.fbx** file, but weren't imported because their texture effects weren't recognized.

Merge with existing items	Add the file contents to the scene without removing existing items. Uncheck this to overwrite the scene.
Load geometry	Load the model geometry.
Load normals	Load Vertex Normals. If unchecked, the normals are calculated by Modo.
Load mesh smoothness	This imports any smoothing information. Modo uses this to create a subdivision map.
Load blend shapes	Loads any blend shapes that are in the model. **Note:** Blend shapes are a way of animating a mesh through capture of deformations and subsequent interpolation. They are equivalent to morph maps.

Load polygon parts	Loads any parts that have been defined on the mesh. See Polygon Set Part for more information.
Load selection sets	Loads all Using Selection Sets.
Load materials	Loads all Materials. See Material for more information.
Invert material's transparency amount	If transparency is used in the source materials, invert the transparency amount. (Some applications use a reverse transparency scheme to Modo.)
Use material transparency color as transparency amount	If the source materials use transparency color, convert this to Transparency Amount in Modo. **Note:** Modo uses the intensity of the red component in the transparency color to determine the transparency amount.
Use RGBA effect for 32 bit diffuse textures	Change a **Diffuse Color** effect (see Layer Effects) to an **RGBA** effect for textures with an alpha channel.
Load cameras	Loads all cameras as **Camera Items**.
Load lights	Loads all lights. See Lights for more information.
Load animation	Restores animation if used.
Sample animation	• If the file uses animation, check this to re-sample the animation curves with continuous keyframes defined by the **Sample animation rate**. **Note:** Sampling is recommended as Modo has a different method of computing the animation curve compared to FBX. This difference results in different animation curves, and therefore different animations. By using sampling, this can be avoided.
Sample animation rate	• Select the rate at which to sample the animation curves: • **1x FPS** - Sample at the frame rate. • **2x FPS** - Sample at twice the frame rate.
Global scale	The factor by which the imported items are scaled. This is useful if you want to restore geometry at a different size to the source geometry.
Import units	By default, the import procedure restores the geometry with the same units as the FBX file. You can change this using the **Import units** setting.

FBX Export

UV maps are sorted by name during file export. This is for compatibility with the Unity engine. Unity supports only two numbered UV maps per mesh, UV1 and UV2. Sorting UV sets by name before exporting allows game artists to select which Modo UV set will be exported as UV1 and UV2.

The animation for the **Enable** option in a Morph Influenceitem is now included in the **.fbx** export.

The **.fbx** equivalent of Modo's Vertex Maps are geometry elements, which are organized into layers.

Note: A single layer holds all the elements of the same type. For example, to store two vertex color maps, three UV maps and one vertex normal map, three layers are required.

Applications may support a limited number of layers. For compatibility reasons, it is recommended that you produce the smallest number of layers possible.

FBX IO Preferences	
FBX Preset	Select the FBX preset to use. The following preset types are available: • (none) • Maya 2017 • Unity Animated • Unity Static • Unreal Animated • Unreal Static You can also add, store, and remove presets.
Use legacy plugin for file import	When enabled, Modo uses the Autodesk FBX 2015 plug-in for import. Leave this unchecked to use the Autodesk FBX 2018 plug-in. For more information on the plug-ins, see FBX Import.
Export File Format	Choose which **.fbx** version to export when saving **.fbx** format files. The default is set to **Use Latest**.
Export Type	Choose which elements from the scene are exported: • **Export All** - All items within the scene. • **Export Selection** - Only the selected items.

FBX IO Preferences	
	• **Export Selection with Hierarchy** - Selected elements and their children.
Save as Text Format	Generates human-readable TXT formatted **.fbx** files. When disabled, Modo binary encoded **.fbx** files are generated. **Note:** TXT readable files can be many times the size of binary encoded FBX files.
Save Only Animation	Export animation only, excluding other content such as geometry, camera, materials and so on. This overrides all other Save checkboxes. **Tip:** This is useful if you want to export the animation separately to the other content.
Save Geometry	Export all polygonal geometry (mesh) items.
Save Materials	Export materials at the best compatibility that can be achieved. **Note:** Due to the nature of different rendering systems, complex material setups can't be transferred, but baking can be used to create image maps that can be transferred.
Save Cameras	Export all camera items.
Save Lights	Export all light items.
Save Animation	Export keyframed animations. **Note:** If the animation is not saved, the currently selected frame in the timeline is exported.
Surface Refining	Specifies how to refine surface on export. The following options are available: • **None** - No surface refining is used during export. • **Save Mesh Smoothness** - Modo saves subdivision surface weights to FBX. • **Triangulate Mesh** - Triangulate the mesh during file export. This may be

FBX IO Preferences	
	required for certain engines, and it avoids manual, temporary conversion to triangles prior to export. **Note:** • The Exporter stores triangulated versions of the original geometry and does not change the scene in any way. • You cannot apply the triangulation selectively during export, every mesh is triangulated. • The results cannot be influenced or previewed. • This option allows quad-based meshes to be directly baked to, and exported to, real-time engines (such as Unreal or Unity).
Save Polygon Parts	Export polygon **Part** tags.
Save Selection Sets	Export defined selection sets.
Save Smoothing Groups	Export any smoothing groups that are assigned to the geometry.
Save Morph Maps	Export morph maps as compatible blend shapes in the **.fbx** format.
Save Tangent Basis	When enabled, the tangent and bitangent normal basis states are written to the FBX file. **Note:** If both the real-time engine (or other 3rd party applications) and Modo support compatible tangent basis algorithms (like Mikk or Unity), it is not recommended to export the tangent basis to FBX, since they are automatically created when the FBX file is imported into the 3rd party application. This option should be used only if the destination application does not support Modo's provided tangent basis algorithms, or if there are problems when rendering normal maps in the 3rd party application.
Units	The unit of measurement that the exporter converts values to. By default, FBX values are measured in centimeters. This option allows other units to be used.
Scale Multiplier	The factor by which all scene objects are scaled. This is useful if you want to

FBX IO Preferences	
	export geometry at a different size to the source geometry.
Export RGBA as Diffuse Color	Convert all **RGBA** texture layers to **Diffuse Color**. (FBX does not have support for Modo's **RGBA** texture layer type.) **Note:** File import can convert the other way around (from diffuse color to RGBA). See FBX Import.
PBR XML Export	Export Physically Based Rendering (PBR) materials for use with the Unity and Unreal Engines. This feature invokes a Python script to generate an **.xml** file, which describes PBR material properties.
Sample Animation	When enabled, Modo re-samples the animations, resulting in a baked type of animation, with continuous keyframes defined by the **Animation Sample Rate**.
Animation Sample Rate	Select the rate at which to sample the animations: • **1x**: Sample every frame. • **2x**: Sample twice every frame.
Export Actions	Choose how to export **Actions**: • **Don't Export Actions** - No actions are exported. • **Export Actions Into Separate Takes** - Convert actions into FBX format **Takes**. • **Export Current Action** - Export only the currently selected action. This is converted into an FBX **Take**. **Note:** You cannot currently export both action-based and regular timeline-based animation in the same file.

GLTF V2 I/O

Export Type	Specify the export type. The following options are available:

	• **Export All** - Exports the whole scene with hierarchy. • **Export Selected** - Exports selected items in the scene without hierarchy (all items are considered as nodes without parents). • **Export Selected with hierarchy** - Exports selected items in the scene with hierarchy. **Note:** Make sure you freeze any transformations you've made to your skinned mesh before exporting into glTF. Unfrozen skinned mesh transformations are ignored during import into any application that supports glTF according to the glTF v2 specifications.
Save Geometry	Controls which items are exported. When enabled, geometry is saved to the **.glb** file.
Save Materials	When enabled, materials are saved as either a glTF V2 (GL Transmission Format) (***.glb**) or a glTF V2 (GL Transmission Format) (***.gltf**) and (***.bin**). **Note:** Due to the nature of different rendering systems, complex material setups can't be transferred, but baking can be used to create image maps that can be transferred.
Save Animation	When enabled, animation is saved to the **.glb** file.
Save Weight Maps	When enabled, exports weight vertex maps. This option is also used to disable and enable skinned animation.
Save Skinned Meshes in World Space	Applies mesh transformation matrix to mesh vertices during export and moves them to bind position in world space. World space mesh vertices are transformed by inverse bind matrices of joints and their vertex weight influence correctly. **Tip:** Enable this option for skinned animation in glTF v2 for compatibility with external previewers (for example, Babylon.js).
Save Tangent Basis	When enabled, the tangent and bitangent normal basis states are written to the **.glb** file.

Indices data type	Specify a data type for geometry indices data for GLTF exporter. This option is used for compatibility with older GL implementation. The following options are available: • **Unsigned Short** - Exports polygon indices as 2 byte arrays per element. • **Unsigned Integer** - Exports as 4 bytes array per element.
Animation Sample Rate	Enables more animation samples per frame for better interpolation results.
Export Actions	Specifies the type of action to export. The following options are available: • **Don't Export Actions** - Creates animation samples from the current timeline key frames without naming glTF animation layers. • **Export Actions into Separate Animation Layers** - Exports all animation actions as separate layers that are named by actions names. • **Export Current Action** - Exports currently selected animation action to sample from.

HPGL Plotter I/O

Projection	Since HPGL is a flat, 2D format, the **Projection** option defines the plane that it is converted to for the final export. **Automatic** tries to determine the best guess plane, with options to define a specific plane or the UVs for the object.
Scale	You can define a default scaling amount when importing HPGL files, eliminating the need to manually scale drawing after they are inserted into a scene.

Image I/O

JPEG	
Quality	This option determines the image/compression quality of images saved from Modo into **.jpg**. 100% is maximum quality possible with **.jpg**, lower values reduce the file's overall size, but also degrade image quality.

JPEG	
Sub-Sampling	The JPEG file format compresses images by separating the color and luminance (brightness) information. Most people are more sensitive to contrast than they are to color variations, so this information is separated into channels, where each can be compressed differently. This compression amount is expressed through the sampling or sub-sampling value. The default save option, **4:1:1**, represents each channel where the luminous channel has four times as much information as two combined color channels. This can make image files quite small, but produces softness on areas of high color contrast. The **4:2:2** option provides double the amount of color fidelity, and the **4:4:4** has the highest quality, saving the same number of data samples for each channel, but also produces the largest of the JPEG compressed file sizes.
Image Cache	
Limit Image Load Size	When this option is enabled, Modo limits the size of large bitmap images when imported and rendered, based on the setting of the **Maximum Image Load Size**. As large images are loaded, they are resized, so they take about the same memory as a square image with the given dimensions. This can be useful if you're editing a model that's been textured with very large maps and don't need to see them in detail. ⚠ **Warning:** This preference affects rendering.
Maximum size that an image will be loaded	Determines the maximum pixel length for any image imported. Images larger than this setting are resized to fit the specified value.
Image Tile Cache Size	The amount of space dedicated to holding resized images in memory. The more images get loaded and resized, the more memory is necessary to hold them all.
Image Format	
Default Image Format	You can use this option to determine the default format when saving Modo-generated bitmap images. You can select alternate format types in the file **Save** dialog, this option only determines the default when you use **Save as**.
PNG	

JPEG	
Compression Level	The **.png** format uses a lossless compression scheme similar to ZIP. Values span from 1 to 9, with 9 being the greatest amount of compression. The compression amount doesn't affect the visual quality of the images, but it affects compression and decompression performance, which can be useful when playing back image sequences.
OpenEXR Export	
Compression	When saving OpenEXR images, you are provided with a number of different compression offerings: uncompressed, lossless, and lossy compression types. It is best to test the setting for the intended purpose to make sure the format chosen is compatible with the intended target.
DWA Compression Level	This option determines the amount of lossy compression applied to the EXR image when saved, the higher the value the greater the compression. but also the more likely to introduce visible artifacts. The default value of 45 should be visibly lossless, but with a smaller saved file size.
Tile Size	When the **Generate Mip Map Levels** option is enabled, the **Tile Size** option defines the size of the pre-generated tiles of the resulting Mipmapped EXR image. 16 or 32 sized tiles are good values for textures ultimately used in Modo.
Generate Mip Map Levels	If this option is enabled, mip map levels are generated when saving OpenEXR files from Modo. Mipmapping is an optimized collection of pre-calculated image maps that are filtered and scaled, and used to reduce image overhead when loading and rendering high resolution files. When disabled, only the full resolution file is saved.
OpenEXR Loader	
Limit Tile Memory	When this option is enabled, you can use the **EXR Tile Memory Limit** setting to define a maximum amount of memory that Modo uses to load all EXR images into a scene. When disabled, Modo fully loads all EXR images into memory.
EXR Tile Memory Limit	When the **Limit Tile Memory** option is enabled, you can define an upper memory limit used to cache OpenEXR mip map tiles. When the maximum amount is reached, the EXR image loader discards the oldest tiles based on when they were last used. If a tile is required later on, it is reloaded into

JPEG	
	memory and older tiles get discarded, if necessary.
Cache Tiles to Disk	When this option is enabled, purged mip map tiles are stored in the drive's temporary directory in a raw, uncompressed format until they are needed again. This reduces overhead associated with opening and decompressing full size EXR images. This information is destroyed and recreated each time Modo is closed and started, respectively.

LightWave I/O

Load SubPatch as Subdiv	When this option is enabled, any geometry generated in LightWave using SubPatch geometry smoothing technique automatically converts to Modo's standard Subdiv when imported.
Save Subdiv as SubPatch	Any geometry modeling Modo as a Subdivision Surface can be converted to LightWave's Subpatch mode upon export to LightWave format.
Content Directory	LightWave supports a content directory structure to make moving scene files around easily. When loading scene dependent on Content Directories, it is helpful to specify the LightWave equivalent path prior to loading and saving the file to aid Modo in locating the appropriate assets (namely image maps) into and out of Modo.
Save Flat Transforms	Saving full Modo scenes to LWO object files means saving out all the meshes to mesh layers. Typically, the item transformations applied in Modo are not applied to the mesh, so it is positioned as if the item mode rotation, scale and positioning were all reset. Turning on this option freezes all the layers of item transformation, so the entire hierarchy is exported as if modeled in that pose. This behavior may be undesirable if you are rigging the layers later on.
Texture Conversion	This option attempts to reproduce the LW texturing in the **Shader Tree**. Use this if exporting textures unchanged back to LWO is not important.
Text Encoding	Set the text encoding. The available options are **UTF-8** and **System Defaults**.

Movie I/O

Video Format	
Default Video Format	You can use this option to determine the default format when saving Modo-generated video files. You can select alternate format types in the file **Save** dialog, this option only determines the default when you use the **Save As** command.

STL I/O

Interpret Export units as	The STL file format has no inherent scale information, so systems generally default to several standard values. Depending on the target, you can use the **Interpret units as** option to define a scale unit when exporting STL format files.
Save as ASCII	When the **Save as ASCII** option is enabled, saving a file generates the less common ASCII (human readable text) format. When disabled, resulting STL files are saved in the more common binary format.
Interpret Import units as	The STL file format has no inherent scale information, so systems generally default to several standard values. Depending on the target, you can use the **Interpret import units as** option to define a scale unit when importing an STL format file.

USD I/O

Convert Distance Units	When enabled, distance units in USD files are converted during import with respect to the authored metersPerUnit value of the USD file. For example, if metersPerUnit in the USD file is set to 0.01 (the default), then one unit in the USD file is equal to one centimeter. If you import a mesh that is 200 units wide, Modo translates this to 200 centimeters, or two meters.

	When disabled, the distance unit conversion is not applied upon import. For example, if you have set the metersPerUnit value of the USD stage to 100, Modo ignores this and still considers each unit in the USD file to be one centimeter (or 0.01 of a meter).
Convert Light Intensity	When enabled, light intensity from lights within USD files are scaled automatically by the **Light Intensity Scale Factor**. When disabled, no intensity scaling is applied.
Light Intensity Scale Factor	When **Convert Light Intensity** is enabled, scale the intensity of all lights within the USD file by this amount.

Wavefront I/O

Wavefront Object Import	
Import as Static Mesh	When this option is enabled, OBJ geometry is imported as non-editable static Mesh Items. This is useful when importing huge geometry files that can slow down scene interactivity.
Import Groups as Separate Meshes	When this option is enabled, defined groups within the OBJ file generate multiple Mesh Item layers for each.
Suppress Import Options Dialog	This option, when enabled, suppresses the pop-up window that opens when importing an OBJ file.
Import Units	You can determine a scaling amount for imported elements automatically, by choosing from the various **Import Units**.
Wavefront Object Export	
Save Meshes as Groups	When enabled, the export utilizes the **G** group tag to define geometry sections within the file created by using the Polygon **Set Part** tool. When disabled, the export uses the **O** object tag. **Note:** For more information on the Set Part command, see Polygon Set Part.

Wavefront Object Import	
Save Material File	When this option is enabled, an additional MTL file is generated when exporting OBJ files that contains the basic surface definitions and links to textures. When disabled, the MTL file is skipped.
Save deformed geometry at current scene time	With this option enabled, Modo freezes the current deformation state of the exported element at the current Timeline position, producing a deformed mesh.
Export Units	You can determine a scaling amount for exported elements automatically by choosing from the various **Export Units**.

Input Preferences

You can customize how Modo handles input.

To access these preferences:

- On Windows, open **System** > **Preferences**.
- On Mac OS, open **Menu** > **Preferences**.

3D Mouse

Navigation Style	The 3D Mouse provides two main navigation styles: • **Camera Walkthrough** - Moving the puck moves the viewport's camera as though you were moving the camera itself. Pushing the puck to the right moves the camera to the right, and thus the scene appears to move to the left, similarly to playing a first-person video game. • **Object Orbit** - Moving the puck moves the viewport's camera so that you seem to be moving the item in the view. Pushing the puck to the right moves the camera to the left, thus making it appear that the object is moving to the right. Twisting the puck rotates the camera around the object, as though the object was spinning around in front of you. The center of rotation is

	determined by the blue dot, called **Auto Rotation Center**, which appears at the center of the viewport when you start manipulating the puck. If the item under the dot fills more than 80% of the view, the blue dot represents a point on the surface of the model that is tracked in 3D space until you release the puck, at which point the blue dot returns to the center of the viewport when you touch the puck again). If less than 80% of the view is filled with the object, the object's bounding box center is used as the center of rotation.
Orthographic View Style	The following options are available to set for orthographic view. • **World Space** - The translation axes of the puck move in world space, irrespective of the current orthographic view. Thus, pushing left or right on the puck always moves on the world X axis, even if the orthographic view is showing the left or right (and in which case you won't appear to see any movement, since you're moving perpendicular to the view). Â Best used/demonstrated with the Model Quad view. • **Screen Space** - The translation axes move relative to the current view: x20left/right moves on the horizontal axis, and up/down moves in the vertical axis. Pushing in/out changes the view's zoom. • **Screen Space (Swap Y and Z Axes)** - Pushing the puck left/right moves left/right just like in **Screen Space** mode, but pushing the puck forward/backward moves the screen up/down, while pulling up/down zooms in and out. Basically, the Y and Z axes are swapped.
Use Modifiers to Toggle Styles	When enabled, holding down **Alt** toggles between the two navigation styles. Similarly, holding down **Shift** and **Ctrl** toggle between the two **Orthographic View Styles.**
Speed	This is a global speed modifier, a scaling value applied to the 3D mouse's input values to make the mouse move faster or slower. This is not capped to 100%, so you can set it higher if you want it to go faster.
Swap Y and Z Axes	This option switches the Y axis with the Z axis, useful for people who think of the screen as lying on a desk (usually CAD/architecture users), as opposed to those who think of the screen as vertical (usually video-centric users).
Focus on Viewport Under Mouse	When enabled, the 3D mouse movement applies to whichever viewport the mouse cursor is currently over. If disabled, you must first interact with the viewport by clicking in it with the mouse or by rolling the mouse wheel, before the 3D mouse switches focus. This is indicated by the orange color of the **Thumb**. In both cases, if the mouse is not over a view that can accept 3D

	mouse input, the last view that can accept such input continues to get input.
Show Rotation Center Marker	When enabled, a blue dot is displayed to indicate the orbit rotation center.
Lock Horizon	Maintains a level horizon when navigating (this is different than just disabling the Z axis rotation).
Lock Axes	
Dominant Axis Only	Only the axis that has the most force applied to it is used, all others remain idle. This is useful if you only want to move on one axis at a time, but don't want to have to keep toggling the **Lock** buttons.
Invert Axes	
Invert All Axes	A simple script that toggles the state of all of the **Invert** buttons. There are similar scripts (and buttons) to lock all the translation or rotation axes.
Advanced: Thresholds	
Translation/Rotation Threshold	The 3D mouse has a range of 0-100%. Thresholds allow you to require a certain amount of force on an axis before it registers as movement. If set to 10%, you need to push the mouse harder before you see any movement, at which point the remaining 10%-100% range is expanded to be the new 0%-100% range. **Tip:** If you're a new user, this may help to avoid accidentally moving on multiple axes at once, and can be thought of as training wheels. However, it also can make the puck feel unresponsive due to the extra force required to see any movement, and may keep you from becoming proficient at controlling multiple axes at once. Mostly, you should leave these at 0%.
Advanced: Speeds	
Translation/Rotation Speed	Per-axis speeds, should you want to change the speed of an individual axis. The global speed control is applied on top of these values.
Advanced: Non Linearity	
Global Exponent	Globally scales the non-linear exponents in the same way that the global

Multiplier	speed scales all speeds. So, it provides an easy, high-level way to further adjust all of the exponents at once.
Translation/Rotation Exponent	Determines the non-linearity of the 3D mouse movement. The original linear implementation made it difficult to do both well-controlled small movements and larger movements. With non-linearity, more of the 0-100% input range of the 3D mouse is used for fine control, but coarse control is still available by simply pushing the puck further.

Drop Mappings

The Drop Mapper options allow you to change the order of drop actions relative to each other within specific modifier combos. This allows different drop actions to be assigned to different modifier combos during a drag and drop operation. It is meant mainly for plug-in developers and advanced users wishing to further customize Modo for specific tasks.

The Drop Mapper is just a list of all possible modifier combos, plus “unmodified". Each combo contains all possible drop actions (as identified by tags on the drop servers; more on that later), as we can’t know which actions may be available for a given drop. You can drag and drop (D&D) to re-arrange both the combos and the actions relative to each other (combos to combos and actions to actions). You can also move an action to the “never map” section to indicate that it will never be available on this modifier combo.

For example, the **Copy Files** drop action for the **File to File System** drop server is never allowed on an unmodified drop, so if you drag and drop a file inside its own parent directly in the Preset Browser, you are not able to drop it, as no actions match the input. Instead, you have to hold **Ctrl** to copy it.

When you move the mouse during drag-and-drop, the list of drop actions that match the current source and destination is built as normal. The code then loops through the first combo in the drop mapper and tests each action within, in order against the current drag-and-drop actions. Once a match is found, it is assigned that modifier combo and we move on to the next combo, and so on. The order of the combos and the order of the actions inside the combos determines which action gets associated with which combo first.

Viewport Overrides: Viewport overrides provide a way to change the order of the actions on a per-viewport basis, as well as excluding them from a viewport entirely. If an action isn't shown in a viewport's override, it falls back to the global mappings (meaning, those when the **Viewport Overrides** dialog is set to **(none)**). When the Source Filter is on, only actions related to that source are shown.

To add a viewport override, click the **(add action)** entry to open a dialog and choose one. Disabled entries indicate that they are already used on in this combo. To remove a viewport override, click the **x** icon on the right of its row.

The override actions in a combo can be re-arranged just like normal actions.

Note: Combos cannot be reordered to an override; they are always in the same global order.

Source Filter: Allows you to show only actions that apply to specific drop sources. Since usually only one source type is being dragged at a time, this limits the list of actions to just those that may show up in the popover together, although depending on the specifics of the source only a subset of those may be shown.

Making Actions Available For Mapping

To show up in the drop mapper, we need to know what actions are available. This is done via the new LXsDROP_ACTIONNAMES tag on the drop server. The format is simple enough:

```
      "1@myAction1 2@myAction2"
```

So, it's a space-delimited list, where each entry consists of an action code, the "@" character, and the message name. The message is looked up from a message table with the same name as the server. The action code is the arbitrary number the server passes to ILxAddDropAction::AddAction() when their ILxDrop::ActionList() method is called.

Currently, only the **File to File System** drop server defined by the Preset Browser implements these server tags. Any servers that don't have these tags are only selectable from the dropover.

Performing Drops With Modifiers

Modifier keys can be used to change drop actions from both in and out of the dropover.

From the Dropover: The dropover now displays the modifier keys associated with an action, as well as "(unmodified)" for the default drop. You can change the currently-selected action by pressing modifier keys, or you can also choose the current action by mousing over it (whichever is done most recently is the one that's chosen).

Without the Dropover: You can also use modifiers outside of the dropover, but you don't get any feedback.

Game Input Mappings

You can configure the game navigation controls. See on how to use this.

You can customize the keys assigned to the various actions. You can also select between **QWERTY** and **AZERTY** keyboards for game navigation. This switches the **Q** and **A** keys to **Z** and **Q**, respectively, to match French and other similar keymaps.

Remapping

Mouse Input Preset	If you're coming to Modo from another 3D application, you can adjust the mouse navigation controls to mimic those of the chosen application with this option. A variety of options are available representing a number of popular 3D programs.
Input Device	This option determines how Modo interprets input commands. **Mouse** style allows you to continue hauling offscreen. **Tablet** style stops at the edge of the screen. **Hybrid** mode leverages the Mouse style for hauling actions and Tablet style for any viewport navigation. This can help improve performance in the viewport. **Note:** If you're using a tablet with relative positioning, you should set this option to **Mouse**.
Allow Modifier Chording	When enabled, modifier keys can trigger chording. When disabled, only mouse buttons can trigger chording. Chording is the use of multiple mouse buttons simultaneously. **Note:** Chording has been widely recognized as a key contributor to carpal tunnel syndrome and other repeated stress injuries. The Foundry recommends that you avoid chording to prevent unnecessary muscle strain.
Numeric Enter as Tab	When enabled, pressing **Enter** on the numeric keypad functions the same way as pressing **Tab** does, by advancing the cursor to the next logical data field.
Open Dialogs Under Mouse	This option forces new dialogs to always open at the current mouse position, for instance the **Save As** and **Open** file dialogs. When disabled, dialogs open in the same position as the last open dialog.
Tooltip Delay	Defined in seconds, this option determines the length of time before the tooltip info display opens when hovering the mouse pointer over a button or function of Modo. The default value is 0.8 seconds.
Mouse Wheel	This percentage value determines the speed at which Modo zooms in or out

Acceleration	when using a mouse's scroll wheel. Higher values zoom faster, while lower values zoom more slowly.
Wheel Zoom at Mouse Cursor	When this option is enabled, zooming in any OpenGL viewport with the mouse scroll wheel center the zoom based on the mouse's position over the viewport, when disabled, the mouse's position is disregarded and the viewport zooms based on its center.
Invert Mouse Wheel Zoom	This option inverts the zooming behavior of the mouse wheel. Pushing forward on the wheel pushes out, instead of in.
Reverse Pan Widget Direction	This option inverts the direction of the Pan widget.
Zoom Drag Axis	This option determines the directional behavior when hauling in a 3D viewport when zooming. The options are **Left to Right**, **Right to Left**, **Bottom to Top**, and **Top to Bottom**.
Viewport Rotation	
Trackball Rotation	This options determines the default navigation rotational type. When disabled (the default), viewport rotations are limited to two axes. When enabled, the viewport can be rotated on multiple axes based on the mouse's viewport position, as if the viewport itself was a giant trackball. You can toggle this setting for each individual viewport separately in **Viewport Properties** > **Drawing and Control** > **Mouse Control**, if **Override Viewport-Level Trackball Option** is disabled. **Note: Trackball Rotation** is **Off** by default. Upon first launch of Modo and rotating the viewport for the first time, a dialog appears asking if you'd like to try out Trackball Rotation to see which rotation style you prefer. This dialog disappears after 17 seconds.
Override Viewport-Level Trackball Option	When enabled, this option disables an individual viewport's ability to override the **Viewport Rotation** preference setting, locking all viewport rotation options to the same preference setting.
Trackball Speed Multiplier (Mouse/Tablet)	This option determines the rotation speed for using trackball rotation when navigating in an OpenGL viewport. There are separate settings for mouse and tablet.

Tree, Browser and Form Navigation	
Invert Drag Scrolling	This option inverts the direction of scrolling for trees, browsers, forms, and similar views when panning with **Alt**+**Shift**+drag.
Invert Mouse Wheel Scrolling	This option inverts the direction of scrolling for trees, browsers, forms, and similar views when panning with the mouse wheel.

Selection

Paint Selection	
Polygon Selection	This option determines the behavior of lasso or paint selection modes when selecting components in the OpenGL viewports. The following options are available: • **Frontface & Wireframe** - Default behavior is only forward-facing polygons are selected, except when in **Wireframe** mode, where both the front and back-facing polygons are selected. • **Frontface & Backface** - Default behavior is front and back faces are always selected regardless of viewport style. • **Frontface only** - Default behavior is to select only front-facing geometry regardless of viewport style. • **Always Raycast** - Paint selection does occlusion testing for hit element regardless of shading style. **Wireframe** view is treated as **Solid** shading view. If the hit element is occluded by polygon on wireframe view, it is not selected. • **No Raycast** - Paint selection does not do occlusion testing for hit element regardless of shading style. **Shaded** view is treated as **Wireframe** shading view. Even if the hit element is occluded by polygon on **Shaded** view, it is not selected.
Double Sided Selection	When this option is enabled, you can paint select any polygon tagged with a material that is defined as **Double Sided** from their back side. When disabled, you cannot paint select polygons from the back.
Selection Hit Size	This value determines the area around a selectable (pre-highlighted) element where clicking the mouse selects it. This value is determined as screen size

Paint Selection	
	pixel scale.
Point Selection Priority	This is for controlling the multi component selection mode used in tools such as Element Move. When the cursor is close to the corner of a polygon, the distances from the cursor to vertex and the connected edges are very close. The **Point Selection Priority** option gives priority to point selection over edge selection.
Lasso Selection	
Ignore Shading Style	The typical behavior of polygon selection in Modo is to select what is visible to the user. This means that when lasso selecting a sphere in **Shaded** display style, only forward-facing polygons are selected, but while in **Wireframe** display style, the same selection selects both forward and backward-facing polygons (as the rear-facing polygons are now visible to the user). When **Ignore Shading Style** is enabled, Modo's selection mode works identically regardless of the viewport display style.
Allow Partial Polygons Selection	In order to select a polygon when using the lasso selection style, you need to completely surround the target with the lasso in order to successfully select the intended geometry. With the **Allow Partial Polygons Selection** option enabled, all selectable geometry that is intersected by the selection line, even by a tiny bit, gets selected.

Units

Distance Units	
Unit System	You can choose your preferred measurement units system based on several options. Once set, this is the measurement input method Modo uses for all numerical input values. • **SI** - The International System of Units (abbreviated 'SI') is the modern form of the metric system, complies to universal base units. • **Metric** - A universal system of measurement based on powers of 10 - millimeter, centimeter, meter, kilometer. • **English** - A historical measurement system based on Imperial Units -mils,

Distance Units	
	inch, foot, yard and mile • **Game Units** - An arbitrary unit of measurement (defined by the **Meters per Game Unit** setting). • **Unitless** - An arbitrary decimalized unit of measurement based on cubic meters. 1 unit = 1m.
Default Unit	This option is dependent on the **Unit System** and determines the default unit used when no unit is specified.
Inch Scale Cutoff	When the **English** unit system is selected, the **Inch Scale Cutoff** determines the cutoff where Modo displays inches as feet and inches. For example, when set to the default 36, 27 inches display as 27", but 42 inches display as 3' 6".
Meters per Game Unit	When utilizing the arbitrary **Game Units** system, use this value to determine the scale of a single unit. This allows you to work with even whole numbers. So if a 1 m equivalent is really 1.375 real world meters in Modo, you can enter 1.375 here and then specify units normally in the numerical input fields.
Dynamic Units	
Force Unit System	This option determines the default force unit Modo uses when specifying dynamic force.
Default Force Units	You can specify alternate units of force to work with using this control. Modo defaults to the selected unit when a unit is not specified in the related input field.
Acceleration Units System	This option determines the default Acceleration unit Modo uses when specifying acceleration.
Default Acceleration Units	You can specify alternate units of Acceleration to work with using this control. Modo defaults to the selected unit when a unit is not specified in the related input field.
Speed Unit System	This option determines the default speed unit Modo uses when specifying speed (distance traveled over time).
Default Speed Units	You can specify alternate units of speed to work with, using this control. Modo defaults to the selected unit when a unit is not specified in the related input field.

Distance Units	
Mass Units System	This option determines the default unit of mass Modo uses when specifying an object's mass.
Default Mass Units	You can specify alternate units of mass to work with, using this control. Modo defaults to the selected unit when a unit is not specified in the related input field.
Light and Color Units	
Light Unit System	This option determines the default light unit Modo uses when specifying a light's brightness. • **Radiometric** - Measurement of the electromagnetic spectrum, which includes visible light, defined by power of radiation. • **Photometric** - Measurement of brightness of light as perceived by the human eye, defined by luminous intensity.
Color Unit System	Determines the color unit system used for specifying colors in Modo. • **Floating Point** - Determines colors based on decimals, uses 0-1 range and beyond for HDR colors. • **Percentage** - Determines colors based on percentages, uses 0%-100% range. • **Integer** - Determines colors based on 8bit value scale, uses 0-255 range (2 to 8th power = 256 values). • **Hexadecimal** - A computer numerical system similar to binary, represented by 16 characters. 2bit Hex produces 256 values, similar to Integer color units. Widely used in specifying color on the web, uses 00.00.00-ff.ff.ff range.
Percentages	
Factor or Range	This option determines surfacing and material value units. You can specify values as a percentage 0%-100% or as a float value 0.0-1.0.
Accuracy	
Coordinate Rounding	This option controls how mouse input is converted through the **Work Plane**, into 3D coordinates. You can choose from the following options: • **None** - No coordinate rounding is done. Every mouse move gives unclamped coordinates, typically with lots of decimals. This option is the raw 2D to 3D transform. This is useful for working freehand.

Distance Units	
	• **Normal** - This option attempts to give clean, round coordinates, based on your view transform. As the mouse moves, you can see values in your current unit system display in the information tab. The step size gets smaller or larger as you zoom in or out. You may need to move the mouse cursor two or three pixels to see values update. • **Fine** - This option is similar to **Normal**, but optimizes for closer to one step of coordinate rounding from one pixel of mouse movement. This gives you finer grained input, but can be difficult to hit exact values. • **Fixed** - This option uses the **Fixed Increment** preference to put a lower limit on both coordinate rounding and the grid. If you set the fixed increment to 10 mm, that means the grid never gets finer than 10 mm, and all input is rounded to the nearest 10 mm even at high zoom. When you zoom out, however, the grid shows larger values but the step size always remains a multiple of the fixed increment. • **Forced Fixed** - This option is similar to **Fixed**, but forces the size of the grid and the input step to match the increment exactly, no matter the zoom level.
Fixed Increment	When the **Coordinate Rounding** is set to either of the fixed options, this value determines the fixed coordinate rounding grid.
Coordinate System	
Up Axis	This option determines the major axis that is considered the default Up direction for Modo. 3D programs typically use the Y axis as up, where CAD applications typically use Z as the up direction. When importing geometry, if it is always 90° off, changing the Up axis to match the originating application may resolve the issue. This can also be set separately for each scene in the Scene Item.

Work Plane

You can favor certain planes when using the **Work Plane**, especially when you don't want to lock the plane, but it would be helpful to have it face a certain direction, such as modeling a rough city. You could rotate the views, but keep the Work Plane mostly facing upwards, as if it was the ground, when creating new geometry.

Work Plane and Grid	
Grid Type	An option for selecting which type of grid to display in the viewport. The following options are available: • **Fixed** - The grid is displayed in regular intervals. • **Dynamic** - The grid is dynamic depending on the zoom level, a small display in the lower right corner gives you the real world equivalent measurements for each square unit.
Size	This option is enabled when the **Grid Type** is set to **Fixed**, allowing you to specify the size of the grid squares displayed in the viewport. **Note:** When zooming out, grid lines can merge together when they become too small. When this occurs, Modo converts fixed grids to dynamic grids.
Alignment	
Preferred Plane	When the **Preferred Plane Bias** option is set above 0%, you can choose a preferred **Work Plane**. Based on the bias amount, Modo attempts to keep the work plane positioned at the preferred plane determined by its two major axes.
Preferred Plane Bias	When working with the **Work Plane**, Modo tries to keep the plane perpendicular to the viewport window when navigating and rotating the view. Increasing the **Plane Bias** option increases Modo's favor of the preferred plane. Setting a very high value means Modo almost always favors the preferred plane.
Lock Preferred Plane	You can lock the **Work Plane** to a specific plane-facing direction by enabling this option. Once fixed, rotating the viewport does not affect the position or angle of the **Work Plane**.
Locked Plane Offset	When the **Work Plane** is locked, you can additionally set an offset from the origin, providing accurate control over the position and placement of the **Work Plane** when used as a construction plane.
Plane and Grid Increments	As you zoom in and out of a scene, the **Work Plane** grid adjusts the density of the grid divisions, adding or removing subdivisions based on the plane increments. You can modify this behavior using this option.

Rendering Preferences

The **Rendering** preferences contains all settings that determine the rendering behavior of Modo.

To access these preferences:

- On Windows, open **System** > **Preferences**.
- On Mac OS, open **Menu** > **Preferences**.

Color Management

Default OCIO Color Config	This option controls the default configuration type for new scenes. Each separate configuration controls which color spaces are available for use by items in the scene. Each type contains a different set of profiles. You can also designate a config for individual scenes manually in the Scene Item.
Default 8-bit/16-bit/Float Colorspace	These options designate the scene-default profile assigned when reading and writing the associated bit depth of image for any items that have their colorspace control set to **Default**. You can also designate colorspace profiles individually on the Image Map item in the **Image Still** subtab.
Default Numeric Colorspace	This option allows you to set a default colorspace to use when displaying images. It defines the colormapping assigned to UI elements, including the **Properties Image Still** tab **Color** > **Colorspace**. The default value is set to **sRGB**. The following formats are available: • Linear • sRGB • sRGBf • Rec. 709 • Cineon • Gamma1.8 • Gamma2.2 • Panalog • REDLog • ViperLog

	• AlexaV3LogC • PLogLin • SLog • Raw
Default View Colorspace	These options control the colormapping assigned to user interface elements, including the Color Picker Viewport, Color Swatches, the Render Display window and the interactive Preview Viewport.
Affect Modo Color Swatches and Picker	When this option is enabled, the **Default View Colorspace** option adjusts the display of individual color input fields in the **Properties** form to match the colors of the **Render** display.
Affect System Color Dialog	When this option is enabled, the **Default View Colorspace** option adjusts the display of the **Color Picker** viewport pop-ups and windows to match the colors of the **Render** display.
Apply Color Correction Profile to Legacy LXO files	Enable this to apply the default color correction profile to pre-Modo 801 **.lxo** files. These files pre-date the use of OCIO. Consequently, it may be desirable not to use Modo's default color correction on loading the files as they may already have been corrected using an alternative method. For this reason it is off by default. **Note:** Modo normally initializes scene files if OCIO color configurations are not stored correctly in the file.

Final Rendering

VDBVoxel Cache	
Cache Mode	Fully optional, but can be used to speed up the display of VDB sequences. There are three options: **None** disables the cache functionality, **Fast** stores a limited data structure that allows for faster display, although less accurate, and **Full** stores the entire VDB file in memory for faster access, but requires the most RAM.
Cache Size	Defines the amount of dedicated RAM to be used for the stored data.

VDBVoxel Cache	
Final Rendering	
Automatic Render Threads	When enabled, Modo determines the number of threads automatically based on the number of CPU cores and other factors, such as Hyper-Threading support.
Render Threads	When **Automatic Render Threads** is disabled, you can manually set the number of render threads, which reflect Modo's ability to simultaneously calculate rendering information, each calculation taking place as its own thread. The greater the number of threads, the faster scenes render. However, there is a point of diminishing returns, where overhead of thread management outweighs benefits of the extra threads. Generally speaking, the best performance is obtained when the number of threads equals the number of total cores, virtual or otherwise.
Geometry Cache Size	This value, set as GB (Gigabytes or 1 billion bytes) determines the maximum amount of system RAM Modo uses to parse scene geometry for rendering. When the amount of geometry exceeds this value, an error message is displayed. You can increase this value to render geometry-intensive scenes, such as those using fur and subdivision surfaces with Micro Poly Displacement. For 32-bit systems, this cannot exceed the maximum amount of addressable RAM, apart from that needed by the system itself, typically around 3GB. For 64bit systems, the amount can be much greater, depending on the amount of available system RAM.
Use Network Render Nodes	When enabled, Modo seeks out additional online Modo workers, and use the network attached systems for additional rendering power. For more information on the setup and use of network rendering, see Network Rendering.
Render With Network Nodes Only	When enabled, this setting prevents the master system from rendering any buckets or frames, and manages the process of distributing the data to the worker systems. This still ties up the Modo process on the master system, but reduces CPU overhead there, allowing you to work in other apps while rendering in Modo. **Note:** As the scene being rendered is still loaded on the master system, memory use may still be high, depending on the scene.

VDBVoxel Cache	
Send Whole Frames to Network Render Nodes	When this option is enabled, Modo sends the entire frame to the network workers for rendering; calculating the IC, evaluating the scene and returning a completed frame back to the master system. **Note:** This option only works for saving individual frames, not when rendering to a movie container.
Network Render Group	The **Network Render Group** is an identifier that allows you to pool the resources of like-named systems for network rendering, giving you a degree of control over how the network resources are allocated. For example, if there are 24 systems available on a network for rendering via worker mode, an administrator could give twelve of them the **Network Render Group** name "Klaatu" and the other twelve the group name "Barada". When any network rendering jobs are submitted on a system named "Klaatu", only the 12 systems with the same group name can be utilized for network rendering, leaving the other 12 available for rendering only within the "Barada" group.
Network Job Size	This setting determines the number of buckets distributed to the worker machines. The general rule is to take the total number of worker cores for rendering and multiply it by two, producing double the number of total buckets to be distributed. Depending on the scene makeup and network speed, the default 200% value (2x) may not be optimal for load balancing across the network. If the master machine tends to sit idle, it may help to increase this setting.
Send and Accept Assets for Network Job	In Modo 501 and earlier, network rendering files were propagated by way of a shared network folder, where all workers could access the scene and its assets. You can now enable the **Send and Accepts Assets for Network Job** option and Modo sends the scene and any assets directly to a worker instead, using networks sockets. This avoids the permission issues caused so often by a shared network folder. When enabled, it is not necessary to specify a **Network Shared Directory**.
Network Shared Directory	When using network rendering with the **Send and Accept Assets for Network Job** option disabled, you need to define a folder (directory) that each worker across the network uses to read and write to. Once a folder has been determined, use the **Network Shared Directory** option to specify this

VDBVoxel Cache	
	location.
Discover Render Nodes via Bonjour	With this option enabled, Modo seeks out systems connected to the same network, that are placed into worker mode and use those systems for network rendering.
Discover Render Nodes via Host List	With this option enabled, Modo connects directly to systems listed in the Host List for network rendering. In instances where **Bonjour** fails to connect networked systems, the **Host List** is a way to bypass **Bonjour** and connect to systems directly if the IP address is known.
Default Output Gamma	When adding new render output layers to the Shader Tree, this value determines the default Gamma setting for the item. You can override this value by adjusting the Gamma value within the render output item itself.
Color Picker Stops	This option sets the default **Stops** value for the **Color Picker** viewport. **Stops** are basically an exposure control that allows you to more easily view and select high dynamic range colors.
Affect Color Control Swatches	When this option is enabled, the UI advanced color selectors are affected by the **Color Picker Stops** value.
Default Field Rendering	When rendering animations that require interlaced frames, this option determines the default field rendering state for new scenes. You can adjust the setting individually, per scene within the render item. Most modern broadcast mediums do not require the use of interlaced frames, so unless you are certain you need them it is best to leave this option at the default **Off** value.
Bake UV Border Size	When Baking a texture, or applying the **Expand UV Seams** command, this option determines the amount of expansion around the UV border. The default value of 3 pixels is generally fine in most instances. When generating very high resolution images, or in cases where texture seams are visible, you may benefit from increasing this value.
Max Render Slots	Determines the maximum number of frame buffers (renders) saved on disc in the **Image Browser**. Once this value is exceeded, the oldest render is deleted, unless locked. If all frames are locked, then they are treated as if all are unlocked once the maximum number is met. This option is not affected by images loaded into the browser.

VDBVoxel Cache	
Frame Cache Size	This option controls how much of the rendered image is loaded into memory at once. Rendered images are organized internally as a set of tiles, so with this option, you can adjust the balance between faster redraw speed and lower memory consumption. Setting this option to a value that is smaller than the amount of memory necessary to display the rendered image can result in some tiles being purged when zooming or panning, which can greatly affect redraw speed (depending on storage speed and how many tiles need to be reloaded as the image is drawn in the viewport).
Write Buckets Directory	When you enable the **Write Buckets to Disk** function in the render item's settings, this option determines the specific directory where these files are written. For more information on the **Write Buckets to Disk** function, see Render Item: Frame.
Recent Frames Directory	Modo saves its rendered results in the user-specified **Recent Frames** directory. These rendered image files can be selected, navigated, modified and saved from their proprietary Modo rendering format using the **Render Display** window. Rendered image files are persistent across Modo sessions until you intentionally remove them or the **Max Render Slots** value is achieved.
Default Output Pattern	This option defines the initial **Output Pattern**, an option to automatically name rendered frames in any new render output item that is created. You can override this for each output by editing the value directly in the render output.
Block System Sleep While Rendering	This option, when enabled, blocks the system from entering low power usage **Sleep Mode** when rendering. When disabled, Modo may stop functioning properly when the system enters **Sleep Mode** while rendering.
Auto Add	This option defines the initial default setting for the **Auto Add** option of the Render Passes functionality.

Host List

When using the Network Rendering function, the **Host List** is the list of possible target computer systems that can be used for rendering (The **Host List** option must be enabled in order to define hosts). This system can bypass the traditional **Bonjour** automatic node discovery, allowing you to manually enter a system's IP address and test it. This is done by finding the target system's IP address using the computer's networking tools. With the IP information in hand, use the **Add Host** button to open a dialog where the **Host Name** or

IP address is entered and then press **OK**. Once you have entered a host's address, yo can test it with the **Test Selected** button, or test all addresses with the **Test Hosts** button. For any successful connections, you can use the **Connect Selected** button for systems that don't connect automatically. For more information on Network Rendering setup, see Network Rendering.

Preview

Settings	
Use Secondary Irradiance Cache	The Secondary Irradiance Cache is used to increase the quality of the preview render, but it takes additional cycles to calculate the additional irradiance values and therefore slows down the preview. If you want to use preview like a final render tool and get the additional IC passes like you get when pressing **F9** to generate a final render, this option enables it. Combined with **Extended Refinement Passes**, it acts more like progressive rendering engines, where Modo continues to improve the image quality as long as you allow it. Disable this option when you want the preview to be more responsive.
Extended Refinement Passes	This value determines the maximum number of rays when using the **Extended Refinement Passes** option for **RayGL** and **Preview** quality. Once Modo arrives at the number of samples as determined by the item settings, it continues firing rays for IC, MC, Reflections, Refractions, Transparency and Subsurface Scattering up to the number of rays set here. This option allows for interactive style rendering when using the **Preview** viewport.
Bucket Order	This option determines the update order of buckets. There are two options you can choose from. • **Semi-Random** - Updates buckets in a semi-random order, providing a more iterative update to the render. • **Scanline** - Updates the buckets in a left to right, top to bottom order, producing a more regular update pattern.
Start Paused	This option determines the initial state when a Preview viewport is opened. In some cases with complex scenes loaded, switching to a workspace with a Preview window, users might experience a temporary freeze as the scene is evaluated for Preview rendering, where Modo will hang. Starting Preview in the Paused state will eliminate this often unexpected hiccup in performance.
Preview Thumbnail	Determines the default rendering style for preview thumbnail icons when saving preset materials.

Settings	
Render Subdivison Level	
Use Subdivs Render Level	In a mesh item's properties, you can define a separate render level for Subdivision Surface geometry. This is a global override option to disable or enable this functions. The option must be enabled here for the associated function to work. When disabled, this value overrides the settings of the Mesh Item.
Use Catmull-Clark Render Level	In a mesh item's properties, you can define a separate render level for Catmull-Clark subdivisions. This is a global override option to disable or enable this functions. The option must be enabled here for the associated function to work. When disabled, this value overrides the settings of the Mesh Item.
Use Cage Mode	An option to disable the display and rendering of the subdivisions altogether without affecting their settings.
Render Proxy	
Render Proxy Items (Initial Value)	Proxy items are a means to save individual item layers to external files, making the source scene file smaller. In order for proxy items to render, they need to be loaded into memory, which may slow the scene's performance down, depending on the scene size and other factors. By default, the **Render Proxy Items** option is enabled, so proxies are rendered. you can set the **Render Proxy Items (Initial Value)** to disabled, meaning that preview does not render proxies by default. This can also be changed directly in the **Preview** viewport under **Options**. For more information on proxy items, see Render Proxy.

File Commands

Modo provides all the standard controls necessary for opening, closing and managing files to and from disk. You can perform most options from operating system-specific dialog boxes. File management is largely accomplished through the commands in the menu bar under the **File** menu. Click on a menu command to perform the operation.

Menu Item	Description
New	Creates a new empty scene file and adds it to the Items (Scenes) List. By default, the scene is named **Untitled**. If other currently opened scenes are also named **Untitled**, Modo appends a number to the name, for example, **Untitled (2)**. Use **Save As** to name the scene.
Open	Loads any previously saved Modo scene files, as well as variety of other 3D object and image file formats. Scene files are added to the **Items** list, as are other 3D object formats. Images are added to the **Images** list of the currently active scene file.
Open Recent	Directly opens a file from a list of files previously opened or saved files. Choose the appropriate file name from the list. **Note:** If you have moved the file to a different location and then you try to open it from **Open Recent**, an error message displays, letting you know that the file is no longer at that location. In this case, use the **Open** command and navigate to the file's new location. The list of recent files are stored in the config file, **MODOXXX.cfg**. If you delete this file, your list of recent files also disappears. **Note:** A backup of configuration file is created at shutdown, when changes are saved. This backup file can be renamed to restore the previous state if configuration changes cause issues. When a corrupt configuration is detected, it is renamed, instead of deleted, which allows you to recover data from corrupted configuration file. Configuration files are found in the following locations: Windows: **C:\Users\**<user name> **\AppData\Roaming\Luxology** Mac: **/Users/qa/Library/Preferences/** Linux: **/home/**<username>**/.luxology**
Import	Appends a saved file into the current scene. Items are added to the bottom of a scene in the **Item List** into their own subgroup with their names corresponding to the imported scene or object.

Menu Item	Description
Close Window	Dismisses the current floating window viewport.
Close Scene	Closes the currently active scene. If the file was not recently saved, Modo asks you to save it. If the closed scene was the only one currently in the Item List, Modo automatically creates a new blank scene.
Close All	Closes all currently open items in Modo, including images and scene files. Modo defaults to a new empty scene. If any scene or image file has changed since the last save, a pop-up dialog prompts you to save recent changes.
Save	Saves the currently active scene. An operating system-specific dialog opens if the file has not been previously saved, requesting a file name as well as a location. You can also specify a format. The native format for Modo is **.lxo**. This should be used for files requiring further refinement in Modo later. For more information on other formats, see Importing and Exporting File Formats.
Save As	Opens a dialog, allowing you to specify a file name, format and location for the currently active scene file. The native format for Modo is **.lxo**. This should be used for files requiring further refinement in Modo later.
Save All	Saves all currently opened scene and image files in one step. For scenes not previously saved, a dialog pops up where you can specify a file name, format and path.
Save Incremental	Saves a version of the file with a numeric value appended to the end of the file name. You can save iterations of a scene's progress this way. Each iteration is saved as a new file with the numerical value incremented by one. Once saved, the file becomes the active file and all future **Save** commands save to this increment.
Export As	Allows you to save your work in different formats. The options include LightWave **.lwo**, **.x3d**, Collada **.dae**, **.dxf**, **.obj** or **.fbx** file formats. This is different than using **Save As** because it doesn't change the current setting of the file as it is inside Modo. This means that when you save a model in a particular format, Modo remembers it for future saves. Export simply allows you to quickly export in different formats without changing the internal file.
Revert	Removes the currently active scene from memory and loads in the last saved version, abandoning all subsequent changes made since the last time the file was saved. A dialog prompts you to confirm this procedure, because it can't be undone. Selecting **Yes** loads in the previous version, and selecting **No** cancels the

Menu Item	Description
	procedure.
Reset	Closes any currently open scene, prompting you to save if necessary, and then restores the user interface to the same state as when Modo was previously opened. Since exiting Modo prompts the application to write any changes in memory to the user configuration file, this command is useful for when a number of undesired changes have been inadvertently applied to the interface and you wish to clear them from memory.
Consolidate Scene	Saves a copy of the current scene in its current location, and copies all associated linked files, such as imported images and **.mdd** files to a sub-folder of the save location. The duplicate file has the text **_bundled** added to its file name. **Consolidate Scene** also adjusts the internal file link locations to relative paths. This aids you in transferring files between workstations, ensuring that all the associated files are transferred with the base scene. To share your scene, compress the bundled file and the **imported_images** folder.

Edit Commands

You can undo and redo changes, cut, copy, paste and delete elements in the **Edit** menu.

Undo/Redo

Undo steps back one step at a time through the sequence of previous revertible operations. All commands that change model state can be undone, except for special exceptions which make changes too large to be undone. Model commands that cannot be undone display a warning, which must be dismissed. Commands that change only the user interface state cannot be undone. You can view a list of irrevertible commands in the Command History viewport.

To **Undo** a step, press **Ctrl**/**Cmd**+**Z**, or in the menu bat, select **Edit** > **Undo**.

To **Redo** a step, press **Ctrl**/**Cmd**+**Shift**+**Z**) or in the menu bat, select **Edit** > **Redo**.

Note: The maximum number of revertible steps can be specified in the Preferences.

Cut, Copy, and Paste

To cut, copy, or paste an element, make a selection and then in the menu bar, click the appropriate command under **Edit**, or use the associated keyboard shortcut:

- **Cut**: (**Ctrl/Cmd**+**X**) Removes the selected element and stores it in the clipboard.
- **Copy**: (**Ctrl/Cmd**+**C**) Copies the selected element, leaving the original in place and stores it in the clipboard.
- **Paste**: (**Ctrl/Cmd**+**V**) Inserts the contents of the clipboard into the target layer or field.

Tip: When managing large scene files, or files with multiple layers, the **Import** option may be more easier to transfer multiple layers simultaneously.

Delete/Remove

Delete (**Del**) applies to any type of selected data. For geometric data this is exactly like **Cut**, removing the data except that it is not moved to the clipboard. Deletion of vertices removes the vertices, but leaves any remaining vertices or polygons. If all the vertices of a polygon are deleted, then the polygon itself is removed. Deletion of polygons also deletes the associated vertices if they are not needed for any other polygons that stays behind. Deleting an edge removes any adjoining polygons leaving a hole. When in **Items mode**, deleting an item removes the layer, including all geometry contained within from the **Items** list.

Remove (**Backspace**) is effectively the same function as **Delete**, but when working with edges, removing them eliminates the edge itself, but leaves the neighboring polygons in place, resulting in a polygon merge, joining the neighboring polygons.

Project Directories

Modo supports the use of project directories, a hierarchical arrangement of folders for related files, meant to ease the management and transfer of complex projects. By using the project directory commands in Modo, you can move projects freely from system to system, and platform to platform without fear of missing files or lost link associations, provided that you adhere to the generated directory structure. The setup and use of Project directories has been designed to be as transparent and intuitive as possible within Modo. The basic methodology for project directories is to first define the project and its folder structure with the **New Project** command before saving a scene. Then, when saving the scene, Modo uses the **Scenes** folder within the project by default, images are saved to the **Images** folder, and so on. To load a previously saved project, you only need to specify the project directory using the **Choose Project** command, and then open any files requiring further editing or rendering. Modo bases the default paths on those specified by the project file.

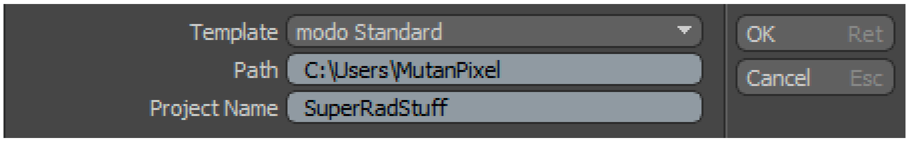
New Project

To work with a project directory, you first need to generate the directory structure.

1. To create a new project, in the menu bar, click **File** > **New Project...**.

   A file browser opens, where you can select the project folder.
2. Specify the location and click **OK**.

   The **Instantiate Project Template** dialog opens.



Template: modo Standard

Path: C:\Users\MutanPixel

Project Name: SuperRadStuff

3. Modify the **Path**, if necessary (this is the path as specified in the previous window) and set the **Template** and **Project Name**. The project name becomes the root project directory folder. Click **OK**.

   This generates the default folders within the specified location, and sets Modo to use the new created structure. Once a Project is specified, Modo's **Save As** dialog opens the appropriate project directory folder by default.

Choose Project

To open or utilize a previous or transferred Modo project, you only need to place the files into the appropriate location on your hard drive.

1. In Modo, use the menu bar command **File** > **Choose Project**.

   This opens a file browser for selecting the project.
2. Navigate to the saved project and select the root folder, then click **OK** to set the project.

   From now on, when opening any project files, Modo adheres to the paths specified in the **luxproject config** file automatically saved into the project directory.

Edit Commands

Undo/Redo

You can reverse an action by using the **Undo** command. **Undo** steps back one step at a time through the sequence of previous revertible operations. All commands that change model state can be undone, except for special exceptions which make changes too large to be undone. Model commands that cannot be undone display a warning, which must be dismissed. Commands that change only the user interface state cannot be undone. They can be mixed with model commands, but they do not undo or redo as you steps backwards and forwards through prior actions. You can view a list of irrevertible commands in the Command History viewport.

Note: The maximum number of revertible steps can be specified in the Preferences.

To **Undo** a step, press **Ctrl**/**Cmd**+**Z**, or in the menu bat, select **Edit** > **Undo**.

To **Redo** a step, press **Ctrl**/**Cmd**+**Shift**+**Z**) or in the menu bat, select **Edit** > **Redo**.

Clipboard

Modo provides a special memory buffer for temporarily holding data referred to as the **clipboard**. A clipboard is a useful function when moving data around from layer to layer and scene to scene. Modo adheres to the standard **Cut**, **Copy** and **Paste** methodologies universal to nearly all Windows and Macintosh applications. You can use the commands to manipulate text or numerical values when working in data input fields, or with geometry when working in the GL viewports.

Elements in the clipboard cannot be saved directly and are cleared from memory when Modo quits. Text and numerical data is not application-specific and can travel in and out of Modo. Any geometric data stored in the clipboard is relevant only to Modo. This is because each data type has its own buffer. Alphanumeric characters are copied to the system clipboard, while geometric data uses a Modo-specific clipboard.

The Modo geometry clipboard only stores component geometry data. Other item types such as Cameras and Lights cannot be stored in the clipboard. Whatever is selected and cut/copied is what gets pasted except with edges; when cut/copy/pasting edges the result is polylines (two point polygons) and not true edges because edges cannot exist without a polygon. These polylines can be selected in both the **Edges** and **Polygons Selection modes**. When selecting elements in **Items** mode, cut/copy/paste works on all component types at once and paste them in the same state as how they were cut/copied. The clipboard support all associated Vertex Maps and tags as well, so cut/copy/paste retains selection sets, material/part tags and when pasted into a different scene, related material surfacing comes along for the ride as well.

To use, user need only to make a selection of some type and then invoke the command from the menu bar under **Edit**, or by using the associated keyboard shortcut.

Cut: (**Ctrl**/**Cmd**+**X**) This option removes the selected element and stores it in the clipboard.

Copy: (**Ctrl**/**Cmd**+**C**) This option copies the selected element, leaving the original in place and stores it in the clipboard.

Paste: (**Ctrl**/**Cmd**+**V**) This option inserts the contents of the clipboard into the target layer or field.

Tip: When managing large scene files, or files with multiple layers, the 'Import' option may be more easier to transfer multiple layers simultaneously.

Delete

The Delete command applies to any type of selected data. For geometric data this is exactly like the cut command removing the data except that it is not moved to the clipboard. Deletion of vertices removes the vertices, but leaves any remaining vertices/polygons (if applicable). If all the vertices of a polygon are deleted, then the polygon itself is removed. Deletion of polygons also deletes the associated vertices if they are not needed for any other polygons that stays behind. Deleting an edge removes any adjoining polygons leaving a hole. When in **Items mode**, deleting an item removes the layer, including all geometry contained within from the Item List. When text is selected in an input field, delete removes the selected section.

Delete: (**Delete** key) Deletes the selected element from the scene entirely.

Remove

Remove is effectively the same function as Delete, however the one notable difference is that when working with edges, removing them eliminates the edge itself, but leaves the neighboring polygons in place essentially resulting in a polygon merge, joining the neighboring polygons.

Remove: (**Backspace** key) Removes the selected element from the scene.

Import Reference

The **Import Reference** option places elements of a saves scene file into a new scene, similarly to a regular import action. However, with referencing, a link is created between the two scenes. If the referenced scene is modified in any way, for example, an item is added or its surfacing is changed, the referenced elements automatically update in the master scene upon re-opening, reflecting the changes made. You can also initiate the update from the menu, without closing and opening the scene.

There are a number of reasons where this type of workflow is advantageous, such as building up complex scenes from many simple elements, making it easier to update individual elements without the overhead of the entire scene. Referencing also simplifies the update process, as one reference scene can be placed into many master scenes. Updates then only need to be applied to the single file. It is also beneficial when working with groups, where each user has a specific task that they work on, and a technical director combines final elements for rendering. Keeping elements as separate scene files also allows flexibility for later asset reuse. However it is used, referencing saves time and makes using Modo easier.

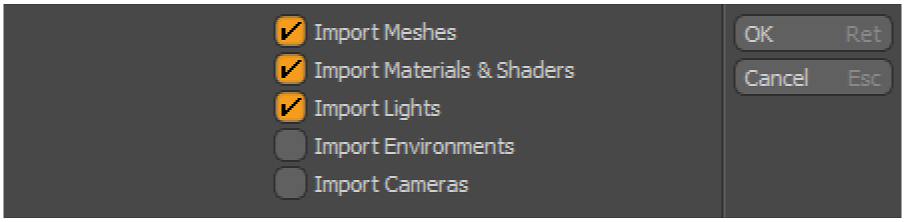
Using Import Reference

Any saved scene file can be imported as a reference, including other formats, such as **.obj** and **.fbx**. Files are loaded through the **Import Reference** command found in the menu bar.

1. Click on **File** > **Import Reference...**.

   An OS-specific dialog opens, where the saved scene file is located.

2. Select the file and click **Open** to open a dialog box, where you can specify the elements to import.



☑ Import Meshes
☑ Import Materials & Shaders
☑ Import Lights
☐ Import Environments
☐ Import Cameras

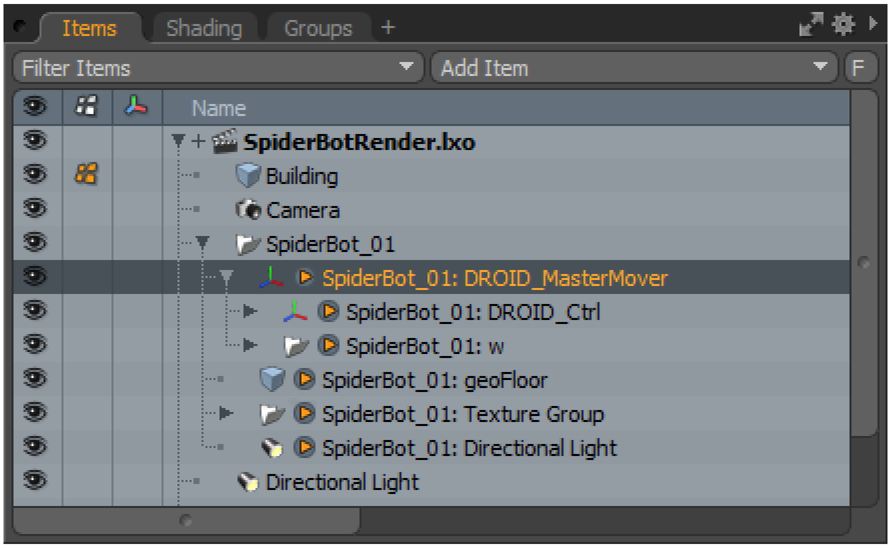
OK Ret
Cancel Esc

3. In this dialog, check the box next to the items you would like to import into the master scene.

   By default, **Import Meshes**, **Import Materials & Shaders** and **Import Lights** are enabled.

4. Click **OK** to start the import.

   Aside from the designated elements, any associated scene items are also imported, including locators, all necessary Shader Tree settings, like materials, shaders, light settings and environment settings, Library items, deformers, assemblies, bitmap images, sound files, dynamic items and particle simulations. All of the imported elements become visible in the 3D Viewports as first class citizens of the scene and appear in their designated lists, placed under an appropriate grouping element named after the referenced scene. You can open the groups to access and select the referenced items. References are clearly identified in the lists by a preceding small arrow ▶ icon and the referenced scene's name appended to the item layer name.



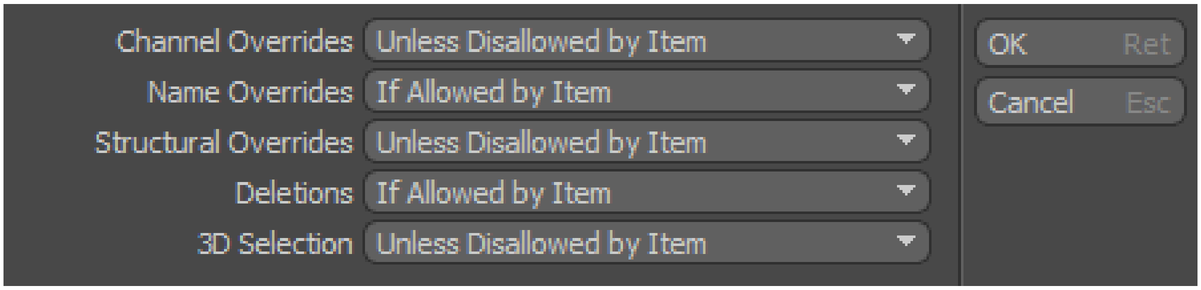
Items
Shading
Groups
+
Filter Items
Add Item
F
Name
SpiderBotRender.lxo
Building
Camera
SpiderBot_01
SpiderBot_01: DROID_MasterMover
SpiderBot_01: DROID_Ctrl
SpiderBot_01: w
SpiderBot_01: geoFloor
SpiderBot_01: Texture Group
SpiderBot_01: Directional Light
Directional Light

Updates and Overrides

Once a scene is imported as a reference, you can select, transform and render any of the imported elements normally. All referenced items appear as regular items within the master scene. When it is necessary to modify the referenced elements outside of standard transforms (position, rotate and scale), there are two ways to go about making the edits that depend on the type of editing needed and the desired results.

For changes that should specifically remain an integral part of the referenced scene (if necessary, so they can be propagated to other master scenes) or changes to the geometry or surfacing boundary, you should always edit the original source scene file. Locate and open the original referenced file, make the desired edits and save the file. The next time any master scene that contains that reference (also referred to as a sub-scene) is opened, the changes are automatically reflected in the referenced items. To manually update referenced elements within an open scene, use the command in the menu bar under **Item** > **References** > **Reload Imported References**.

The second way to update a referenced item is using an override. An override means any edit that is assigned to a referenced item within the master scene. Overrides are saved as part of the master scene itself and are not reflected in the original source reference file. These types of edits include channel value changes, structure changes, such as hierarchy and parenting changes, item deletions and item name changes. The ability to override channel values is governed by the **Reference Override** in Data Preferences, which is scene-specific. To open the **Reference Override Settings** dialog box, go to the menu bar and click **Item** > **References** > **Edit Item Overrides**.


Channel Overrides: Unless Disallowed by Item
Name Overrides: If Allowed by Item
Structural Overrides: Unless Disallowed by Item
Deletions: If Allowed by Item
3D Selection: Unless Disallowed by Item

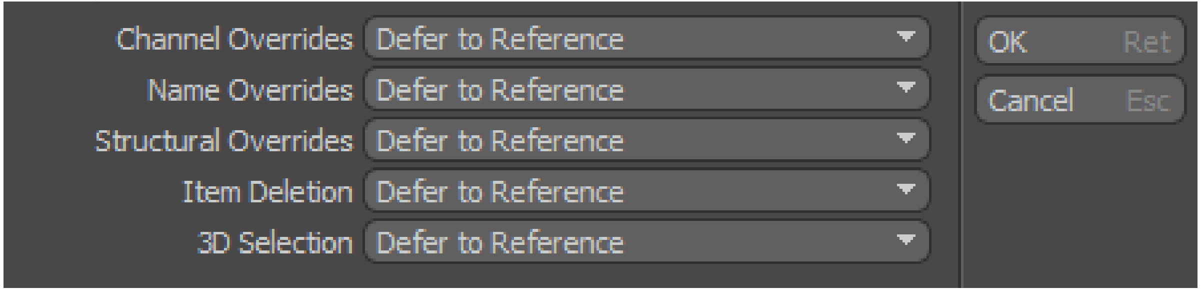
The panel, shown here in its default state, lists the types of override operations with a control to determine whether the operation is allowed. The following operations are available:

- **Channel Overrides**: changes made to any of the associated attributes of the item itself, such as modifications to the transform values or item-specific changes.
- **Name Overrides**: changes to the referenced item's names (overrides are tracked by an internal ID assigned when the items are imported).
- **Structural Overrides**: changes to the hierarchies of the referenced elements, meaning modifications to their ordering in parent child relationships or their positions within the **Items** lists.
- **Deletions**: control whether referenced items can be removed from the master scene.
- **3D Selections**: controls the selectability of the referenced elements in the 3D Viewports.

Each of these operations have four possible states governing them:

- **Always Allowed**: override changes can always be made of the designated type.
- **Never Allowed**: overrides cannot be assigned, editing of the referenced items is locked.
- **If Allowed by Item**: overrides cannot be applied unless explicitly set as **Enabled** by the reference item setting.
- **Unless Disallowed by Item**: Overrides can always be applied unless explicitly set as **Disabled** by the reference item setting.

The latter two settings are defined directly on the items, either within their source scenes or as referenced items within the master scene. This is assigned by selecting the target element to control, and then selecting the menu bar command **Items** > **References** > **Item Override Settings**, which opens the **Item Override Settings** dialog box.



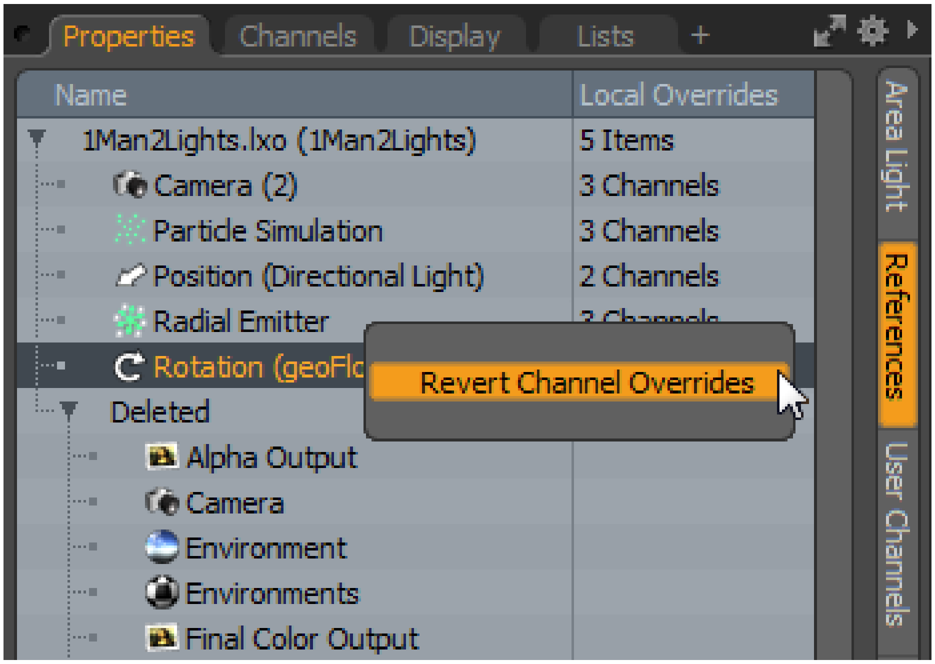
Channel Overrides: Defer to Reference
Name Overrides: Defer to Reference
Structural Overrides: Defer to Reference
Item Deletion: Defer to Reference
3D Selection: Defer to Reference

The item override settings are defined per item and apply to any selected items when the menu is invoked, and saved with the item until explicitly changed. These settings provide a finer degree of control over the editability of the items when referenced in another scene (based on the operation types, explained above). The control states are:

- **Enabled**: overrides can always be made unless set otherwise in the master scene.
- **Disabled**: overrides cannot be made unless the setting allowing it is explicitly set in the master scene.
- **Defer to Reference**: override assignments are controlled exclusively by the **Reference Override Settings** in the master scene.

Tip: If you need different default reference override settings are desired, you can modify the **Reference** preferences setting to control the initial state of overrides in a scene.

With the settings that govern how overrides are controlled, applying overrides is a rather simple process. As a matter of fact, overrides are just normal edits made when the item is selected, either by modifying an attribute in the **Properties** panel, or by modifying any values in the **Channels** viewport, if allowed. When not allowed, the edit fields show up as dimmed input fields that cannot be selected. Name and structure overrides are assigned normally when changing the layer order of the referenced elements or by renaming the item in the normal way.

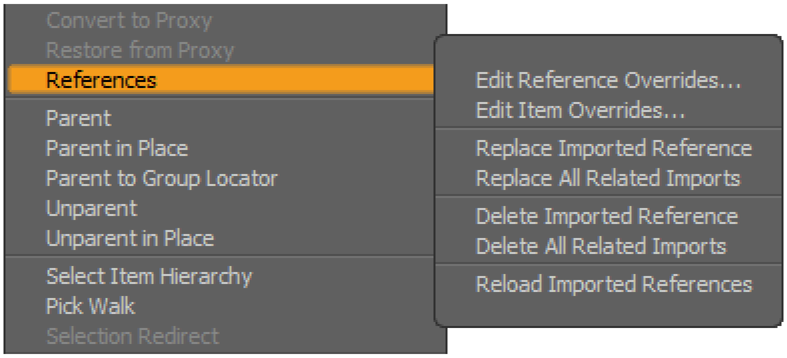

Properties
Channels
Display
Lists
+
Area Light
References
User Channels
Name
Local Overrides
1Man2Lights.lxo (1Man2Lights)
5 Items
Camera (2)
3 Channels
Particle Simulation
3 Channels
Position (Directional Light)
2 Channels
Radial Emitter
Rotation (geoFlo
Revert Channel Overrides
Deleted
Alpha Output
Camera
Environment
Environments
Final Color Output

All overridden values are stored in a separate action state within the master scene. If you need to remove an override, you can revert the settings to those of the source file in the **References** section of the **Properties** panel, appearing as a sub-tab whenever any referenced item is selected in the **Items** list (it doesn't matter which item is selected, as long as it is a reference). The panel itself displays a list of all the referenced elements within the scene and denoted how many modifications have been assigned and of which type.

Within the panel, you can right-click over a selected element to open a contextual menu that allows them to **Revert Channel Overrides**. Selecting the option removes the edits.

Note: Structure and name overrides cannot be restored this way.

References Contextual Menu


Convert to Proxy
Restore from Proxy
References
Parent
Parent in Place
Parent to Group Locator
Unparent
Unparent in Place
Select Item Hierarchy
Pick Walk
Selection Redirect
Edit Reference Overrides...
Edit Item Overrides...
Replace Imported Reference
Replace All Related Imports
Delete Imported Reference
Delete All Related Imports
Reload Imported References

There are several options related to references, in the menu bar under **Item** > **References**. The same options are also available in the **Item** contextual menu by right-clicking over the target element in the **Items** list and selecting the **References** section from the pop-up menu. The following commands are available:

Edit Reference Overrides...: opens the scene-wide reference override settings panel, explained in Updates and Overrides.

Edit Item Overrides...: opens the item-specific reference override settings panel.

Replace Imported Reference/Replace All Related Imports: these two options allow you to easily replace a referenced scene while retaining any override settings, if possible. Selecting either option opens an OS specific dialog that allows you to select an alternate file to be imported to replace the desired elements. Once selected, click **Open** to display a dialog box allowing you to suppress different items from the new reference source scene.

The difference between the two commands is the **All Related** option of this panel. When enabled, all the associated items within the same sub-scene are replaced by the new import. When disabled, only the selected elements are replaced. If overrides have been applied to the referenced items in the scene, when the new items are loaded, Modo tries to apply the same settings to the corresponding items, resolved by item name. If matching names cannot be found, the overrides are discarded.

Delete Imported Reference/Delete All Related Imports: These options allow you to remove referenced items one by one, or by all associated items within the same sub-scene.

Reload Imported References: This option forces Modo to reload the sub-scene into an open Master Scene, based on the sub-scene associated to the selected referenced item when the command is invoked. This picks up any changes made to the referenced sub-scene, if they were make externally while the master scene was opened. Users should note that referenced sub-scenes cannot be loaded while a master scene that contains that reference is open.

Note: Referenced items cannot be duplicated or instanced by standard means in the master scene. If there is a need to have more than a single item from any referenced scene, it's best to use the **Import Reference** option again to create the additional copy.

General Notes for Working with References

A referenced source scene cannot be opened at the same time as its master scene. You must first close any master scenes that contain the reference before opening them for editing. However, in a workgroup situation, it is possible for a referenced scene, saved to a network, to be edited on another machine and then saved back to the network. The **Reload Imported References** command triggers the update to the master file.

Items in Modo do not have transform channels by default (position, rotate, scale). They are added automatically when a transform is applied or the channels are explicitly added to the item. In order for referenced item transforms to be overridden in a master scene, the channels must first be present on the item when referenced, otherwise any animation on the item in the master scene is treated as a local transform.

Geometry component edits such as adding new polygons to an object cannot be assigned as an override. Geometry changes must be applied in the original source scene.

When overrides are applied to an item in the Master scene that is then deleted in the source scene, the override values are discarded, and are no longer saved with the Master scene.

Due to the way that Modo searches for links when loading files (using relative paths, for example, relative to their save location), referenced files with identical names, but in different folders may load incorrectly. To avoid such issues, it is highly recommended to give all files unique names.

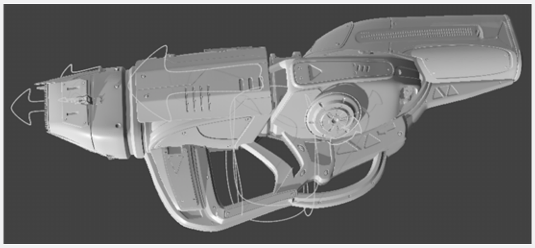
Certain global settings cannot be imported when files are referenced, like preference settings, scene-specific settings (such as FPS and scene length, and color management; although setting on individual items are retained), render outputs, or the settings of the render item itself, such as the frame size and resolution.

Importing and Exporting File Formats

Artists rarely work in a single application, so it is important to support multiple input and output formats. Modo supports a wide range of formats for transferring data. For more information about these formats, see Supported Formats.

Note: Modo's default unit is 1 m, whereas some applications, such as Maya, Houdini, or Nuke use 1 cm as default. This can cause problems with size when exporting/importing files from Modo to other applications. For example, a 1 m unit cube exported from Modo displays as a 1 cm unit cube in another application, whereas importing a 1 m unit cube into Modo from another application results in a 100 m cube. You can use the **Scale** option in the **Preferences** > **File I/O** > **Alembic I/O** to resolve this. See Alembic I/O.

You can read the various formats by opening them directly by clicking **File** > **Open** and selecting the proper file or by importing it by clicking **File** > **Import...**. The difference is that opening files makes them their own scene, while importing merges the saved file into the current scene. The following **Open** and **Import** formats are available:

Format	Extension	Notes
LightWave Object (LWO2)	*.**lwo**	
LightWave Object (LWOB)	*.**lwo**	
Alembic Format	*.**abc**	
COLLADA 1.4.1	*.**dae**	
Autodesk DXF	*.**dxf**	
Adobe Illustrator EPS	*.**eps** *.**ai**	
Protein Database	*.**pdb**	
Scalable Vector Graphics	*.**svg**	
McNeel Rhino 3DM	*.**3dm**	See Work with CAD Assets Inside Modo for more information.
Pixar Universal Scene Description USD	*.**usd** *.**usda** *.**usdc** *.**usdz**  *Model courtesy of Aaron Sims Creative*	During import, if the up-axis of the incoming scene does not match the up-axis of Modo, the incoming scene is transformed to match Modo's up-axis. If a conversion occurs, a message is logged in **System** > **Advanced** > **Event Log**. During import, distance units inside the USD file are converted based on the metersPerUnit value of the USD file. You can disable this behaviour inside **Preferences**

		> **USD I/O**. For more information about Pixar USD file preferences, see File I/O Preferences. Actor, Blob, Camera, Curve, Light, Mask, Material, Texture, Texture Locator, Locator and Mesh items are exported to USD as custom data on their respective primitive types. The data is preserved within Modo on import. **Tip:** Modo's USD import functions also support UV tiling **Real World Size** information in the Texture Locator tool's properties viewport. See Texture Locator for more information.
Autodesk FBX	*.**fbx**	
Sterolithography STL	*.**stl**	
VideoScape GEO	*.**geo**	
SolidWorks Model	***.sldprt *.stdasm**	
Wavefront OBJ	*.**obj**	
X-Rite AxF	*.**axf**	

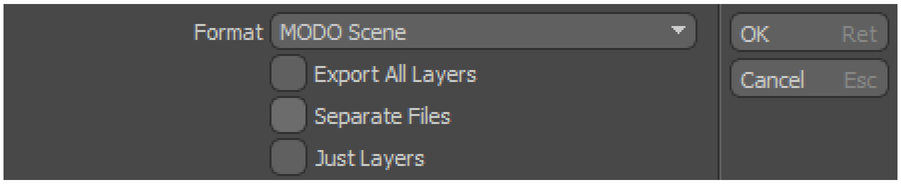
You can save a scene to a specific format with the **File** > **Save As** or the **Export As** command. The difference between them is that saving a file in a specific format replaces whatever is in memory, and subsequent saves continue to overwrite the saved file in the same format, while exporting keeps the same scene in memory, only writing the selected format to disk. The following **Save As** and **Export** formats are available:

Save As	**Export**
LightWave Object (*.**lwo**)	LightWave Object (*.**lwo**)
Alembic Format (*.**abc**)	Web3D Standard X3D (*.**x3d**)
Alembic Format (HDF) (*.**abc**)	Alembic Format (*.**abc**)
COLLADA 1.4.1 (*.**dae**)	Alembic Format (HDF) (*.**abc**)
Autodesk DXF (*.**dxf**)	COLLADA 1.4.1 (*.**dae**)
McNeel Rhino 3DM (*.**3dm**)	Autodesk DXF (*.**dxf**)
Pixar USD .usd (*.**usd**)	Pixar USD .usd (*.**usd**)
Pixar USD .usda (*.**usda**)	Pixar USD .usda (*.**usda**)
Pixar USD .usdc (*.**usdc**)	Pixar USD .usdc (*.**usdc**)
Pixar USD .usdz (*.**usdz**)	Pixar USD .usdz (*.**usdz**)
Autodesk FBX 2015 (*.**fbx**)	HPGL Plotter File (*.**plt**)
Autodesk FBX 2013 (*.**fbx**)	Layered PSD (*.**psd**)
VideoScape GEO (*.**GEO**)	SVG: Scalable Vector Graphics (*.**svg**)
glTF V2 (GL Transmission Format) (*.**glb**)	glTF V2 (GL Transmission Format) (*.**glb**) glTF V2 (GL Transmission Format) (***.gltf**) and (***.bin**)
Wavefront OBJ (*.**obj**)	McNeel Rhino 3DM (*.**3dm**)

Tip: Modo's USD import functions also support UV tiling **Real World Size** information in the Texture Locator tool's properties viewport. See Texture Locator for more information.

	Autodesk FBX 2015 (*.**fbx**)
	Autodesk FBX 2013 (*.**fbx**)
	Stereolithography STL (*.**STL**)
	VideoScape GEO (*.**GEO**)
	Wavefront OBJ (*.**obj**)

Additionally, you can right-click on any layer in the **Item List** and choose the **Export Selected Layers** option from the contextual menu for control over saving individual parts of a scene.



Format MODO Scene

☐ Export All Layers

☐ Separate Files

☐ Just Layers

Option	**Description**
Format	Specifies the format for the exported layers.
Export all Layers	Exports all layers in the scene.
Separate Files	Exports each selected layer to a separate file.
Just Layers	Exports only the mesh layers. No default items, such as cameras and lights, are exported.

Exporting Pixar USD Files

Modo supports the import and export of Pixar's Universal Scene Description (**.usd**) files. The following Modo item types are supported:

- cameras,
- group locators,
- instances,
- meshes,
- replicators,
- all Modo light types except Portal and Mesh lights,

- the Principled material type,
- Blob volumes.

Tip: Modo's USD export functions also support **Real World Size** information in Texture Locator tool properties viewport. See Texture Locator for more information.

The USD exporter also supports joint-based animations. Due to USD requirements, you may have to make the following changes to your models to ensure the animations work as expected in other USD supported applications:

- In order to create a USD supported skeleton, you need an actor which contains the joint hierarchy. See Creating an Actor.
- In order to create a USD supported animation, you need an action, which contains keyframes of the entire timeline for the actor. See Creating Actions.
- Joints that affect a mesh must use a General Influence. See General Influence.
- General Influences must be ordered by hierarchy in the Normalizing folder, and match the order in the actor group. See Normalization.

Note: Modo ships with the 20.02 release of the USD plug-in, therefore USD Python bindings are not exposed and cannot be accessed by users.

To export a **.usd** file out of Modo:

1. Select **File** > **Export As**.

   This opens the **Export As** dialog.
2. Enter a name into the **File name** field.
3. Select **Pixar USD .usd** from the **Save as type** dropdown.

The exporter offers three other formats:

- **.usda** - a plain text ASCII file format,
- **.usdc** - a binary encoded file format,
- **.usdz** - a binary encoded zip-compressed file containing **.usd** files, image files, and audio files, making it easier to transfer your assets between machines.

See the Universal Scene Description documentation for more information on each of the supported file formats.

Note: Hidden items in the Items list are exported with the **.usd** file. If you don't want items to appear in the **.usd** file, you must remove them from the **Items** list in the **Model** layout, or **Scenes** list in the **Modo** layout.

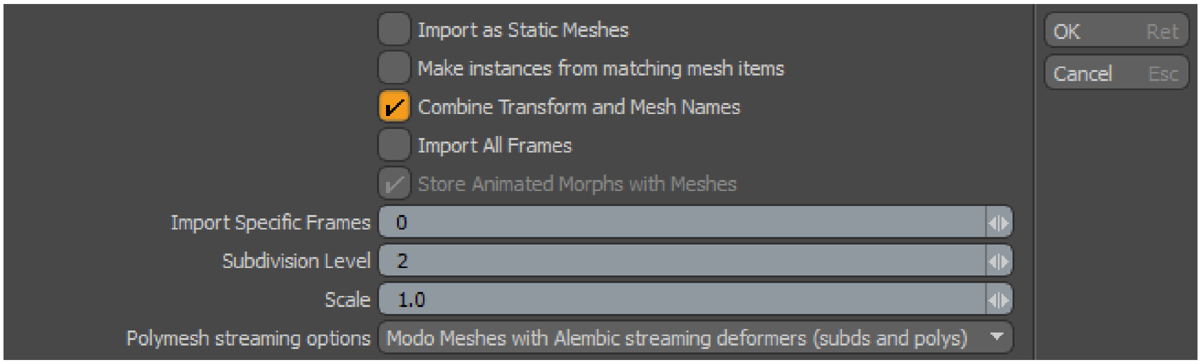
Importing and Exporting Alembic Files

Modo provides a method of importing and exporting particle systems and geometry. The following additional attributes are imported and exported behind the Modo scene:

- **age** - A float value for each particle containing the particle age information.
- **mass** - A float value for each particle containing the particle mass information.
- **force** - A 3D float vector for each particle containing particle force information.
- **modoParticleId** - A float value for each particle containing native Modo particle ID in a range from zero to one.
- **xform** - A 3 x 3 matrix for each particle containing rotation and scale information.
- **dissolve** - A float value for each particle containing the dissolve coefficient information.
- **pathLength** - a float value for each particle containing the path length information.
- **rate** - float value, per-particle, containing particle rate
- **angVel** - A 3D float vector for each particle containing the particle's angular velocity information.
- **torque** -A 3D float vector for each particle containing the particle's torque information.
- **posPrev** - A 3D float vector for each particle containing the particle's previous frame position information.
- **luminosity** - a float value for each particle containing the particle luminosity information.
- **rgb** - A 3D float vector for each particle containing the particle's color value information.

Importing Alembic Files

Importing Alembic geometry into Modo is done by clicking **File** > **Open** and selecting the appropriate ***.abc** file. Once selected, the **Alembic Load Options** dialog displays.



☐ Import as Static Meshes
☐ Make instances from matching mesh items
☑ Combine Transform and Mesh Names
☐ Import All Frames
☑ Store Animated Morphs with Meshes
Import Specific Frames: 0
Subdivision Level: 2
Scale: 1.0
Polymesh streaming options: Modo Meshes with Alembic streaming deformers (subds and polys)

The following import options are available for **Alembic** files:

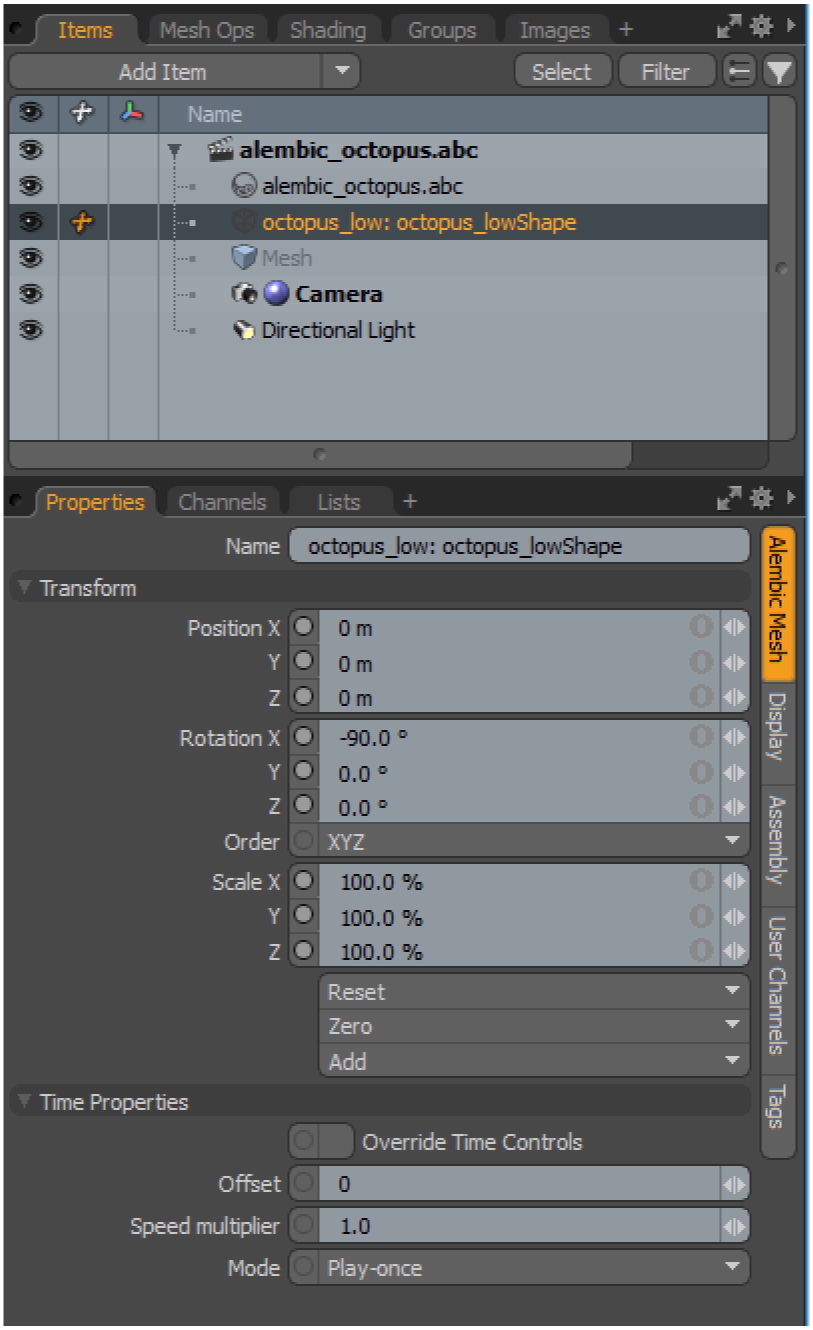
Option	Description
Import as Static Meshes	Converts the loaded scene items to the more efficient, though non-editable Static Mesh Item type. This does not apply to items with deformations.
Make Instances from Matching Mesh Items	Creates instances in cases where identical meshes are encountered during loading.
Combine Transform and Mesh Names	With this option enabled, the resulting mesh node becomes a combination of the names of the two items, the transform and the mesh. With it disabled, it only uses the name of the mesh alone.
Import All Frames	When loading an animated scene, enabling this option loads the transformations and deformations for the entire sequence. If disabled, Modo only loads the scene for the frame specified in **Import Specific Frames.**
Store Animated Morphs with Meshes	When enabled, Modo loads all the mesh deformations into the scene file directly. If disabled, the deformations are referenced from the original Alembic file and are loaded as necessary.
Import Specific Frames	Loads the scene with the transformations and deformations for the defined frames only.
Subdivision Level	When loading subdivision meshes, this value defines their current subdivisions level amount.
Scale	Sets the factor used to scale vertices, velocities, and object position. For non-streaming items, scale is applied only during import. For streaming meshes, scale can be modified in the Properties panel for

	the **Alembic File** Item.
Polymesh streaming options	Imports geometry items and deformers into the current scene, allowing you to layout and render very large animated Alembic files exported from Modo and other DCC software packages. In addition to particles, simulated meshes, and curves, multiple UV sets, materials, and parts are also imported into Modo. **Note:** The Alembic geometry, ***.abc** file listed in the **Item List**, is read only. The following options are available: • **Modo Meshes with Alembic streaming deformers (subds and polys)** - Imports animated geometry, subdivision meshes, and polygonal meshes. Once imported, a procedural **Mesh Item** displays in the **Items List** and you can change the transform, time, and scale properties in the **Alembic Streaming Deformer** tab. For more information, see Alembic Import Properties. • **Alembic Streaming Meshes (polys only)** - Imports non-topology changing polygonal meshes into Modo as a procedural **Mesh Item**. Once imported, two procedural **Mesh Items** display in the **Items List**. You can apply procedural operations to each of these mesh items. When you select the ***.abc** file in the **Item List**, the **Alembic File** tab displays in the **Properties** panel. When you select the ***_lowShape** item in the **Item List**, the **Alembic Mesh** tab displays in the **Properties** panel for you to make the appropriate changes. For more information, see Alembic Import Properties. • **Alembic Streaming Meshes (polys only, force all)** - Imports only polygonal meshes, and all of the Alembic data. It also forces non-animated Alembic meshes to load into Modo. The ***.abc** file is imported, an Alembic mesh, and a procedural mesh are created and listed in the **Items List**. You can apply procedural operations to each of these mesh items.

	When you select the ***.abc** file in the **Item List**, the **Alembic File** tab displays in the **Properties** panel. When you select the ***_lowShape** item in the **Item List**, the **Alembic Mesh** tab displays in the **Properties** panel for you to make the appropriate changes. For more information, see Alembic Import Properties.

Alembic Import Properties

Once your Alembic file is imported, you can set the transform, time, and scale properties for the meshes, particles, and lines used in your scene. The following options are available:


Items
Mesh Ops
Shading
Groups
Images
Add Item
Select
Filter
Name
alembic_octopus.abc
alembic_octopus.abc
octopus_low: octopus_lowShape
Mesh
Camera
Directional Light
Properties
Channels
Lists
Alembic Mesh
Display
Assembly
User Channels
Tags
Transform
Reset
Zero
Add
Time Properties

Name octopus_low: octopus_lowShape

Position X 0 m

Y 0 m

Z 0 m

Rotation X -90.0 °

Y 0.0 °

Z 0.0 °

Order XYZ

Scale X 100.0 %

Y 100.0 %

Z 100.0 %

Override Time Controls

Offset 0

Speed multiplier 1.0

Mode Play-once

Option	**Description**
Transform	
Position X/Y/Z	Use the X, Y, and Z input fields to apply specific offset distance values (calculated from the position the object was in when the tool was activated). When adjusting an object interactively in the viewport, the **Position** input fields display the current offset distance applied,

	providing useful user feedback.
Rotation X/Y/Z	Use the X, Y, and Z input fields to apply specific rotation values. When adjusting an object interactively in the viewport, the **Rotation** input fields display the current offset distance applied, providing useful user feedback.
Scale X/Y/Z	Use the X, Y, and Z input fields to apply specific scaling values. When adjusting an object interactively in the viewport, the **Scale** input fields display the current scale factor applied, providing useful user feedback.
File Properties	
Geometry scale	Sets the factor used to scale vertices, velocities, and object position.
Time Properties	
Override Time Controls	When enabled, the "per file" time controls, set on the Alembic file item, are overwritten. **Note:** This option is not available for selected ***.abc** file in the **Item List**.
Offset	Sets the offset value used in the timeline on which the Alembic animation starts from.
Speed multiplier	Sets the speed of the playback for the Alembic animation. The Default is set to 1.0.
Mode	Sets the playback mode to change the type of animation playback. The following options are available: • **Single-frame** - Plays only the single frame. • **Play-once** - Plays only once. • **Loop** - Repeats the playback. • **Ping-pong** - Playbacks normally and then reverse the playback.

Exporting Alembic Files

Modo provides an Alembic exporting workflow for better interop between Modo and other DCC applications, such as Katana and Houdini.

You can select whether to export Alembic replicas as **Geometry**, **Particles**, or both **Particles and Geometry**. Exporting replicas as a particle system is a space effective method and is usually used for moving replicas into other DCC applications. Particles and geometry can then be connected to native replicators and easily modified. Exporting replica geometry is usually done for rendering purposes.

Note: The Alembic file format automatically deduplicates all geometry items that are the same.

How it works:

Exporting Alembic files is done by clicking the **File** > **Export As** menu and selecting the appropriate ***.abc** file format. Once done, navigate to the appropriate directory and click **Save**.

Modo provides a number of different export options for Alembic files. For example, setting the preferences for the **Export Replicas As** option is done in the **System** > **Preferences** menu by expanding **File I/O** > **Alembic I/O**. Once done, set **Export Replicas As** to either **Geometry**, **Particles**, or **Particles and Geometry**. For more information, see Alembic I/O.

Modo's particle systems are created from replicators by adding an Alembic scalar string property called **ModoReplicator**. This property is enabled by default. Apart from standard Alembic point cloud attributes (positions, velocities, widths, and IDs), Modo exports the following additional data behind the scene:

- **xform** - A 3 x 3 matrix for each particle containing rotation and scale information.
- **dissolve** - The float value for each particle containing the dissolve co-efficient information.
- **itemIndices** - The integer value for each particle containing the index replica item names and item Alembic object path arrays.
- **itemNames** - Non-animated string arrays containing replica item names.
- **itemAbcPaths** - Non-animated string arrays containing the replica item Alembic object paths. Replicas using the same names are identified.
- **modoParticleId** - The float value for each particle containing a Modo specific floating point particle ID.

Supported Formats

Format	Description
Alembic (**.abc**)	Alembic is a data format that saves geometry and animation data. Established by Sony Imageworks and ILM, Alembic is an open computer graphics interchange framework. Alembic distills complex, animated scenes into a non-procedural, application-independent set of baked geometric results. The format currently supports the import and export of

	both static and animated meshes, including subdivision surfaces and camera data. While the format does not directly support material information, part and material tags are exchanged with the mesh data. When importing an **.abc** file, the files remain external to the scene, streaming in on an as-needed basis. When importing files, Modo displays a dialog. For more information about Alembic file preferences, see File I/O Preferences. For information about importing Alembic geometry, see Importing and Exporting Alembic Files.
Autodesk® fbx (**.fbx**)	Originally a file format developed by Kaydara for their FiLMBOX® application, now owned by Autodesk. It has been adopted as a universal exchange format between digital content creation applications, because of its robust support of object, and animation data. Modo reads and writes the libraries supplied by Autodesk, that support interchange of geometry, lights, cameras, basic material and surfacing, UVs and transform animations. For more information about **.fbx** file preferences, see File I/O Preferences.
Autodesk® dxf (**.dxf**)	A simplistic geometry format developed for the interchange of data between CAD applications. DXF format does not support surfacing, materials, Vertex Maps or animation. Modo both reads and writes this format.
COLLADA (**.dae**)	COLLADA™ is an open source 3D interchange format managed by the Khronos™ Group. The Modo COLLADA exporter can read and write geometry with normals and multiple UV texture coordinate sets, materials, lights, cameras, effects, and transform animations, all using a common technique that is compatible with a wide variety of third-party tools. There are a number of File I/O Preferences that support this format. For more information about COLLADA file preferences, see File I/O Preferences.
Encapsulated PostScript (**.eps**)	A 2D vector format developed by Adobe® for exchange of vector drawing information created in their Illustrator® application. Though generally used for print applications, **.eps** files are imported into Modo as

	curves and you can freeze them into faces for further manipulation, render the curves directly, or use them as a basis for patch modeling. Modo does not export into **.eps**, except for the Export UVs to EPS command in the menu bar: **Texture** > **Export UVs to EPS**.
glTF (.***glb**,*.**gltf** and ***.bin**)	The **glTF** (GL Transmission Format) shading model is a file format for 3D scenes and models using the JSON standard. The glTF format exports locator nodes with transformations and hierarchy, meshes with normals, tangents, UVs, and color maps and a placeholder PBR metallic material attached to all meshes. Multiple color and UVs maps can be exported with the meshes. For more information see, glTF Shader and GLTF V2 I/O
HPGL Plotter File (**.plt**)	A 2D vector-based plotter format originally devised for use with AutoCAD®. Modo only exports to this format. For more information about HPGL file preferences, see File I/O Preferences.
LightWave Object (**.lwo**)	Modo reads and writes **.lwo2** format for geometry, supporting a full array of features including UV, weight, morph, and color Vertex Maps. Equivalent surfacing data is translated as well, when applicable. Unsupported information is encapsulated and retained, as long as you remain in the **.lwo** format. Saving to any other format discards the extra data. The loading and saving of LightWave's scene files is not currently supported. For more information about LightWave file preferences, see File I/O Preferences.
McNeel Rhino® (**.3dm**)	Rhino is a NURBS-based 3D modeling application. It has the ability to internally convert its curve based NURBS models into a triangulated polygon model. Modo reads the triangulated polygon data from any Rhino 3DM format file (Modo cannot read NURBS data directly). Modo can both Read and Write to the **.3dm** file format.
Pixar USD (**.usd**, **.usda**, **.usdc**, **.usdz**)	Universal Scene Description (USD) is a system for authoring, composing, and reading hierarchically organized scene descriptions. This format enables users to share USD files freely between applications that support it. See Introduction to USD for more information.

	Note: Currently, Modo can only import Pixar USD file formats. For more information about Pixar USD file preferences, see File I/O Preferences.
Protein Database (**.pdb**)	These files represent the three-dimensional structural data of large biological molecules, such as proteins and nucleic acids. PDB files are common interchange format for displaying these structures and can be readily found on the Internet for most any molecule type.
Scalable Vector Graphic (**.svg**)	SVG is an open standard XML based vector image format, that supports both curve data and image map data. Modo reads the vector data stored in the format and import files as bezier curves into a scene. SVG files are generated in a vector drawing package such as Adobe Illustrator. Once imported, the Mesh Item layer is stored in a folder named after the SVG file. Imported curves can be converted to Curve Polygons or converted to regular polygons using the 'Freeze' command. Additional enabling the render curves option will also make the resulting curves visible to the render engine.
Stereolithography (**.stl**)	The Stereolithography format is the language of 3D printers and CNC milling machines, commonly used with rapid prototyping and computer-aided manufacturing. STL format only describes the surface of the geometry, ignoring any shading or surfacing values. The STL I/O preference options allow you to define how unit values are interpreted and what format the resulting file is written in binary format (**Save as ASCII** option disabled), or in the ASCII text format (**Save as ASCII** option enabled).
VideoScape (**.GEO**)	Aegis VideoScape 3D was an early 3D rendering and modeling program on the Amiga (precursor to LightWave). You can both read and write geometry in this legacy format, though it is most useful for reading, as very few modern programs support the format.
Wavefront Object (**.obj**)	The **.obj** format is a file format developed by Wavefront Technologies for its Advanced Visualizer software package. The file format, being open, has been adopted by many other 3D applications because of its simplicity to implement, and for the most part has become a universal exchange format. **.obj** supports geometry, layers and simple surfacing. Vertex

	normal maps are also supported. The OBJ format provides only a single UV map per object, and all vertices must be mapped (no partial UVs are allowed and they are ignored on export). Modo loads and saves **.obj** format. For information about Wavefront file preferences, see File I/O Preferences.
Web3D Standard (**.x3d**)	An XML-based 3D standard and is the successor to VRML on the Internet. It supports geometry and basic surfacing. Modo is only able to write out to this format.

Work with CAD Assets Inside Modo

Modo can import and export **.3dm** geometry, allowing you to work flexibly on your CAD assets. Modo is compatible with all versions of Rhino up to Rhino 7.

Importing Geometry from Rhino

Certain geometry types in Rhino have direct mappings to geometry in Modo. For example:

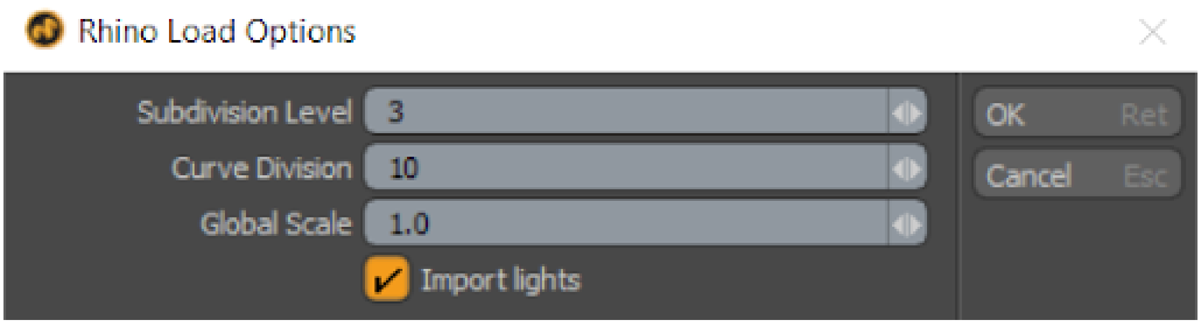
- Meshes
- Subdivision Surfaces
- Curves
- Points

Other geometry types such as NURBS and Extrusions do not have an equivalent in Modo, and are converted to Modo-usable geometry such as curves on import.

Note: When working with Rhino geometry that does not have a native type in Modo, we recommend importing Render Geometry created in Rhino. The topology of render meshes can be adjusted in Rhino to better optimize them for Modo. More information is available here.

To import geometry from Rhino into Modo, you can drag and drop a .3dm file into the viewport or navigate to **File** > **Import** and select the file from the explorer.

Once the **.3dm** file is detected, the **Rhino Load Options** window opens.



Subdivision Level 3
Curve Division 10
Global Scale 1.0
☑ Import lights

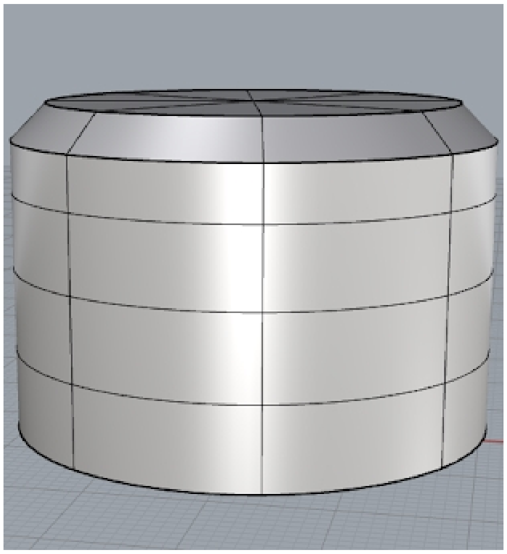
The options present on import are as follows:

- **Subdivision Level** - Allows you to specify the subdivision level of Catmull-Clark polygons created during import.
- **Curve Division** - For NURBS curves converted into Polylines. Each curve segment is divided by this division number.
- **Global Scale** - Scales the model on import.
- **Import Lights** - When enabled, import any lights present in the Rhino scene. When disabled, ignore any lights in the Rhino scene.

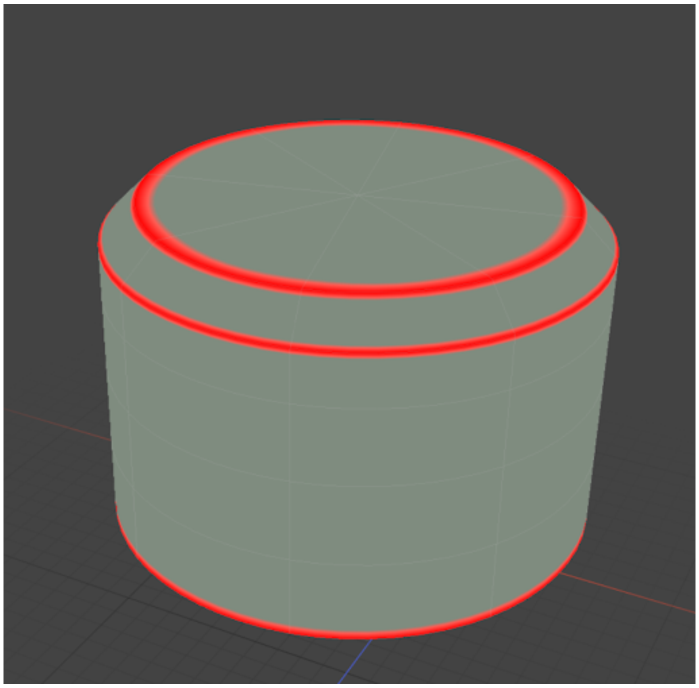
Working with Subdivided Surfaces

When imported into Modo, Rhino surfaces are treated as Catmull-Clark subdivision surfaces. Any edge weighting or creases present in the model are treated as weighted subdivision edges with a value of 100%.

When exporting meshes out of Modo back to Rhino, your asset is converted into a Rhino SubD model. Any edge weight values in Modo are also treated as a Crease Edge in Rhino SubD.



Asset in Rhino



Rhino Asset Imported into Modo, with vertex map enabled to show edge weighting

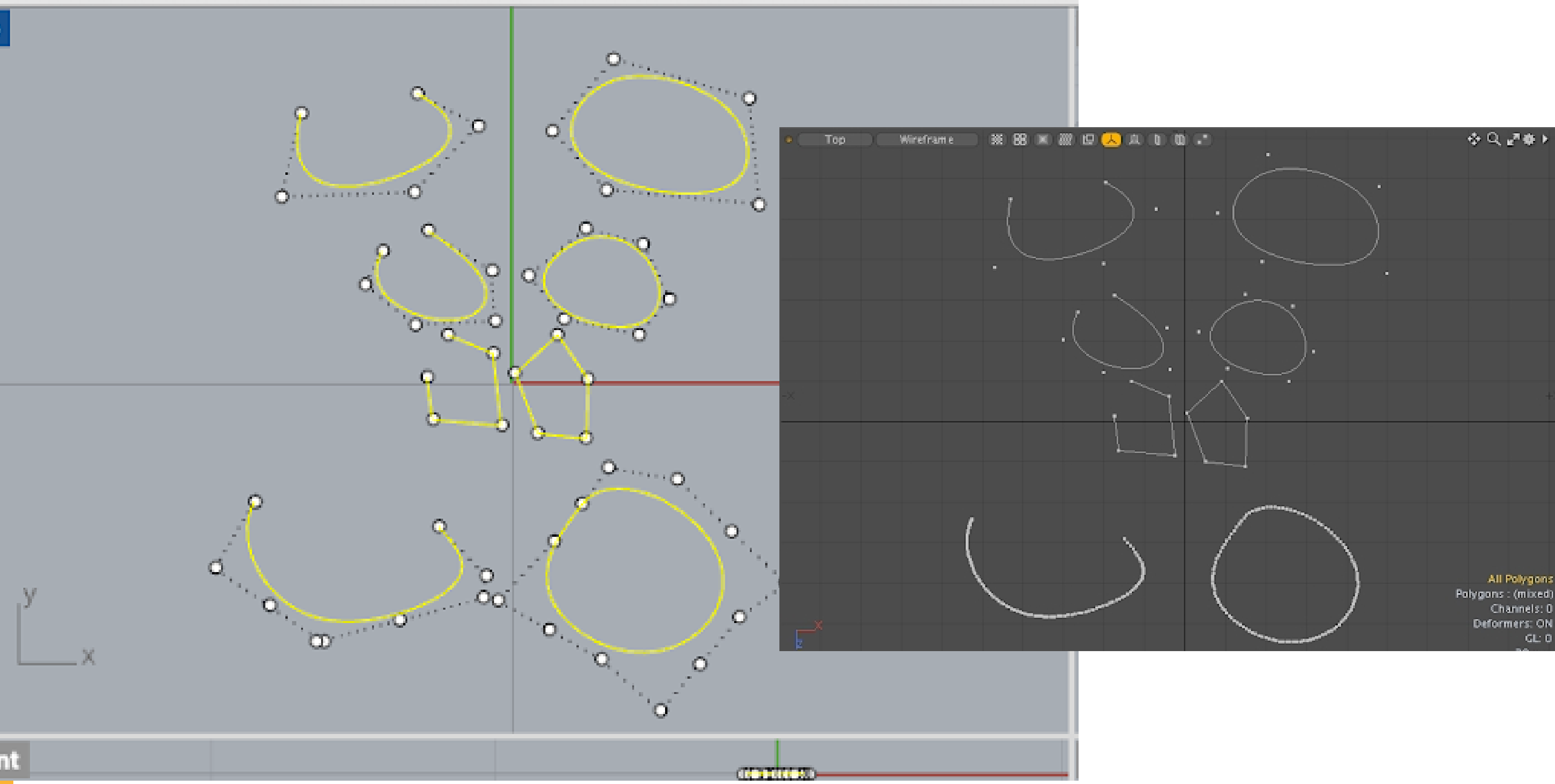
N-gons, or polygons with five or more vertices on one face, present on imported models remain as is, while n-gons present on a model exported from Modo are treated as an n-gon and are not decimated.

Working with Curves

Modo also supports importing and exporting curves to and from Rhino.

NURBS curves in Rhino assets imported into Modo are converted to a B-Spline curve. Weighted NURBS curves are not currently supported. Rhino Polyline Curves are treated as Polyline when imported into Modo.

When exporting assets from Modo for Rhino, B-Spline, Bezier, and regular Curves are converted into NURBS curves. Polylines in Modo are converted into Polyline curves in Rhino.



Top Six Curves: NURBS Curves in Rhino

Bottom Two Curves: Polyline Curves in Rhino

Top: NURBS curves converted to Bezier curves

Bottom: Polyline Curves converted to Polylines

Bitmap Image Formats

Bitmap images are an essential component of texturing, painting and image-based sculpting. They are also useful as backdrops for modeling, providing reference for shape and scale. They can also be used to light the scene, where information captured in the image can illuminate rendered images. With so many different uses, it is necessary to support a great number of image formats.

There are several ways you can load images into Modo. One is from the **Images** viewport, typically located near the **Item List** and **Shader Tree** viewports on the right side of the default interface. Using the **Add Clips** button in the **Images** viewport, you can open an OS-specific dialog box to locate and select an image. Once loaded, images in the **Clips** list appear throughout the application in applicable menus. You can also load images by simply dragging and dropping them from the operating system into Modo. When you drag and drop an image onto geometry, it appears as a texture on a material. If dragged into a 3D viewport, it appears as a backdrop item. To load a bitmap that is not applied to geometry or a backdrop item, drag it into a non-3D viewport. Items can be dragged from the 3D views directly onto groups (where they become members of the group), or into the **Schematic** viewport. You can also load images directly from any menu for selecting images, using the **Add Image** command present in the menu. Once an image is loaded, you can reference the dedicated documentation page for any particular function, specific to how the image is applied and used.

Read Formats

This is a listing of the image map formats Modo can read and use for texturing, backdrops, brushes, and a variety of other functions.

Format Name	Extension	Bit Depth per Channel	Compressed	Type
Flexible Precision Image Format	.flx	HDR	Yes	Lossless
Targa	.tga	8 bit + Alpha	No	
Windows BMP	.bmp	8 bit + Alpha	No	
Graphic Interchange Format	.gif	256 colors	Yes	Palette
Radiance High Dynamic Range	.hdr	HDR	Yes	Lossless
JPEG	.jpg	8 bit	Yes	Lossy
Portable Network Graphic	.png	8 bit + Alpha	Yes	Lossless
Portable Network Graphic 16 bit	.png	16 bit + Alpha	Yes	Lossless
PSD (Photoshop Document)	.psd	8 bit	No	
SGI RGB	.sgi	8 bit	No	
Scalable Vector Graphic (bezier curves only)	.svg	resolution independent format	Yes	Lossless

Format Name	Extension	Bit Depth per Channel	Compressed	Type
Tagged Image File Format	.tif	8 bit + Alpha	Yes	Lossless
16-Bit TIF	.tif	16 bit + Alpha	Yes	Lossless
16-Bit Uncompressed TIF	.tif	16 bit + Alpha	No	
Open EXR	.exr	HDR	Yes/No	
Direct Draw Surface	.dds		No	

Write Formats

This is a listing of formats Modo supports for creating new images.

Format Name	Extension	Bit Depth per Channel	Compressed	Type
Flexible Precision Image Format	.flx	HDR	Yes	Lossless
Targa	.tga	8 bit + Alpha	No	
Windows BMP	.bmp	8 bit + Alpha	No	
Radiance High Dynamic Range	.hdr	HDR	Yes	Lossless
JPEG	.jpg	8 bit	Yes	Lossy*
Portable Network Graphic	.png	8 bit + Alpha	Yes	Lossless*
Portable Network Graphic 16 bit	.png	16 bit + Alpha	Yes	Lossless*
PSD (Photoshop Document)	.psd	8 bit	No	

Format Name	**Extension**	**Bit Depth per Channel**	**Compressed**	**Type**
SGI RGB	.sgi	8 bit	No	
Tagged Image File Format	.tif	8 bit + Alpha	Yes	Lossless
16-Bit TIF	.tif	16 bit + Alpha	Yes	Lossless
16-Bit Uncompressed TIF	.tif	16 bit + Alpha	No	
Open EXR (+layers)	.exr	HDR	Preference	Lossless*

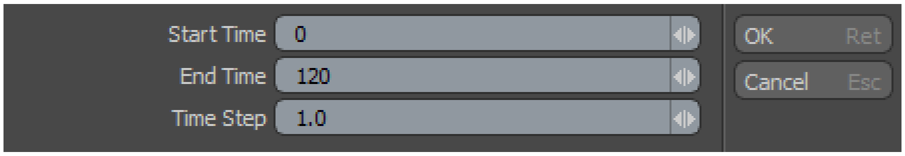
*You can set the amount of compression used by Modo in **System** > **Preferences...** > **File I/O** > **Image I/O**.

Export MDD

MDD is a very simple point cache format that stores a position in 3D space for every vertex in a model for every frame in the animation. It acts as a universal animation transfer format as it is supported by nearly every major 3D application. You can animate your model in your application of choice, and then transfer the geometry to the rendering app of choice and load the geometry and then apply the MDD data as a deformer. In Modo, the MDD format saves all deformation applied to the target by way of deformers. Deformation applied by way of shading or displacement is not taken into consideration for export.

Usage

The general workflow for working with MDD files is to first complete the animation process of the Mesh Item, using any of the available deformers. Next, select the item in the **Items** list, and then invoke the menu bar command **Animate** > **Export MDD...** to save the actual MDD file. Here the **Start Time** and **End Time** are defined and the **Time Step** selected. Click **OK** to open an OS-specific dialog and set the save location. The resulting MDD file is named after the Item name that was initially selected. Once an MDD file has been saved, export the geometry using a method appropriate to the target application. In that application the exported geometry should be imported and the MDD file assigned for rendering. You can also remove the deformers from the item in Modo and utilize the MDD deformer to then re-apply the animation to the object.



Start Time 0

End Time 120

Time Step 1.0

Option	Description
Start Time	This option defines the Timeline position of the first frame of the animation that will be exported into the MDD file.
End Time	This option defines the Timeline position of the last frame of the animation that will be exported into the MDD file.
Time Step	This option defines the frame step of the export. A **Time Step** of '1' would sample the deformation at every frame, a **Time Step** of 2 would skip ever other frame and so on.

Game Asset Exporter

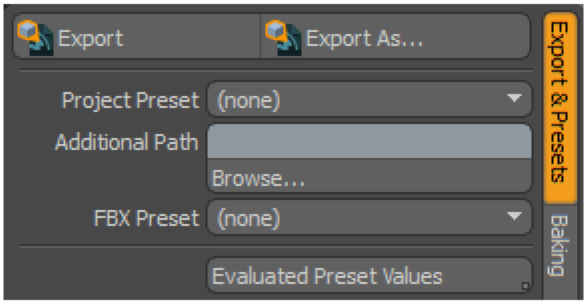
Modo's Game Asset Exporter is available on the **Export & Presets** tab in the left panel of the **Game**layout. The exporter generates **.fbx** files and includes presets for Unreal Engine 4 and Unity. Engine presets are stored in a configuration file and new presets can be created and managed using the UI.

Note: In order to use the asset exporter, you need to have Modo Material Importer for Unreal Engine or Modo Material Importer for Unity installed.

When you export, an additional **.xml** file is generated with the same name, using the suffix **_mat.xml**, in the same directory as the **.fbx** file. For example, **myStuff.fbx** is renamed **mystuff_mat.xml**.

Note: **Procedural** and **Enhance:modo** textures cannot be exported or imported. If you want to preserve the appearance of these textures, you can bake them to the mesh(es).

Export and Presets Settings



- **Export** - Triggers the export to the project root. The scene name is used as the filename.
- **Export As ...** - Opens a file browser, which allows you to select a filename and directory.
- **Project Preset** - Specifies the export settings for a particular engine. The settings include all the parameters on the panel. Presets are included for **Unreal Engine 4** and **Unity**. The **Default** preset maintains the settings for general use.
- **Additional Path** - Specifies an additional path, which is appended to the project's root path to get the full **.fbx** export path. For example, if the project preset's root path is **C:\MyGameProject\Content** and your scene's additional path is **Meshes\Props**, then the full path that the **.fbx** file is exported to is **C:\MyGameProject\Content\Meshes\Props**. This way, you don't need to change the project's export path everytime you want to export a scene to a different directory or file.

  Click **Browse** to select a directory.

  Tip: **Ctrl/Cmd**+click this field to open the location in a file explorer and access your exported data quickly.

  You can also type the directory path into the **Additional Path** field.

  Note: You can specify an absolute path here, which is used instead, ignoring the project's root path.

  Note: For setting up the project's root path, see Editing Project Presets.

- **FBX Preset** - Selects the FBX preset to use for exporting this scene. Each scene stores the name of the FBX preset selected, so when loading or changing scenes, the appropriate FBX preset is selected automatically. This makes having multiple FBX presets for one project, such as for static or animated meshes, easier to deal with.

- **Evaluated Preset Values** - Lists all relevant export settings, so you can easily check whether the final settings are correct before export.

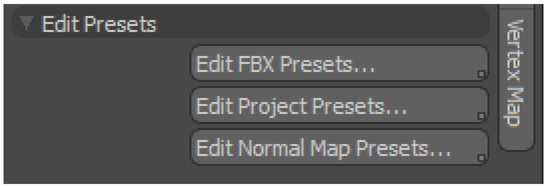
Note: You can't edit the options in **Evaluated Preset Values** directly, they are for information only.

Editing Presets

All **Export & Presets** tab settings can be saved to a preset. This allows you to customize the settings for a particular game engine and save them for future use.

The **Edit Presets** section of the **Export & Presets** tab lets you choose existing presets and create new ones. You can edit the FBX, project, and normal map presets in separate panels.

- Click **Edit FBX Presets...** to select and edit the properties of the FBX preset to be used.
- Click **Edit Project Presets...** to edit settings related to the whole project.
- Click **Edit Normal Map Presets...** to select and edit the properties of the normal map preset to be used.



In each of the following preset panels, the **Edit Preset** dropdown shows the current preset being edited, and gives you the following options to manage presets.

- **(save as)** - saves the current settings as a new preset.
- **(save)** - saves the current settings to the current preset.
- **(revert)** - reverts any changes to the current preset.
- **(delete)** - deletes the current preset.

Note: You cannot remove built-in presets.

Note: Only preset names are stored with the scene, not the settings themselves. This means that if you modify the preset with that name, then any scenes using that preset will use the updated settings.

Editing FBX Presets

The options found in the **Edit FBX Presets...** panel are also available in the **Preferences**. From the menu bar, click **System** > **Preferences** > **File I/O** > **FBX I/O**.

For detailed information on each setting, see FBX Export.

Editing Project Presets

The project preset contains settings relevant to an entire project. Ideally, you set these up once for each project and then you don't have to edit them further.

In **Edit Project Presets...** you can set the following options:

Project Root Path	Sets the root path for the project preset. This is the base directory where all FBX exports go. For Unity projects, this is the project's **Assets** folder, for Unreal projects, this is the **Content** folder.
PBR XML Settings	
Relative Texture Paths in XML	If you have enabled **Export PBR XML** in your FBX settings, this toggles whether texture file paths are stored relative to the exported FBX file or as absolute paths. When enabled, relative paths are used for texture images in the sidecar **.xml** file. This is needed for the Unity and Unreal material importers. For more information on these plug-ins, see Modo Material Importer for Unity and Modo Material Importer for Unreal Engine.
Use Material Tag to Match Materials	If you have selected **Export PBR XML** in your FBX settings and enable this option, the importer matches up polygons to their materials using the polygon material tag of that material, as opposed to the material group's name in the **Shader Tree**. This is primarily used in engines such as Unreal Engine 4, which use the polygon's material tag to match up polygons to their materials.
Texture Copying Settings	

Copy Textures on Export	Enables textures to be copied to the project along with the **.fbx** file at export. The following options are available: • **Never** - never copies textures over to the project. Use this if you prefer to handle textures manually. • **Different Only** - only copies the texture over to the project if the texture file that Modo uses has been updated compared to the destination texture. Any textures which do not exist in the destination are also copied. • **Always** - always copies textures over to the project, even if they haven't been updated.
Texture Path	If any textures are to be copied over at export, this specifies where they are stored. This is closely linked to the **Texture Path Mode** setting.
Texture Path Mode	This specifies how to interpret the **Texture Path** setting. There are three available options: • **Absolute** - the texture path represents an absolute path that textures should be copied into. • **Relative to Project Root** - similar to the scene's **Additional Path** setting, appends the texture path to the project's root path to get the full destination for texture copies. • **Relative to Export** - appends the texture path to the scene's full FBX export path.
Force Alphanumeric Names	This is an aid for some game engines that prefer to use alphanumeric names for assets, and don't handle special characters well. Unreal Engine 4 is an example of this. When enabled, it strips out all special characters from copied texture files except underscores (_) and dashes (**-**), and replaces any spaces with underscores.
Use Image Clip Filenames	When enabled, copied texture files are named to match the name of the texture layer, rather than retaining the image's original filename.

Normal Map Settings	
Auto Apply Normal Preset	When enabled, normal maps are automatically set up according to the normal map preset selected for the project.
Normal Map Preset	The name of the normal map preset to be used with the project. The following options are available: • Source • Unity • Unity Legacy • Unreal

Editing Normal Map Presets

The normal map preset specifies default settings for normal maps and tangent bases.

In **Edit Normal Map Presets...** you can set the following options:

Tangent Basis Command	Selects the default command to generate tangent basis maps. This command is assigned as the default choice for the for the tangent basis map creation button in the **Mesh Tangents** section of the **Vertex Map** tab. See Creating Mesh Tangent Maps for details.
Per-Pixel Bitangent	Sets whether the bitangent vector of the tangent basis should be generated per-pixel rather than per-vertex, in order to achieve an exact match with the target game engine's normal mapping methods. **Note:** By default, this is enabled for Unreal and disabled for Unity. Other game engines may vary.
Invert Green Channel	Sets whether the green channel of the normal map should be inverted to match the game engine's requirements.

Keyboard Shortcuts

Keyboard shortcuts can really streamline the user experience, by mapping specific functions to simple keyboard key presses. Any command in Modo can have a keyboard shortcut assigned to it. These keyboard shortcuts can also be re-assigned, customizing Modo to a users specific needs. By also allowing the **Ctrl**, **Shift** and **Alt** key qualifiers (including combinations thereof), you have a wide array of keys to choose from.

Note: On Mac machines that use a limited keyboard, such as a laptop, replace the **Ctrl** key with the **Cmd** key.

To assign a keyboard shortcut to a command, perform the command from the interface, then expand the Command History viewport, located in the lower-right hand corner of the interface. Locate the command in the list (usually visible at the bottom of the history), then right-click and choose **Map Command to Key**. You can also assign keyboard equivalents using the **Input Editor** found on the **System** menu on the menu bar. This is useful if you want to create keyboard shortcuts for scripts or tools that don't actually appear in the default interface.

List of Keyboard Shortcuts

General/System

Keystroke(s)	Action
F1	Help (Click on the interface element to open the related help file.)
Ctrl+**N**	New scene
Ctrl+**O**	Open saved scene
Ctrl+**I**	Open saved image
Ctrl+**W**	Close current scene
Ctrl+**Shift**+**W**	Close all open scenes

Keystroke(s)	**Action**
Ctrl+**S**	Save current scene
Ctrl+**Shift**+**S**	Save scene as...
Ctrl+**Q**	Quit Modo
Alt+**F12**	Config file save
Shift+**F5**	Run script
Ctrl+**Z**	Undo
Ctrl+**Shift**+**Z**	Redo
Ctrl+**X**	Cut
Ctrl+**C**	Copy
Ctrl+**V**	Paste
Delete	Delete (not available in **Items** mode)
Ctrl+**R**	Brings up the last tool used
Alt+**R**	Brings up the last tool used, ignoring UI interactions and selections
Shift + **Ctrl**+**Tab**	Layout switcher, organized by most recent

Viewports

Navigation

Keystroke(s)	**Action**
Alt+click	Rotate view
Alt+middle-click	Rotate viewport Z axis (bank) view
Alt+right-click	'Flick' rotate
Shift+**Alt**+left-click	Pan view

Keystroke(s)	Action
Shift+**Alt**+right-click	Pan up/down only
Ctrl+**Alt**+left-click	Zoom to mouse position
Ctrl+**Alt**+right-click	Box zoom
. (period)	Zoom in (only in component modes)
Shift+**.** (period)	Zoom in x2
, (comma)	Zoom out (only in component modes)
Shift+**,** (comma)	Zoom out x2
G	Center the view on the cursor's position
/	Turntable (only in component modes)
Shift+**/**	When turntable is running, displays upright turntable
Mouse Wheel Scroll	Zoom view in and out to mouse pointer
A	Fit (zoom to scene extents/all items)
Shift+**A**	Fit selected (zooms to selected elements)
Ctrl+**A**	Align selected (aligns view to the current selection)
Ctrl+**Shift**+**A**	Fit and align selected (zooms and aligns the view to the current selection)

3D viewport Options

Enter values on the numeric keypad and move the pointer over the target viewport to select.

Keystroke(s)	Action
O	Open the viewport options pop-over
Alt	Modifies the **Viewport Settings** button in the top left of the viewport to show shading styles for inactive meshes

Keystroke(s)	**Action**
0 (zero)	Maximize/minimize the viewport within the group
. (decimal)	Perspective
1	Top (press again to toggle top/bottom view)
(Mac): **Shift**+**1** (Windows): **Windows** +**1** (Linux): **Shift**+**1**	Bottom
2	Front (press again to toggle front/back view)
(Mac): **Shift**+**2** (Windows): **Windows** +**2** (Linux): **Shift**+**2**	Back
3	Right (press again to toggle right /left view)
(Mac): **Shift**+**3** (Windows): **Windows** +**3** (Linux): **Shift**+**3**	Left

Keystroke(s)	Action
4	Shaded
5	Texture
6	Reflection
7	Wireframe
8	Solid
9	Vertex Map
/	Wireframe None
Shift+**/**	Wireframe Light
Ctrl+**/**	Wireframe Dark
= (equals)	Toggle viewport smoothing

Tabbed Viewport

Note: The following keyboard shortcuts only work on Mac.

Keystroke(s)	Action
Ctrl+**Shift**+**Home**	Open first tab
Ctrl+**Shift**+**End**	Open last tab
Ctrl+**Shift**+**Page Up**	Open next tab
Ctrl+**Shift**+**Page Down**	Open last tab

Item List

Keystroke(s)	**Action**
F	In **Items** selection mode, locate selected item (layer). In other modes flip polygons
Shift+click ▸	Expand/close all (also in Shader Tree)
' (apostrophe)	Swap foreground/background layers
Ctrl+**D**	Duplicate selected layer(s) (only in **Items** mode). If you have a procedural mesh or procedural operation selected, this command activates **Duplicate Procedural Stack**.
Shift+**D**	Instance selected layer(s) (only in **Items** mode)
N	New Mesh Item
L	New locator item (only in **Items** mode)

Channel List

Keystroke(s)	**Action**
I	Select modifier input
O	Select modifier output
C	Channel haul on (only in **Items** mode)

Floating Windows

Keystroke(s)	**Action**
F2	Modeling toolbox
F3	Sculpt/paint toolbox

Keystroke(s)	**Action**
F4	Setup toolbox
F5	Command History viewport
F6	Presets viewport
F7	Graph Editor
F8	Render preview
F11	Snapping pop-over
` (backtick)	Toggle window visibility

Pie Menus

Pie menus are quick contextual pop-up menus that appear using keystroke combinations. Menu items are laid out in a wheel so you can access them gesturally for speedier workflow. To open, press the keyboard shortcut, then click in the 3D viewport to open. If you want the pie menu to appear immediately, go to **System** > **Preferences** > **Input Remapping** and enable **Open Pies Immediately**.

Keystroke(s)	**Action**
Ctrl+` (backtick)	Layout switch favorites
Ctrl+**1**	GL viewport toggles
Ctrl+**2**	GL viewport style
Ctrl+**Spacebar**	GL viewport view type
Ctrl+**3**	Open pop-over
Alt+**Q**	Selection mode/Transform tools
Alt+` (backtick)	Open palette
Shift+` (backtick)	Properties and options
B	Brushes pie menu (only when a Brush tool is active)
Alt+**Spacebar**	Animation shortcuts (only in the Animation layout)

Modeling

General Modeling

Keystroke(s)	Action
Q	Drop active tool
Ctrl/Cmd+**Tab**	Toggle the Tool HUD
K	Tool properties pop-over **Note:** The tool needs to be active for this shortcut to work.
Esc	Escape current operation/drop active selection
Spacebar	Cycle-through **Component** modes
Shift+**Spacebar**	Toggle **Component** mode to **Item** mode (or last mode)
= (Equals)	Tool handles larger
- (Minus)	Tool handle smaller
0 (zero)	Toggle tool handle visibility (not on the numeric keypad; only on Mac)
Ctrl+**D**	Reset tool attributes
Shift+**0-9**	Select layer 1-10
Alt+**Spacebar**	Quick access pop-over
M	Assign material tag. This applies when working in both direct and procedural modeling.
S	Enables the Slide / Edge Slide tool when in a component mode (Polygons, Edges, or Vertices).

Keystroke(s)	Action
	Note: This shortcut only activates the Slide tool when in component mode. Pressing **S** in item mode keys a selected channel. For more information, see Using Channels.
Tab	Toggle subdivision surfaces
Shift+**Tab**	Toggle Catmull-Clark subdivision surfaces
Numeric Keypad **+/-**	Increase/decrease subdivision level

Action Center/Axis

Keystroke(s)	Action
Alt+**W**	Origin
Alt+**E**	Pivot
Alt+**A**	Automatic
Alt+**S**	Selection
Alt+**D**	Selection center, auto axis
Alt+**Q**	Displays pie menu
Alt+**F**	Screen
Alt+**Z**	Element
Alt+**X**	Local
Alt+**G**	Pivot center, parent axis

Selection

Keystroke(s)	**Action**
Alt+**Shift**+**A**	Select all
Up arrow	More (select next plausible element in loop)
Shift+Up arrow	Select expand
Down arrow	Less (de-select last element)
Shift+Down arrow	Shrink
Left arrow	Select previous loop
Shift+Left arrow	Select previous loop add
Right arrow	Select next loop
Shift+Right arrow	Select next loop add
L	Select loop
Alt+**L**	Select ring
]	Connected
[	Invert selection
Shift+**]**	Close loop
Shift+**G**	Select between
Alt+**1**	Convert selection to vertices
Alt+**2**	Convert selection to edges
Alt+**3**	Convert selection to polygons
; (semicolon)	Select through toggle

Hide/Lock Geometry

Keystroke(s)	Action
H	Hide selected
Shift+**H**	Hide unselected
Ctrl+**H**	Hide invert (toggle state)
Shift+**Q**	Hide Fusion Source Mesh
U	Show
J	Lock selected
Shift+**J**	Lock unselected
Ctrl+**J**	Lock invert (toggle state)
I	Unlock

Work Plane

Keystroke(s)	Action
Home	Align to geometry under cursor
Shift+**Home**	Align to selection (average of all selected)
Ctrl+**Home**	Align to viewport
End	Reset the **Work Plane**
Shift+**End**	Open **Work Plane** pop-over
Page Up	Walk +
Page Down	Walk -
Numeric Keypad ***** (asterisk)	Toggle visibility
Alt+**O**	Offset **Work Plane** to geometry position under cursor

Snapping

Keystroke(s)	Action
X	Toggle snap state (hold with tool active for temporary snap)
F11	Open snapping pop-over
Ctrl+Shift+T	Element snap

Tools

Keystroke(s)	Action
Transform	
W	Move
E	Rotate
Alt+Shift+E	Planar rotate
R	Scale
Shift+R	Uniform scale
Y	Transform
Deform	
Ctrl+E	Bend
T	Element move (Tweak)
Ctrl+T	Soft drag
Shift+Y	Flex tool
Shift+S	Smooth tool
Duplicate	
Shift+X	Extrude

Keystroke(s)	Action
Shift+L	Radial sweep
Shift+Q	Grow quad row from edge
Z	Extend edge
Shift+V	Mirror tool
Mesh Edit	
Backspace	Delete
Shift+Backspace	Remove
Ctrl+Backspace	Collapse
B	Bevel
Shift+B	Polygon bevel
Alt+B	Sketch extrude
Ctrl+B	Create polygon of given type from vertex selection
G	Edge slice
Shift+C	Polygon slice
Alt+C	Loop slice
D	Subdivide selected geometry
Shift+D	Open subdivide options
F	Flip polygon normal
Ctrl+L	Split face by selected vertices
P	Make polygon
Shift+T	Triple polygons (convert to triangles)
V	Spin edge
Alt+V	Spin quads

Keystroke(s)	Action
Curves	
Shift+**O**	Make open curve
Ctrl+**P**	Make closed curve
Vertex Tools	
Shift+**W**	Weight tool
Shift+**M**	Morph tool

Paint & Sculpt

Keystroke(s)	Action
B	Open brush tips pie menu (only when tool is active)
O	Open large color picker pop-over (hover over UI color picker)
S	Swap foreground/background colors
Shift + click and drag	Reposition the node and any child nodes further down the chain.
Ctrl + click	Create a new node between an existing one.
Middle mouse button + click	Remove individual nodes.
E/**R**	Image ink rotate left/right
D/**F**	Image ink scale up/down
Shift+**Z**	Adjust offset amount (sticky key, only while **Brush** tool is active)
Multi-Res Sculpting	
Alt+(numeric pad) **+**	Subdivision level up
Alt+(numeric pad) **-**	Subdivision level down
Ctrl+(numeric pad) **+**	Current level up
Ctrl+(numeric pad) **-**	Current level down

Camera

Keystroke(s)	Action
Shift+**Alt**+click+drag	Move camera item
Shift+**Alt**+right-click+drag	Move up/down direction only
Alt+click+drag	Rotate around target distance
Alt+right-click+drag	Rotate around Z axis
Ctrl+middle-click+drag	Image ink scale up/down
Ctrl+**Alt**+click+drag	Move forward/backward only
Ctrl+**Alt**+right-click+drag	Adjust focal length
Roll Mouse Wheel	Zooms to cursor
Ctrl+**A**	Auto focus camera (only in a viewport defined as **Camera** and with auto-focus enabled)
Ctrl+**F**	Restart preview (only in the **Preview** viewport)

Animation

Selection

Keystroke(s)	Action
Shift+**Z**	Select all animated items
Z	Select all animated channels

Hierarchy

Keystroke(s)	Action
Ctrl+**G**	Group selected items (under group locator)
P	Parent in place
Ctrl+**P**	Parent
Shift+**P**	Unparent in place
Ctrl+**Shift**+**P**	Unparent

Timeline

Users must be in **Items** selection mode for these to work.

Keystroke(s)	Action
Shift+Up arrow	Go to first frame
Ctrl+Up arrow	Go to first keyframe
Ctrl+Left arrow	Go to previous keyframe
Shift+Left arrow	Go back one frame
/	Start/stop animation playback
Shift+Right arrow	Go forward one frame
Ctrl+Right arrow	Go to next keyframe
Ctrl+Down arrow	Go to last keyframe

Setting Keyframes

Keystroke(s)	**Action**
Shift+**Y**	Key transforms
Shift+**E**	Key rotations
Shift+**W**	Key positions
Shift+**R**	Key scale
S	Key selected channels
Shift+**S**	Key selected items

Graph Editor

Keystroke(s)	**Action**
A	Fit selected
Ctrl+**F**	Fit visible
Shift+**F**	Fit visible width
Alt+click	Pan view
Ctrl+**Alt**+click	Scale view
Alt+right-click	Scale graph size
Ctrl+**Alt**+right-click	Box zoom (drag box to zoom)
Ctrl+**Shift**+**V**	Paste insert keyframes to selected channels
Ctrl+**Alt**+**V**	Paste replace keyframes to selected channels

Render

Keystroke(s)	Action
F9	Render
Ctrl+**F9**	Render visible
Shift+**F9**	Render selected
F10	Render current view
Alt+**Shift**+**F9**	Open render display window
P	Pauses mPath renders while the render window is open

Preview Viewport

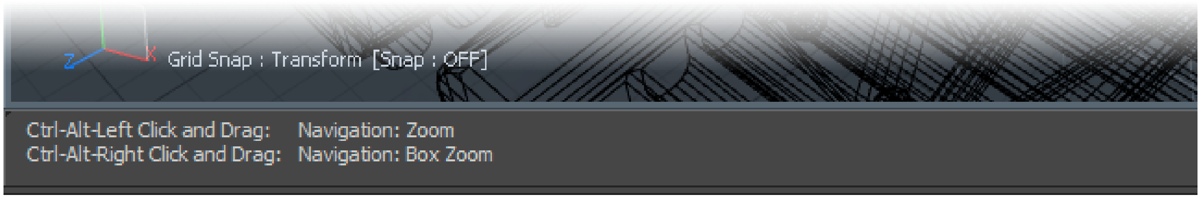
(**F8** to open)

Keystroke(s)	Action
Ctrl+**F**	Focus under cursor
mouse wheel	Preview zoom
middle-click+drag	Jump the preview out to the original focal length (only when zoomed in)

Mouse Standards

Modo has many keyboard-based shortcut keys. Despite this fact, you may still largely interact with the application by way of the mouse (or pen and tablet, if that's what is preferred). When working in the various 3D Viewports, Modo has two main styles of mouse input: hauling and handle manipulation. Hauling is when you press and hold the left mouse button, and then drag anywhere within a 3D viewport while a tool is active. (In this instance, active means selected, and activated tools are highlighted in the toolbox.) Handle manipulation is when you hover over a tool's manipulator handle, which is present in the viewport when a tool is active. This action highlights the handle in yellow, wherein you may then click and drag the handle, constraining the action to the particular function that the handle represents.

To make your workflow more fluid, Modo employs a set of standards for all mouse behaviors, depending on the type of tool. These standards allow Modo to leverage all three mouse buttons as well as qualifier key combinations to produce variations on tool behaviors. There are several basic types of tools - geometry creation tools, geometry editing (transform) tools, and curve creation tools. The standards for mousing are slightly different between these categories, so that the most commonly desired behaviors could be mapped to the primary mouse inputs for each style of tool. When you activate a tool in Modo you can always see the mousing behaviors listed at the bottom of the Modo interface in the Tool Tips viewport.



Grid Snap : Transform [Snap : OFF]

Ctrl-Alt-Left Click and Drag: Navigation: Zoom
Ctrl-Alt-Right Click and Drag: Navigation: Box Zoom

The Status Bar appears at the bottom of the default interface providing feedback on the current tool. The basic rules of using a mouse are quite simple. Understanding these rules allows you to take full advantage of the flexibility and speed of the tools. The following tables quickly illustrate the various modes, followed by detailed descriptions and examples of each operation type.

Transform Tools

Click	Middle-click	Right-click	Shift+click	Ctrl+click
Reset tool and haul parameters	Clone and haul parameters	None (*reserved for falloff modifiers*)	Continue hauling parameters without reset	Constrain parameters

Reset Tool and Haul Parameters

Definition: On click, the tool properties are reset. As the mouse is dragged, the values are changed.

Example: When using the **Axis Rotate** tool with the Automatic Action Center, clicking in the 3D viewport sets the pivot position for the rotation. Click and drag resets the tool properties to 0° and sets the pivot for the rotation under the mouse. Dragging then changes the amount of rotation.

Clone and Haul Parameters

Definition: Re-applies the current/last properties of the tool and allows continued editing from those values.

Example: The Axis Rotate tool has been used to rotate the selection by 45°. Using clone and haul parameters re-applies that rotation so the selection is then at 90° from its original position. The tool widget is still at 45° since the value was cloned. You are now able to continue hauling to edit the value.

None

Definition: The right mouse button, when transforming geometry, has no direct action. This option is reserved for setting the action center and defining the range of an active falloff.

Continue Hauling Parameters Without Reset

Definition: The tool properties are edited from the current value without resetting any of the tools parameters or action center info.

Example: When using Axis Rotate, this mode leaves the current rotation amount in place, as well as the Rotate tool's pivot. Dragging the mouse in this mode simply increases or decreases the rotation value.

Constrain Parameters

Definition: Uses initial mouse drag to determine which properties to edit, and constrains the mouse to only edit those values.

Example: When hauling the **Move** tool, two properties are edited simultaneously depending on the current Work Plane. If you are editing on the XY plane, hauling edits both the X and Y move values. Using Constrain parameters, the first direction the mouse is moved determines the editable axis. In this case, holding **Ctrl** and dragging to the right edits the X move value, and the Y value remains unchanged.

Geometry Creation Tools

Click	Middle-click	Right-click	Shift+click	Ctrl+click
Haul parameters	Clone and haul parameters	Haul alternate parameters	Reset tool and haul parameters	Constrain parameters

Haul Parameters

Definition: The tool properties are edited from the current value without resetting any of the tools parameters or action center info.

Example: Using the **Polygon Bevel** tool, the first click activates and allows you to haul both the inset and shift values freely. Subsequent clicks simply add to the existing edit values. This enables you to click and drag repeatedly on a single bevel to edit the values.

Clone and Haul Parameters

Definition: Re-applies the current properties of the tool and allows continued editing from those values.

Example: The **Polygon Bevel** tool can be applied repeatedly with the same inset and shift properties by using Clone and Haul parameters without additional dragging.

Haul Alternate Parameters

Definition: Some creation tools provide hauling edits for secondary properties by right-clicking to haul.

Example: The Cube primitive tool allows the segment value to be edited by right-clicking, then dragging, while the tool is active. The axis is determined by the orientation of the work plane.

Reset Tool and Haul Parameters

Definition: Any current tool action is applied, the tool properties are reset and a new application of the tool begins.

Example: When using the polygon bevel tool this behavior allows you to begin a new bevel and immediately haul the inset and shift values. With this method you can quickly apply a series of bevels, each with its own inset and shift values.

Constrain Parameters

Definition: Constrains multiple parameters of the geometry creation tool simultaneously.

Example: When creating a sphere, holding down the **Ctrl** key constrains all three axes to the same value creating an equally sized sphere.

Curve Tools

Click	Middle-click	Right-click	Shift+click	Ctrl+click
Create new curve node	Delete Node	None	Drag Node+Child Node	Move Constrained

Create New Curve Node

Definition: Click with a curve-type tool (Curve, Bezier, Tube, etc) creates a new curve point *after* the currently-selected point. To add a point between two existing points, first select the point that you intend to be “behind” the new point. Click+dragging on a curve point edits the location of that point.

Example: The Tube tool allows you to click repeatedly in the 3D viewport to define new tube control points. Clicking on an existing point selects it so that you can add new points in the middle of the tube.

Delete Node

Definition: Middle-clicking directly on a curve handle deletes it immediately.

Example: When working with the Curve tool you may add more nodes that you need. Simply middle-click on a node deletes it from the curve.

None

Definition: Right-click has no direct action when editing curves.

Drag Node and Child Nodes

Definition: Holding down the **Shift** key while clicking to move a curve node also moves any nodes that are “children” of that node.

Example: When using Solid Sketch you may want to move an entire “branch”. To do this, simply hold the **Shift** key while click+dragging the node.

Move Constrained

Definition: **Ctrl**+clicking a curve control point constrains your movement along the first axis you drag toward.

Example: When using the Tube tool, hold the **Ctrl** key down and click on a control point in the tube. Dragging along an axis *constrain*s the move to that axis so that you can not adjust the position of the point on the other axis.

Modo Markup for Forms

Modo Markup is a simplified text markup feature that exposes some Rich Text features for tooltips and message tables. It supports tooltips in command dialogs and forms, and any message table lookups.

Syntax

text	Bold
text	Italics
\`text\`	Fixed-width font
~text~	Small font
==text==	Heading (bold and centered)
*** text**	Bullet

	Note: There is a space after the asterisk. Must be at the beginning of a line.
![icon][iconResource]	Icon Resource
![inputevent][eventHash]	Key assigned to the an event hash. Hash format is the same as is stored in the config.
![inputcommand][commandString]	Key assigned to the command string in the global input map
****	Escape character. Any character typed after this is not parsed, and inserted directly.
{{{...}}}	Content inside the curly braces is not parsed.

Restrictions

- Font styles cannot be nested. You can't do *_text_* to get a bold italics. This is a limitation of the underlying rich text system.
- **![][]** inserts can be wrapped in the font styles for situations where you want to display input mappings or icons with bold or italic text, for example.
- Fixed-width fonts are slightly offset, due to the underlying implementation.
- inputcommand key assignments are always found in the global map, not in view-specific maps.
- Special hidden Rich Text codes are inserted into parsed strings, which means that they can be parsed a second time without issues. However, these hidden codes can result in unprintable characters being displayed for clients that don't support Rich Text, such as window titles, edit fields, or OpenGL. Most internal code detects and deletes these codes where necessary.

Frequently Asked Questions

Have a problem that's perplexing you with Modo? It's likely others have had the same problem. This appendix lists some of the most common questions about Modo. If your question isn't answered here, review the documentation and the release notes, or contact Foundry customer support for help.

Accuracy

How do I find out my model's dimensions?

Absolute dimensions for the overall bounding volume of any selection can be found using the Dimension Tool under **View** > **Dimensions Tool**, and described inUsing Utilities. Invoking the command displays the values over a ruler, displayed directly in the 3D viewport. To confine the display to a specific section, simply select it using the appropriate selection mode, described in Selecting Items. When nothing is selected, the Dimension defaults to the overall bounding box volume of all visible items.

How do I scale my model to a certain size?

You can scale objects, based on the selection's bounding box volume using the Absolute palette, found at the bottom of the Snaps and Precision Palette tool. Use the **Grab Size** command to find the current size, then enter the new value in the appropriate input field, and press enter to apply the new dimension.

How can I snap items in a scene?

Modo has a snapping function that is based on the snapping of tool handles. Proper snapping requires an understanding ofSpecifying Action Centers and Falloffs settings. The snapping functionality is covered under Applying Snapping in the Modeling section.

Alignment and Positioning

How can I accurately align my object onto the surface of another object?

For simple geometry in the same item layer as the target surface, you can use the Tack tool with the **Move Connected Vertices** command enabled. For more complex objects, it's easier to use theHierarchy functionality found in the Animate interface tab. Both the source and target need to be separate item layers.

1. You can set the **Center** of the source item to the appropriate position in relation to the geometry, typically using the command **Edit** > **Center to Bounding Box** > **Bottom**.
2. Set the **Drop Action** function to the **Place and Align** option.
3. Without any tools active, simply click on the source mesh and drag it over the target surface.
4. Release the mouse button, setting the object to the surface.

I can't get a background constraint to work, how should I set it up?

Background constraints need some background geometry. The target geometry needs to be in the foreground, meaning in thelayer is visible and selected in the **Item List**. If you're not familiar with foreground, background, and visibility states, please reference the Items (Scenes) List. To enable a background constraint:

1. Press the **F11** key and choose **Background** from the **Constraint Mode** options.

   By default, the background constraint only constrains tool handles.
2. To constrain geometry as well, choose either **Screen Axis** or **Vector** options under **Geometry Constraint** in Tool Properties.

How can I specify specific positions for elements?

This is dependent on the particular selection mode. While in **Items** selection mode, you can define the position of individual layers in the Items properties panel. This sets the center position to the specific location with the associated geometry remaining in proper relation to the center element. Individual vertices can be explicitly positioned in the Info and Statistics viewport. For Edges and Polygons, transforms are applied as an offset from the present position using the Move tool. To assign a specific position, simply center the selection prior to applying the offset.

How can I tell what size my object/selection is?

Refer to How do I find out my model's dimensions?

Animation

My Item isn't rotating/scaling from where I want it to. How can I fix this?

This is most likely because the Center isn't positioned correctly in relation to the geometry. You can use the Center to Bounding Box commands to easily reposition the Center or simply Center the geometry using the appropriate command.

Why don't my deformers show up?

You can define how [MDD, Morph and Weight] Deformers display on a per-viewport basis. By default, most viewports are designated to not display **animated** deformers (only the **Animation** and **Rendering** interface tabs do). You can open the options panel for a particular viewport by hovering the mouse over the target viewport and press **O** on the keyboard. On the viewport options pop-over, you need to set the **Enable Deformers** option at the bottom of the **Drawing and Control** section. If they still aren't working, after doing so, make sure that you are working in **Items** selection mode when adjusting the deformer values.

Documentation

Why does Internet Explorer give me an error every time I open the documentation?

For security purposes, Internet Explorer disables the execution of scripts that are served locally (from your local system, not the Internet). If you are viewing the documentation from your local drive, you need to **Allow Blocked Content** each time the documentation is accessed. Advanced users may wish to enable the **Allow Active Content to Run in Files on My Computer** setting in the **Advanced Internet Options** dialog.

Interface

How can I customize my interface?

You can customize your interface easily but, before doing so, understanding the functions of the **Thumb** and **Widget** are essential. Those functions are covered under Using Layout Controls. Additionally, you can follow a walk-through for building a custom interface under Customizing Your Layout.

How can I map my own keyboard shortcuts?

Keyboard shortcuts can be assigned within the **Input Editor** found in the menu bar under **System** > **Input Editor**. An easier way to accomplish the same task is to utilize the Command History viewport. To map a command to a custom key:

1. Perform the command in Modo by either invoking it from the menu bar or within the Modo UI itself.
2. Within the **Command History** viewport, locate the appropriate command. Right-click the command and choose the **Map Command to Key** option from the contextual menu.

   This opens the **Key Assignment** dialog where you can define the particular key you want to use.
3. Press **OK** to set the new definition.

   You should note that this new value is saved in the Modo configuration file (MODOXXX.cfg). Removal of this file also removes custom keyboard shortcuts.

How can I change my viewport colors?

Viewport colors are changed in the Preferences (on Windows: **System** > **Preferences**, on Mac OS: **Menu** > **Preferences**) under Display > Colors. For more information on display preferences, refer to Display Preferences. You can modify the colors and then use the **Save to Scheme** button and assign a new name to the color scheme.

To apply the scheme to a viewport, you need to click within the target viewport to select it. Selected viewports are denoted by the orange highlighting of the **Thumb** icon. Then, in the menu bar, select the command **View** > **Viewport Color Scheme** and choose the named scheme defined earlier in the Preferences.

How can I change the default lights in the 3D Viewports for modeling?

Modo offers you the ability to adjust the default lighting in 3D Viewports using the options found in the menu bar under **View** > **Edit Viewport Lights**. Once the function is active, you can click and drag in the viewport to change the current light's position. Additional properties are available in Tool Properties for **Color** and **Intensity**. The option **Current** defines the particular light being edited. You can use the commands **View** > **Add Viewport Light** and **View** > **Remove Viewport Light** to modify the number of lights visible.

Why does my 'X' keyboard shortcut not work?

Certain keyboard shortcuts require Modo to be in a specific state or mode. For example, the Timeline shortcuts require Modo to be in **Items** selection mode, or the brush tip pie menu requires a Paint tool to be active. Make sure the proper tool is selected and the correct selection mode is active before applying the keyboard shortcut.

After I've changed an option, how can I return its setting to the default value/state?

Each animate-able attribute value is preceded by a small dot or circle that defines its keyframe status. You can right-click on the option to open the contextual menu and select the **Remove All** option returning any attribute back to its default state.

Modeling

What is a bounding box?

The bounding box is the virtual rectangular volume encompassing all the geometry in any Mesh Item layer, defined by the maximum and minimum vertex values on all three axes.

What size should I make my object?

In most cases, models are best created at real-world scale. The way lighting attenuates in a scene is all based off real-world values. To get the most realistic results, building to the correct size produces the best results. However, there are situations where working to scale has benefits, such as when modeling anything smaller than a few millimeters in size, when the model should be scaled up, or anything larger than a few city blocks should be scaled down. Items beyond these sizes push the boundaries of Modo's accuracy. You definitely don't want to be modeling sub-atomic particles or entire planets to actual scale.

How do I insert an image into my viewport for modeling?

Users can insert custom background images into a viewport using the Backdrop item added in the **Item List** viewport. Within the Background items properties, specific images can be applied and the settings determine how it's displayed in the 3D viewport. For more information, please refer to Modeling with Images.

How do I change the default behavior for a tool?

By default, tools should remember their settings between uses, so that the last-used settings become the default. If you want to change the system default values for a tool this can be done within the Tool Pipe viewport. With the tool active, and the appropriate new settings defined, right-click over the command name in the Tool Pipe and choose the **Save Preset** option. Keep the default name value in the resulting save dialog to overwrite the default behavior. Make sure only the tool you want to modify is active. Any additional Action Center and falloff settings are included if also active.

Why do my OBJ files import as all triangles? Why can't I edit my imported OBJ file?

Modo has an option to import **.obj** files as a Static Mesh Item, which makes it much easier to deal with multi-million-polygon objects. This option limits the modifications available to only **Items** mode transforms. However, the process also triangulates the geometry when generating the **Static Mesh** from the imported geometry. If you want to further modify imported **.obj** files, you should make sure this option is disabled (the default state) before importing any **.obj** format geometry.

When I duplicate an object it changes size, Why?

Understanding the difference between Modo's **Component** and **Item** selection modes is very important to successfully using Modo. If you aren't sure of the differences, you can read more about it under Understanding Items vs. Components. What can often happen is that an item layer is accidentally scaled while in **Items** mode without realizing it. Subsequent duplicates then default to the zero base transform values, thus changing the size of the duplicated objects. If this happens, you can use the **Freeze** > **Scale** option in the items properties panel before duplicating the item layer.

Why are my Replicators positioned away from the Point Source when I assign the Prototype?

Replicators base their position information in relation to the Prototype's **Center** position. If the geometry was created at an arbitrary position away from the Center, you can use the Tools and Commands to easily reposition the Center, or simply use the Center Commands to center the geometry.

Shading/Rendering

Do I still need to de-gamma my images with Color Management?

No, color management completely eliminates the need to remove gamma from images; this is done automatically in the background. Gamma settings are still present for image adjustment purposes, but should mostly be left at the default **1.0** value. For more information on Color Management, please see Color Management Overview.

I can't get Color Management to turn 'on' when opening older Modo scenes?

To ensure 100% compatibility when rendering older scenes, Color Management is disabled automatically. If you want to manually enable it, the **Gamma** setting on all image maps needs to be returned to 1.0, as well as the output gamma on the Render output. Selecting the **Scene** item (the clapper board icon at the top of the Items list), define a color control in the **OCIO config** section, such as **Nuke Default**. Then, under the Default Colorspace settings for each bit-depth, define the appropriate colorspace, as these settings are then applied to the images in the scene. For more information on Color Management, please see Color Management Overview.

I've applied some preset materials to an object but when I duplicate it, the clone is just gray. Why does this happen?

Material Presets default to limiting their application to a single Mesh Item, by way of an Item Mask. Disabling the Item mode masking function in the material Group item alleviates this problem. Simply change the item name listed under the **Item** to **(all)** to resolve the issue, allowing the textures to globally affect the scene.

How can I get rid of those terrible splotches in my render?

Splotchiness is generally an indicator that the number of Irradiance Rays is too small to accurately depict the surrounding environment when calculating Irradiance Caching. Try increasing the value to better capture the surrounding lighting. Lowering the Irradiance Rate, and increasing the Irradiance Ratio, can also help in certain scenes. Enabling the multi-importance sampling within the global illumination settings can also help to increase the sampling quality of Irradiance Caching global illumination.

Why is my 'X' surface so noisy in a final render?

Noise that is confined to a specific area, be it a shadow, a reflective surface, or a transparent object, is typically the result of the number of samples related to that particular option. Modo offers an extremely flexible shading solution, providing you the ability to adjust the number of rays used to calculate individual effects. The default values provide speedy renders but, on occasion, don't provide enough sampling latitude to produce smooth results.

Lights that generate soft shadows individually have a **Samples** setting that, when raised, increases overall rendering quality. The Material item offers a **Samples** setting when rendering soft reflection, as does the Refraction Roughness setting. Volumetric light does as well. When trying to reduce noise in a scene, typically there is some investigation required to determine the likely culprit. Noise related directly to shading can be reduced by adjusting the Shading Rate, while noise produced by geometry, such as using Replicators of fur, can be reduced by increasing the amount of anti-aliasing.

Note: Keep in mind that when increasing these settings, render times are likely to increase as well.

Why, when I add a Replicator, do my Irradiance pre-pass times go through the roof?

When rendering lots of very tiny details, using Global Illumination, the default calculation method, **Irradiance Caching** (IC) calculates the irradiance value every time a surface normal changes. When there is a lot of detail due to Replicators, Fur, or Displacements, the expensive IC calculation takes much longer to generate. You should assign affected surfaces to use the less-expensive Global Illumination option **Monte Carlo**. To do this:

1. Create a Selection Set of all the affected geometry.
2. Add a Material Group item to the Shader tree and place it above the **Base Shader** item.
3. In the Material Group properties, set the **Polygons Tag Type** to **Selection Set** and under **Polygon Tag** select the previously created Selection Set.

4. Add a Shader to a new Material Group by navigating to **Add Layer** > **Special** > **Shader** and, in the Shaders Properties, set the **Indirect Illumination Type** to **Monte Carlo**.

   Fur, by default, utilizes Monte Carlo shading. The Fur Item has an option **Use Irradiance Cache** that disables this behavior.

My scene won't render. What can I do?

The number one reason Modo usually fails to render a scene is that it has run out of available memory. Often times this is because of too many polygons. To reduce the memory overhead, try:

- Reducing the number of subdivisions for Subdivision Surfaces geometry in the Item properties panel for certain high resolution objects.
- Disabling **Adaptive Subdivision** and setting the subdivision level manually for each item.
- Setting the **Displacement as Bump** option (if using displacement), enabled in Render items under Settings, and reducing the number of subdivisions.

  The resulting image has far fewer polygons but is visually similar to the higher-level render.
- Removing fur outside the camera view on fur items by using the **Fur Frustrum Culling** option.

  If there are no reflective surfaces in the scene this can be a huge memory saver.

Other options include rendering multiple Limited Region sections and assembling them in an image editor.

Plug-Ins

Modo Bridge - A plug-in for the Unreal Engine Editor and Unity.

Modo Material Importer for Unity - A Unity plug-in that sets up standard materials in Unity.

Modo Material Importer for Unreal Engine - A plug-in for Unreal Engine 4 that you can use to load Modo material .xml files.

ProRender Beta - AMD's GPU accelerated path-tracer, the Radeon ProRender is now a part of Modo's collection of powerful rendering options.

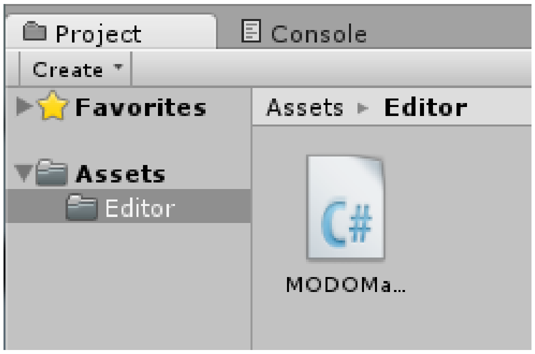
Modo Material Importer for Unity

Modo Material Importer is a Unity plug-in that sets up standard materials in Unity, ensuring that their settings match those of your materials in Modo. If the script is unable to find a viable matching texture in your project, it automatically imports texture files that are stored outside of your Unity project.

Installing the Material Importer

You can download the plug-in from the Unity Asset Store.

After creating a project in Unity, create a folder called **Editor** in the **Assets** folder of your project, and copy the script into it. When you open your project, you can see the material importer script in the bottom area of the interface, under **Assets** > **Editor**.

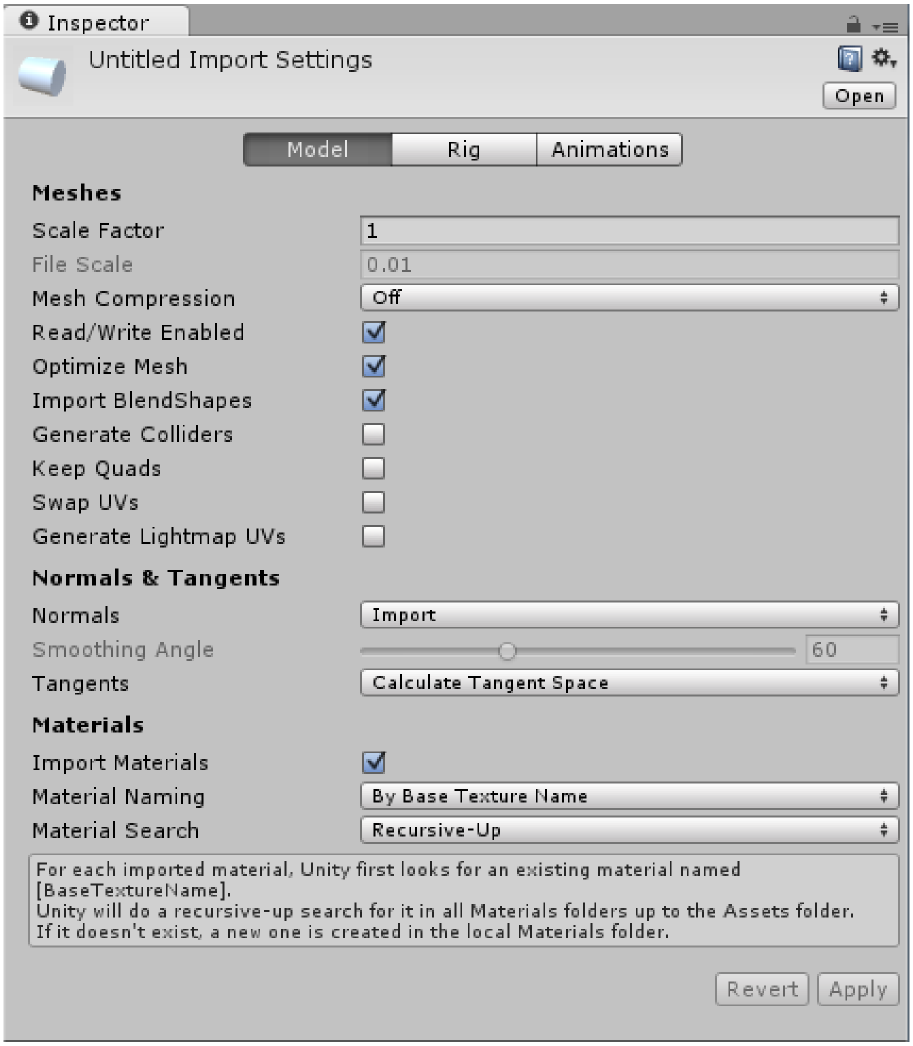


Using the Material Importer

After exporting the **.fbx** and **.xml** files from Modo, place them into the **Assets** folder of your Unity project. The importer automatically detects the **.xml** files that are exported alongside the **.fbx** files from Modo, based on their matching file names.

If either the **.fbx** or the **.xml** file is updated, the plug-in automatically applies the material settings.

The regular Unity settings of your mesh for material naming and texture searching are respected. You can check these by selecting your imported **.fbx** file and opening the **Inspector** tab on the right side of the Unity interface.



Untitled Import Settings

Meshes

Scale Factor: 1
File Scale: 0.01
Mesh Compression: Off
Read/Write Enabled ☑
Optimize Mesh ☑
Import BlendShapes ☑
Generate Colliders ☐
Keep Quads ☐
Swap UVs ☐
Generate Lightmap UVs ☐

Normals & Tangents

Normals: Import
Smoothing Angle: 60
Tangents: Calculate Tangent Space

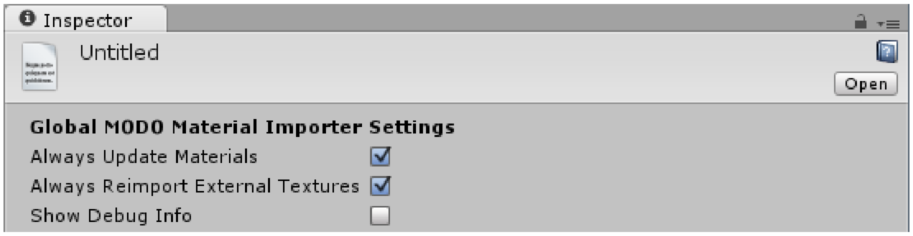
Materials

Import Materials ☑
Material Naming: By Base Texture Name
Material Search: Recursive-Up

For each imported material, Unity first looks for an existing material named [BaseTextureName].
Unity will do a recursive-up search for it in all Materials folders up to the Assets folder.
If it doesn't exist, a new one is created in the local Materials folder.

Importer Settings

There are two global options for the script. You can access these by opening an **.xml** file exported from Modo in Unity's Inspector. The **Inspector** tab appears on the right side of the screen. These options affect all **.xml** files, not just the selected one.



Untitled

Global MODO Material Importer Settings

Always Update Materials ☑
Always Reimport External Textures ☑
Show Debug Info ☐

- **Always Update Material Values** - With this option enabled, the importer always re-imports the material settings. When disabled, the settings of existing materials remain intact, and only new or missing materials have their settings applied from the XML values.

> **Note:** If a matching texture is found inside your project, the importer doesn't attempt to import an external texture to replace it.

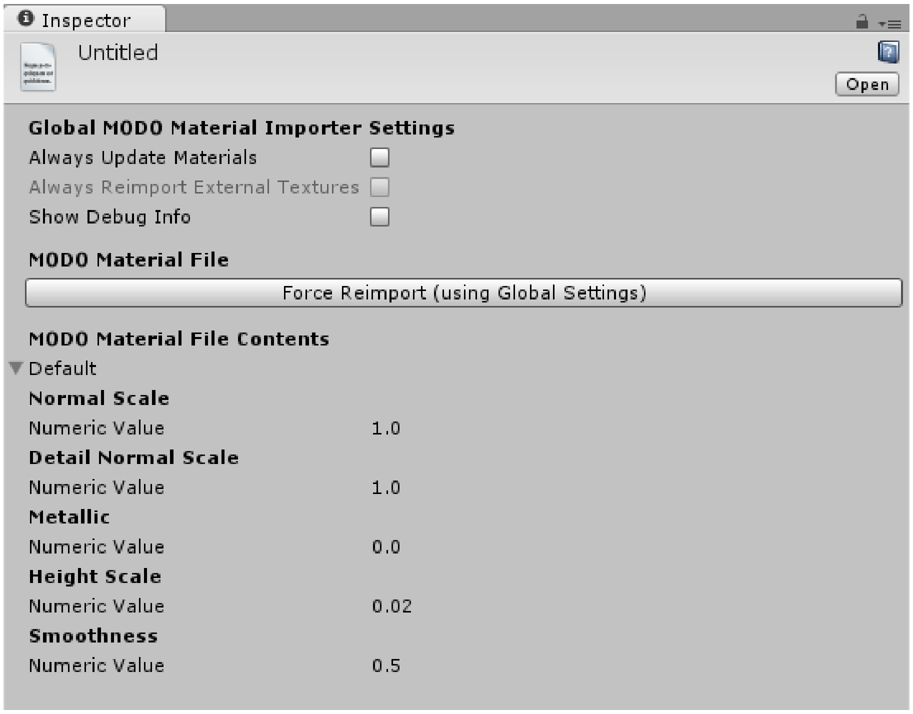
- **Always Reimport External Textures** - When enabled, the plug-in always re-imports textures from an external location. When disabled, if an existing texture is found, the material uses that. Only new or missing textures are imported from an external source.

Script Inspector

Selecting an **.xml** asset displays a custom Inspector containing options for the global settings.

Clicking the **Force Reimport** button re-imports the material settings from the currently selected **.xml** file.

Below are the current settings for the materials defined in the **.xml** file. They cannot be edited, they are for information only.



Global MODO Material Importer Settings

Always Update Materials ☐

Always Reimport External Textures ☐

Show Debug Info ☐

MODO Material File

MODO Material File Contents

▼ Default

Normal Scale

Numeric Value 1.0

Detail Normal Scale

Numeric Value 1.0

Metallic

Numeric Value 0.0

Height Scale

Numeric Value 0.02

Smoothness

Numeric Value 0.5

Material Values

The following rules apply for material parameters:

- **Transparency** - Transparency is currently not supported for Unity in Modo, nor is it exported to the **.xml** file. However, you can use the regular **Transparent Amount** texture effect with **Invert** enabled. To match Unity, set your **Transparent Amount** texture to be the **Albedo** texture, but swizzled to output just the Alpha channel. To do this, check **Swizzling** and set the **Channel** to **Alpha**. Then, in Unity, manually set your material's blend mode to **Fade** or **Transparent**. For more details on channel swizzling, see Channel Swizzling.

Note: The standard shaders in Unity use the alpha of the **Albedo** texture for their opacity mask.

- **Color** - Your material's base Albedo color is only applied if there is no Albedo texture defined in your material. Similarly, your material's **Emissive** color is only applied if your material has an **Emissive** color texture defined. These are both set to white if a texture is defined.
- **Normal Maps** - Textures set to **Normal** or **Detail Normal** are automatically set to normal maps at import if imported from an external location.
- **Height/Emissive** - Most other settings are enabled correctly depending on whether textures or values are present (**Height** for parallax mapping, **Emissive** for lightmap baking).

Exported Materials and Textures

Almost all material settings and textures are exported:

- **Texture Wrap** - In Unity, all of the main texture UVs are derived from the texture settings of the Albedo texture. Similarly, all of the detail texture UVs are derived from the texture settings of the Detail Albedo texture.

  You can export the **Wrap U**/**Wrap V** settings of a texture layer from Modo into Unity.

  However, like Unity, Modo only uses the **Wrap U**/**Wrap V** values sampled from Albedo and Detail Albedo textures. This means that if you have an Albedo with 2.0 **Wrap U** and 2.0 **Wrap V**, but a Normal map with 1.0, when exported to Unity, the Normal map takes the 2.0 **Wrap U** and **Wrap V** settings of the Albedo texture.
- **Detail UV** - Modo assumes that all of your main textures are using the first set of UVs on your mesh. This works similarly in **Texture Wrap** values. If the UV map of your Albedo texture and Detail Albedo texture differ, Modo assumes that your detail map uses a second UV set. The material importer script then sets the detail maps to use the second UV set.

What Doesn't Get Exported

Unity has hard-coded settings for certain texture map channels it takes its values from.

For example, Unity expects the Albedo texture to contain the Albedo color in the RGB channels and the transparency in the alpha channel. If you have your transparency set up to use a different texture in Modo, then it doesn't carry over to Unity, and the alpha channel of the Albedo texture is used instead.

Matching Modo textures with Unity Expectations

The following table contains groups of values and the corresponding texture of Modo, from which the Unity importer takes them. Each group is hard-coded in Unity to come from the same texture.

Value	Texture
Albedo Color and Transparency	Modo: Albedo Albedo Color: RGB Transparency: A
Metalness and Glossiness	Modo: Metalness Metalness: R Glossiness: A
Ambient Occlusion	Modo: Ambient Occlusion Ambient Occlusion: G
Height	Modo: Height Height: G
Detail Mask	Modo: Detail Mask Detail Mask: A

Tip: Modo's channel swizzling feature allows you to set your textures up in Modo the way that Unity requires them to be. For more information, see Channel Swizzling.

Modo Material Importer for Unreal Engine

Modo Material Importer is a plug-in for Unreal Engine 4 that you can use to load Modo material **.xml** files and apply them to 3D meshes of a game level in the Unreal Engine 4 editor.

If you have a Modo scene, you can export 3D meshes in the form of **.fbx** files, materials in **.xml** files, and texture files. You can then apply the exported materials to the **.fbx** 3D meshes in Unreal Engine, using the Material Importer plug-in.

> **Note:** For more information on exporting game assets from Modo, see Game Asset Exporter.
>
> For information on using Modo Bridge to communicate with both the Modo server and an Unreal Engine client, see Modo Bridge for Unreal Engine.

Installing the Material Importer

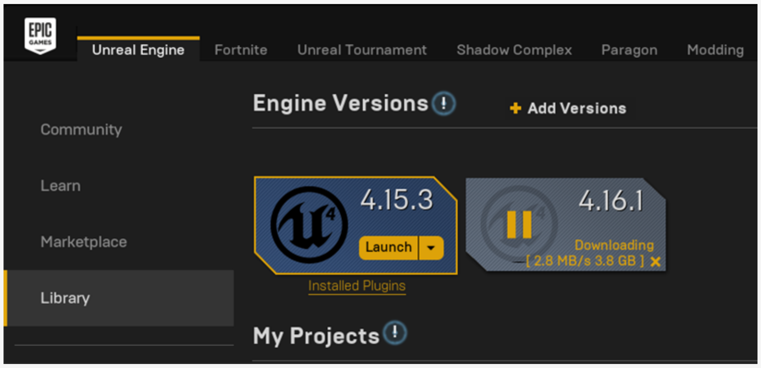
You can download the plug-in from the Unreal Asset store, using the Epic Games Launcher, which is installed on your computer along with Unreal Engine.

Installing from Epic Games Launcher (Recommended)

To install the Modo Material Importer for Unreal from the Epic Games Launcher:

1. Start Epic Games Launcher, then click **Marketplace** and search for **Modo Material Importer**.
2. On the **Content Detail** page, click the **Free** button to download the plug-in, then click **Install to Engine** to automatically install it to Unreal Engine.

Tip: If you currently have an older version of the Epic Games Launcher installed, you can quickly update to the latest Unreal Engine. With the Epic Games Launcher launched, on the left panel, click **Library** and then click **Add Versions** at the top of the view. Locate the latest version and click **Install**.


Unreal Engine
Fortnite
Unreal Tournament
Shadow Complex
Paragon
Modding
Community
Learn
Marketplace
Library
Engine Versions
+ Add Versions
4.15.3
Launch
Installed Plugins
4.16.1
Downloading
[2.8 MB/s 3.8 GB]
My Projects

Manual Install

You can also download the Material Importer from our Git repository.

Note: If you want to use pre-build binaries from the Git repository, please make sure the version matches the major version of your Unreal Engine, for example 4.16.1 the major version is 4.16. You can build the plug-in from the source code. Note, changes may be needed for different Unreal Engine versions you use. For more information, see Code in Plugins.

There are two ways you can install the plug-in manually:

- as a project plug-in
- as an engine plug-in

Installing as a Project Plug-in

This process installs the importer as a project plug-in. This means that the plug-in is installed for the current project only, and you need to install it again for each new project you'd like to use it in.

1. Copy the **Plugins\ModoMaterialImporter** folder to the **Plugins** folder of your Unreal project: <Your Unreal Project>**\Plugins**

Note: If you don't have a **Plugins** folder, you need to create one.

For example, for Unreal Engine 4.9 on Windows, the path looks like this:

C:\Users\<user>**\Documents\Unreal Projects\MyProject\Plugins\**

2. When you open the Unreal Editor, the plug-in loads automatically.

Note: If the plug-in is not loaded automatically, load the project in the Unreal Editor and enable **Edit** > **plugins** > **Modo Material Importer Plugin**.

You need to re-install the plug-in for every new project in which you need it.

Installing as an Engine Plug-in

This process installs the importer as an engine plug-in, which means that once installed, the plug-in is used in all new projects.

Note: Due to module dependencies used by the importer, some constraints in Unreal Engine 4 may prevent the plug-in from working properly. Therefore, if this installation doesn't work for you, please install the material importer as a project plug-in, as described in Installing as a Project Plug-in.

1. Copy the **Plugins\ModoMaterialImporter** folder to the **Plugins** folder of the Unreal Engine files: <Unreal4 install location>**\Engine\Plugins\**

   For example, for Unreal Engine 4.9 on Windows, the path looks like this:

   C:\Program Files\Epic Games\4.9\Engine\Plugins\
2. When you open the Unreal Engine 4 editor, the plug-in is loaded automatically.

Using the Plug-in

The plug-in allows you to import an **.fbx** file and assign materials to it, or import materials directly. You can import materials to any folder you select in Unreal Editor's **Content Browser**.

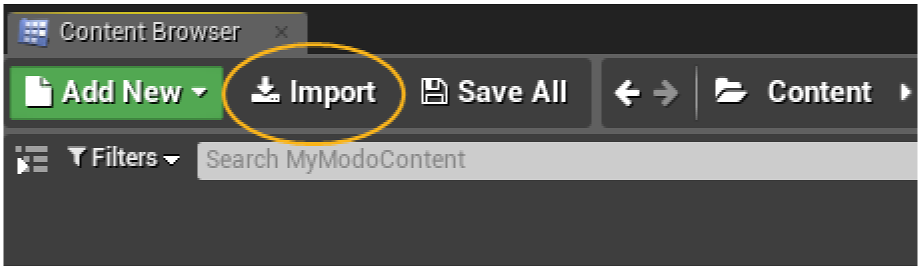
To import the **.fbx** file and assign materials to it:

1. In Unreal Editor's **Content Browser**, navigate to the folder where you want to save the files.

Tip: To create a new folder, right-click in the **Content Browser** and click **New Folder**.

2. Click the **Import** button at the top of the **Content Browser**.

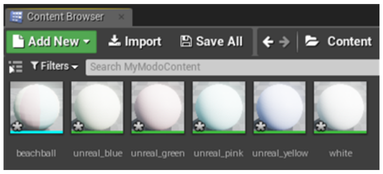
Note: Importing files this way is supported in Unreal Engine 4.15 and later.


Content Browser
Add New
Import
Save All
Content
Filters
Search MyModoContent

A file browser opens.

3. Navigate to your **.fbx** file, select it, and click **Open**.
4. Click **Import All** in the **FBX Import Options** dialog.

   Your mesh and your materials appear in the **Content Browser**.


Content Browser
Add New
Import
Save All
Content
Filters
Search MyModoContent
beachball
unreal_blue
unreal_green
unreal_pink
unreal_yellow
white

The materials imported with the **.fbx** file are dummy materials. The names of these materials are then used as tags for material assignment.

5. Import the materials by clicking **Import** in the **Content Browser**, or clicking the **Modo Material** button in the toolbar.

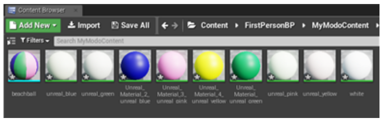
Note: Using the **Import** button does not allow the import of materials from an **.xml** file that has the same filename as the **.fbx** file, unless FBX objects from the **.fbx** file are not saved (marked with an asterisk (*****) in the **Content Browser**).
Modo exports a separate **.xml** file, which uses the suffix **_mat.xml** and saves it in the same directory as the **.fbx** file. For example, **myStuff.fbx** is renamed **myStuff_mat.xml**.
In previous releases, the package names were typeless, meaning that the names of the **.fbx** and **.xml** files were identical. To avoid this problem with older versions of your files, rename the **.xml** file, or use the **Modo Material** button in the toolbar to import the **.xml** file.

6. Navigate to your **_mat.xml** file, select it, and click **Open**.

   Your materials are imported and assigned to the **.fbx** file.

Note: Only Unreal materials are imported and assigned to the **.fbx** file. Other materials are ignored.

Note: Exporting with **Use Material Tag to Match Materials** enabled in the Game Asset Exporter's project preset settings, the dummy materials imported in the **.fbx** file are overwritten by the materials in the **.xml** file. Exporting without this, the dummy materials imported in the **.fbx** are kept, but after importing the materials in the **.xml** files, the **.fbx** meshes use the **.xml** materials instead of the dummy materials.
For more information on this option, see Use Material Tag to Match Materials.


Content Browser
Add New
Import
Save All
Content
FirstPersonBP
MyModoContent
Filters
Search MyModoContent
beachball
unreal_blue
unreal_green
Unreal_Material_2_unreal_blue
Unreal_Material_3_unreal_pink
Unreal_Material_4_unreal_yellow
Unreal_Material_unreal_green
unreal_pink
unreal_yellow
white

To import materials directly into Unreal:

1. In Unreal Editor, click the **Import** button in the **Content Browser**, or the **Modo Material** button in the toolbar to open the plug-in.

   A file browser displays, where you can select your material ***mat.xml** file to import.

Tip: When you export, a separate sidecar **.xml** file is generated with the same name, using the suffix **_mat.xml**, in the same directory as the **.fbx** file. For example, **myStuff.fbx** is renamed **mystuff_mat.xml**.

2. Select your file and click **Open**.

   Your materials are displayed in the **Content Browser**.

   To import more **.xml** files, click the **Modo Material** button again.

Sometimes, you may only want to re-import specific materials that have been updated. To do this, in Unreal's **Content Browser**, select these materials directly or select the imported FBX mesh using these materials, then import the **.xml** file as described above.

Note: You can modify the **.xml** files manually, but you must follow the format, otherwise the files may not load correctly.

Note: The plug-in automatically imports any textures referenced in the **.xml** file, even if the effects are not supported. Instead of linking the textures to material slots, unsupported textures are left as isolated in the material editor.

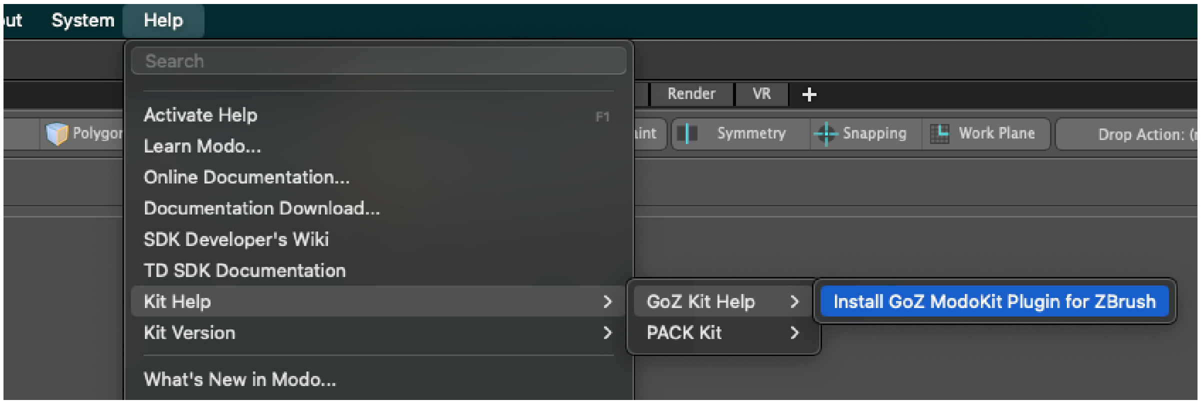
Transferring assets with GoZ Kit

The GoZ Kit, which allows easy transfer of assets between Modo and ZBrush, has been updated and bundled with Modo 17.1.

While the GoZ Kit is built in to Modo, you will need to install the ModoKit to your ZBrush installation and enable transfers from Modo to ZBrush.

To install:

1. Open **Help** in the top bar.
2. Select **Kit Help** > **GoZ Kit Help**.
3. Select **Install GoZ ModoKit Plugin for ZBrush** option to complete the kit setup.


ut
System
Help
Search
Render
VR
+
Polygon
aint
Symmetry
Snapping
Work Plane
Drop Action: (
Activate Help
F1
Learn Modo...
Online Documentation...
Documentation Download...
SDK Developer's Wiki
TD SDK Documentation
Kit Help
Kit Version
What's New in Modo...
GoZ Kit Help
PACK Kit
Install GoZ ModoKit Plugin for ZBrush

To enable transfers:

To enable transfer from ZBrush to Modo, you will need to update your ZBrush Preferences to point to this new Modo version.

1. That setting can be found in the ZBrush System Menu.
   Select: **Preferences** > **GoZ** > **path to Modo (New)**.
2. You can then pick a Modo build from the list (which scans for installed versions of Modo), or click 'Browse...' and navigate to the location of your Modoinstall directly.

Article: For more information and troubleshooting - Q100331: Troubleshooting the GoZ plug-in for Modo.

Modo Bridge

Modo has the ability to communicate with Unreal Engine and Unity in real-time. To improve the workflow for game engine users, you can use a network protocol to share data between applications. The connection can be on the same machines, across multiple machines on the LAN, or even possibly remotely. An entire scene or selected elements from a scene can bi-directionally be pushed and updated between the client and server. Mesh geometry, normals, UVs, textures, materials, as well as scene hierarchy, lights, and cameras can be synchronized between Modo and either game engine.

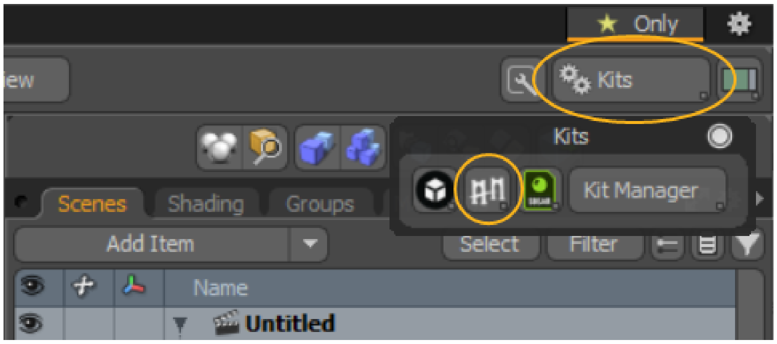
Modo provides an Unreal Engine and a Unity client, **Modo Bridge**, as a plug-in for Unreal Engine and Unity respectively. Once downloaded and installed, you can start the communication between both applications. You can then push changes made to the connected application.

The benefits of using this tool:

- Transferring mesh items, vertex maps, materials and textures, from Modo to Unreal Engine/Unity.
- Transferring static mesh actors and vertex maps from Unreal Engine/Unity to Modo.

- Transferring modified selected **Polygons**, **Edges**, and **Vertices** from Modo to Unreal Engine/Unity. For more information, see Selecting Items.
- Transferring data changes made in Unreal Engine/Unity to Modo. For example, repositioning a mesh.
- Transferring data using a separate server and client computers on the same network.
- Transferring items, materials, and textures from multiple Modo scenes to Unreal Engine/Unity.

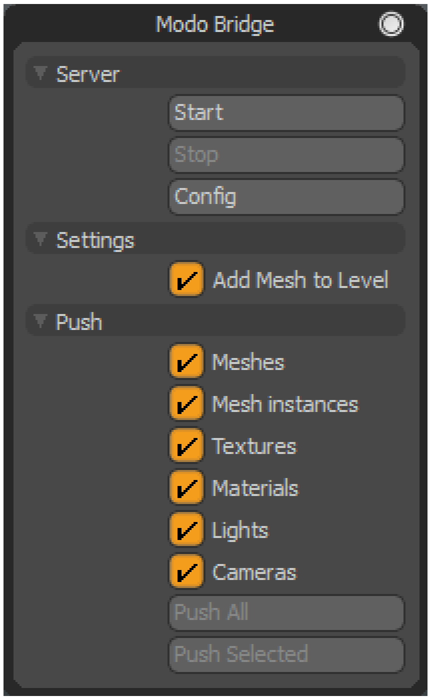
To access Modo Bridge in Modo, in the top right corner of the interface, click **Kits**, then click the Modo Bridge button.



For information on how to use Modo Bridge with your chosen game engine, refer to the Modo Bridge for Unreal Engine or the Modo Bridge for Unity topic.

Modo Bridge Options

When you open the **Modo Bridge** panel, you can find the following options:


Modo Bridge
Server
Start
Stop
Config
Settings
☑ Add Mesh to Level
Push
☑ Meshes
☑ Mesh instances
☑ Textures
☑ Materials
☑ Lights
☑ Cameras
Push All
Push Selected

Server	
Start	Starts the communication between Modo and the game engine client using a network protocol.
Stop	Stops the communication between Modo and the game engine client.
Config	Allows you to configure the server's listening IP address and port number for incoming client connections. These are the same settings configured in the **System** > **Preferences** > **Default** > **Application** > **Foundry Link Settings** category. For more details, see Defaults Preferences.

	Note: You cannot change the IP address and port information while the Modo Bridge is connected.
Settings	
Add Mesh to Level	When enabled, meshes pushed to the game engine are added to the game engine's content browser, as well as the current level. When disabled, meshes are only added to the content browser.
Push	
Meshes/Mesh instances / Textures /Materials/Lights/Cameras	Select which scene items are transferred. Click **Push All** or **Push Selected** to push to the game engine client.
Push All	After specifying which items to push, Meshes/Mesh instances/Textures/Materials/Lights/Cameras (meshes, cameras, and lights), Modo Bridge, and/or Modo Bridge, all corresponding items in your scene are pushed to the game engine client. All items pushed are converted to actors in the game engine client. **Note:** If multiple checkboxes are checked, then the order of pushing is always **Meshes**, **Textures**, and then **Materials**.
Push Selected	After specifying which items to push, items selected in the Shader Tree or the **Items** list in Modo are pushed to the game engine client. All pushed items are converted to actors in the game engine client. **Tip:** You can select individual geometry items or selected polygons, edges, or vertices, materials, and textures to transfer them.

Modo Bridge for Unreal Engine

Modo has the ability to communicate with the Unreal Engine in real-time and allows you to transfer meshes, Instances, textures, materials, and cameras between Modo and the Unreal Engine.

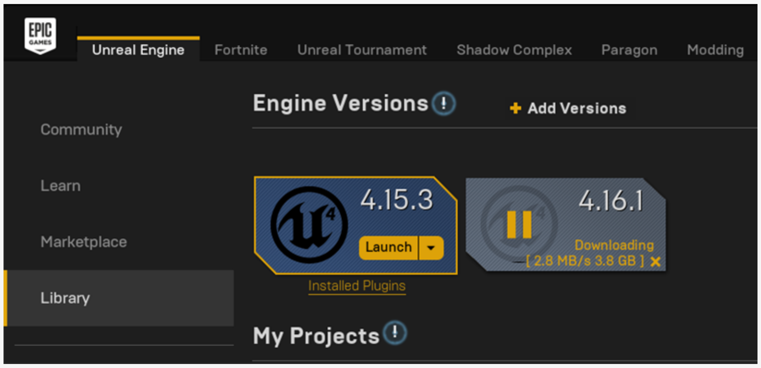
Installing the Modo Bridge Plug-in

Installing from Epic Games Launcher

To install the Modo Bridge plug-in for Unreal from the Epic Games Launcher:

1. Start Epic Games Launcher, then click **Marketplace** and search for **Modo Bridge**.
2. On the **Content Detail** page, click the **Free** button to download the plug-in, then click **Install to Engine** to automatically install it to the Unreal Engine.

Tip: If you currently have an older version of the Epic Game Launcher installed, you can quickly update to the latest Unreal Engine. With the Epic Games Launcher running, on the left panel, click **Library** and then click **Add Versions** at the top of the view. Locate the latest version and click **Install**.


EPIC GAMES
Unreal Engine
Fortnite
Unreal Tournament
Shadow Complex
Paragon
Modding
Community
Learn
Marketplace
Library
Engine Versions
+ Add Versions
4.15.3
Launch
4.16.1
Downloading
[2.8 MB/s 3.8 GB]
Installed Plugins
My Projects

Installing Modo Bridge as a Project Plug-in

This process installs **Modo Bridge** plug-in as a project plug-in. This means that the plug-in is installed for the current project only, and you need to install it again for each new project you'd like to use it in.

1. Copy the **Plugins\ModoBridge** folder to the **Plugins** folder of your Unreal project: <Your Unreal Project>**\Plugins**

Note: If you don't have a **Plugins** folder, you need to create one.

For example, for Unreal Engine 4.16 the path looks like this:

Windows:

C:\Users\<user>**\Documents\Unreal Projects\MyProject\Plugins\**

Mac OS:

/Users/<username>**/Documents/Unreal Projects/<Your Unreal Project>/Plugins/**

Note: Create the **Plugins** folder, if it does not exist.

2. When you open the Unreal Editor, the plug-in loads automatically.

Note: If the plug-in is not loaded automatically, load the project in the Unreal Editor and enable **Edit** > **Plugins** > **Modo Bridge Plugin**.

Note: You need to re-install the plug-in for every new project in which you need it.

Installing Modo Bridge as an Unreal Engine Plug-in

This process installs the Modo Bridge plug-in as an Unreal Engine plug-in, which means that once installed, the plug-in is used in all new projects.

Note: Due to module dependencies used by the Modo Bridge, some constraints in Unreal Engine 4 may prevent the plug-in from working properly. Therefore, if this installation doesn't work for you, please install Modo Bridge as a project plug-in, as described in Installing Modo Bridge as a Project Plug-in.

1. Copy the **Plugins\ModoBridge** folder to the **Plugins** folder of the Unreal Engine files: <Unreal4 install location>**\Engine\Plugins\**

   For example, for Unreal Engine 4.16 the path looks like this:

Windows:

C:\Program Files\Epic Games\EU_4.16\Engine\Plugins\

Mac OS:

/Users/Shared/Epic Games/UE_4.16/Engine/Plugins/

2. When you open the Unreal Engine 4 editor, the plug-in is loaded automatically.

Installing Modo Material Importer (Optional)

You can also install the Modo Material Importer plug-in on your Unreal Engine client to transfer material updates.

For more information, see Installing the Material Importer.

Note: If the plug-in is not loaded automatically, load the project in the Unreal Editor and enable **Edit** > **plugins** > **Modo Material Importer Plugin**.

Compiling Modo Bridge from Source with Unreal Engine

This process compiles the plug-in source with Unreal Engine 4 for your operating system and installs Modo Bridge plug-in as a project plug-in or as an Unreal Engine plugin.

Compiling with a new C++ Unreal Engine project

1. Create a new C++ project to compile the plug-in source.
2. Download the plug-in package from here and extract it.
3. Copy the **ModoBridge** folder to the **Plugins** folder of your new C++ Unreal project: **<Your Unreal Project>\Plugins\**

Note: If you don't have a **Plugins** folder, you need to create one.

For example, for Unreal Engine 4.20 the project Plugins path looks like this:

Windows:

C:\Users\<user>\Documents\Unreal Projects\MyProject\Plugins\

Mac OS:

/Users/<username>/Documents/Unreal Projects/<Your Unreal Project>/Plugins/

4. Open the C++ project in your IDE of choice and start the compiling process.

   To start the compiling process on Windows:

   Open the C++ Unreal project and in the Visual Studio 2015 or 2017 main menu click **Build** > **Build Solution**.

   To start the compiling process on Mac OS:

   Open the C++ Unreal project and in the XCode main menu, click **Product** > **Build**.

5. When compiling is finished, copy the **ModoBridge** folder from the new C++ Unreal project **Plugins** folder to the **Plugins** folder of your Unreal project, or to the **Plugins** folder of your Unreal Engine. Copying to the **Plugins** folder of your Unreal Engine enable the plug-in for all Unreal projects.

   For example, for Unreal Engine 4.20 the path looks like this:

   Windows:

   C:\Program Files\Epic Games\EU_4.20\Engine\Plugins\

   Mac OS:

   /Users/Shared/Epic Games/UE_4.20/Engine/Plugins/

Compiling with the Unreal Engine source

Note: Compilation with Unreal Engine source may take longer.

1. Follow the instructions on how to download and build Unreal Engine from source.
2. Download the plug-in package from here and extract it.
3. Copy the **ModoBridge** folder to the **Plugins** folder of the Unreal Engine files: **<Unreal4 install location>\Engine\Plugins\**

   For example, for Unreal Engine 4.20 the project Plugins path looks like this:

   Windows:

   C:\Program Files\Epic Games\EU_4.20\Engine\Plugins\

   Mac OS:

   /Users/Shared/Epic Games/UE_4.20/Engine/Plugins/

4. Open the Unreal Engine C++ project in your IDE of choice and start the compiling process.

   To start the compiling process on Windows:

   Open the C++ Unreal project and in the Visual Studio 2015 or 2017 main menu click **Build** > **Build Solution**.

   To start the compiling process on Mac OS:

   Open the C++ Unreal project and in the XCode main menu, click **Product** > **Build**.

5. When compiling is finished, open the Unreal Editor to automatically load the plug-in.

Configuring the Server and Client

Both the Modo server and the Unreal Engine client must use the same port number and corresponding addresses to communicate with each other. The following topics outline the setup and connection process.

Configuring Modo Preferences

To configure Modo:

1. Launch **Modo**.
2. Click **Systems** > **Preferences** and expand **Application** on the left panel.
3. Under **Foundry Link Settings**, in the **Address** field, type the TCP server IP address.

   By default the TCP server address is **127.0.0.1**. This is the loopback Internet Protocol (IP) address used to point back to your computer's TCP/IP network configuration.
4. In the **Port** field, type the port number for your TCP connection.

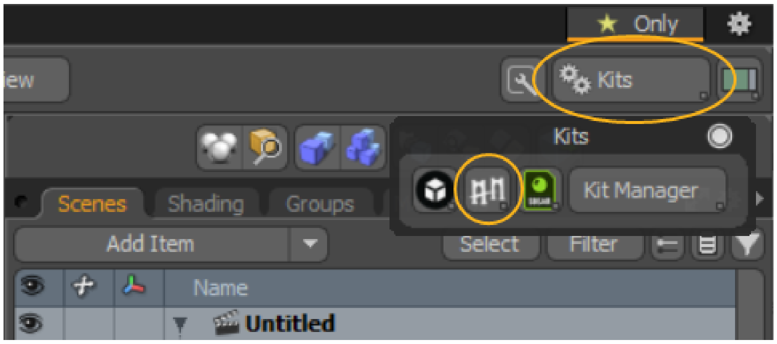
   By default, the **Port** number is **12000**.

Starting the Communication between Modo and the Unreal Engine Client

Note: You must enable the TCP protocol communication on the Modo server or on the Unreal Engine client.

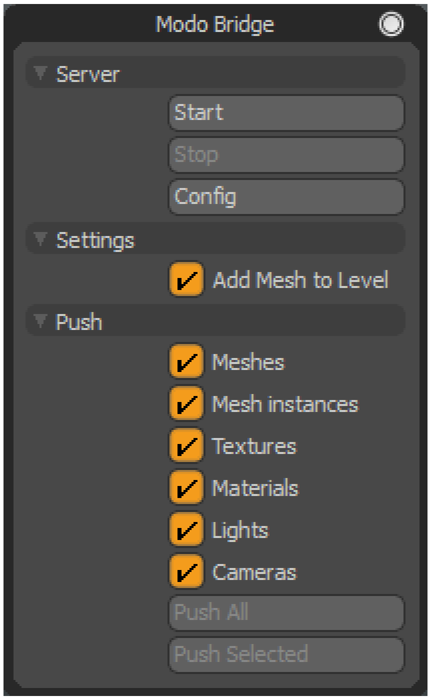
To enable communication on the Modo server:

1. Launch Modo.
2. In the top-right corner of the UI, click **Kits**, then click the Modo Bridge button.



   This opens the **Modo Bridge** options.

Note: For more information on each option, see Modo Bridge.



Modo Bridge
Server
Start
Stop
Config
Settings

☑ Add Mesh to Level

Push

☑ Meshes
☑ Mesh instances
☑ Textures
☑ Materials
☑ Lights
☑ Cameras

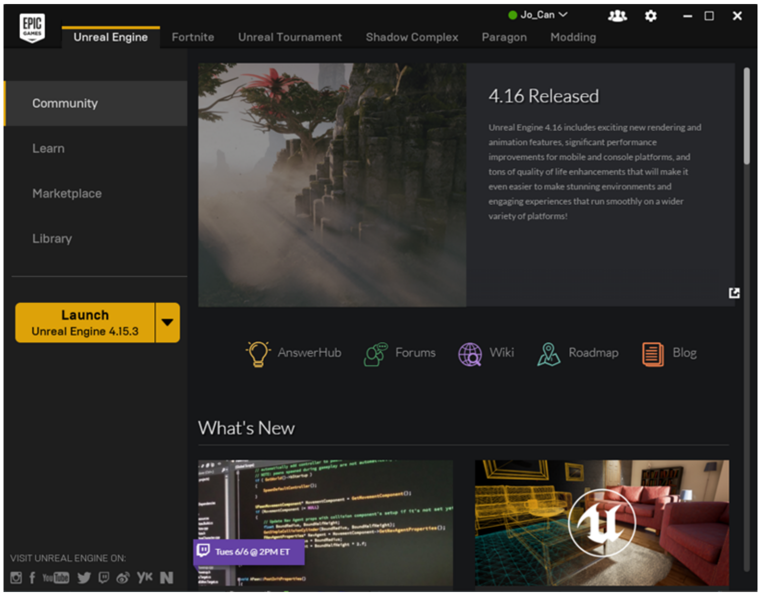
Push All
Push Selected

3. Under **Server**, click **Start**.

   The Modo Bridge server is connected to the IP address and port specified in the **Preferences** and waits for incoming client connections.

To enable communication on the Unreal client:

1. Launch **Epic Games** and then click **Launch Unreal Engine**.



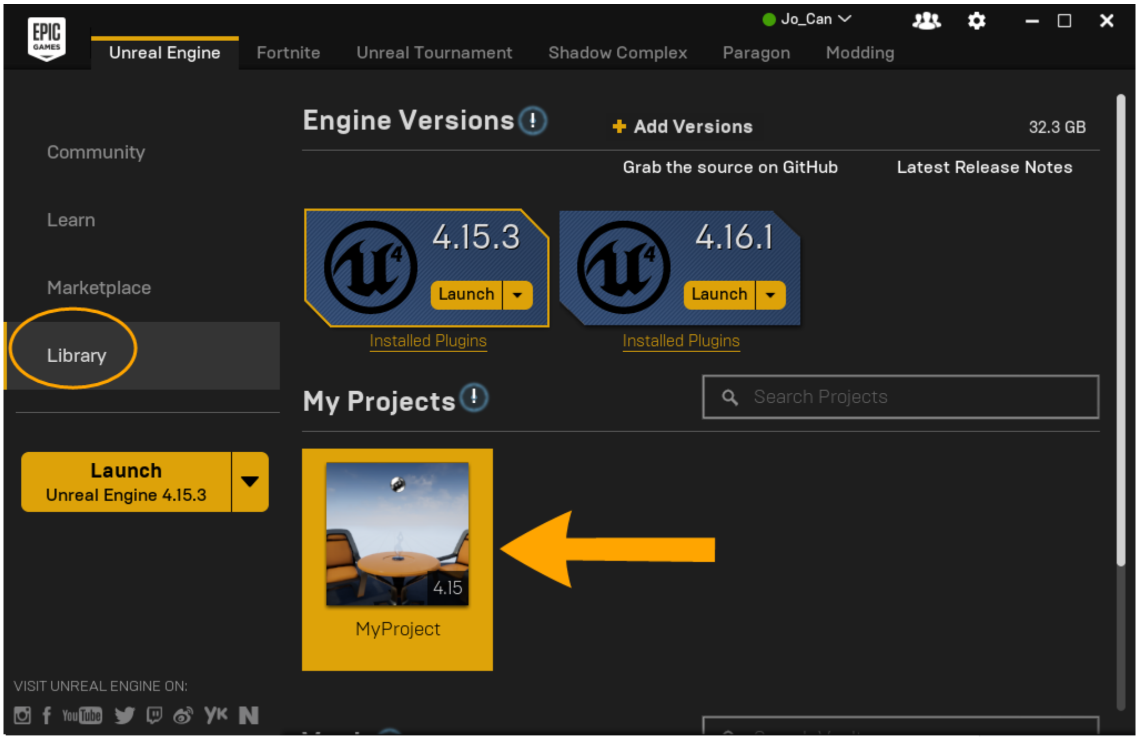
EPIC GAMES
Jo_Can
Unreal Engine
Fortnite
Unreal Tournament
Shadow Complex
Paragon
Modding
Community
Learn
Marketplace
Library
Launch
Unreal Engine 4.15.3
AnswerHub
Forums
Wiki
Roadmap
Blog
What's New
Tues 6/6 @ 2PM ET
VISIT UNREAL ENGINE ON:

4.16 Released

Unreal Engine 4.16 includes exciting new rendering and animation features, significant performance improvements for mobile and console platforms, and tons of quality of life enhancements that will make it even easier to make stunning environments and engaging experiences that run smoothly on a wider variety of platforms!

The Projects tab displays MyProjects.

2. On the left panel, click **Library** and then click **MyProject**.

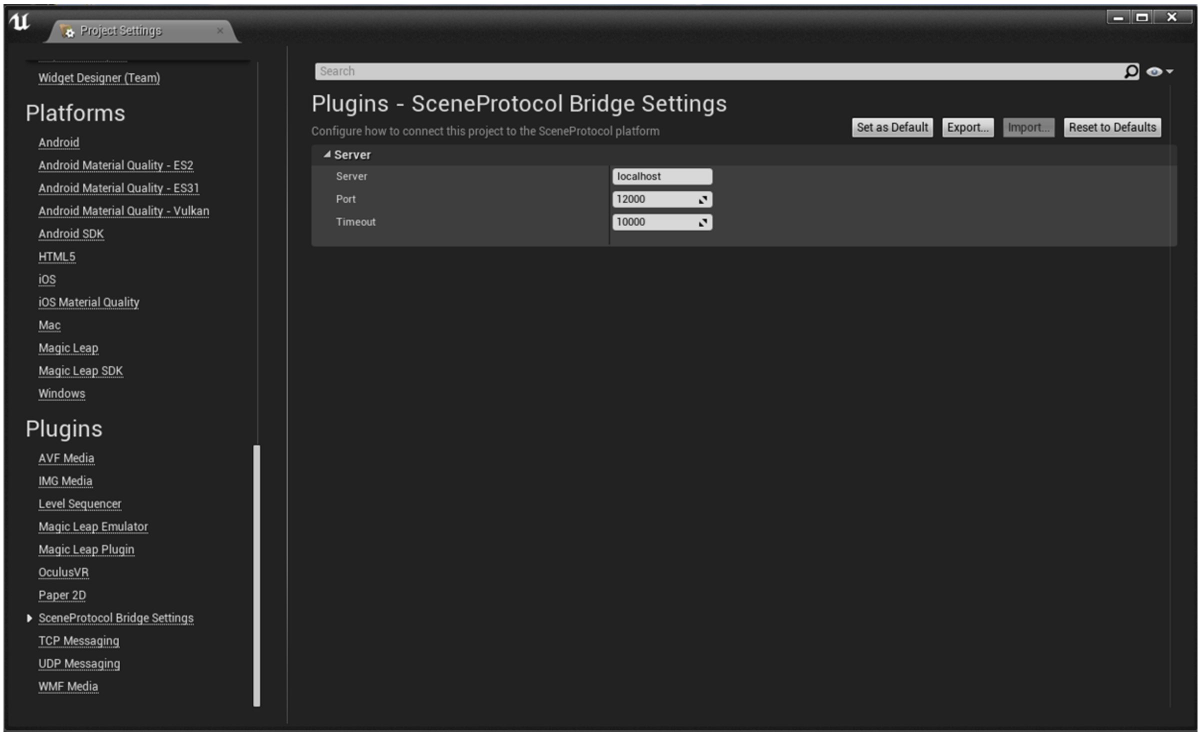

Jo_Can
Unreal Engine
Fortnite
Unreal Tournament
Shadow Complex
Paragon
Modding
Community
Learn
Marketplace
Library
Launch
Unreal Engine 4.15.3
Engine Versions
Add Versions
32.3 GB
Grab the source on GitHub
Latest Release Notes
4.15.3
Launch
Installed Plugins
4.16.1
Launch
Installed Plugins
My Projects
Search Projects
4.15
MyProject
VISIT UNREAL ENGINE ON:

3. In the Unreal Engine menu bar, click **Edit** > **Project Settings**.

   This opens the **Project Settings** dialog

4. Under **Plugins**, click **Scene Protocol Bridge Settings** and enter the **Server**, **Port**, and **Timeout** information.

Note: The information used by the Modo server is automatically displayed.


Project Settings
Search
Widget Designer (Team)
Platforms
Android
Android Material Quality - ES2
Android Material Quality - ES31
Android Material Quality - Vulkan
Android SDK
HTML5
iOS
iOS Material Quality
Mac
Magic Leap
Magic Leap SDK
Windows
Plugins
AVF Media
IMG Media
Level Sequencer
Magic Leap Emulator
Magic Leap Plugin
OculusVR
Paper 2D
SceneProtocol Bridge Settings
TCP Messaging
UDP Messaging
WMF Media
Set as Default
Export...
Import...
Reset to Defaults

Plugins - SceneProtocol Bridge Settings

Configure how to connect this project to the SceneProtocol platform

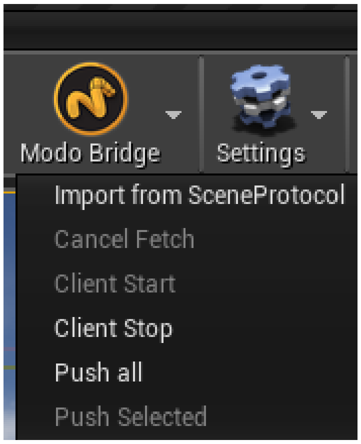
Server

Server: localhost

Port: 12000

Timeout: 10000

5. On the top of the Unreal Engine Editor, click **Modo Bridge** and click **Client Start**.


Modo Bridge
Settings
Import from SceneProtocol
Cancel Fetch
Client Start
Client Stop
Push all
Push Selected

The Unreal client connects to the server's IP address and port. The communication between Modo and Unity is now established.

Using the Modo Bridge Plug-in

Updates made in both applications, Modo and the Unreal Engine client, are transferred simultaneously using the **Push** functionality. You have the ability to select individual geometry items or selected polygons, edges, or vertices, materials, and textures to transfer them to the other application.

The following procedures provide workflow examples.

Creating Geometry and a Material

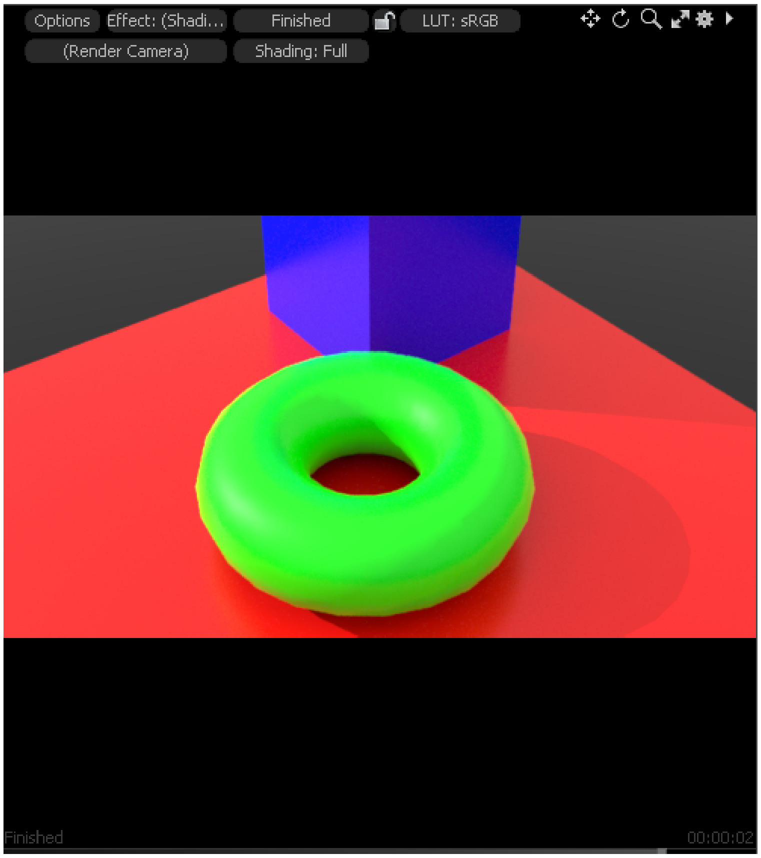
First, let's load an example scene containing basic primitives and assigned materials.

1. Launch **Modo** and open the **Model** layout.

   If you don't have a scene, you can download our example scenes and load **unreal_bridge_example_scene.lxo**.

2. Click **Render** > **Open Preview** (alternatively press **F8**) and click the arrow to render your scene.

   These are the mesh items and the assigned materials that we will use to transfer to the Unreal Engine client.


Options
Effect: (Shadi...
Finished
LUT: sRGB
(Render Camera)
Shading: Full
Finished
00:00:02

Push the Scene From Modo to the Unreal Engine Client

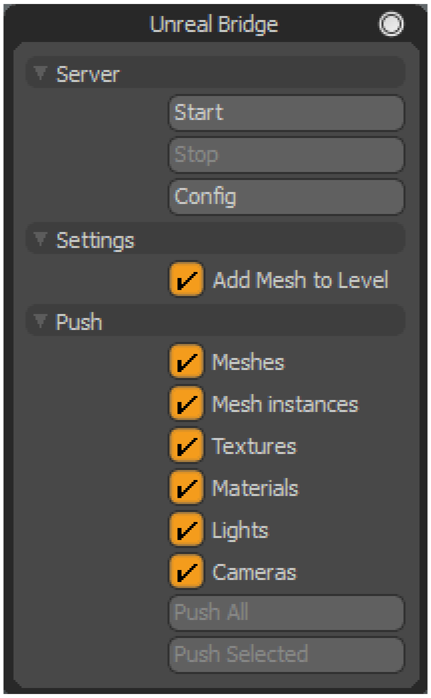
In this procedure, we will walk you through the process of transferring Modo geometry and materials to the Unreal Engine client.

Note: You must complete the steps of Configuring the Server and Client before doing the following procedure.

To push a scene from Modo:

1. In Modo, in the top-right corner, click **Kits**, then click the Unreal Engine button.

   This opens the Modo Bridge.



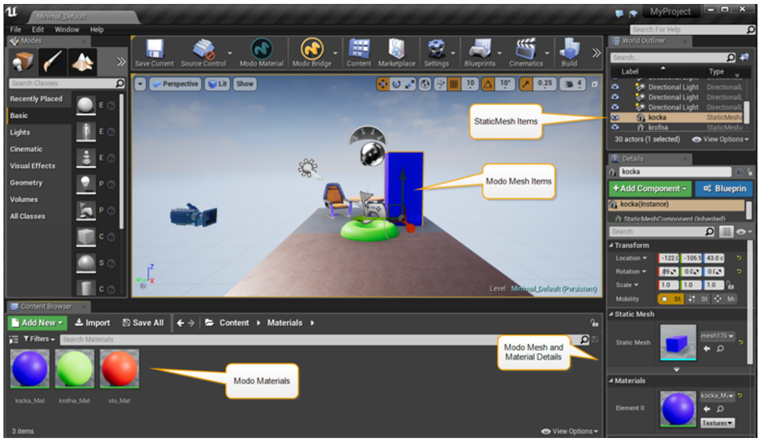
☑ Add Mesh to Level

☑ Meshes
☑ Mesh instances
☑ Textures
☑ Materials
☑ Lights
☑ Cameras

2. Under the **Push** category, enable **Meshes**, and click **Push All**.
3. Open your Unreal Engine project.

   The Modo geometry is added to your scene.
4. Next, update your materials by enabling **Push** > **Materials** and clicking **Push All** or select the materials you want to update and click **Push** > **Push Selected**.

Note: If the materials are not assigned to geometry, empty materials are created, otherwise, existing materials are used.



All transferred assets are stored on the Unreal Engine client in the **Content Browser** in the **Scene Protocol** folder.

Tip: You can push the **Camera** from Modo to the Unreal Engine client, if you want to have the same view in both applications.

If you have textures applied to geometry in your scene, enable **Push** > **Textures** and then click **Push All**. The textures are then found and assigned to the appropriate materials.

Push a Transformed Static Mesh from Unreal to Modo

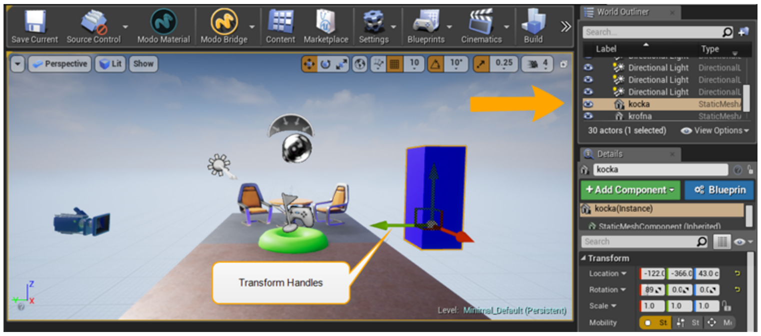
In this procedure, we will walk you through the process of changing the position of one of the mesh items in Unreal and transferring the updates back to Modo.

Tip: You can use the **Modo Material Importer** plug-in to export modified materials made in Unreal and then import them into Modo.

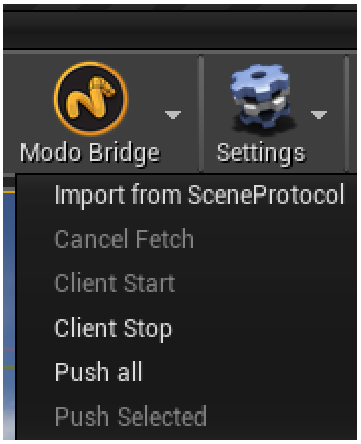
To push a transformed mesh from Unreal:

1. With the Unreal Engine Editor open and your project displayed, on the right panel select the **Kocka** item.
2. Use the transform handles in the view to move this item to a new location.

   The **Kocka** item is displayed in a new position in your project scene.


Save Current
Source Control
Modo Material
Modo Bridge
Content
Marketplace
Settings
Blueprints
Cinematics
Build
Perspective
Lit
Show
World Outliner
Details
kocka
+Add Component
Blueprint
Transform
Location
Rotation
Scale
Mobility
Transform Handles

3. On top right corner of the interface, click **Save Current**.
4. On the top of the Unreal Engine Editor, click **Modo Bridge** and click **Push Selected**.


Modo Bridge
Settings
Import from SceneProtocol
Cancel Fetch
Client Start
Client Stop
Push all
Push Selected

5. Open your Modo session to view the updates.

Push Transformed Polygons from Modo to Unreal

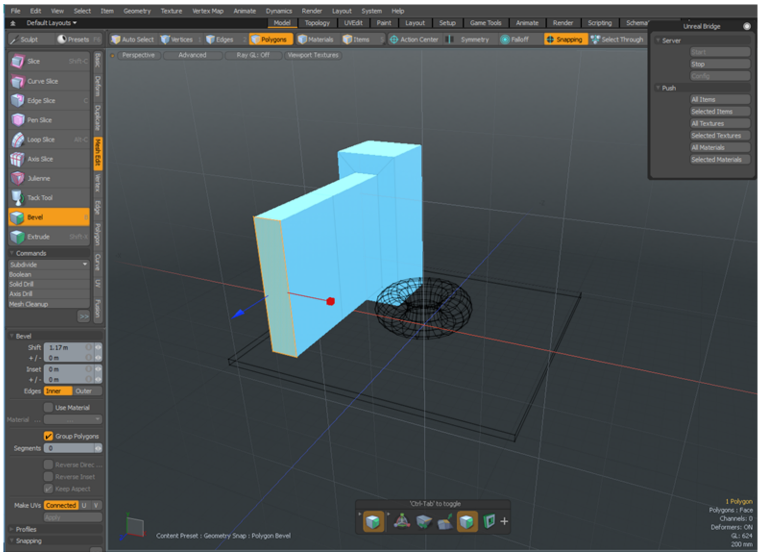
In this procedure, we will walk you through the process of transforming a selection of Polygons in Modo, using the Bevel tool, and transferring the changes to Unreal.

> **Tip:** You can also transform **Edges** and **Vertices** on a **Mesh** item in Modo and push your changes back to the Unreal Engine client.

To push transformed polygons from Modo:

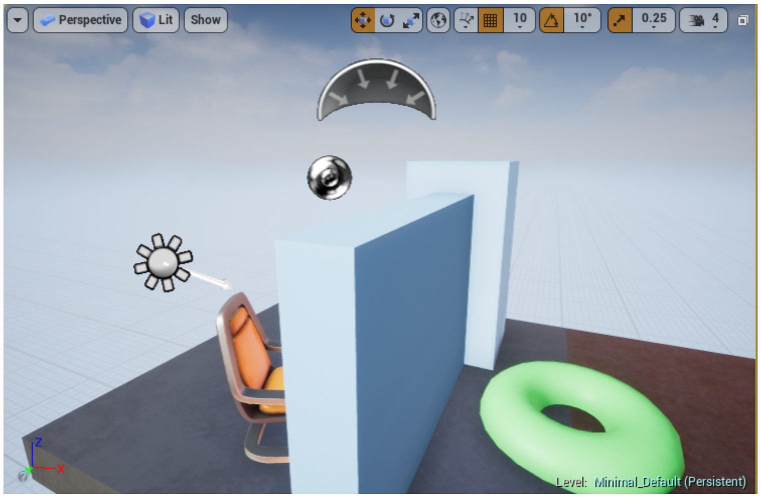
1. In Modo, while in **Model** layout, click **Polygons** to change your selection mode.
2. Select one polygon on the **kocka** mesh item.
3. On the Toolbar, located on the left panel, open the **Mesh Edit** sub-tab and click **Bevel**.
4. Click and hold the transform handle and drag.

   The selected polygon is updated.



5. Click **File** > **Save**.
6. From the **Modo Bridge** options, click **Push** > **All Items**.
7. Open the Unreal Engine client.

   The mesh is updated in the Unreal Engine client.


Perspective
Lit
Show
10
10°
0.25
4
Level: Minimal_Default (Persistent)

Push a UV Map from Modo to Unreal

In this procedure, we will walk you through the process of transferring a UV Map from Modo to Unreal.

Note: Modo meshes pushed into the Unreal Engine client have auto generated UVs applied to the **Light Map**, which always overwrites UV Channel 1. You can disable this behavior by turning off the auto generator for the **Light Map** in Unreal Engine 4.

To push a UV map from Modo:

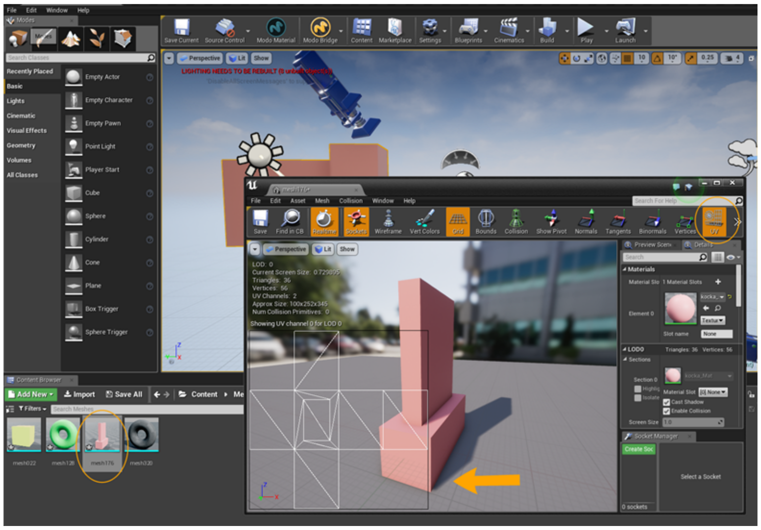
1. In Modo, change your layout to **UVEdit**.
2. On the Toolbox, located on the left panel, open the **UV** sub-tab and select **Map** > **(new)**.

   The **Create New Vertex Map** windows displays.
3. Type a name for your Vertex Map, ensure the **Vertex Map Type** is set to **UV Map**, and then click **OK**.

> **Tip:** You can also use other **Vertex Map Types**, such as **RGB Map** and set the **Color** value. This data can also be pushed to the Unreal Engine client. To see the color specified for the RGB Map in Unreal, open the **StaticMesh** item in the **Content Browser** > **Meshes** directory and click **Vert Colors**.

4. On **Items** tab, on the right panel, select **kocka** and **krofna**.
5. On the Toolbox, located on the left panel, click **Project**.
6. Click **File** > **Save**. Alternatively, press **Ctrl/Cmd** + **S**.
7. In the **Modo Bridge** options window, click **Push** > **All Items**.
8. Open your Unreal Engine client, in the **Content Browser** located at the bottom of the view, open **Content** > **Scene Protocol** > **Meshes** and double-click on a Mesh item.
9. In the preview window, click **UV**.

   The UV Map displays.



Modo Bridge for Unity

Modo has the ability to communicate with Unity in real-time and allows you to transfer meshes, mesh instances, textures, materials, and cameras between Modo and Unity.

Installing the Modo Bridge Plug-in

Installing from the Unity Asset Store

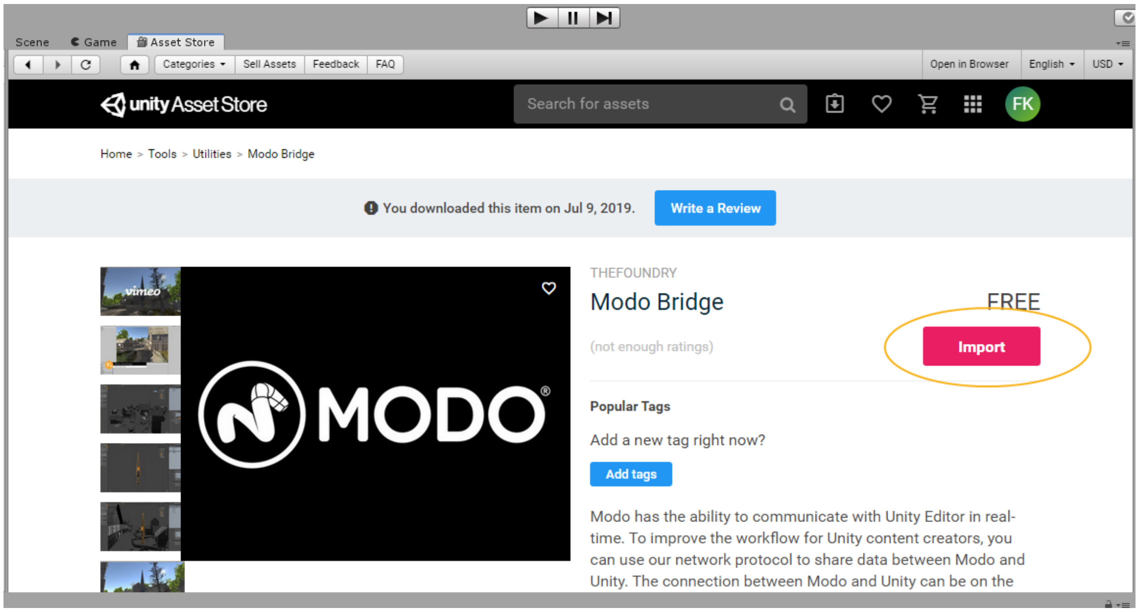
You can download the Modo Bridge plug-in from the Unity Asset Store.

1. Click **Open in Unity**.

   This opens the Unity project browser.

2. Select a project to open or create a new project.

   This opens the project and displays the store page for the Modo Bridge plug-in in a Unity browser.



Scene
Game
Asset Store
Categories
Sell Assets
Feedback
FAQ
Open in Browser
English
USD
unity Asset Store
Search for assets
FK
Home > Tools > Utilities > Modo Bridge
Write a Review
THEFOUNDRY
Modo Bridge
FREE
(not enough ratings)
Import
Popular Tags
Add a new tag right now?
Add tags
vimeo
MODO

You downloaded this item on Jul 9, 2019.

Modo has the ability to communicate with Unity Editor in real-time. To improve the workflow for Unity content creators, you can use our network protocol to share data between Modo and Unity. The connection between Modo and Unity can be on the

3. Click **Import**.

   The Import Unity Package dialog opens.

4. Click **Import** again.

   The plug-in is imported into Unity. A **ModoBridge** folder is added in the project's **Assets** folder.

5. Find your Unity project folder on your local drive and create a folder called **Plugins** in the **Assets** folder.

   Windows: **C:\Users\**<username>**\Documents**<Unity Project Name>**\Assets\Plugins** macOS: **/Users/**<username>**/**<Unity Project Name>**/Assets/Plugins/**

6. Place the **ModoBridge** folder into the **Plugins** folder of your Unity project.

   Modo Bridge for Unity is now installed.

Configuring the Server and Client

To get started, the Modo Bridge needs to connect through the network. Both the Modo server and the Unity client must use the same port number and corresponding addresses to communicate with each other.

In most cases where Modo and Unity are started on the same machine, it is sufficient to just start the Modo Bridge server and the Modo bridge client in Unity. In more complex cases where the connection is made on the local network or over the internet, the **Address** and **Port** options in Modo, and the **Server IP/name** and **Port** options in Unity should be changed according to the network and firewall specifications.

For example, for LAN networks, **Server IP/name** could be "192.168.1.100" or "192.168.20.32" , and for internet connections it could be different, like "216.27.61.137".

Here's how to configure the server and the client:

Configuring Modo Preferences

To configure Modo:

1. Launch **Modo**.
2. Click **Systems** > **Preferences** and under **Defaults**, click **Application**.
3. Under **Foundry Link Settings**, in the **Address** field, type the TCP server IP address.

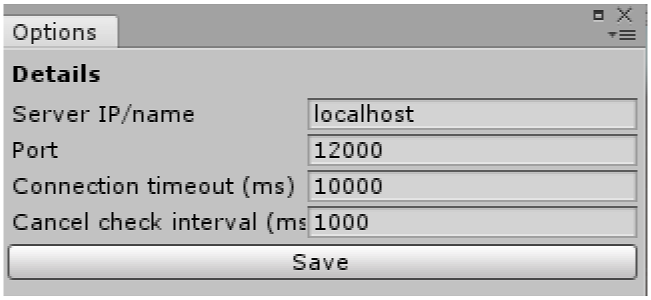
   By default the TCP server address is **127.0.0.1**. This is the loopback Internet Protocol (IP) address used to point back to your computer's TCP/IP network configuration.
4. In the **Port** field, type the port number for your TCP connection.

   By default, the **Port** number is **12000**.

Configuring the Unity Client

1. In Unity's menu bar, click **Tools** > **Modo Bridge** > **Set Client Options**.

Note: This option is only accessible while the client is off.



Options

Details

Server IP/name localhost

Port 12000

Connection timeout (ms) 10000

Cancel check interval (ms 1000

Save

2. Set the followings:

   Server IP/name: **127.0.0.1**

   Port: **12000**

   Other settings you can edit are:

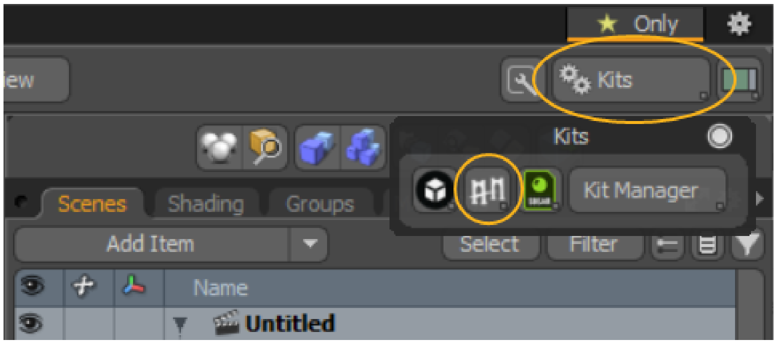
   - **Connection timeout (ms)**- Sets the number in milliseconds when the client should break the connection with the server if no response has been received.
   - **Cancel check interval (ms)** - Sets the number in milliseconds when the client should check if the user clicked the **Cancel Import** command.

3. Click **Save**.

Starting the Communication between Modo and the Unity Client

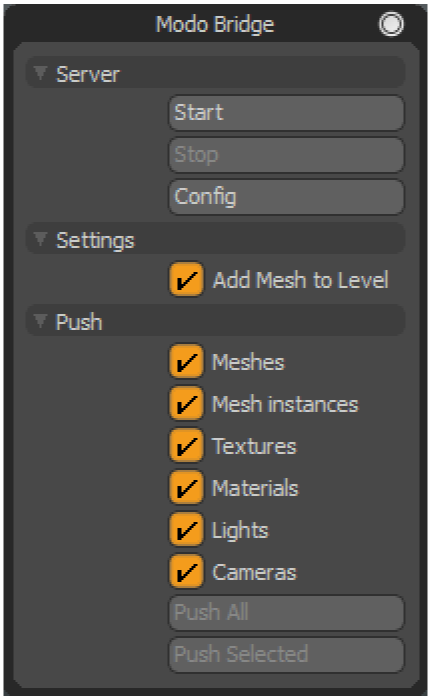
To enable communication on the Modo server:

1. Launch Modo.
2. In the top-right corner of the UI, click **Kits**, then click the Modo Bridge button.



   This opens the **Modo Bridge** options.

Note: For more information on each option, see Modo Bridge.



Modo Bridge

Server

Start

Stop

Config

Settings

☑ Add Mesh to Level

Push

☑ Meshes

☑ Mesh instances

☑ Textures

☑ Materials

☑ Lights

☑ Cameras

Push All

Push Selected

3. Under **Server**, click **Start**.

   The Modo Bridge server is connected to the IP address and port specified in the **Preferences** and waits for incoming client connections.

To enable communication on the Unity client:

1. Launch Unity.

   The **Projects** tab displays your available projects.

2. Click the project you want to load.

   Unity opens the selected project

3. In the menu bar, click **Foundry** > **Start Client**.

   The Unity client connects to the server's IP address and port. The communication between Modo and Unity is now established.

Using the Modo Bridge Plug-in

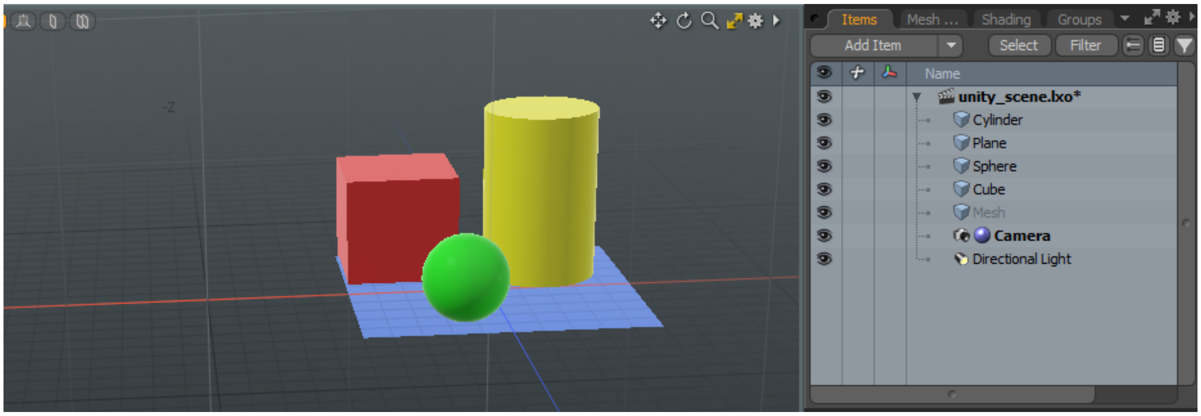
Updates made in both Modo and Unity are transferred simultaneously using the **Push** functionality. You can select individual geometry items or selected polygons, edges, or vertices, materials, and textures to transfer them to the other application.

By using unique Foundry Tokens, items created in Modo and sent to Unity preserve their mapping and reconnect in the event of server-client connection break, level, or item renames, changes in the scene graph hierarchy, and so on. Taking the examples below, pushing the **Cylinder** mesh always updates the **Cylinder** GameObject in Unity, rather than creating a new cylinder GameObject.

Pushing Objects from Modo to Unity

Meshes can be transferred dynamically from Modo to Unity. Mesh transfer supports geometry data, as well as normal and UV vertex maps and surface sets used for materials.

Note: You must complete the steps of Configuring the Server and Client before doing the following procedure.


Items
Mesh ...
Shading
Groups
Add Item
Select
Filter
Name
unity_scene.lxo*
Cylinder
Plane
Sphere
Cube
Mesh
Camera
Directional Light

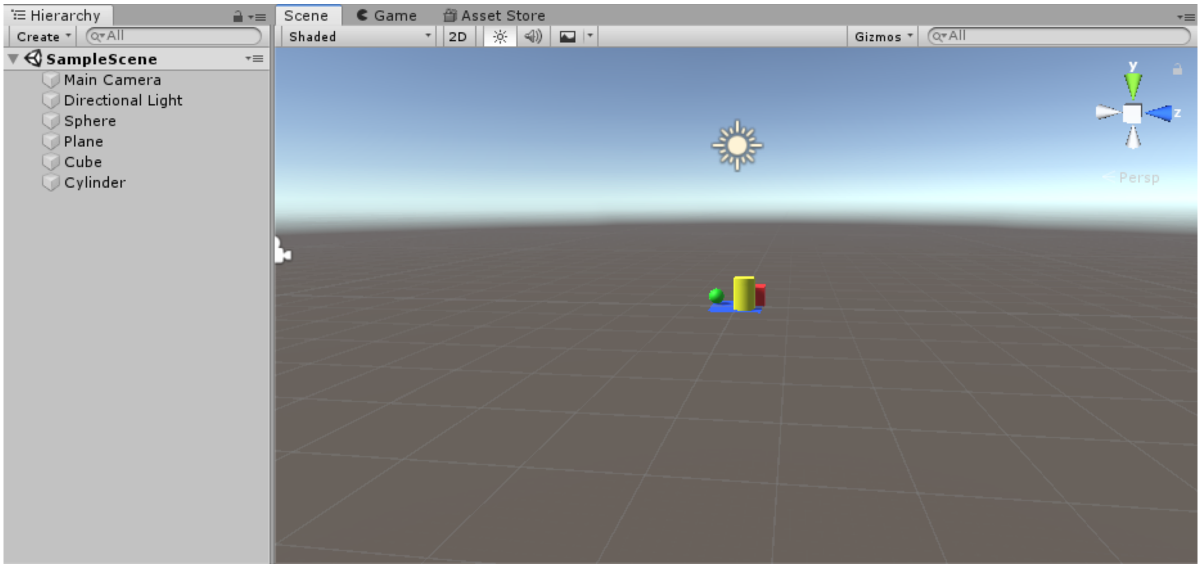
To push a scene from Modo:

1. In Modo, in the top-right corner, click **Kits**, then click the Modo Bridge button.

   This opens the **Modo Bridge** options dialog.

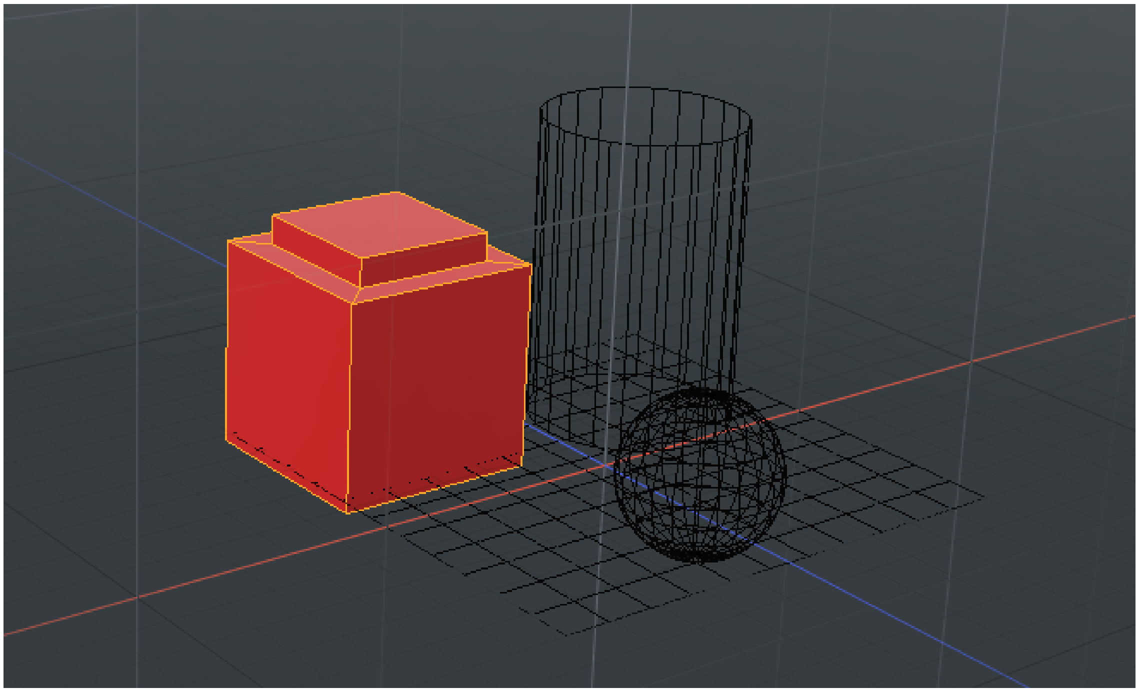
2. Under the **Push** category, enable the objects you want to transfer, and click **Push All**.

   The items are transferred to your Unity project.


Hierarchy
Create
All
SampleScene
Main Camera
Directional Light
Sphere
Plane
Cube
Cylinder
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
Persp

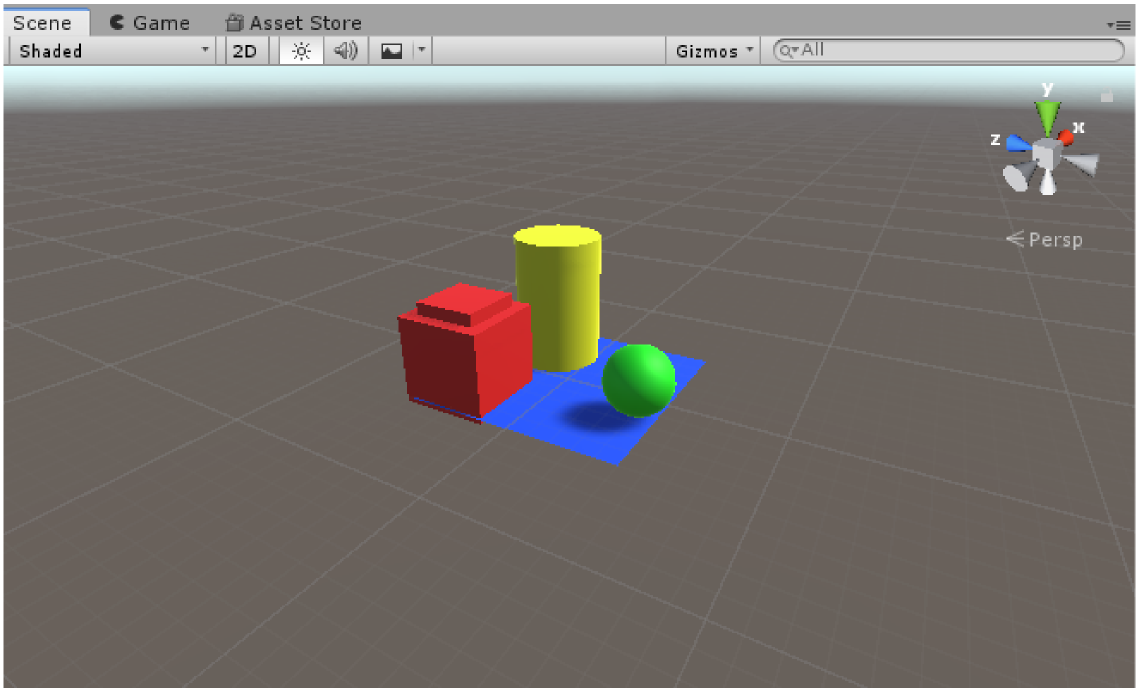
Transferring meshes from Modo to Unity generates new Prefab assets with Unity mesh objects as children in the **Project** browser. This workflow preserves mesh data, creates Unity GameObject components and settings connecting materials to Unity sub-meshes (material surfaces of the mesh). Unique Foundry Tokens are created to preserve links between objects in Modo and Unity, and the fully shaded mesh GameObject becomes available to be used in the level.

If you make any modifications to your scene in Modo, you can repeat the transfer process by transferring only the modified item. In this example, we bevelled the top of the cube:



1. Save your changes in Modo.
2. Select the item, and in the **Modo Bridge** dialog, click **Push Selected**.

   The item is updated in your Unity scene.


Scene
Game
Asset Store
Shaded
2D
Gizmos
All
y
x
z
Persp

Pushing Objects from Unity to Modo

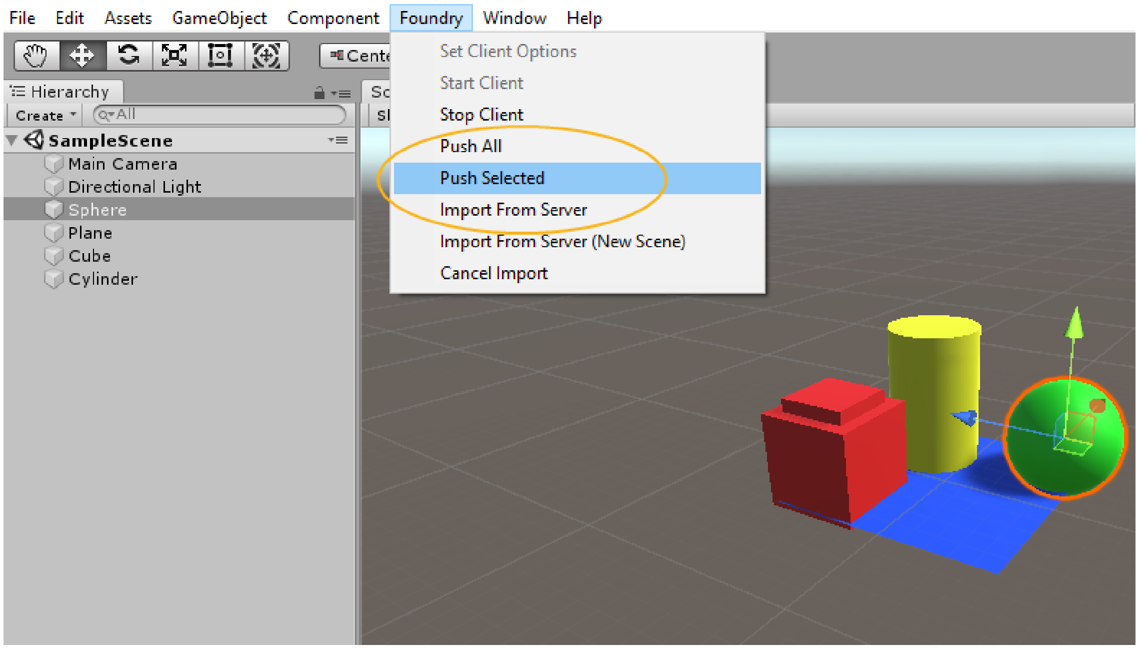
When you've made changes to your transferred scene in Unity, you can update your scene in Modo. You can transfer the entire scene, or only selected objects.

To transfer selected objects:

1. Select your object in Unity.
2. In the Unity menu bar, click **Foundry** > **Push Selected**.

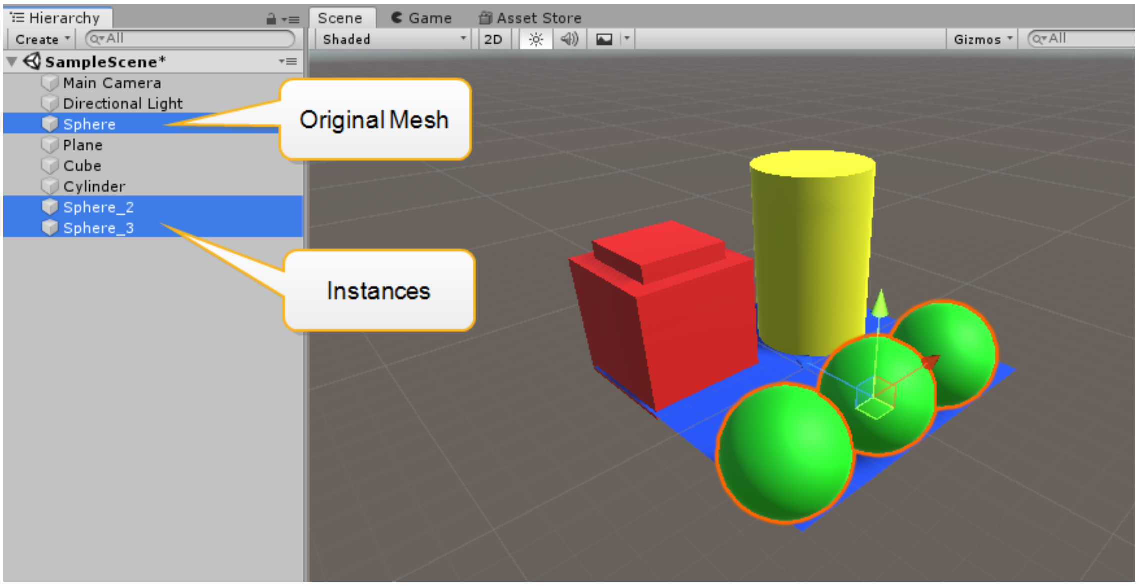
   Your scene is updated in Modo.

To transfer the entire scene, in the menu bar, click **Foundry** > **Push All**.


File
Edit
Assets
GameObject
Component
Foundry
Window
Help
Set Client Options
Start Client
Stop Client
Push All
Push Selected
Import From Server
Import From Server (New Scene)
Cancel Import
Hierarchy
Create
SampleScene
Main Camera
Directional Light
Sphere
Plane
Cube
Cylinder

Transferring Mesh Instances

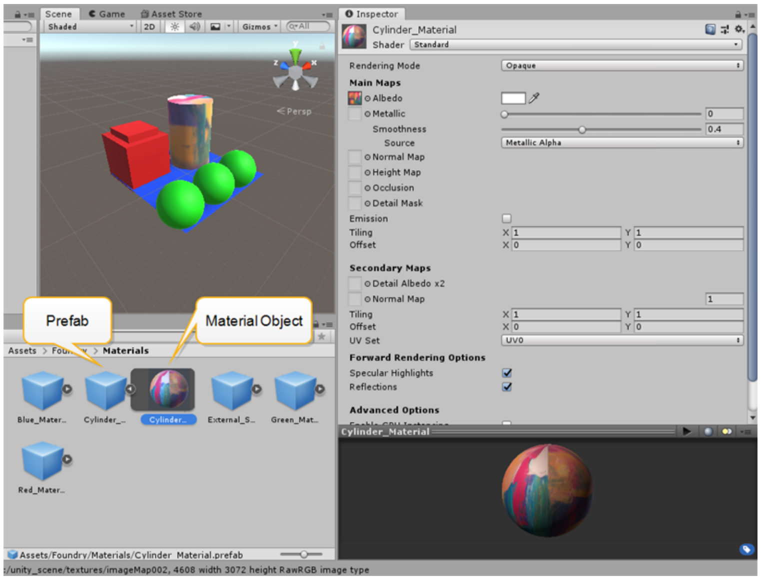
You can transfer Mesh Instances back and forth between the two application, just like regular meshes. In Modo they are displayed as mesh Instance items and in Unity they appear as Unity GameObjects in the current level.


Hierarchy
Create
SampleScene*
Main Camera
Directional Light
Sphere
Plane
Cube
Cylinder
Sphere_2
Sphere_3
Scene
Game
Asset Store
Shaded
2D
Gizmos
Original Mesh
Instances

Note: Meshes or mesh Instances in Modo only generate a scene graph hierarchy in the current level in Unity when the GameObject is first created, not on subsequent updates. There is limited support for meshes and mesh Instances parented to other meshes and mesh Instances.

Transferring Materials

When you transfer materials from Modo to Unity, Unity generates Prefabs with material objects as children. Textures are sent alongside material data and automatically connect to the resulting material in Unity with the Unity standard shader.



Cylinder_Material
Shader: Standard
Rendering Mode: Opaque
Main Maps
Albedo
Metallic: 0
Smoothness: 0.4
Source: Metallic Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission ☐
Tiling X 1 Y 1
Offset X 0 Y 0
Secondary Maps
Detail Albedo x2
Normal Map: 1
Tiling X 1 Y 1
Offset X 0 Y 0
UV Set: UV0
Forward Rendering Options
Specular Highlights ☑
Reflections ☑
Advanced Options
Cylinder_Material

Assets/Foundry/Materials/Cylinder_Material.prefab
:/unity_scene/textures/imageMap002, 4608 width 3072 height RawRGB image type

Note: Subsequent material pushes are currently not supported. You can only update materials by first deleting the created material in Unity.

Note: When transferring materials without mesh data, the materials don't automatically get connected to the mesh, you have to assign the material to the required mesh in Unity.

The currently supported Unity material data are:

- Base Albedo color value
- Metallic float value
- Smoothness float value
- Emissive color value

The currently supported Unity texture effects are:

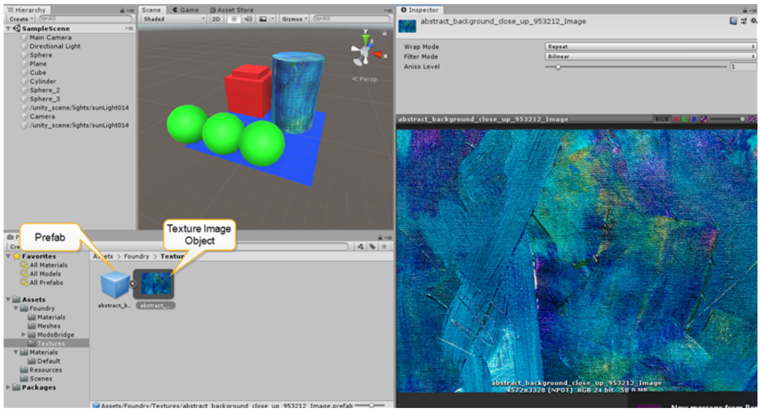
- Unity Albedo
- Unity Bump
- Unity Metallic
- Unity Normal
- Unity Ambient Occlusion
- Unity Emission
- Unity Detail Mask
- Unity Detail Albedo x2
- Unity Detail Normal

Note: The above two data sets are currently mutually exclusive. For example, Base Albedo is overridden by Unity Albedo texture effect.

Transferring Texture Images

When you transfer texture images from Modo to Unity, Unity generates Prefabs with texture image objects as children.

Once you've saved your images in Modo, click the **Push All** button or select the texture effect in the Shader Tree in Modo and click **Push Selected**. This creates or updates the texture in Unity.



Subsequent transfers of the same texture effect update the already existing texture image in Unity.

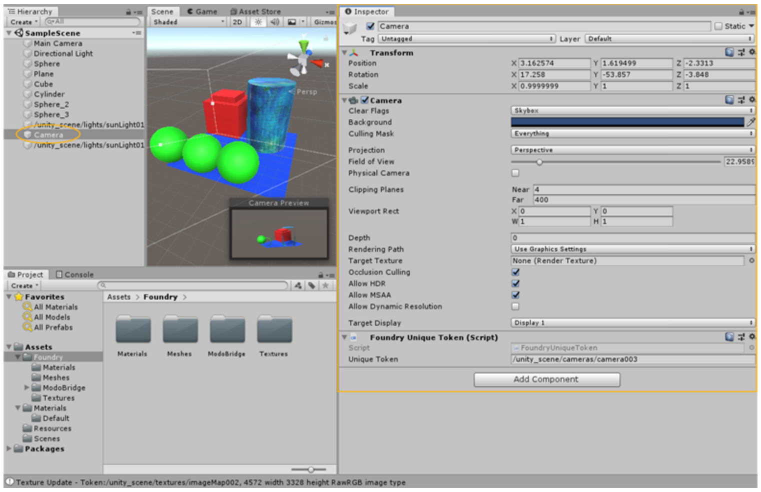
Note: Make sure that you save the edited image before transfer.

Transferring Camera Data

You can transfer camera data from Modo to Unity. Unity generates a GameObject in the current open level with the camera component. The only supported **Projection Type** is **Perspective**.

The following camera data are transferred:

- Angle of View (Field of View)
- Near Clipping Plane
- Aspect Ratio


Hierarchy
Create
All
SampleScene
Main Camera
Directional Light
Sphere
Plane
Cube
Cylinder
Sphere_2
Sphere_3
/unity_scene/lights/sunLight01
Camera
/unity_scene/lights/sunLight01
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp
Camera Preview
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Foundry
Materials
Meshes
ModoBridge
Textures
Materials
Default
Resources
Scenes
Packages
Assets > Foundry >
Materials
Meshes
ModoBridge
Textures
Inspector
Add Component

☑ Camera ☐ Static
Tag: Untagged Layer: Default

Transform
Position X 3.162574 Y 1.619499 Z -2.3313
Rotation X 17.258 Y -53.857 Z -3.848
Scale X 0.9999999 Y 1 Z 1

☑ **Camera**
Clear Flags: Skybox
Culling Mask: Everything
Projection: Perspective
Field of View: 22.9589
Physical Camera: ☐
Clipping Planes: Near 4, Far 400
Viewport Rect: X 0 Y 0 W 1 H 1
Depth: 0
Rendering Path: Use Graphics Settings
Target Texture: None (Render Texture)
Occlusion Culling: ☑
Allow HDR: ☑
Allow MSAA: ☑
Allow Dynamic Resolution: ☐
Target Display: Display 1

Foundry Unique Token (Script)
Script: FoundryUniqueToken
Unique Token: /unity_scene/cameras/camera003

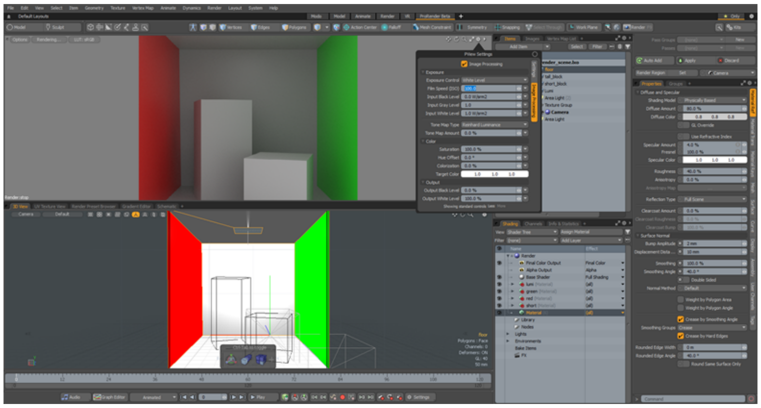
```
Texture Update - Token:/unity_scene/textures/imageMap002, 4572 width 3328 height RawRGB image type
```

ProRender Beta

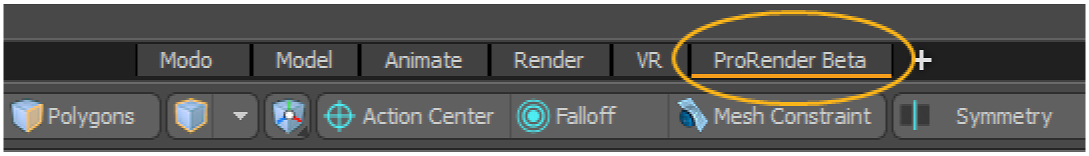
ProRender Beta is a kit that contains the AMD Radeon™ ProRender plug-in for Modo. It's a GPU renderer that uses OpenCL/Metal compute APIs to perform ray-tracing. The ProRender kit is developed for Windows and macOS and supports both AMD and NVIDIA graphics cards.

Note: NVIDIA graphics cards only work on Windows due to a driver limitation in macOS.

Note: The current version of the Radeon™ ProRender kit is SDK version 1.35.1.



The kit has a dedicated **ProRender Beta** layout, where you can access the PView Viewport. To open the **ProRender Beta** layout, click the **ProRender Beta** tab on the switcher bar, or in the menu bar, click **Layout** > **Layouts** > **ProRender Beta**.


Modo
Model
Animate
Render
VR
ProRender Beta
+
Polygons
Action Center
Falloff
Mesh Constraint
Symmetry

Note: To learn more about the controls in the **ProRender Beta** viewport, see PView Viewport.

Note: For information on ProRender's preferences, settings, and limitations, see ProRender Settings and Limitations.

Installing ProRender Beta

Windows

1. Download the correct **.msi** installation file from our website at www.foundry.com.

   The file is saved to your preferred download location.
2. Double-click on the file to install ProRender.
3. Follow the on-screen instructions in the setup dialog.

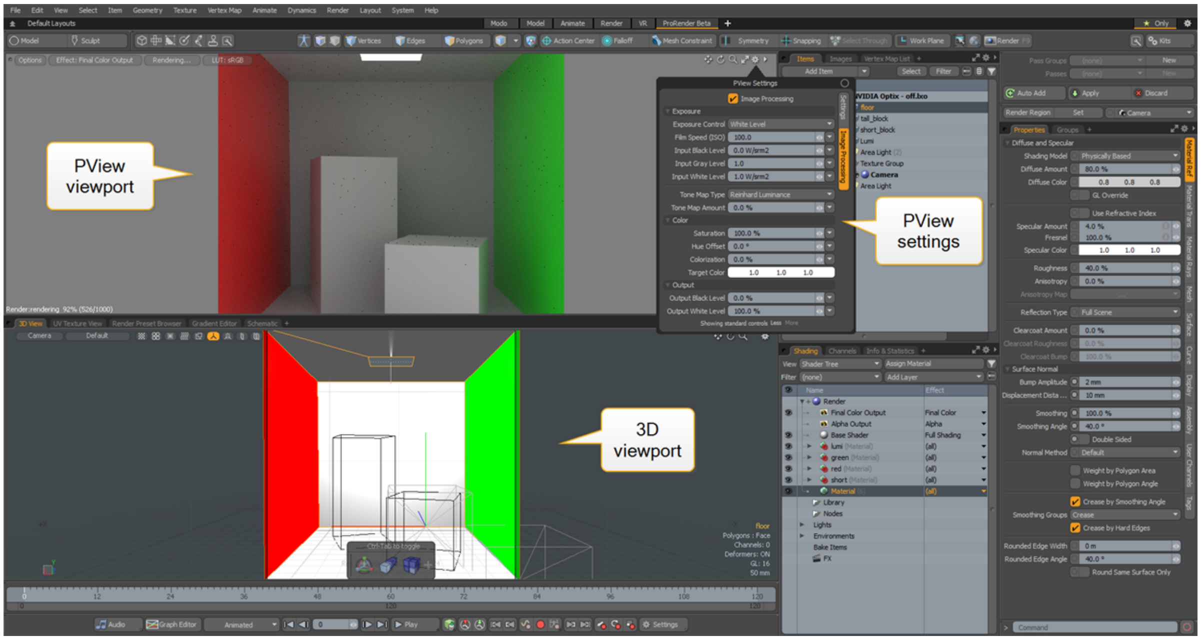
   By default, ProRender is installed to *drive letter*:\Users\<*username*>\AppData\Roaming\Luxology\Kits\ProRender13.0\
4. Open Modo for the changes to take effect.

Mac

On macOS, ProRender is installed with Modo by default. For more information on installing Modo on Mac, see Installing Modo on Mac.

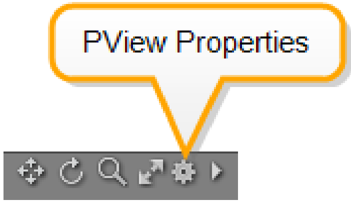
PView Viewport

The PView viewport uses OpenGL for displaying an image rendered by a custom renderer. The PView viewport can be opened as a separate palette by pressing **Alt**+**F8**. Alternatively, you can access the PView viewport in the new ProRender layout by clicking **ProRender Beta** under the main menu bar or by selecting **Layout** > **Layouts** > **ProRender Beta**.

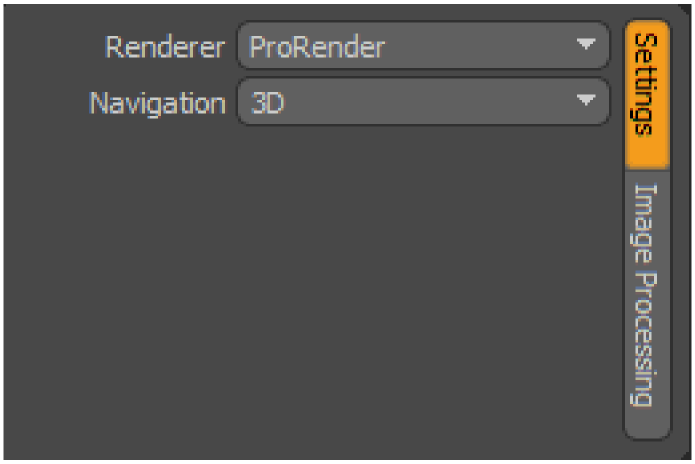


Pview Properties

To access the PView properties, click the gear icon on the top right corner of the view.


PView Properties

Settings

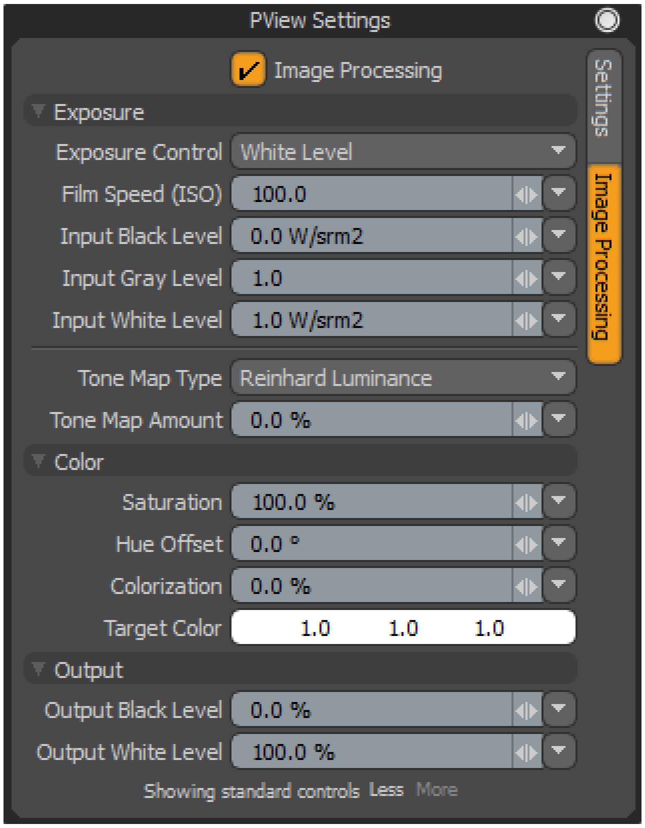

Renderer: ProRender

Navigation: 3D

Settings
Image Processing

You can choose which custom **Renderer** is being used in the PView viewport and enable 2D or 3D **Navigation**. 3D is the camera navigation and 2D is navigation in image space (pan/zoom).

Image Processing



PView Settings

☑ Image Processing

Exposure

Exposure Control: White Level
Film Speed (ISO): 100.0
Input Black Level: 0.0 W/srm2
Input Gray Level: 1.0
Input White Level: 1.0 W/srm2
Tone Map Type: Reinhard Luminance
Tone Map Amount: 0.0 %

Color

Saturation: 100.0 %
Hue Offset: 0.0 °
Colorization: 0.0 %
Target Color: 1.0 1.0 1.0

Output

Output Black Level: 0.0 %
Output White Level: 100.0 %

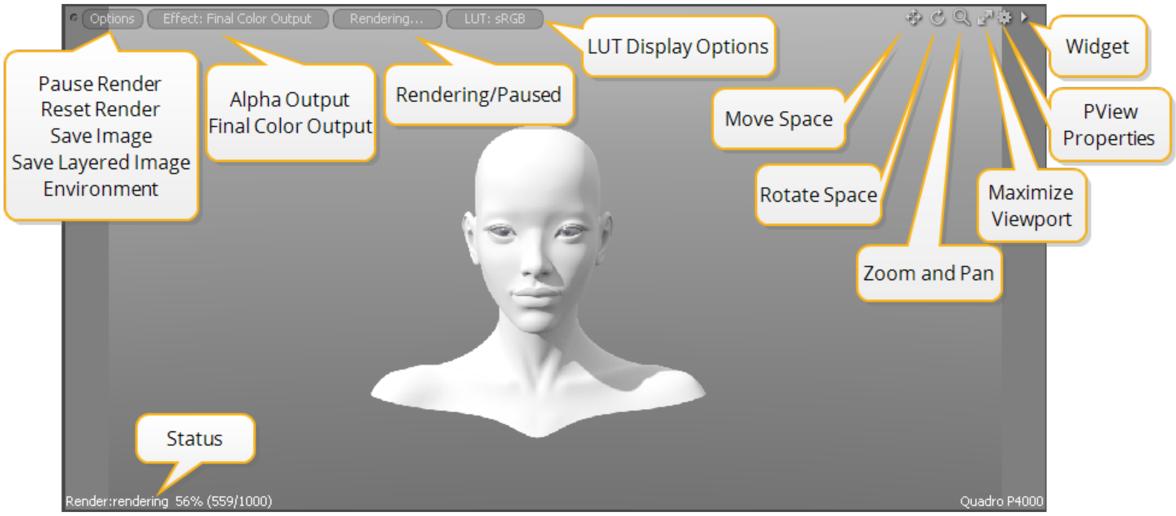
Showing standard controls

The **Image Processing** options provide non-destructive controls to adjust, correct, and tone the rendered image. PView image processing options were implemented using the OpenGL Shading Language (GLSL). In using GLSL to perform post-processing, the effort has been moved from CPU to GPU.

Note: For more information on these options, see Render Display.

Note: Not all of the functionality is currently implemented. Vignette, and Bloom effects, and the vectorscope tools for setting RGB levels are not supported.

PView Buttons and Controls

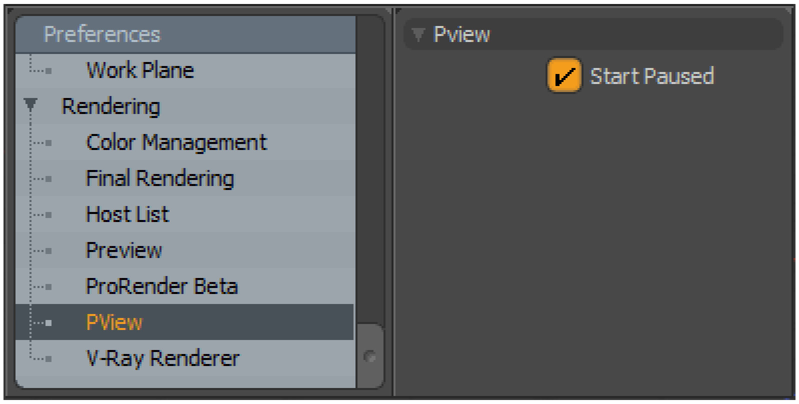

Options
Effect: Final Color Output
Rendering...
LUT: sRGB
LUT Display Options
Widget
Pause Render
Reset Render
Save Image
Save Layered Image
Environment
Alpha Output
Final Color Output
Rendering/Paused
Move Space
PView Properties
Rotate Space
Maximize Viewport
Zoom and Pan
Status
Render:rendering 56% (559/1000)
Quadro P4000

- **Options**
  - **Pause Render** - Pauses current rendering
  - **Reset Render** - Resets the rendering and re-initializes the scene data
  - **Save Image** - Triggers save image to disk
  - **Save Layered Image** - Saves a layered image of all render outputs. This only works for Final Color and Alpha render output types. Only render outputs that are immediate children of a render item are saved.
  - **Environment** - Shows/hides status bar.
- **Effect** - Switch between Final Color and Alpha render outputs.
- **Render/Pause** - Toggles the rendering process on/off
- **LUT** - Sets the LUT for display purposes on your monitor
- Status bar - Contains messages from the custom render plug-in and render devices used by the plug-in.

Pview Preferences

Similarly to the Preview viewport, you can choose to start the PView viewport in a paused state. In some cases with complex scenes loaded, switching to the PView viewport, might result in a temporary freeze as the scene is evaluated for preview rendering. Starting PView in the paused state eliminates this.

You can enable or disable **Start Paused** in the **Preferences**. In the menu bar, click **System** > **Preferences**, then expand **Rendering**, and click **PView**.


Preferences
Work Plane
Rendering
Color Management
Final Rendering
Host List
Preview
ProRender Beta
PView
V-Ray Renderer
Pview

☑ Start Paused

ProRender Settings and Limitations

ProRender has a dedicated **Preferences** section, and you can also edit the Render item's ProRender settings in the **Properties** tab.

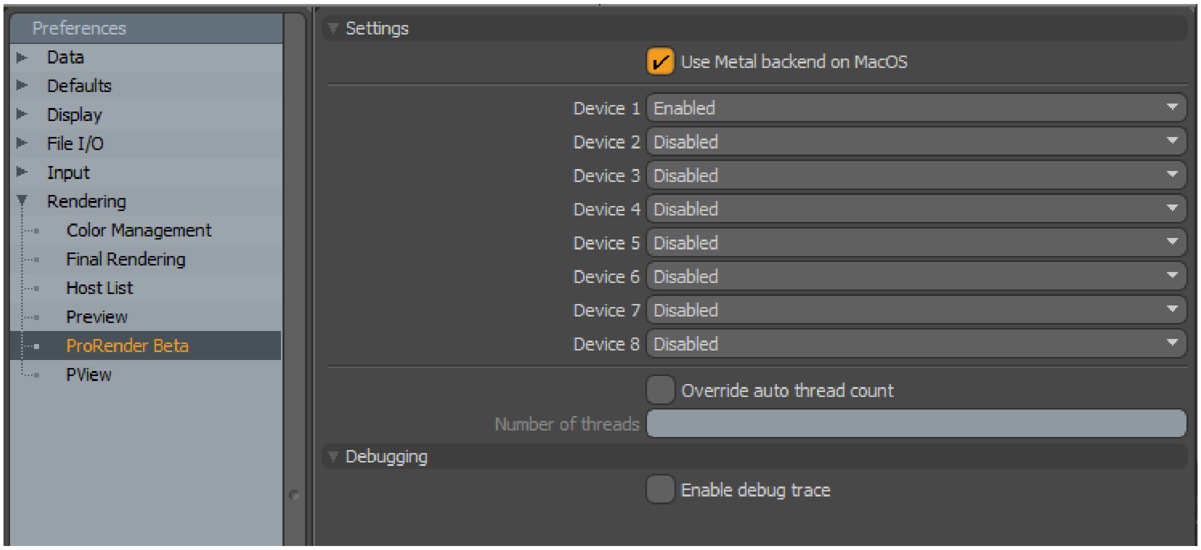
While ProRender supports most of the Modo Shader Tree items, geometry, instances, replicators, cameras, and lights, some features are not supported. For a list of not, or not fully supported features, see Features Not Supported.

ProRender Preferences

The ProRender preferences determine the rendering behavior of Modo.

To access the ProRender preferences:

- On Windows, open **System** > **Preferences**, expand **Rendering**, and click **ProRender Beta**.
- On macOS, open **Menu** > **Preferences**, expand **Rendering**, and click **ProRender Beta**.


Preferences
Data
Defaults
Display
File I/O
Input
Rendering
Color Management
Final Rendering
Host List
Preview
ProRender Beta
PView
Settings
Debugging

☑ Use Metal backend on MacOS

Device 1 Enabled
Device 2 Disabled
Device 3 Disabled
Device 4 Disabled
Device 5 Disabled
Device 6 Disabled
Device 7 Disabled
Device 8 Disabled

☐ Override auto thread count

Number of threads

☐ Enable debug trace

The **ProRender Beta** preferences.

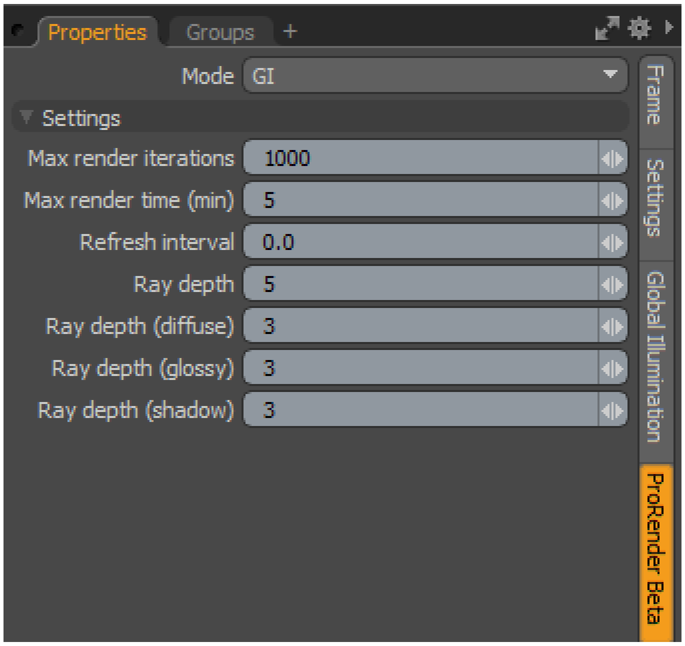
You can:

- Enable Metal implementation on MacOS only (10.13.5+ is required).
- Enable/disable detected devices.
- Set the maximum number of CPU threads used by ProRender.
- Enable debug tracing (useful information for the AMD development team).

Render Settings

You can find the render settings in the **ProRender Beta** tab on the Render Item.

To access the ProRender render settings, open the Shader Tree on the right panel, click **Render**, and open the **ProRender Beta** tab in the **Properties** viewport.



Mode GI

Settings

Max render iterations 1000

Max render time (min) 5

Refresh interval 0.0

Ray depth 5

Ray depth (diffuse) 3

Ray depth (glossy) 3

Ray depth (shadow) 3

You can:

- Set output type (GI/No GI, Wireframe, etc.).
- Set the maximum number of iterations and render time.
- Refresh interval (in milliseconds) it allows ProRender to utilize more GPU resource prior sending the rendering data to PView viewport.
- Set ray depths for certain rendering effects.

Render Devices

Render devices are detected on Modo startup. For each render device ProRender creates a GPU kernel cache directory and cached files.

Although ProRender can use multiple GPUs to render, for interactivity, it is recommended to use a fast graphics processing unit. Since multiple GPUs take more time to initialize the data and compile the GPU kernels (programs), more CPU resources are required.

Note: CPU is not used as a rendering device. Only GPU devices are supported.

Features Not Supported

ProRender supports most of the Modo Shader Tree items, geometry, instances, replicators, cameras, and lights. The following table lists the features currently not or not fully supported by ProRender, and provides additional information where relevant.

Geometry Items

Basic		
Fur Material Cylinder	Not supported	
RPC Mesh	Not supported	Deprecated in Modo 16.1
Volumes		
Blob	Not supported	
Ground Plane	Not supported	
Render Boolean	Not supported	
Sprite	Not supported	
Volume	Not supported	

Light Items

Light Types		
Area	Partially supported	Shape settings are not working.
Cylinder	Not supported	
Dome	Not supported	
Photometric	Not supported	

Point	Partially supported	Works, but does not respond to radius settings, which significantly affects shadow falloff.
Portal	Not supported	Portals don't actually produce light. They direct rays through openings like windows to improve renders. Low priority.
Spot	Partially supported	Basics of the light work. Cone angle and soft edge work, however soft edge gives different results.
Common Light Channels		
Render	Not supported	Visibility can be enabled/disabled as an alternative.
Dissolve	Not supported	
Radiant Intensity	Partially supported	Slightly different results between ProRender and Modo's renderer.
Falloff Type	Not supported	Behavior difference.
Shadow Type	Not supported	
Samples	Not supported	Samples only apply to Modo's Raytrace renderer.
Importance	Not supported	Importance only applies to Modo's Raytrace renderer.
Visible to Reflection Rays	Partially supported	Always visible in ProRender.
Visible to Refraction Rays	Partially supported	Visibility in ProRender depends on visibility to camera.
Area Light Channels	Not supported	
Shape	Not supported	
Cylinder Light Channels		
Directional Light Channels	Not supported	
Shadow/Photon size	Not supported	
Spread Angle	Partially supported	Works, but only after changing the channel value. The initial load shows no spread.

Physical Sun	Partially supported	The direction matches, but the color doesn't.
Haze	Not supported	Not working for light. Behavior is different when used in conjunction with an environment.
Clamp Intensity	Not supported	Not working for light. Behavior is different when used in conjunction with an environment.
Gamma	Not supported	
Photometric Light Channels	Not supported	
Point Light Channels	Not supported	
Portal Light Channels	Not supported	
Spot Light Channels	Not supported	
Radius	Not supported	Affects softness in conjunction with a Cone Angle. Soft Edge can confuse this, as it perceptually has a similar effect but isn't physically accurate.
Soft Edge	Partially supported	Produces a different result to the Modo renderer.
Light Material Channels		
Shadow Color	Not supported	
Affect Diffuse	Not supported	
Affect Specular	Not supported	
Affect Caustics	Not supported	
Affect Subsurface	Not supported	
Volumetrics	Not supported	

Geometry and Light Visibility

Geometry and Light

Render Channel on Mesh/Light	Partially supported	Works on meshes, but not on lights.
Dissolve Channel on Light	Not supported	

Environment

Environment Item		
Visible to Camera	Partially supported	Toggles environment, but also affects **Visible to Indirect Rays**.
Visible to Indirect Rays (Diffuse Only)	Partially supported	Has no effect.
Visible to Reflection Rays	Partially supported	Has no effect.
Visible to Refraction Rays	Partially supported	Has no effect.
Environment Material		
Invert	Not supported	
Blend Mode	Not supported	
Opacity	Not supported	
Sun Light	Partially supported	Works when the light is selected, but doesn't work when light is set to **none**.
Solar Disc Size	Not supported	
Haze Amount	Partially supported	Incorrect behavior.
Clamp Sky Brightness	Not supported	
Sky Gamma	Not supported	
Disc In-scatter	Not supported	
Environment Fog	Not supported	

Camera

Camera View Tab		
Projection Type		
Orthographic	Not supported	
Cylindrical Map	Partially supported	Render very different from Modo Preview.
Cylindrical VR	Not supported	
Spherical Map	Partially supported	Works, but it's rotated 90 degrees.
Spherical VR	Partially supported	Works, but defaults to **Over Under** despite stereoscopic options not being enabled.
Other Channels		
Lens Distortion	Not supported	
Override Render Resolution	Not supported	
Foreground Image	Not supported	
Background Image	Not supported	
Camera Effects Tab - Depth of Field		
Iris Blades	Not supported	
Iris Rotation	Not supported	
Edge Weighting	Not supported	
Camera Effects Tab - Motion Blur	Not supported	
Camera Effects Tab - Stereoscopic	Not supported	
Camera Effects Tab - Clipping Plane	Not supported	

Shader Tree

Masking with		
Replicator	Not supported	
Special Texture Effects	Not supported	
Shader Control Effects	Not supported	
Surface Shading Effects		
Clear-coat Bump	Partially supported	
Displacement	Partially supported	
Rounded Edge Width	Not supported	
Vector Displacement	Not supported	

Nodal Shading

Not supported.

Note: Channel modifier items and Shader Input items that are not explicitly mentioned are not supported. Some Channel modifiers can still be used as long as their values are constant for all shading samples.

Materials and Shader Items

Material Ref - Diffuse and Specular		
Refractive Index	Partially supported	Render different from Modo Renderer.

Roughness	Partially supported	0% roughness does not produce a bright specular highlight on a sphere, but 0.1% does.
Anisotropy	Not supported	
Anisotropy Map	Not supported	
Match Specular	Not supported	
Reflection Type	Partially supported	Full scene reflection works but **Environment only** doesn't.
Clearcoat Bump	Not supported	
Material Ref - Surface Normal		
Smoothing	Partially supported	
Smoothing Angle	Partially supported	
Double Sided	Not supported	
Rounded Edge Width	Not supported	
Round Same Surface Only	Not supported	
Material Trans - Transparency		
Transparent Amount	Partially supported	Render different from Modo renderer.
Absorption distance	Not supported	
Refractive Index	Partially supported	Incorrect results.
Dispersion	Not supported	
Refraction Roughness	Not supported	
Refraction Rays	Not supported	
Material Trans - Sub-surface Scattering		

Subsurface Amount	Partially supported	Works, but having a bright sub-surface color, and a high amount shows visible differences to Modo preview.
Scattering Distance	Partially supported	A **Scattering Distance** of 0 produces a significantly different render, as do large distances.
Maximum Depth	Partially supported	A very shallow depth shows subtle differences (Modo preview has no sign of SSS, but RPR does).
Front Weighting	Not supported	
Samples	Not supported	
Same Surface Only	Not supported	
Material Rays - Ray Tracing	Not supported	
Material Rays - Volumetrics	Not supported	

Unreal Material

Not supported.

Unity Material

Not supported.

Shader Item

Not supported.

Image Map

Image Map Types		
Layered	Not supported	

Sequence	Not supported	
UDIM	Not supported	
Substance	Partially supported	Works, but enabling **Height Output** causes Modo to crash.
Image Still Channels		
UDIM	Not supported	
Color-space	Partially supported	
Image Map Channels		
Anti-aliasing	Not supported	
Anti-aliasing Strength	Not supported	
Minimum Spot	Not supported	
Low/High Value	Not supported	
Brightness	Not supported	
Contrast	Not supported	
Clamp	Not supported	
Invert Red/Green/Blue	Not supported	
Swizzling	Not supported	

Textures

Processing Sub-menu		
Gradient	Not supported	
Occlusion	Not supported	
Process	Not supported	

Tension Texture	Not supported	
Variation Texture	Not supported	
Weigth Map Texture	Not supported	
Gradient Texture Supported Inputs	Not supported	
Variation Texture Supported Inputs	Not supported	

Texture Locator

Solid	Not supported	
Planar	Not supported	
Cylindrical	Not supported	
Spherical	Not supported	
Cubic	Not supported	
Front Projection	Not supported	
Light Probe	Not supported	
Implicit UV	Not supported	
Box	Not supported	
Other Options		
Transform channels	Not supported	
World Transforms	Not supported	
World Coordinates	Not supported	
Front Culling	Not supported	
Projection Axis	Not supported	

Use Clip UDIM	Not supported	
U/V Tile Offset	Not supported	
Horizontal/Vertical Repeat	Not supported	
UV Rotation	Not supported	
Random Texture Offsets	Not supported	
Texture Offset Amplitude	Not supported	

Third-Party Software

Note: If, for any reason, you think Foundry is not entitled to use these libraries or fonts, please contact support via Foundry's Support Portal.

Third-Party Library Licenses

The following table lists third-party libraries and their licenses.

Library	Version	License
Alembic	1.8.3	Alembic TM & © 2009-2015 Lucasfilm Entertainment Company Ltd. or Lucasfilm Ltd. All

Library	Version	License
		rights reserved. Industrial Light & Magic, ILM and the Bulb and Gear design logo are all registered trademarks or service marks of Lucasfilm Ltd. © 2009-2015 Sony Pictures Imageworks Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Industrial Light & Magic nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. -- ALEMBIC ATTACHMENT A — REQUIRED NOTICES FOR DISTRIBUTION

Library	Version	License
		The Alembic Software is distributed along with certain third party components licensed under various open source software licenses ("Open Source Components"). In addition to the warranty disclaimers contained in the open source licenses found below, Industrial Light & Magic, a division of Lucasfilm Entertainment Company Ltd. ("ILM") makes the following disclaimers regarding the Open Source Components on behalf of itself, the copyright holders, contributors, and licensors of such Open Source Components: TO THE FULLEST EXTENT PERMITTED UNDER APPLICABLE LAW, THE OPEN SOURCE COMPONENTS ARE PROVIDED BY THE COPYRIGHT HOLDERS, CONTRIBUTORS, LICENSORS, AND ILM "AS IS" AND ANY REPRESENTATIONS OR WARRANTIES OF ANY KIND, WHETHER ORAL OR WRITTEN, WHETHER EXPRESS, IMPLIED, OR ARISING BY STATUTE, CUSTOM, COURSE OF DEALING, OR TRADE USAGE, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT WILL THE COPYRIGHT OWNER, CONTRIBUTORS, LICENSORS, OR ILM AND/OR ITS AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE OPEN SOURCE COMPONENTS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Boost C++ Libraries -- Boost Software License – Version 1.0 August 17th, 2003 Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

Library	Version	License
		The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. MurmurHash3 -- The MIT License (MIT) Copyright (c) <year> <copyright holders> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER

Library	Version	License
		IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
AxF	1.5.1	The AxF Decoding SDK is used under license from X-Rite
Boost	1.80.0	Use of this software is granted under one of the following two licenses, to be chosen freely by the user. 1. Boost Software License - Version 1.0 - August 17th, 2003 === Copyright (c) 2006, 2007 Marcin Kalicinski Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following: The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. 2. The MIT License

Library	Version	License
		=== Copyright (c) 2006, 2007 Marcin Kalicinski Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Bullet Physics	3.2.4	(C) 1995-2024 Jean-loup Gailly and Mark Adler This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

Library	Version	License
		2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution. Jean-loup Gailly Mark Adler jloup@gzip.org madler@alumni.caltech.edu
Curl	7.72.0	COPYRIGHT AND PERMISSION NOTICE Copyright (c) 1996 - 2024, Daniel Stenberg, daniel@haxx.se, and many contributors, see the THANKS file. All rights reserved. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.
Eigen	3.4.0	**Mozilla Public License** Version 2.0 **1. Definitions** 1.1. “Contributor”

Library	Version	License
		means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software. 1.2. “Contributor Version” means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution. 1.3. “Contribution” means Covered Software of a particular Contributor. 1.4. “Covered Software” means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof. 1.5. “Incompatible With Secondary Licenses” means that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License. 1.6. “Executable Form” means any form of the work other than Source Code Form. 1.7. “Larger Work” means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software. 1.8. “License” means this document.

Library	Version	License
		1.9. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License. 1.10. “Modifications” means any of the following: any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or any new file in Source Code Form that contains any Covered Software. 1.11. “Patent Claims” of a Contributor means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version. 1.12. “Secondary License” means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses. 1.13. “Source Code Form” means the form of the work preferred for making modifications. 1.14. “You” (or “Your”) means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than

Library	Version	License
		fifty percent (50%) of the outstanding shares or beneficial ownership of such entity. **2. License Grants and Conditions** 2.1. Grants Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license: under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version. 2.2. Effective Date The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution. 2.3. Limitations on Grant Scope The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1 (b) above, no patent license is granted by a Contributor: for any code that a Contributor has removed from Covered Software; or for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or under Patent Claims infringed by Covered Software in the absence of its Contributions.

Library	Version	License
		This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4). 2.4. Subsequent Licenses No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3). 2.5. Representation Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License. 2.6. Fair Use This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents. 2.7. Conditions Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1. **3. Responsibilities** 3.1. Distribution of Source Form All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form. 3.2. Distribution of Executable Form

Library	Version	License
		If You distribute Covered Software in Executable Form then: such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License. 3.3. Distribution of a Larger Work You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s). 3.4. Notices You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies. 3.5. Application of Additional Terms You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor.

Library	Version	License
		You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction. **4. Inability to Comply Due to Statute or Regulation** If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it. **5. Termination** 5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice. 5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

Library	Version	License
		5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination. **6. Disclaimer of Warranty** Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer. **7. Limitation of Liability** Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You. **8. Litigation** Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's

Library	Version	License
		ability to bring cross-claims or counter-claims. **9. Miscellaneous** This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor. **10. Versions of the License** 10.1. New Versions Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number. 10.2. Effect of New Versions You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward. 10.3. Modified Versions If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License). 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Library	Version	License
		Exhibit A - Source Code Form License Notice This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/. If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice. You may add additional accurate notices of copyright ownership. **Exhibit B - “Incompatible With Secondary Licenses” Notice** This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.
Embree	3.13.5	Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions

Library	Version	License
		granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

Library	Version	License
		2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminateas of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices

Library	Version	License
		that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

Library	Version	License
		8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

Library	Version	License
		http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
FBX	2020.3.1	This software contains Autodesk® FBX® code developed by Autodesk, Inc. Copyright 2017 Autodesk, Inc. All rights, reserved. Such code is provided “as is” and Autodesk, Inc. disclaims any and all warranties, whether express or implied, including without limitation the implied warranties of merchantability, fitness for a particular purpose or non-infringement of third party rights. In no event shall Autodesk, Inc. be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of such code.
FFMpeg	5.1	FFmpeg: Most files in FFmpeg are under the GNU Lesser General Public License version 2.1 or later (LGPL v2.1+). Read the file COPYING.LGPLv2.1 for details. Some other files have MIT/X11/BSD-style licenses. In combination the LGPL v2.1+ applies to FFmpeg. Some optional parts of FFmpeg are licensed under the GNU General Public License version 2 or later (GPL v2+). See the file COPYING.GPLv2 for details. None of these parts are used by default, you have to explicitly pass --enable-gpl to configure to activate them. In this case, FFmpeg's license changes to GPL v2+. Specifically, the GPL parts of FFmpeg are - libpostproc - libmpcodecs - optional x86 optimizations in the files libavcodec/x86/idct_mmx.c

Library	Version	License
		- libutvideo encoding/decoding wrappers in libavcodec/libutvideo*.cpp - the X11 grabber in libavdevice/x11grab.c - the swresample test app in libswresample/swresample-test.c - the texi2pod.pl tool - the following filters in libavfilter: - f_ebur128.c - vf_blackframe.c - vf_boxblur.c - vf_colormatrix.c - vf_cropdetect.c - vf_decimate.c - vf_delogo.c - vf_geq.c - vf_histeq.c - vf_hqdn3d.c - vf_kerndeint.c - vf_mcdeint.c - vf_mp.c - vf_owdenoise.c - vf_perspective.c - vf_phase.c - vf_pp.c - vf_pullup.c - vf_sab.c - vf_smartblur.c - vf_spp.c - vf_stereo3d.c - vf_super2xsai.c - vf_tinterlace.c

Library	Version	License
		- vf_yadif.c - vsrc_mptestsrc.c Should you, for whatever reason, prefer to use version 3 of the (L)GPL, then the configure parameter --enable-version3 will activate this licensing option for you. Read the file COPYING.LGPLv3 or, if you have enabled GPL parts, COPYING.GPLv3 to learn the exact legal terms that apply in this case. There are a handful of files under other licensing terms, namely: * The files libavcodec/jfdctfst.c, libavcodec/jfdctint_template.c and libavcodec/jrevdct.c are taken from libjpeg, see the top of the files for licensing details. Specifically note that you must credit the IJG in the documentation accompanying your program if you only distribute executables. You must also indicate any changes including additions and deletions to those three files in the documentation. external libraries ================== FFmpeg can be combined with a number of external libraries, which sometimes affect the licensing of binaries resulting from the combination. compatible libraries -------------------- The following libraries are under GPL: - frei0r - libcdio - libutvideo - libvidstab - libx264 - libxavs

Library	Version	License
		- libxvid When combining them with FFmpeg, FFmpeg needs to be licensed as GPL as well by passing --enable-gpl to configure. The OpenCORE and VisualOn libraries are under the Apache License 2.0. That license is incompatible with the LGPL v2.1 and the GPL v2, but not with version 3 of those licenses. So to combine these libraries with FFmpeg, the license version needs to be upgraded by passing --enable-version3 to configure. incompatible libraries ---------------------- The Fraunhofer AAC library, FAAC and aacplus are under licenses which are incompatible with the GPLv2 and v3. We do not know for certain if their licenses are compatible with the LGPL. If you wish to enable these libraries, pass --enable-nonfree to configure. But note that if you enable any of these libraries the resulting binary will be under a complex license mix that is more restrictive than the LGPL and that may result in additional obligations. It is possible that these restrictions cause the resulting binary to be unredistributeable.
fmt	8.1.1	Copyright (c) 2012 - present, Victor Zverovich and {fmt} contributors Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be

Library	Version	License
		included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. --- Optional exception to the license --- As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into a machine-executable object form of such source code, you may redistribute such embedded portions in such object form without including the above copyright and permission notices.
GLEW	1.13.0	The OpenGL Extension Wrangler Library Copyright (C) 2002-2007, Milan Ikits <milan.ikits@ieee.org> Copyright (C) 2002-2007, Marcelo E. Magallon <mmagallo@debian.org> Copyright (C) 2002, Lev Povalahev All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Library	Version	License
		* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * The name of the author may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Mesa 3-D graphics library Version: 7.0 Copyright (C) 1999-2007 Brian Paul All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Library	Version	License
		THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL BRIAN PAUL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Copyright (c) 2007 The Khronos Group Inc. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to permit persons to whom the Materials are furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Materials. THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.
HDF5	1.8.7	Copyright Notice and License Terms for HDF5 (Hierarchical Data Format 5) Software Library and Utilities --- HDF5 (Hierarchical Data Format 5) Software Library and Utilities Copyright 2006-2011 by The HDF Group. NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities Copyright

Library	Version	License
		1998-2006 by the Board of Trustees of the University of Illinois. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution. 3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change. 4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by The HDF Group and by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and credit the contributors. 5. Neither the name of The HDF Group, the name of the University, nor the name of any Contributor may be used to endorse or promote products derived from this software without specific prior written permission from The HDF Group, the University, or the Contributor, respectively. DISCLAIMER: THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall The HDF Group or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage. --- ---

Library	Version	License
		Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois, Fortner Software, Unidata Program Center (netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip), and Digital Equipment Corporation (DEC). --- Portions of HDF5 were developed with support from the Lawrence Berkeley National Laboratory (LBNL) and the United States Department of Energy under Prime Contract No. DE-AC02-05CH11231. --- Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials: This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL. DISCLAIMER: This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately- owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Library	Version	License
		--
Hosek Skylight Model		Copyright (c) 2012 - 2013, Lukas Hosek and Alexander Wilkie All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
IMath	3.1.4	Copyright Contributors to the OpenEXR Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this

Library	Version	License
		list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
JPEG	9e	Independent JPEG Group License LEGAL ISSUES In plain English: 1. We don't promise that this software works. (But if you find any bugs, please let us know!) 2. You can use this software for whatever you want. You don't have to pay us. 3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

Library	Version	License
		In legalese: The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy. This software is copyright (C) 1991-1998, Thomas G. Lane. All Rights Reserved except as specified below. Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions: (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation. (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group". (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind. These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us. Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software". We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

Library	Version	License
		ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do. The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable. It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code. The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders. We are required to state that "The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."
libIGL	2.4.0	**Mozilla Public License** Version 2.0

Library	Version	License
		1. Definitions 1.1. “Contributor” means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software. 1.2. “Contributor Version” means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution. 1.3. “Contribution” means Covered Software of a particular Contributor. 1.4. “Covered Software” means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof. 1.5. “Incompatible With Secondary Licenses” means that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License. 1.6. “Executable Form” means any form of the work other than Source Code Form. 1.7. “Larger Work” means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

Library	Version	License
		1.8. “License” means this document. 1.9. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License. 1.10. “Modifications” means any of the following: any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or any new file in Source Code Form that contains any Covered Software. 1.11. “Patent Claims” of a Contributor means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version. 1.12. “Secondary License” means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses. 1.13. “Source Code Form” means the form of the work preferred for making modifications. 1.14. “You” (or “Your”) means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is

Library	Version	License
		under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity. **2. License Grants and Conditions** 2.1. Grants Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license: under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version. 2.2. Effective Date The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution. 2.3. Limitations on Grant Scope The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1 (b) above, no patent license is granted by a Contributor: for any code that a Contributor has removed from Covered Software; or for infringements caused by: (i) Your and any other third party’s modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

Library	Version	License
		under Patent Claims infringed by Covered Software in the absence of its Contributions. This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4). 2.4. Subsequent Licenses No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3). 2.5. Representation Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License. 2.6. Fair Use This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents. 2.7. Conditions Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1. **3. Responsibilities** 3.1. Distribution of Source Form All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients’ rights in the Source Code Form.

Library	Version	License
		3.2. Distribution of Executable Form If You distribute Covered Software in Executable Form then: such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License. 3.3. Distribution of a Larger Work You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s). 3.4. Notices You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies. 3.5. Application of Additional Terms You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However,

Library	Version	License
		You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction. **4. Inability to Comply Due to Statute or Regulation** If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it. **5. Termination** 5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice. 5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors

Library	Version	License
		for the Covered Software under Section 2.1 of this License shall terminate. 5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination. **6. Disclaimer of Warranty** Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer. **7. Limitation of Liability** Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You. **8. Litigation** Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and

Library	Version	License
		such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims. **9. Miscellaneous** This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor. **10. Versions of the License** 10.1. New Versions Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number. 10.2. Effect of New Versions You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward. 10.3. Modified Versions If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License). 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

Library	Version	License
		If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached. **Exhibit A - Source Code Form License Notice** This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/. If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice. You may add additional accurate notices of copyright ownership. **Exhibit B - “Incompatible With Secondary Licenses” Notice** This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.
libTIFF	4.3.0	Copyright (c) 1988-1997 Sam Leffler Copyright (c) 1991-1997 Silicon Graphics, Inc. Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics. THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND,

Library	Version	License
		OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
libXML2	2.9.9	Copyright (c) 2018 The Meson development team Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so. THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
LittleCMS	2.12	Copyright <YEAR> <COPYRIGHT HOLDER> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so. THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	Version	License
mDNS Responder	1557.120.7	Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that

Library	Version	License
		is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity

Library	Version	License
		(including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminateas of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work

Library	Version	License
		otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You

Library	Version	License
		may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
msgpack	4.1.1	Boost Software License - Version 1.0 - August 17th, 2003 Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

Library	Version	License
		The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MiniZip	1.1	(C) 1995-2024 Jean-loup Gailly and Mark Adler This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution. Jean-loup Gailly jloup@gzip.org Mark Adler madler@alumni.caltech.edu
MuParser	2.3.2	Copyright 2020 Ingo Berg

Library	Version	License
		All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

Library	Version	License
		THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Oculus	1.19.0	Oculus SDK License Agreement Effective date: October 29, 2021 Copyright © Facebook Technologies, LLC and its affiliates. All rights reserved. The text of this may be found at: https://developer.oculus.com/licenses/oculussdk/ This Oculus SDK License Agreement (“Agreement”) is a legal agreement between you and Oculus governing your use of our Oculus Software Development Kit. Oculus Software Development Kit means any application programming interfaces (“APIs”), tools, plugins, code, technology, specification, documentation, Platform Services, and/or content made available by us to others, including app developers and content providers (collectively, the “SDK”). By downloading or using our SDK, you are agreeing to this Agreement along with other applicable terms and conditions such as the additional terms or documents accompanying the SDK and the Terms of Service, and acknowledging our Privacy Policy (collectively, the “Terms”). If you use the SDK as an interface to, or in conjunction with other Oculus products or services, then the terms for those other products or services also apply. Here, "Oculus" means Facebook Technologies, LLC, formerly known as Oculus VR, LLC, a Delaware limited liability company with its principal place of business at 1 Hacker Way, Menlo Park, California 94025, United States unless set forth otherwise. We may refer to "Oculus" as "we", "our", or "us" in this Agreement. You may not use the SDK and may not accept this Agreement if (1) you are a person with whom Oculus is prohibited from transacting business under applicable law, or (2) you are a person barred from using or receiving the SDK by Oculus or under the applicable laws of the United States or other countries including the country in which you are resident or from which you use the SDK. If you are using the SDK on behalf of an entity, you represent and warrant that you have authority to bind that entity to this Agreement and by accepting this Agreement, you are doing so on behalf of that entity (and all

Library	Version	License
		references to "you" in this Agreement refer to that entity). This Agreement requires the resolution of most disputes between you and Oculus by binding arbitration on an individual basis; class actions and jury trials are not permitted. 1. License Grant 1.1 License. Subject to the Terms and the restrictions set forth in this Agreement, Oculus hereby grants you a limited, royalty-free, non-exclusive, non-transferrable, non-sublicensable (except as otherwise set forth in this Agreement), revocable copyright license (“License”) during the term of this Agreement to use and reproduce the SDK solely to develop, test, and/or distribute your Application (defined below) and to enable you and/or your end users to access Oculus features through your Application. You may only use the SDK to develop Applications in connection with Oculus approved hardware and software products (“Oculus Approved Products”) unless the documentation accompanying the SDK expressly authorizes broader use such as with other third-party platforms. 1.1.1 If the SDK includes any libraries, sample source code, or other materials that we make available specifically for incorporation in your Application (as indicated by applicable documentation), you may incorporate those materials and reproduce and distribute them as part of your Application, including by distributing those materials to third parties contributing to your Application. 1.1.2 The SDK may include other content (e.g., sample code) that is for demonstration, reference, or other purposes and is subject to terms and conditions included with such materials. Such materials will be clearly marked in the applicable documentation. Absent such additional terms and conditions, you may modify, distribute, and sublicense any sample source made available as part of the SDK pursuant to this Agreement and the Terms. 1.1.3 The SDK may include Oculus content that is subject to your additional right to display the content to your end users through the use of the corresponding SDK, as contemplated by the documentation accompanying such SDK. For example, the SDK may include avatars that you may display to your end users. 1.2 General Restrictions. The License grant in this Section is solely for the purpose of developing, testing, and promoting your engines, tools, applications,

Library	Version	License
		content, games and demos, or other products and features (collectively, "Application") and providing you and/or your end users access to Oculus services and features through your Application as contemplated by applicable documentation accompanying the SDK. You may not (or allow those acting on your behalf to): 1.2.1 modify or create derivative works from any SDK or its component (other than sample source code described in this Section or expressly authorized by the documents accompanying the SDK); 1.2.2 misrepresent or mask either your identity or your Application's identity when using the SDK or developer accounts; 1.2.3 attempt to circumvent any limitations implemented within or documented with the SDK (e.g., limiting the number of requests you may make or end users you may serve); 1.2.4 reverse engineer, decompile, disassemble, or otherwise attempt to extract the source code from the SDK, except to the extent that applicable law expressly permits such actions despite this limitation; 1.2.5 alter, restrict, or interfere with the normal operation or functionality of the SDK, the Oculus hardware or software, or Oculus Approved Products, including, but not limited to: (a) the behavior of the "Oculus button" and "XBox button" implemented by the Oculus system software; (b) any on-screen messages or information; (c) the behavior of the proximity sensor in the Oculus hardware implemented by the Oculus system software; (d) any Oculus hardware or software security features; (e) any end user's settings; and (f) Health and Safety Warnings; 1.2.6 use the SDK or your Application in a manner that violates: (a) the Oculus Data Use Policy (where applicable); (b) the Oculus Content Guidelines, or other applicable terms and policies made available on our Developer Policy portal; (c) any rights of Oculus or third parties; (d) applicable laws (such as laws regarding import, export, privacy, health & safety); or (e) other terms of service with Oculus or its affiliates; 1.2.7 remove, obscure, or alter any Terms or any links to or notices of those

Library	Version	License
		Terms; or 1.2.8 use or redistribute the SDK or any portion thereof in any manner that would cause the SDK (or any portion thereof) or Oculus to become subject to the terms of any open source license or other restrictions. 1.3 Distribution and Sublicense Restrictions. The redistribution and sublicense rights under this Section are further subject to the following restrictions: (1) redistribution of sample source code or other materials must include the following copyright notice: “Copyright © Facebook Technologies, LLC and its affiliates. All rights reserved;” and (2) if the sample source code or other materials include a "License" or "Notice" text file, you must provide a copy of the License or Notice file with the sample code. 1.4 Privacy and Security. 1.4.1 You are responsible for the data collection, processing and disclosure by your Application and agree to comply with all applicable privacy and data protection laws, as well as our applicable terms and policies, particularly the Oculus Data Use Policy. You represent and warrant that you have provided robust and sufficiently prominent notice to users regarding (i) data processing that includes, at a minimum, that third parties, including Oculus and its affiliates, may collect or receive information from your Application, and (ii) any other information required to be disclosed to users by applicable privacy and data protection laws. You represent and warrant that you will not back up or make available to the Oculus Cloud Backup feature any information that you know or reasonably should know (i) is from or about children under the age of 13, or (ii) includes data concerning health, financial information, or other categories of sensitive information (including any information defined as special or sensitive under applicable laws, regulations, and applicable industry guidelines). 1.4.2 For purposes of the GDPR, you acknowledge and agree that you are a separate and independent controller of the Developer User Data (as defined in the Oculus Developer Data Use Policy) and Facebook Ireland Limited (an affiliate of Oculus) is a separate and independent controller for any processing of personal data, except as provided in Section 1.4.3, including the Oculus User Data (as defined in the Oculus Developer Data Use Policy). The parties do not

Library	Version	License
		and will not process Developer User Data or Oculus User Data as joint controllers. Each party shall comply with the obligations that apply to it as a controller under the GDPR, and each party shall be individually and separately responsible for its own compliance. 1.4.3 Notwithstanding the foregoing, where (a) we process Developer User Data that contains personal data to (i) store, host or otherwise backup Developer User Data through Oculus Cloud Backup; (ii) provide and operate the Spatial Audio VoIP API; (iii) or provide other services described in the Data Processing Terms (the “Services”), and (b) (i) our processing of such personal data is subject to the GDPR, you instruct Facebook Ireland Limited to process such personal data in order to provide the Services pursuant to this Agreement and the Data Processing Terms, which are incorporated herein by reference, including for product improvement for your benefit and to anonymize the data so that it is no longer Personal Data for the purposes of GDPR, and you understand and agree that we may retain such anonymized metadata for our own legitimate purposes, including improvement and development of the Services; or (ii) such personal data is considered “personal information,” “personal data,” “personally identifiable information” or similar terms and is subject to the California Consumer Privacy Act of 2018 (“CCPA”), or other applicable privacy and data protection laws (excluding the GDPR), we will only retain, use and disclose such personal data for the purposes of providing those Services to you and improving those Services on your behalf or as otherwise permitted by the CCPA and such other applicable privacy and data protection laws. 1.4.4 “Personal data,” “controller,” “processor,” and “process” in this Section 1.4 have the meanings set out in the Data Processing Terms. 1.5 You have no obligations under this Agreement to license or make available your Application to Oculus, its affiliates, or any third parties. Nothing in this Agreement obligates Oculus or its affiliates to enable you or any of your Applications to access, interact with, or retrieve or publish content to any Oculus platform or service. However, Oculus and/or its affiliates may require you to agree to additional terms as a condition of providing you with such platform services in connection with your use of the SDK. You acknowledge and agree that Oculus and its affiliates may develop products or services that may compete with your Application or any other products or services of yours.

Library	Version	License
		1.6 Experimental Features. From time to time, Oculus may, in its sole discretion, make available to you as part of the Oculus Software Development Kit, certain experimental, test or beta software, APIs or features on a limited or test basis ("Experimental Features"). Experimental Features can only be used for experimental or testing purposes and cannot be incorporated into a production build unless (i) the Experimental Feature has been released or included in Oculus software production builds or (ii) otherwise permitted by Oculus in writing. Your use of any Experimental Feature is voluntary. You agree that all use of any Experimental Feature is at your sole risk. You agree that once you use an Experiment Feature, your content, data and/or systems may be affected, and you may be unable to revert back to a prior version of the same or similar feature. Additionally, if such reversion is possible, you may not be able to return or restore data created or transferred using the Experimental Feature back to the prior version. The Experimental Features may not work in the same way as a final production version. Oculus and its affiliates make no representations or warranties that the Experimental Features will function or be free from errors. The Experimental Features are provided on an "as is" basis and may contain errors or inaccuracies that could cause failures, corruption or loss of data and information from any connected device or service. Oculus and its affiliates have no obligation to correct bugs, defects, or errors or otherwise support or maintain Experimental Features. Oculus and its affiliates may discontinue, update, modify or remove access to any Experimental Feature at any time in its sole discretion, and may not release a final version of an Experimental Feature in its sole discretion. 2. Oculus Platform Services Oculus and/or its affiliates makes certain Platform Services (defined below) available to you to include and enable in your Application on our Platform. An Application that enables or includes any Platform Service must implement the Oculus Platform Framework (defined below) with the Application. Once your Application has been authorized for use of the Platform Services, you are not required to update your Application to include new Platform Services that Oculus and/or its affiliates may make available as part of the Oculus Platform Framework. For more information, please visit https://developer.oculus.com. 2.1 For the purpose of this Section,

Library	Version	License
		2.1.1 “Application Services” means services provided by Oculus and/or its affiliates associated with the Platform, including, but not limited to, in-app purchasing, multiplayer matchmaking, friends, leader boards, achievements, Virtual Reality Real Time Systems (“VERTS”), voice over IP and Cloud Backup, which list may be changed from time to time in Oculus' or its affiliates’ sole discretion. 2.1.2 "Oculus Platform Framework" means the suite of Oculus platform services, including, but not limited to, the Oculus file distribution and update system (enabling distribution and updates of Applications by Oculus and/or its affiliates, including through generated activation Keys), entitlement system, and account authentication, which list may be changed from time to time in Oculus' or its affiliates’ sole discretion. 2.1.3 "Platform" means the virtual, mixed, and augmented reality platform made available by Oculus and/or its affiliates, including, but not limited to, the user experience, user interface, store, and social features, usable on hardware approved by Oculus or its affiliates or any third-party device or operating system, including, but not limited to, iOS, Android, Windows, OS X, Linux, and Windows Mobile. 2.1.4 "Platform Services" means the Oculus Platform Framework and the Application Services. 2.2 Key Provision and Redemption. If you request that Oculus generate activation keys for your Application on the Platform ("Keys") and Oculus agrees, you hereby grant Oculus and its affiliates (1) the right to generate Keys for you and (2) a license to make available, reproduce, distribute, perform, and display the Application to end users who have submitted a Key to Oculus or its affiliates. Oculus agrees to authenticate and make the Application available to any end user supplying a valid Key (or have its affiliates do so) (unless the Application has been removed or withdrawn). 2.3 Platform Services Requirements. You will not make any use of any API, software, code or other item or information supplied by Oculus or its affiliates in connection with the Platform Services other than to enhance the functionality of your Application. In particular, you must not (nor enable others to): (1) defame,

Library	Version	License
		abuse, harass, stalk, or threaten others, or to promote or facilitate any prohibited or illegal activities; (2) enable any functionality in your Application that would generate excessive traffic over the Oculus network or servers that would negatively impact other users' experience, or otherwise interfere with or restrict the operation of the Platform Services, or Oculus' or its affiliates’ servers or networks providing the Platform Services; (3) remove, obscure, or alter any license terms, policies or terms of service or any links to or notices thereto provided by Oculus or its affiliates; or (4) violate any rights of Oculus, its affiliates, or any third parties. Notwithstanding anything to the contrary set forth in this Agreement, you may not sublicense any software, firmware or other item or information supplied by Oculus or its affiliates in connection with the Platform Services for use by a third party, unless expressly authorized by Oculus or its affiliates to do so. You agree not to use (or encourage the use of) the Platform Services for mission critical, life saving or ultra-hazardous activities. Oculus or its affiliates may suspend operation of or remove any Application that does not comply with the restrictions in this Agreement. 2.4 Changes to Platform or Platform Services. Oculus and/or its affiliates may change the Platform or the functionality of the Platform Services at any time, including discontinuing some of the functionality of the Platform Services, and your continued use of the Platform or Platform Services or use of any modified or additional Platform Services is conditioned upon your adherence to the terms of this Agreement, as modified by Oculus or its affiliates from time to time. 3. Intellectual Property 3.1 Ownership. As between you and Oculus, Oculus and/or its affiliates or licensors own all rights, title, and interest, including all Intellectual Property Rights (defined below), in and to the SDK (including associated Oculus content and sample code) and all derivatives thereof. Oculus reserves all rights not expressly granted under the License. As between you and Oculus, you and/or your licensors own all rights, title, and interest in and to your Application, (excluding our SDK), including all Intellectual Property Rights. “Intellectual Property Rights” means any and all worldwide rights under applicable laws of patent, copyright, trade secret, trademark, rights of publicity and privacy, and other proprietary rights.

Library	Version	License
		3.2 Third-Party Materials. Our SDK may include third-party software offered under an open source license or third-party content subject to a separate third-party agreement. To the extent any of such third-party terms conflicts with this Agreement, such third-party terms will control solely with respect to such third-party software or content. 3.3 Feedback. If you provide comments, suggestions, recommendations, ideas, know-how or other feedback about our SDK or any other Oculus or affiliate product or service, we (and our affiliates and those we allow) may use such information for any purposes without obligation to you and all intellectual property and other proprietary rights in any such feedback are deemed (and hereby) licensed to Oculus (with the right to sublicense through multiple tiers) for any purpose on a perpetual, irrevocable, worldwide, paid-up,and royalty-free basis and may be used or disclosed for any purpose. 3.4 Brand Attribution. This Agreement does not grant you or any third party permission to use our trade names, trademarks, service marks, logos, domain names, and other distinctive brand features (collectively, “Brand Features”) except as required for reasonable and customary use in describing the origin of the SDK or reproduction of the copyright notice as required under the License grant. You will not use our SDK or make any statement regarding the SDK or your Application which suggests partnership with, sponsorship by, or endorsement by Oculus, its affiliates or any of their employees, contractors, contributors, licensors, affiliates, or partners without our prior written permission. 4. Confidentiality 4.1 Confidentiality. Our communications to you and our SDK may contain Oculus confidential information, which includes information that is marked confidential or that would normally be considered confidential under the circumstances. If you receive any such information, you will not disclose it to any third party without Oculus' prior written consent. Oculus confidential information does not include information that you independently developed, that was rightfully given to you by a third party without a confidentiality obligation with regard to such information, or that becomes public through no fault of your own. You may disclose Oculus confidential information when compelled to do so by law if you

Library	Version	License
		provide us reasonable prior notice, unless a court order prohibits such notice. 5. Termination 5.1 Termination. The term of this Agreement will begin on the date on which you click accept, download, or use the SDK or any of its components and will continue until terminated as set forth in this Agreement. Oculus reserves the right to terminate this Agreement with you, or to discontinue or suspend the SDK or any portion or feature or your access thereto in the event you breach any material provisions of this Agreement or the Terms, without liability or other obligation to you. 5.2. Discontinuation of SDK. Oculus reserves the right to discontinue all or part of the SDK at any time, in our sole discretion, without notice to you, and without liability or other obligation to you. This Agreement will terminate automatically and without notice to you in the event that the SDK is discontinued in its entirety. 5.3 Effect of Termination. Upon termination of this Agreement, you will immediately stop using, distributing, or otherwise making available the SDK and all Applications that incorporate the SDK or any of its components, cease all use of the Oculus Brand Features, and destroy or return any cached or stored content, software, or other materials obtained through our SDK. 5.4 Surviving Provisions. When this Agreement terminates, those terms that by their nature are intended to continue indefinitely will continue to apply, including, but not limited to, Section 3 (Intellectual Property), Section 4 (Confidentiality), Section 5 (Termination), Section 6 (Liability) and Section 7 (General Provisions). 6. Liability 6.1 Indemnification. Unless prohibited by applicable law, you will indemnify and (at Oculus's option), defend Oculus, its affiliates and subsidiaries, and the agents, licensors, contributors, directors, officers, employees, suppliers, and distributors thereof (collectively, "Oculus Parties") against all liabilities, damages, losses, costs, fees (including legal fees), and expenses relating to any allegation or third-party legal proceeding arising from: (1) your use of the SDK, or any negligence or

Library	Version	License
		misconduct, by you or your employees, agents, vendors, or contractors (collectively “Developer Parties”); (2) any Developer Parties’ violation of this Agreement, Terms, or any applicable law and regulation; (3) any use of your Application; or (4) Developer User Data (defined in Developer Data Use Policy). 6.2 WARRANTIES. EXCEPT AS EXPRESSLY SET OUT IN THE TERMS, THE SDK IS PROVIDED “AS IS” WITHOUT ANY SPECIFIC PROMISES, REPRESENTATIONS, GUARANTEES OR WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY COMMITMENTS ABOUT THE CONTENT ACCESSED THROUGH THE SDK, THE SPECIFIC FUNCTIONS OF THE SDK OR OUR PLATFORM SERVICES, OR THEIR RELIABILITY, AVAILABILITY, OR ABILITY TO MEET YOUR NEEDS. THE OCULUS PARTIES HEREBY DISCLAIM ANY IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. SOME JURISDICTIONS DO NOT PERMIT THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES, SO YOU MAY HAVE ADDITIONAL RIGHTS. 6.3 LIMITATION OF LIABILITY. TO THE EXTENT PERMITTED BY APPLICABLE LAW, OCULUS PARTIES WILL NOT BE RESPONSIBLE FOR LOST PROFITS, BUSINESS OR GOODWILL, REVENUES, OR DATA; FINANCIAL LOSSES; OR INDIRECT, SPECIAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) ARISING AS A RESULT OF THIS AGREEMENT, USE OF THE SDK OR ANY MODIFIED SAMPLE CODE EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. YOU AGREE THAT YOUR REMEDIES UNDER THIS AGREEMENT ARE LIMITED SOLELY TO THE RIGHT TO COLLECT MONEY DAMAGES, IF ANY, AND YOU HEREBY WAIVE YOUR RIGHT TO SEEK INJUNCTIVE RELIEF OR OTHER EQUITABLE RELIEF. IF YOU ARE A CALIFORNIA RESIDENT, YOU AGREE TO WAIVE CALIFORNIA CIVIL CODE § 1542, WHICH SAYS: “A GENERAL RELEASE DOES NOT EXTEND TO CLAIMS THAT THE CREDITOR OR RELEASING PARTY DOES NOT KNOW OR SUSPECT TO EXIST IN HIS OR HER FAVOR AT THE TIME OF EXECUTING THE RELEASE, WHICH IF KNOWN BY HIM OR HER WOULD HAVE MATERIALLY AFFECTED HIS OR HER SETTLEMENT WITH THE DEBTOR OR RELEASED PARTY.” TO THE EXTENT PERMITTED BY LAW, THE CUMULATIVE, AGGREGATE LIABILITY OF OCULUS

Library	Version	License
		PARTIES, FOR ANY AND ALL CLAIMS ARISING UNDER THIS AGREEMENT OR ITS SUBJECT MATTER SHALL NOT EXCEED THE GREATER OF ONE HUNDRED US DOLLARS ($100) OR THE AMOUNT YOU HAVE PAID US IN THE PAST TWELVE MONTHS. IN ALL CASES, OCULUS PARTIES WILL NOT BE LIABLE FOR ANY EXPENSE, LOSS, OR DAMAGE THAT IS NOT REASONABLY FORESEEABLE. 7. General Provisions 7.1 Updates. We may need to update this Agreement from time to time, including to accurately reflect the access or uses of our SDK, and so we encourage you to check this Agreement regularly. By continuing to access or use our SDK after any notice of an update to this Agreement, you agree to be bound by them. Any updates to the Disputes section of this Agreement will apply only to disputes that arise after notice of the update takes place. If you do not agree to the updated terms, please stop all access or use of our SDK. You cannot sidestep your compliance obligations under an updated version of this Agreement by developing against an older release of the SDK or relying on the older Agreement and all updates to your application are subject to the modified Agreement. 7.2 Authorization. You hereby grant Oculus and its contractors and affiliates the authorization reasonably necessary for Oculus to exercise its rights and perform its obligations under this Agreement, including a limited, royalty-free, non-exclusive license to use, perform, and display the Application you provide to Oculus for testing, evaluation, and approval purposes. 7.3 General Provisions. You and Oculus are independent contractors with regard to each other. This Agreement does not create any third-party beneficiary rights or any agency, partnership, employment, or joint venture. We are not liable for failure or delay in performance to the extent caused by circumstances beyond our reasonable control. If you do not comply with this Agreement, and Oculus does not take action right away or does not enforce any provision of this Agreement, this inaction or lack of enforcement will not act as a waiver by Oculus of any rights that it may have (such as taking action in the future) or in any way affect the validity of this Agreement or parts thereof. If a particular provision of this Agreement is deemed unenforceable, it will be deemed modified to the minimum extent necessary to render it enforceable and most nearly reflect the

Library	Version	License
		intent of the original provision, and all other provisions in this Agreement shall remain in full force and effect. You may not assign or delegate this Agreement or any obligations under this Agreement without our advance written consent. Any such prohibited attempted assignment will be void. Oculus may assign or delegate this Agreement and any of its rights or obligations under this Agreement without your consent or notice to you. This Agreement shall bind the parties and their respective heirs, successors, and permitted assigns. This Agreement is the entire agreement between you and Oculus relating to the subject matter herein and supersede any prior or contemporaneous agreements on such subject matter. 7.4 Dispute Resolution. 7.4.1 If you reside outside the US or your business is located outside the US: You agree that any claim, cause of action, or dispute you have against us that arises out of or relates to any access or use of the SDK must be resolved exclusively in the U.S. District Court for the Northern District of California or a state court located in San Mateo County, that you submit to the personal jurisdiction of either of these courts for the purpose of litigating any such claim, and that the laws of the State of California will govern this Agreement and any such claim, without regard to conflict of law provisions. 7.4.2 If you reside in the US or your business is located in the US: You and we agree to arbitrate any claim, cause of action, or dispute between you and us that arises out of or relates to any access or use of the SDK for business or commercial purposes ("commercial claim"). This provision does not cover any commercial claims relating to violations of your or our intellectual property rights, including, but not limited to, copyright infringement, patent infringement, trademark infringement, violations of the brand guidelines, violations of your or our confidential information or trade secrets, or efforts to interfere with our products or engage with our products in unauthorized ways (for example, automated ways). 7.4.3 We and you agree that, by entering into this arbitration provision all parties are waiving their respective rights to a trial by jury or to participate in a class or representative action. THE PARTIES AGREE THAT EACH MAY BRING COMMERCIAL CLAIMS AGAINST THE OTHER ONLY IN ITS INDIVIDUAL

Library	Version	License
		CAPACITY, AND NOT AS A PLAINTIFF OR CLASS MEMBER IN ANY PURPORTED CLASS, REPRESENTATIVE, OR PRIVATE ATTORNEY GENERAL PROCEEDING. You may bring a commercial claim only on your own behalf and cannot seek relief that would affect other parties. If there is a final judicial determination that any particular commercial claim (or a request for particular relief) cannot be arbitrated in accordance with this paragraph's limitations, then only that commercial claim (or only that request for relief) may be brought in court. All other commercial claims (or requests for relief) remain subject to this paragraph. 7.4.4 The Federal Arbitration Act governs the interpretation and enforcement of this arbitration provision. All issues are for an arbitrator to decide, except that only a court may decide issues relating to the scope or enforceability of this arbitration provision or the interpretation of the prohibition of class and representative actions. 7.4.5 If any party intends to seek arbitration of a dispute, that party must provide the other party with notice in writing. 7.4.6 The arbitration will be governed by the AAA's Commercial Arbitration Rules ("AAA Rules"), as modified by this Agreement, and will be administered by the AAA. If the AAA is unavailable, the parties will agree to another arbitration provider or the court will appoint a substitute. The arbitrator will not be bound by rulings in other arbitrations in which you are not a party. To the fullest extent permitted by applicable law, any evidentiary submissions made in arbitration will be maintained as confidential in the absence of good cause for its disclosure. The arbitrator's award will be maintained as confidential only to the extent necessary to protect either party's trade secrets or proprietary business information or to comply with a legal requirement mandating confidentiality. Each party will be responsible for paying any AAA filing, administrative and arbitrator fees in accordance with AAA Rules, except that we will pay for your filing, administrative, and arbitrator fees if your commercial claim for damages does not exceed $75,000 and is non-frivolous (as measured by the standards set forth in Federal Rule of Civil Procedure 11(b)). 7.4.7 If you do not wish to be bound by this provision (including its waiver of class and representative claims), you must notify us as set forth below within 30 days of the first acceptance date of any version of this Agreement containing an

Library	Version	License
		arbitration provision. Your notice to us under this subsection must be submitted to the address here: Facebook Technologies, LLC, 1 Hacker Way, Menlo Park, California 94025 7.4.8 All commercial claims between us, whether subject to arbitration or not, will be governed by California law, excluding California's conflict of laws rules, except to the extent that California law is contrary to or preempted by federal law. 7.4.9 If a commercial claim between you and us is not subject to arbitration, you agree that the claim must be resolved exclusively in the U.S. District Court for the Northern District of California or a state court located in San Mateo County, and that you submit to the personal jurisdiction of either of these courts for the purpose of litigating any such claim. 7.4.10 If any provision of this dispute resolution provision is found unenforceable, that provision will be severed and the balance of the dispute resolution provision will remain in full force and effect.
OpenAL	1.22.0	LGPL version 2.1 license available here: gnu.org/licenses/old-licenses/lgpl-2.1.en.html#SEC1
OpenColorIO	2.1.1	Copyright Contributors to the OpenColorIO Project. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,

Library	Version	License
		INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
OpenEXR	3.1.4	Copyright (c) 2006, Industrial Light & Magic, a division of Lucasfilm Entertainment Company Ltd. Portions contributed and copyright held by others as indicated. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Industrial Light & Magic nor the names of any other contributors to this software may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

Library	**Version**	**License**
		PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
OpenImageDenoise	1.4.3	Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

Library	Version	License
		"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive,

Library	Version	License
		no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminateas of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as

Library	Version	License
		modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

Library	Version	License
		9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
OpenNURBS	7.x	Copyright (c) 1993-2006 Robert McNeel & Associates. All Rights Reserved. Rhinoceros is a registered trademark of Robert McNeel & Associates. THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT EXPRESS OR IMPLIED

Library	Version	License
		WARRANTY. ALL IMPLIED WARRANTIES OF FITNESS FOR ANY PARTICULAR PURPOSE AND OF MERCHANTABILITY ARE HEREBY DISCLAIMED. The openNURBS Initiative provides CAD, CAM, CAE, and computer graphics software developers the tools to accurately transfer 3-D geometry between applications. The tools provided by openNURBS include: • C++ source code libraries to read and write the file format. • Quality assurance and revision control. • Various supporting libraries and utilities. • Technical support. Unlike other open development initiatives, alliances, or consortia: • Commercial use is encouraged. • The tools, support, and membership are free. • There are no restrictions. Neither copyright nor copyleft restrictions apply. • No contribution of effort or technology is required from the members, although it is encouraged. For more information, please see http://www.openNURBS.org.
OpenSSL	1.1.1m	Copyright (c) 1998-2022 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit.

Library	Version	License
		(http://www.openssl.org/)" 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. 5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)" THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. == This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Original SSLeay License ----------------------- Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

Library	Version	License
		All rights reserved. This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-). 4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

Library	Version	License
		THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]
OpenSubdiv	3.5.0	Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

Library	Version	License
		"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently

Library	Version	License
		incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminateas of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the

Library	Version	License
		attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks

Library	Version	License
		associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the

Library	Version	License
		License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
OpenVDB	10.0.1	**Mozilla Public License** Version 2.0 **1. Definitions** 1.1. “Contributor” means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software. 1.2. “Contributor Version” means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution. 1.3. “Contribution” means Covered Software of a particular Contributor. 1.4. “Covered Software” means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof. 1.5. “Incompatible With Secondary Licenses” means that the initial Contributor has attached the notice described in Exhibit B to the

Library	Version	License
		Covered Software; or that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License. 1.6. “Executable Form” means any form of the work other than Source Code Form. 1.7. “Larger Work” means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software. 1.8. “License” means this document. 1.9. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License. 1.10. “Modifications” means any of the following: any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or any new file in Source Code Form that contains any Covered Software. 1.11. “Patent Claims” of a Contributor means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

Library	Version	License
		1.12. “Secondary License” means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses. 1.13. “Source Code Form” means the form of the work preferred for making modifications. 1.14. “You” (or “Your”) means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity. **2. License Grants and Conditions** 2.1. Grants Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license: under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version. 2.2. Effective Date The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such

Library	Version	License
		Contribution. 2.3. Limitations on Grant Scope The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1 (b) above, no patent license is granted by a Contributor: for any code that a Contributor has removed from Covered Software; or for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or under Patent Claims infringed by Covered Software in the absence of its Contributions. This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4). 2.4. Subsequent Licenses No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3). 2.5. Representation Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License. 2.6. Fair Use This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

Library	Version	License
		2.7. Conditions Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1. **3. Responsibilities** 3.1. Distribution of Source Form All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form. 3.2. Distribution of Executable Form If You distribute Covered Software in Executable Form then: such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License. 3.3. Distribution of a Larger Work You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such

Library	Version	License
		Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s). 3.4. Notices You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies. 3.5. Application of Additional Terms You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction. **4. Inability to Comply Due to Statute or Regulation** If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it. **5. Termination** 5.1. The rights granted under this License will terminate automatically if You fail

Library	Version	License
		to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice. 5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate. 5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination. **6. Disclaimer of Warranty** Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer. **7. Limitation of Liability**

Library	Version	License
		Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You. **8. Litigation** Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims. **9. Miscellaneous** This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor. **10. Versions of the License** 10.1. New Versions Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

Library	Version	License
		10.2. Effect of New Versions You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward. 10.3. Modified Versions If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License). 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached. **Exhibit A - Source Code Form License Notice** This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at https://mozilla.org/MPL/2.0/. If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice. You may add additional accurate notices of copyright ownership. **Exhibit B - “Incompatible With Secondary Licenses” Notice** This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.
OpenVR	1.0.8	Copyright (c) 2015, Valve Corporation

Library	Version	License
		All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Optix	7.4.0	DESIGNWORKS NVIDIA SDKS, SAMPLES AND TOOLS AGREEMENT, DISTRIBUTION RIGHTS (V.13.06.2017) SOFTWARE DEVELOPER KITS, SAMPLES AND TOOLS LICENSE AGREEMENT (with distribution rights) IMPORTANT – READ BEFORE DOWNLOADING, INSTALLING, COPYING OR USING THE LICENSED SOFTWARE

Library	Version	License
		READ CAREFULLY: This Software Developer Kits, Samples and Tools License Agreement ("Agreement"), made and entered into as of the time and date of click through action (“Effective Date”), is a legal agreement between you and NVIDIA Corporation ("NVIDIA") and governs the use of the following NVIDIA deliverables to the extent provided to you under this Agreement: API’s, source code and header files, data sets and assets (examples include images, textures, models, scenes, videos, native API input/output files), binary software and/or documentation (collectively, "Licensed Software"). By downloading, installing, copying, or otherwise using the Licensed Software, you agree to be bound by the terms of this Agreement. If you do NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT DOWNLOAD, INSTALL, COPY OR USE THE NVIDIA LICENSED SOFTWARE. IF YOU ARE ENTERING INTO THIS AGREEMENT ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE LEGAL AUTHORITY TO BIND THE ENTITY TO THIS AGREEMENT, IN WHICH CASE “YOU” WILL MEAN THE ENTITY YOU REPRESENT. IF YOU DON’T HAVE SUCH AUTHORITY, OR IF YOU DON’T ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT, THEN NVIDIA IS UNWILLING TO LICENSE THE LICENSED SOFTWARE TO YOU, AND YOU MAY NOT DOWNLOAD, INSTALL, COPY OR USE IT. 1. LICENSE. 1.1 License Grant. Subject to the terms of this Agreement, NVIDIA hereby grants you a nonexclusive, non-transferable, worldwide, revocable, limited, royalty-free, fully paid-up license during the term of this Agreement to: (i) install, use and reproduce the Licensed Software delivered by NVIDIA plus make modifications and create derivative works of the source code and header files delivered by NVIDIA, provided that the software is executed only in hardware products as specified by NVIDIA in the accompanying documentation (such as release notes) as supported, to develop, test and service your products (each, a “Customer Product”) that are interoperable with supported hardware products. If the NVIDIA documentation is silent, the supported hardware consists of certain NVIDIA GPUs; and (ii) incorporate Licensed Software as delivered by NVIDIA (including source code and header files as modified by you) into a Customer Product in binary format

Library	Version	License
		only and sub-license and distribute a Customer Product for use by your recipients only in the hardware products specified by NVIDIA as supported, provided that: (a) all such distributions by you or your distribution channels are consistent with the terms of this Agreement; and (b) you must enter into enforceable agreements with your recipients that binds them to terms that are consistent with the terms set forth in this Agreement for their use of the software binaries, including (without limitation) terms relating to the license grant and license restrictions, confidentiality and protection of NVIDIA's intellectual property rights in and to the software you distributed. You are liable for the distribution and the use of distributed software if you failed to comply or enforce the distribution requirements of this Agreement. You agree to notify NVIDIA in writing of any known or suspected use or distribution of the Licensed Software that are not in compliance with the terms of this Agreement. 1.2 Enterprise and Contractor Usage. Under this Agreement you may allow (i) your Enterprise employees, and (ii) individuals who work primarily for your Enterprise on a contractor basis and from your secure network (each a "Contractor") to access and use the Licensed Software pursuant to the terms in Section 1 solely to perform work on your behalf, provided further that with respect to Contractors: (i) you obtain a written agreement from the Contractor which contains terms and obligations with respect to access to or use of Licensed Software no less protective of NVIDIA than those set forth in this Agreement, and (ii) such Contractor's access and use expressly excludes any sublicensing or distribution rights for the Licensed Software. You are responsible for the compliance with the terms and conditions of this Agreement by your Enterprise and Contractors. Any act or omission that if committed by you would constitute a breach of this Agreement shall be deemed to constitute a breach of this Agreement if committed by your Enterprise or Contractors. "Enterprise" means you or any company or legal entity for which you accepted the terms of this Agreement, and their subsidiaries of which your company or legal entity owns more than fifty percent (50%) of the issued and outstanding equity. 1.3 No Support. NVIDIA is under no obligation to provide support for the Licensed Software or to provide any error corrections or updates to the Licensed Software under this Agreement. 1.4 Product Specific Terms. With respect to the Iray Developer Edition Licensed

Library	Version	License
		Software, a separate license is required from NVIDIA to enable or use the Iray runtime in any given machine. 1.5 Notification. You are required to notify NVIDIA prior to use of the NVIDIA DesignWorks Licensed Software in a commercial application (including a plug-in to a commercial application). Please send notification by visiting https://developer.nvidia.com/sw-notification and submitting the web form requested information. NVIDIA will request company name, DesignWorks software and version used, platform, commercial application release date, and weblink to product/video. Failure to notify NVIDIA pursuant to this section shall be considered a material breach of this Agreement. 2. LIMITATIONS. 2.1 License Restrictions. Except as expressly authorized in this Agreement, you agree that you will not (nor authorize third parties to): (i) copy and use software that was licensed to you for use in one or more devices in other unlicensed devices (provided that copies solely for backup purposes are allowed); (ii) reverse engineer, decompile, disassemble (except to the extent applicable laws specifically require that such activities be permitted) or attempt to derive the source code, underlying ideas, algorithm or structure of software provided to you in object code form; (iii) sell, transfer, assign, distribute, rent, loan, lease, sublicense or otherwise make available the Licensed Software or its functionality to third parties (a) as an application services provider or service bureau, (b) by operating hosted/virtual system environments, (c) by hosting, time sharing or providing any other type of services, or (d) otherwise by means of the internet; (iv) modify, translate or otherwise create any derivative works of any of the Licensed Software; (v) remove, alter, cover or obscure any proprietary notice that appears on or with the Licensed Software or any opies thereof; (vi) use the Licensed Software, or allow its use, transfer, transmission or export in violation of any applicable export control laws, rules or regulations; (vii) distribute, permit access to, or sublicense the Licensed Software as a stand-alone product; (viii) bypass, disable, circumvent or remove any form of copy protection, encryption, security or digital rights management or authentication mechanism used by NVIDIA in connection with the Licensed Software, or use the Licensed Software

Library	Version	License
		together with any authorization code, serial number, or other copy protection device not supplied by NVIDIA directly or through an authorized reseller; (ix) use the Licensed Software for the purpose of developing competing products or technologies or assisting a third party in such activities; (x) use the Licensed Software with any system or application where the use or failure of such system or application can reasonably be expected to threaten or result in personal injury, death, or catastrophic loss including, without limitation, use in connection with any nuclear, avionics, navigation, military, medical, life support or other life critical application ("Critical Applications"), unless the parties have entered into a Critical Applications agreement; (xi) distribute any modification or derivative work you make to the Licensed Software under or by reference to the same name as used by NVIDIA; or (xii) use the Licensed Software in any manner that would cause the Licensed Software to become subject to an Open Source License. Nothing in this Agreement shall be construed to give you a right to use, or otherwise obtain access to, any source code from which the software or any portion thereof is compiled or interpreted. You acknowledge that NVIDIA does not design, test, manufacture or certify the Licensed Software for use in the context of a Critical Application and NVIDIA shall not be liable to you or any third party, in whole or in part, for any claims or damages arising from such use. You agree to defend, indemnify and hold harmless NVIDIA and its affiliates, and their respective employees, contractors, agents, officers and directors, from and against any and all claims, damages, obligations, losses, liabilities, costs or debt, fines, restitutions and expenses (including but not limited to attorney's fees and costs incident to establishing the right of indemnification) arising out of or related to you and your Enterprise, and their respective employees, contractors, agents, distributors, resellers, end users, officers and directors use of Licensed Software outside of the scope of this Agreement or any other breach of the terms of this Agreement. "Open Source License" includes, without limitation, a software license that requires as a condition of use, modification, and/or distribution of such software that the software be (x) disclosed or distributed in source code form; (y) be licensed for the purpose of making derivative works; or (z) be redistributable at no charge. 2.2 Third Party License Obligations. You acknowledge and agree that the Licensed Software may include or incorporate third party technology (collectively

Library	Version	License
		“Third Party Components”), which is provided for use in or with the software and not otherwise used separately. If the Licensed Software includes or incorporates Third Party Components, then the third-party pass-through terms and conditions (“Third Party Terms”) for the particular Third Party Component will be bundled with the software or otherwise made available online as indicated by NVIDIA and will be incorporated by reference into this Agreement. In the event of any conflict between the terms in this Agreement and the Third Party Terms, the Third Party Terms shall govern. Copyright to Third Party Components are held by the copyright holders indicated in the copyright notices indicated in the Third Party Terms. Audio/Video Encoders and Decoders. You acknowledge and agree that it is your sole responsibility to obtain any additional third party licenses required to make, have made, use, have used, sell, import, and offer for sale your products or services that include or incorporate any Third Party Components and content relating to audio and/or video encoders and decoders from, including but not limited to, Microsoft, Thomson, Fraunhofer IIS, Sisvel S.p.A., MPEG-LA, and Coding Technologies as NVIDIA does not grant to you under this Agreement any necessary patent rights with respect to audio and/or video encoders and decoders. 2.3 Limited Rights. Your rights in the Licensed Software are limited to those expressly granted in Section 1 and no other licenses are granted whether by implication, estoppel or otherwise. NVIDIA reserves all rights, title and interest in and to the Licensed Software not expressly granted under this Agreement. 3. CONFIDENTIALITY. Neither party will use the other party’s Confidential Information, except as necessary for the performance of this Agreement, nor will either party disclose such Confidential Information to any third party, except to personnel of NVIDIA and its affiliates, you, your Enterprise, your Enterprise Contractors, and each party’s legal and financial advisors that have a need to know such Confidential Information for the performance of this Agreement, provided that each such personnel, employee and Contractor is subject to a written agreement that includes confidentiality obligations consistent with those set forth herein. Each party will use all reasonable efforts to maintain the confidentiality of all of the other party’s Confidential Information in its possession or control, but in no event less than the efforts that it ordinarily uses with respect to its own Confidential Information of similar nature and

Library	Version	License
		importance. The foregoing obligations will not restrict either party from disclosing the other party's Confidential Information or the terms and conditions of this Agreement as required under applicable securities regulations or pursuant to the order or requirement of a court, administrative agency, or other governmental body, provided that the party required to make such disclosure (i) gives reasonable notice to the other party to enable it to contest such order or requirement prior to its disclosure (whether through protective orders or otherwise), (ii) uses reasonable effort to obtain confidential treatment or similar protection to the fullest extent possible to avoid such public disclosure, and (iii) discloses only the minimum amount of information necessary to comply with such requirements. "Confidential Information" means the Licensed Software (unless made publicly available by NVIDIA without confidentiality obligations), and any NVIDIA business, marketing, pricing, research and development, know-how, technical, scientific, financial status, proposed new products or other information disclosed by NVIDIA to you which, at the time of disclosure, is designated in writing as confidential or proprietary (or like written designation), or orally identified as confidential or proprietary or is otherwise reasonably identifiable by parties exercising reasonable business judgment as confidential. Confidential Information does not and will not include information that: (i) is or becomes generally known to the public through no fault of or breach of this Agreement by the receiving party; (ii) is rightfully known by the receiving party at the time of disclosure without an obligation of confidentiality; (iii) is independently developed by the receiving party without use of the disclosing party's Confidential Information; or (iv) is rightfully obtained by the receiving party from a third party without restriction on use or disclosure. 4. OWNERSHIP. 4.1 Ownership of Licensed Software. The Licensed Software, and the respective intellectual property rights therein, is and will remain the sole and exclusive property of NVIDIA and its licensors, whether the Licensed Software is separate from or combined with any other products or materials. You shall not knowingly engage in any act or omission that would impair NVIDIA's and/or its licensors' intellectual property rights in the Licensed Software or any other materials, information, processes or subject matter proprietary to NVIDIA. NVIDIA's

Library	Version	License
		licensors are intended third party beneficiaries with the right to enforce provisions of this Agreement with respect to their Confidential Information and/or intellectual property rights. 4.2 Modifications. You have no obligation to provide your permitted modifications to NVIDIA. You hold all rights, title and interest in and to the modifications to and derivative works of the NVIDIA source code and header files that you create as permitted hereunder, subject to NVIDIA's underlying intellectual property rights in and to the NVIDIA software; provided, however that you grant NVIDIA, its affiliates and their respective customers an irrevocable, perpetual, nonexclusive, worldwide, royalty-free paid-up license to make, have made, use, have used, reproduce, sell, license, distribute, sublicense, transfer and otherwise commercialize modifications and derivative works including (without limitation) with the Licensed Software or other products, technologies or materials. 5. FEEDBACK. You have no obligation to provide Feedback to NVIDIA. However, NVIDIA and/or its affiliates may use and include any Feedback that you provide to improve the Licensed Software or other NVIDIA products, technologies or materials. Accordingly, if you provide Feedback, you agree that NVIDIA and/or its affiliates may at their option, and may permit its licensees, to make, have made, use, have used, reproduce, sell, license, distribute, sublicense, transfer and otherwise commercialize the Feedback in the Licensed Software or in other products, technologies or materials without the payment of any royalties or fees to you. All Feedback becomes the sole property of NVIDIA and may be used in any manner NVIDIA sees fit, and you hereby assign to NVIDIA all of your right, title and interest in and to any Feedback. NVIDIA has no obligation to respond to Feedback or to incorporate Feedback into the Licensed Software. "Feedback" means any and all suggestions, feature requests, comments or other feedback relating to the Licensed Software, including possible enhancements or modifications thereto. 6. NO WARRANTIES. THE LICENSED SOFTWARE IS PROVIDED BY NVIDIA "AS IS" AND "WITH ALL FAULTS," AND NVIDIA EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND OR NATURE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTIES OF OPERABILITY,

Library	Version	License
		CONDITION, VALUE, ACCURACY OF DATA, OR QUALITY, AS WELL AS ANY WARRANTIES OF MERCHANTABILITY, SYSTEM INTEGRATION, WORKMANSHIP, SUITABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR THE ABSENCE OF ANY DEFECTS THEREIN, WHETHER LATENT OR PATENT. NO WARRANTY IS MADE BY NVIDIA ON THE BASIS OF TRADE USAGE, COURSE OF DEALING OR COURSE OF TRADE. NVIDIA DOES NOT WARRANT THAT THE LICENSED SOFTWARE WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION THEREOF WILL BE UNINTERRUPTED OR ERRORFREE, OR THAT ALL ERRORS WILL BE CORRECTED. YOU ACKNOWLEDGE THAT NVIDIA'S OBLIGATIONS UNDER THIS AGREEMENT ARE FOR THE BENEFIT OF YOU ONLY. Nothing in this warranty section affects any statutory rights of consumers or other recipients to the extent that they cannot be waived or limited by contract under applicable law. 7. LIMITATION OF LIABILITY. TO THE MAXIMUM EXTENT PERMITTED BY LAW NVIDIA OR ITS LICENSORS SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR ANY LOST PROFITS, LOSS OF USE, LOSS OF DATA OR LOSS OF GOODWILL), OR THE COSTS OF PROCURING SUBSTITUTE PRODUCTS, ARISING OUT OF OR IN CONNECTION WITH THIS AGREEMENT OR THE USE OR PERFORMANCE OF THE LICENSED SOFTWARE, WHETHER SUCH LIABILITY ARISES FROM ANY CLAIM BASED UPON BREACH OF CONTRACT, BREACH OF WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR ANY OTHER CAUSE OF ACTION OR THEORY OF LIABILITY. IN NO EVENT WILL NVIDIA'S TOTAL CUMULATIVE LIABILITY UNDER OR ARISING OUT OF THIS AGREEMENT EXCEED THE GREATER OF THE NET AMOUNT NVIDIA RECEIVED FOR YOUR USE OF THE LICENSED SOFTWARE ONE HUNDRED U.S. DOLLARS (US $100). THE NATURE OF THE LIABILITY, THE NUMBER OF CLAIMS OR SUITS OR THE NUMBER OF PARTIES WITHIN YOUR ENTERPRISE THAT ACCEPTED THE TERMS OF THIS AGREEMENT SHALL NOT ENLARGE OR EXTEND THIS LIMIT. THE FOREGOING LIMITATIONS SHALL APPLY REGARDLESS OF WHETHER NVIDIA OR ITS LICENSORS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF WHETHER ANY REMEDY FAILS ITS ESSENTIAL PURPOSE. 8. TERM AND TERMINATION. This Agreement and your licenses hereunder shall

Library	Version	License
		become effective upon the Effective Date and shall remain in effect unless and until terminated as follows: (i) automatically if you breach any of the terms of this Agreement; or (ii) by either party upon written notice if the other party becomes the subject of a voluntary or involuntary petition in bankruptcy or any proceeding relating to insolvency, receivership, liquidation or composition for the benefit of creditors, if that petition or proceeding is not dismissed with prejudice within sixty (60) days after filing, or if a party ceases to do business; (iii) by you, upon ceasing to use the Licensed Software provided under this Agreement; or (iv) by NVIDIA upon written notice if you commence or participate in any legal proceeding against NVIDIA, with respect to the Licensed Software that is the subject of the proceeding during the pendency of such legal proceeding. Termination of this Agreement regardless of cause or nature shall be without prejudice to any other rights or remedies of the parties and shall be without liability for any loss or damage occasioned thereby. Upon any expiration or termination of this Agreement (i) you must promptly discontinue use of the Licensed Software, and (ii) you must promptly destroy or return to NVIDIA all copies of the Licensed Software and all portions thereof in your possession or control, and each party will promptly destroy or return to the other all of the other party's Confidential Information within its possession or control, provided that your prior distributions in accordance with this Agreement are not affected by the expiration or termination of this Agreement. Upon written request, you will certify in writing that you have complied with your obligations under this section. Sections 2 through 9 will survive the expiration or termination of this Agreement for any reason. 9. GENERAL. This Agreement constitutes the entire agreement of the parties with respect to the subject matter hereto and supersedes all prior negotiations, conversations, or discussions between the parties relating to the subject matter hereto, oral or written, and all past dealings or industry custom. Any additional and/or conflicting terms and conditions on purchase order(s) or any other documents issued by you are null, void, and invalid. Any amendment or waiver under this Agreement must be in writing and signed by representatives of both parties. This Agreement and the rights and obligations thereunder may not be assigned

Library	Version	License
		by you, in whole or in part, including by merger, consolidation, dissolution, operation of law, or any other manner, without written consent of NVIDIA, and any purported assignment in violation of this provision shall be void and of no effect. NVIDIA may assign, delegate or transfer this Agreement and its rights and obligations hereunder, and if to a non-affiliate you will be notified. Each party acknowledges and agrees that the other is an independent contractor in the performance of this Agreement, and each party is solely responsible for all of its employees, agents, contractors, and labor costs and expenses arising in connection therewith. The parties are not partners, joint ventures or otherwise affiliated, and neither has any authority to make any statements, representations or commitments of any kind to bind the other party without prior written consent. Neither party will be responsible for any failure or delay in its performance under this Agreement (except for any payment obligations) to the extent due to causes beyond its reasonable control for so long as such force majeure event continues in effect. This Agreement will be governed by and construed under the laws of the State of Delaware and the United States without regard to the conflicts of law provisions thereof and without regard to the United Nations Convention on Contracts for the International Sale of Goods. The parties consent to the personal jurisdiction of the federal and state courts located in Santa Clara County, California. You acknowledge and agree that a breach of any of your promises or agreements contained in this Agreement may result in irreparable and continuing injury to NVIDIA for which monetary damages may not be an adequate remedy and therefore NVIDIA is entitled to seek injunctive relief as well as such other and further relief as may be appropriate. If any court of competent jurisdiction determines that any provision of this Agreement is illegal, invalid or unenforceable, the remaining provisions will remain in full force and effect. Unless otherwise specified, remedies are cumulative. The Licensed Software has been developed entirely at private expense and is “commercial items” consisting of “commercial computer software” and “commercial computer software documentation” provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions set forth in this

Library	Version	License
		Agreement pursuant to DFARS 227.7202-3(a) or as set forth in subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, as applicable. Contractor/manufacturer is NVIDIA, 2701 San Tomas Expressway, Santa Clara, CA 95050. You acknowledge that the Licensed Software described under this Agreement is subject to export control under the U.S. Export Administration Regulations (EAR) and economic sanctions regulations administered by the U.S. Department of Treasury's Office of Foreign Assets Control (OFAC). Therefore, you may not export, reexport or transfer in-country the Licensed Software without first obtaining any license or other approval that may be required by BIS and/or OFAC. You are responsible for any violation of the U.S. or other applicable export control or economic sanctions laws, regulations and requirements related to the Licensed Software. By accepting this SLA, you confirm that you are not a resident or citizen of any country currently embargoed by the U.S. and that you are not otherwise prohibited from receiving the Licensed Software. Any notice delivered by NVIDIA to you under this Agreement will be delivered via mail, email or fax. Please direct your legal notices or other correspondence to NVIDIA Corporation, 2701 San Tomas Expressway, Santa Clara, California 95050, United States of America, Attention: Legal Department.
Perl	5.36.0	Preamble The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications. Definitions: "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification. "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

Library	Version	License
		"Copyright Holder" is whoever is named in the copyright or copyrights for the package. "You" is you, if you're thinking about copying or distributing this Package. "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.) "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it. 1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers. 2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version. 3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following: a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package. b) use the modified Package only within your corporation or organization. c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

Library	Version	License
		d) make other distribution arrangements with the Copyright Holder. 4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following: a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version. b) accompany the distribution with the machine-readable source of the Package with your modifications. c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version. d) make other distribution arrangements with the Copyright Holder. 5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded. 6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

Library	Version	License
		7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language. 8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package. 9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission. 10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
PNG	1.6.37	This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail. COPYRIGHT NOTICE, DISCLAIMER, and LICENSE: If you modify libpng you may insert additional notices immediately following this sentence. This code is released under the libpng license. libpng versions 1.2.6, August 15, 2004, through 1.4.8, July 7, 2011, are Copyright (c) 2004, 2006-2010 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.2.5 with the following individual added to the list of Contributing Authors Cosmin Truta

Library	Version	License
		libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002, are Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors Simon-Pierre Cadieux Eric S. Raymond Gilles Vollant and with the following additions to the disclaimer: There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user. libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors: Tom Lane Glenn Randers-Pehrson Willem van Schaik libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors: John Bowler Kevin Bracey Sam Bushell

Library	Version	License
		Magnus Holmgren Greg Roelofs Tom Tanner libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc. For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals: Andreas Dilger Dave Martindale Guy Eric Schalnat Paul Schmidt Tim Wegner The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage. Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions: 1. The origin of this source code must not be misrepresented. 2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source. 3. This Copyright notice may not be removed or altered from any source or altered source distribution.

Library	Version	License
		The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated. A "png_get_copyright" function is available, for convenient use in "about" boxes and the like: printf("%s",png_get_copyright(NULL)); Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31). Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative. Glenn Randers-Pehrson glennrp at users.sourceforge.net July 7, 2011
PTex	2.4.1	Copyright 2009 Disney Enterprises, Inc. All rights reserved Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: • Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. • Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. The names “Disney”, “Walt Disney Pictures”, “Walt Disney Animation Studios” or the names of its contributors may NOT be used to endorse or promote products derived from this software without specific prior written permission from Walt Disney Pictures. Disclaimer: THIS SOFTWARE IS PROVIDED BY WALT DISNEY PICTURES AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES,

Library	Version	License
		INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT AND TITLE ARE DISCLAIMED. IN NO EVENT SHALL WALT DISNEY PICTURES, THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND BASED ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
ProtoBuf	3.17.2	Copyright 2008, Google Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

Library	Version	License
		PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.
PySide6	6.3.1	GNU LESSER GENERAL PUBLIC LICENSE Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions. As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License. "The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below. An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library. A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

Library	Version	License
		The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version. The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL. You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions. If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version: a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy. 3. Object Code Incorporating Material from Library Header Files. The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following: a) Give prominent notice with each copy of the object code that the Library is

Library	Version	License
		used in it and that the Library and its use are covered by this License. b) Accompany the object code with a copy of the GNU GPL and this license document. 4. Combined Works. You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following: a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. b) Accompany the Combined Work with a copy of the GNU GPL and this license document. c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. d) Do one of the following: 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. 5. Combined Libraries.R

Library	Version	License
		You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following: a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work. 6. Revised Versions of the GNU Lesser General Public License. The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation. If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
Python	3.10.10	A. HISTORY OF THE SOFTWARE ========================== Python was created in the early 1990s by Guido van Rossum at Stichting

<table>
<tr><th>Library</th><th>Version</th><th>License</th></tr>
<tr><td></td><td></td><td>
<p>Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.</p>
<p>In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.</p>
<p>In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.</p>
<p>All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.</p>
<table>
<tr><th>Release</th><th>Derived from</th><th>Year</th><th>Owner</th><th>GPL-Compatible? (1)</th></tr>
<tr><td>0.9.0 thru 1.2</td><td></td><td>1991-1995</td><td>CWI</td><td>yes</td></tr>
<tr><td>1.3 thru 1.5.2</td><td>1.2</td><td>1995-1999</td><td>CNRI</td><td>yes</td></tr>
<tr><td>1.6</td><td>1.5.2</td><td>2000</td><td>CNRI</td><td>no</td></tr>
<tr><td>2.0</td><td>1.6</td><td>2000</td><td>BeOpen.com</td><td>no</td></tr>
<tr><td>1.6.1</td><td>1.6</td><td>2001</td><td>CNRI</td><td>yes (2)</td></tr>
<tr><td>2.1</td><td>2.0+1.6.1</td><td>2001</td><td>PSF</td><td>no</td></tr>
<tr><td>2.0.1</td><td>2.0+1.6.1</td><td>2001</td><td>PSF</td><td>yes</td></tr>
<tr><td>2.1.1</td><td>2.1+2.0.1</td><td>2001</td><td>PSF</td><td>yes</td></tr>
<tr><td>2.1.2</td><td>2.1.1</td><td>2002</td><td>PSF</td><td>yes</td></tr>
<tr><td>2.1.3</td><td>2.1.2</td><td>2002</td><td>PSF</td><td>yes</td></tr>
<tr><td>2.2 and above</td><td>2.1.1</td><td>2001-now</td><td>PSF</td><td>yes</td></tr>
</table>
<p>Footnotes:</p>
<p>(1) GPL-compatible doesn't mean that we're distributing Python under the GPL.</p>
</td></tr>
</table>

Library	Version	License
		All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't. (2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL. Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible. B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON == PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2 -- 1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated documentation. 2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee. 3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

Library	Version	License
		4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS. 5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. 6. This License Agreement will automatically terminate upon a material breach of its terms and conditions. 7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party. 8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement. BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0 --- BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1 1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software"). 2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-

Library	Version	License
		wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee. 3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS. 4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. 5. This License Agreement will automatically terminate upon a material breach of its terms and conditions. 6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page. 7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement. CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1 ---------------------------------------

Library	Version	License
		1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation. 2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013". 3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1. 4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS. 5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON

Library	Version	License
		1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. 6. This License Agreement will automatically terminate upon a material breach of its terms and conditions. 7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party. 8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement. ACCEPT CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2 -- Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright

Library	Version	License
		notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Qt	6	GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions. As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License. "The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below. An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by

Library	Version	License
		the Library. A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version". The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version. The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL. You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions. If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version: a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy. 3. Object Code Incorporating Material from Library Header Files. The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of

Library	Version	License
		your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following: a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. b) Accompany the object code with a copy of the GNU GPL and this license document. 4. Combined Works. You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following: • a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. • b) Accompany the Combined Work with a copy of the GNU GPL and this license document. • c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. • d) Do one of the following: 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the

Library	Version	License
		Linked Version. e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.) 5. Combined Libraries. You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following: a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work. 6. Revised Versions of the GNU Lesser General Public License. The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any

Library	Version	License
		later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation. If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
RadeomImageFilters	1.7.3	Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

Library	Version	License
		"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

Library	Version	License
		3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminateas of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You

Library	Version	License
		may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this

Library	Version	License
		License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Library	Version	License
RapidJSON	1.1.0	Copyright (C) 2015 THL A29 Limited, a Tencent company, and Milo Yip. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so. THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
sse2neon	1.5.1	Copyright (c) 2015-2024 SSE2NEON Contributors Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Substance	8.6.4	Adobe Substance3D Materials SDK is used under license from Adobe Inc.
TBB	2020 Update	Apache License

Library	Version	License
	3	Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions,

Library	Version	License
		annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a

Library	Version	License
		Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminateas of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

Library	Version	License
		5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold

Library	Version	License
		each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
TinyGLTF	2.8.18	Copyright (c) 2017 Syoyo Fujita, Aurélien Chatelain and many contributors Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

Library	Version	License
		IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
TinyXML	2.6.0	TinyXML is released under the zlib license: This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution.
USD	23.05	Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

Library	Version	License
		"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

Library	Version	License
		means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminateas of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

Library	Version	License
		(b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and

Library	Version	License
		reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be

Library	Version	License
		enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
ZeroMQ	4.3.3	GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions. As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License. "The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

Library	Version	License
		An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library. A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version". The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version. The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL. You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions. If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version: a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy. 3. Object Code Incorporating Material from Library Header Files.

Library	Version	License
		The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following: a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. b) Accompany the object code with a copy of the GNU GPL and this license document. 4. Combined Works. You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following: • a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. • b) Accompany the Combined Work with a copy of the GNU GPL and this license document. • c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. • d) Do one of the following: 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library

Library	Version	License
		already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.) 5. Combined Libraries. You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following: a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work. 6. Revised Versions of the GNU Lesser General Public License. The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General

Library	Version	License
		Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation. If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.
ZLib	1.2.11	(C) 1995-2024 Jean-loup Gailly and Mark Adler This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution. Jean-loup Gailly Mark Adler jloup@gzip.org madler@alumni.caltech.edu